Research Activities in Digital Photogrammetry at The Ohio State University

A Collection of Papers Presented At the XVII Congress of ISPRS

Toni Schenk
Editor

Report No. 418

Department of Geodetic Science and Surveying
The Ohio State University
Columbus, Ohio 43210-1247

July 1992

Foreword

Most of the research efforts in photogrammetry are now directed toward digital photogrammetry. The arrival of digital photogrammetric workstations is a clear demonstration of the considerable success that has been achieved in this rapidly developing subfield of photogrammetry.

At The Ohio State University we embarked on research in digital photogrammetry six years ago. From a rather small group with no special equipment we have grown: two faculty, several postdoctoral researchers and fifteen PhD students are now actively involved in digital photogrammetry research projects. Our laboratories are equipped with a high-performance softcopy workstation (Intergraph ImageStation), several UNIX workstations, image processing systems, digital cameras and scanners—all networked together.

It is with great pleasure that I serve as editor of this report. I have gently persuaded the majority of my advisees to submit a paper to the ISPRS Congress 1992. This report is a collection of those contributions. The research in my group is primarily focused on automating photogrammetric processes. Specifically, we are working on *surface reconstruction*, *feature extraction and recognition*, and on *automated aerotriangulation.*

By surface reconstruction I refer not only to automatic DEM collection but include segmenting processes with the goal of grouping the surface into breaklines and smooth patches to support the subsequent processes of object recognition. Several papers contribute toward that goal.

The first two contributions are my invited papers for the ISPRS Congress 1992. They may serve as a framework within which all the other contributions fit. The first paper summarizes the most important concepts and issues of computer vision and relates them to digital photogrammetry. The second paper builds on this overview and focuses on conceptual and algorithmic aspects. There is some repetition because the presentations will not address the same audience.

Zong's paper is concerned with matching edges—in our case zero-crossings. She continued work originally initiated by Jin-Chen Li. The idea is to find corresponding edges by positioning the templet on an edge in one image and by finding the corresponding edge by cross-correlation. The matching results are checked for continuity as it is unlikely that discontinuities occur along edges.

Matched edges are irregularly distributed in object space. Thus, the problem of interpolating the surface arises. **Al-Tahir** investigates surface fitting methods. The thin plate method with weak continuity constraints is of particular interest because it allows detecting breaklines. They are compared with the position of edges which are also potential breaklines. It now becomes possible to verify the hypothesis about breaklines and to use this information on the next level of matching.

Wang analyzes the interpolated surface for objects of a certain vertical dimension, called humps. The surface is segmented into regions of similar elevations followed by comparing the shapes of their boundaries. The boundaries are grouped and classified into near horizontal and vertical edges. Hump detection is important for reconstructing surfaces in large-scale urban areas.

One of the reasons for the astounding capability of the human visual system to reconstruct surfaces is to integrate several depth cues, e.g., apparent size, perspective, motion and tex-

ture. **Lee's** paper is concerned with segmenting the image by analyzing texture. Surface orientation and texture are very closely related.

Most surface reconstruction methods adopt a hierarchical approach, for example by constructing image pyramids. **Stefanidis** examines the hierarchical approach with regard to the scale space theory. He explores the relationship between images and surfaces since both can be represented in scale space.

The goal of our OSU surface reconstruction system is to segment the surface into smooth patches and breaklines and to represent them by a symbolic description. This step is important for the subsequent task of object recognition. **Krupnik** groups matched edges in the object space into straight lines and regular curves. He compares different methods that allow 3-D segmentation.

A fundamental task that occurs at all levels of the computer vision paradigm is comparing shapes. Fourier descriptors have long been used for that purpose. However, there is no real quantitative criterion for measuring the similarity of two objects. **Tseng** employs an innovative approach by embedding shape invariants in a least-squares adjustment procedure that provides not only a superior measure for the goodness of the match but also the transformation parameters between the two shapes.

Late vision processes, such as object recognition and image understanding, are application dependent (or goal-driven, if you prefer) and must incorporate domain-specific knowledge. **Al-Garni's** work of interpreting landforms with the help of a knowledge-based system is an important contribution to our research since we will have the surface reconstruction system under the control of a knowledge-based system.

This report contains two contributions in the area of automated aerotriangulation, a subject of considerable research interest. **Agouris'** paper addresses the problem of matching multiple image patches simultaneously. This important step corresponds to the classical procedure of transferring and measuring points. Considering the notorious problem with point transferring one can expect a significant increase in reliability from multiple image matching. My paper describes general mathematical models which are suitable for matching multiple image patches.

The remaining two papers from **Toth** deal with analytical plotters and their digital counterparts — softcopy workstations. Both workstation types play an important role in our research. So does Charles Toth who developed software systems for analytical plotters that are invaluable not only for research but student laboratories as well. His second paper describes our research efforts on the softcopy workstation to keep the measuring mark (cursor) automatically on the ground. Thus, the operator is relieved from setting the cursor precisely on the ground.

Finally, I want to thank the authors for their contributions. However, this report would not have been possible without the help of Peggy Agouris who spent many night shifts to put everything together. I wish to thank Irene Tesfai who diligently read all the papers. Her comments are appreciated by every writer—none with English as mother tongue. Funding for most of the research reported here was provided in part by the NASA Center for the Commercial Development of Space Component of the Center for Mapping at The Ohio State University.

Toni Schenk

TABLE OF CONTENTS

MACHINE VISION AND CLOSE-RANGE PHOTOGRAMMETRY

Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

ABSTRACT

This paper provides an overview of concepts and methods of machine vision as it may pertain to close-range photogrammetry. The ultimate goal of a machine vision system is to recognize objects from one or several 2-D images. This cannot be achieved in one giant step. Intermediate processes and representations are necessary. Usually, the first goal is to reconstruct the 3-D surface of the object space, with emphasis placed on a symbolic description in which surface properties are made explicit. The surface information aids the subsequent object recognition task. The paper concludes with suggestions on how some of the concepts developed in machine vision can (and should!) be employed in digital close-range applications.

Report No. 418 July 1992

1 INTRODUCTION

Since time immemorial, mankind has been fascinated by the idea to create a machine that would somehow exhibit mental capabilities. The robot is a typical example of such dreams. With the attempt of endowing computers with information processing capabilities similar to those of humans, researchers in artificial intelligence pursue this dream in modern times. Ever since computers became available, researchers tried to mimic the mental faculty of seeing. The endeavour *machine vision* seemed to achieve quick success. Expectations were pushed far beyond what could be delivered and disillusion followed. The problem has been tremendously underestimated—like many other problems tackled by artificial intelligence. We see and interpret scenes without conscious effort, however, this does not mean that the task is easy.

Clearly, the lack of a detailed understanding of vision is the reason why it is so difficult to make computers understand and analyze images. It seems only natural that someone who attempts to solve a vision task should have a good understanding of the human visual system. Admittedly, this view is not shared by every vision researcher.

As the name suggests, digital photogrammetry deals with digital imagery. Great strides have been made during the last ten years due to the availability of new hardware and software, such as image processing workstations, parallel processing, and increased storage capacity. This in turn spurred much interest in research and development. The arrival of digital photogrammetric workstations is a clear demonstration of the progress achieved.

The goal of digital photogrammetry is to capture images and to store, manipulate and process them automatically. In that regard, digital photogrammetry and machine vision have the same goals. The purpose of this paper is to present the major concepts, methods, solutions and issues of machine vision. This may be a risky enterprise, considering the glut of publications in that field, and the high probability that the machine vision research community would not unanimously agree on what the concepts and issues are.

We begin with a summary of human vision for it is a measure beyond all bounds. Most of the material presented is based on recent research results. We conclude the section about human vision with Marr's theory about vision because it is the most advanced approach to date. It has been widely accepted by visual psychologists and the machine vision research community.

The exposition of machine vision begins with the paradigm, followed by the most important concepts, methods, and critical issues. This paves the way for comparing digital close-range photogrammetry and machine vision. We elaborate on a few but very important aspects which the two disciplines share and point out where they differ. It is hoped the concluding remarks stimulate discussions on how digital photogrammetry and machine vision can benefit from each other – more than they do now.

2 HUMAN VISION

For an animal or person to respond properly to a changing environment it must detect objects, events and structures. This ability, called perception, requires that a living organism must be sensitive to different stimuli which carry important information about the environment. Most animals have some visual perception abilities. For people, vision is the most important sense. By the same token, it is by far the most impressive and complicated sense.

We see and analyze our environment continuously, nearly in real-time. That we do this without conscious effort does not imply that we know *how* we analyze and understand scenes, however. In fact, the lack of a detailed understanding of vision is the reason why it is so difficult to program a computer to analyze and understand images. It seems only natural then that someone who attempts to solve part of this task should have a basic understanding of human vision. Consider the following summary as an exciting journey through the fascinating world of vision. Most of the material presented in the next subsection is from Hubel (1988).

2.1 Neurophysiology of Human Vision

Neurophysiology is concerned with the processes that are performed by specialized tis-

sues and cells of the nervous system (Uttal, 1975). Visual information is processed in various stages at centers of specialized nerve cells, from the retina to the primary visual cortex. The processing centers are connected by the visual pathway which can be thought of as a serial link (see Fig. 1).

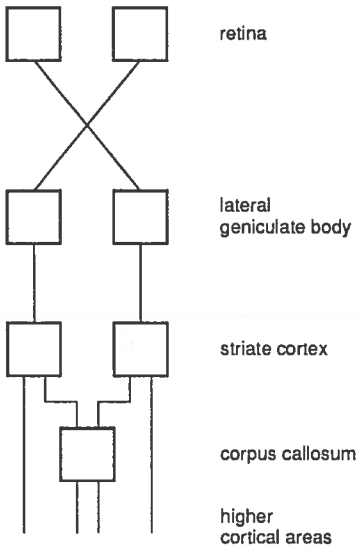


Fig. 1: Visual pathway; each structure consists of millions of cells. Information is sent to one or several higher order structures. (Figure adapted from Hubel, 1988).

Light is focused on the retina to form an image. Approximately 125 million light sensitive photoreceptors (rods and cones) are unevenly distributed over the entire posterior portion of the eyeball. The retina consists of three layers: photoreceptors, middle layer, and ganglion cells whose dendrites are bundled together to form the optic nerve. Oddly, light passes through two layers before it reaches the photoreceptors, except for the site of acute vision, the fovea, a region smaller than a millimeter in diameter.

It is tempting to compare the eye with a camera. The analogy must be met with caution, however. First, the quality of the retinal image is far inferior to that of any cheap Instamatic camera. Aberrations of lens and cornea are responsible for considerable distortions. The curvature of the retina causes straight lines in object space to appear curved, disturbing the metrical relationship between image and object space. Moreover, the constant movements of the eye results in a blurred image. While the purpose of the camera is to render a static snapshot of the world, the eye's and brain's purpose is to extract useful information to guide a person's response to an ever changing environment.

How do the ganglion cells respond to incident light and what is reported back to the next processing centers? First, we note that there are far fewer ganglion cells than photoreceptors (the ratio is approximately 1 : 125). This is a first indication that the retinal image is processed by the cells of the middle layer and the ganglion cells. It also implies that one ganglion cell receives impulses from several photoreceptors.

The receptive field of a ganglion cell refers to those receptors which are "connected" to it. The circular center of a receptive field is surrounded by a ring-shaped region. An on-center ganglion cell reacts most (increases its firing rate) if the center of its receptive field is stimulated, for example by shining a spot of light on the receptors that form the center. The ganglion cell stops firing if the center-surround region of its receptive field is stimulated, but reacts with a burst of impulses when the stimulus is turned off. Off-center cells exhibit the opposite behavior. For example, if their centers are stimulated, firing is suppressed. Both, on- and off-center cells do not respond if their entire receptive field is evenly illuminated.

We conclude that ganglion cells respond to brightness differences within their receptive fields, that is, to local intensity differences. Receptive fields differ in size. As one would expect, the size is smallest in the fovea and progressively increases further out in the visual field. The light intensity changes, transmitted by the optical nerve, are detected by biological filters of the retina. Campbell and Robson (1968) showed that cells are sensitive to different spatial frequencies—a strong indication that the visual input is processed in multiple independent channels.

In the interest of brevity, we skip the next processing stage, the lateral geniculate body, and shift our attention to the primary visual (striate) cortex, a complex substructure of the cerebral cortex. The visual cortex is topograph-

ically organized: an area of about two millimeter square has all the functionality. These areas—self-contained modules of the striate cortex—map out a portion of the visual field. Consequently, if one such area is damaged, the corresponding part of the retinal image is not processed further and the result is local blindness. Neighboring modules do not compensate for the loss. However, the perceptual process "fill in" completes the missing information by interpolating it from the surrounding area.

The specialization of cells in the cortex increases. So does the complexity of their receptive fields. Unlike cells of earlier levels, cortical cells have no circular symmetrical receptive fields, and they respond quite differently too. A simple cell, for example, responds best if a slit of light crosses its receptive field at a specific angle. Changing the orientation and position only slightly evokes no response. Other simple cells respond more strongly if one half of the receptive field is stimulated.

The most commonly found cells in the striate cortex are the complex cells. Like simple cells they respond to properly oriented stimuli. However, the cell's firing rate fades out rather quickly unless the stimulus is moved. So, complex cells are movement sensitive; they respond with a barrage of impulses if a properly oriented slit is swept across their receptive fields. Some complex cells are also direction sensitive. That is, it matters in which direction the slit is moved. That a large population of cells is highly sensitive to movements makes a lot of sense, at least from an evolutionary point of view. After all, to react properly and timely to the environment, moving objects should be discovered promptly.

End-stopped cells are further specialized in that they are sensitive to the length of the stimulus. They respond much more strongly if the slit of light ends or changes direction within their receptive field. Thus, they respond best to corners and curvature.

So far information from the two eyes was treated separately, even though one cortical hemisphere receives information from both eyes. As photogrammetrist we are professionally interested in stereopsis. The corpus callosum is the site of stereovision. Here, binocular cells are found that respond to depth. Some of these cells only fire if the stimulus is roughly as far away as the distance on which the two eyes are focused (zero parallax). Other cells evoke a brisk barrage of impulses if the stimulus is nearer or farther away from the fixation point. Another characteristic feature of these disparity-tuned cells is that they are also orientation and movement sensitive. As one would expect, they do not respond at all if only one eye is stimulated. Though disparity-tuned cells undoubtedly contribute to stereovision they are just a partial explanation of how we perceive depth. One should bear in mind that stereopsis is only one of several depth cues.

Let us interrupt our journey through the visual system for a moment and recapitulate. What reaches the brain is not an image, but information about changes in the scene, e.g., light intensity differences, their orientation, and movement. The specialization of cells and the complexity of their receptive fields increase. How far will this specialization go? After cells were discovered in the visual area of a monkey that responded to the shape of paws, the notion of a grandmother cell arose. Is there a cell that would respond to grandmother's face?

2.2 Visual Perception

Visual perception is the ability of humans to organize and interpret visual sensory information. The psychology of human visual perception was dominated in the late 19th century by associationism. It was thought that perception could be explained by associating simple sensations. This was precisely what the Gestalt psychologists attacked most, for their basic tenet was that "the whole is more than the sum of its parts". They argued that the form and structure of sensations and their interrelationships should be taken into account. The Gestaltists thought that this synergism is accomplished by magnetic force fields between brain events. The Gestalt psychology has fallen into disrepute, mainly because no evidence was found for the force-fields in the brain.

Cognitive psychology adopts a more information theoretical approach where computer models of perceptual processes are legitimate goals for establishing psychological theories. This, together with a more quantitative ap-

proach in research, paves the way for "computational perception", results that can be converted to algorithms.

Perceptual organization

The neurophysiological approach to vision left us with the image decomposed into simple local features, such as edges, corners and some depth information. Such low-level descriptions must be organized into larger perceptual structures. Perceptual organization is the first process of perception (Rock, 1975). It detects groupings and structures in images which in turn are believed to be the input for object recognition and image understanding.

The following are examples of a set of criteria for grouping the image and finding associations. Most of these principles have been advocated by the Gestalt psychologists and are known as the Gestalt laws of organization. *Proximity* groups local features together which are close together. Depth is a very strong for proximity. Things with similar disparity values are grouped together and perceived as belonging to the same surface. *Similarity* groups similar features together. Similarity can override proximity. *Common fate* groups things together which appear to move together. It can be demonstrated by generating randomly distributed dots and superimposing a copy with a slight shift or rotation. The shift or rotation is clearly perceived. Another Gestalt law is *good continuation* which emphasizes smooth continuity over abrupt changes. *Closure* emphasizes a preference for closed figures and *symmetry* groups symmetrical features together. *Figure ground separation* is quite a strong perceptual organization process.

In reality, grouping processes work concurrently on the same image. Two (or more) processes yielding the same interpretation results in a more salient perception. McCafferty and Fryer (1987) showed that a very strong and stable perception results from combining stereo with figure-ground separation.

Other perceptual processes

Here, we mention some other powerful perceptual processes which could be used in computational vision.

Filling in or completion is responsible for us to not perceive the world as a patchwork of edges and blobs (as might be concluded from the neurophysiological discussion about vision). A very illustrative example is the blind spot. Close one eye and fix a point with the open eye. Move a pencil with one hand so that it crosses the visual field. When the pencil is imaged at the blind spot, it disappears, as expected. However, you are not left with a black spot; rather the hole in the retinal image is covered (filled in) by the surrounding background. Filling in appears to belong to a more general perceptual process called surface interpolation (Ramachandran, 1992).

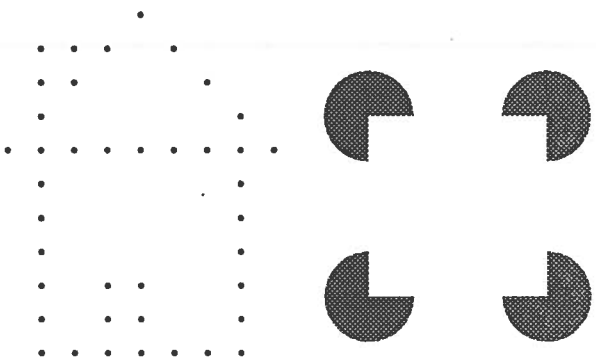


Fig. 2(a): Example for virtual lines. Fig. 2(b) demonstrates the phenomenon of illusionary contours. The figure is perceived as a square and not as four partial circles.

Virtual lines are imaginary lines, linking nearby tokens. Fig. 2(a) is an example. A similar phenomenon are *illusionary contours*, investigated by Kanizsa (1979). In Fig. 2(b) we perceive the structure of a square. The four corners are lying on circles. Another (unlikely) interpretation of this figure is four partial circles.

Texture is a very important but not well understood perceptual process. Texture is strongly related to surfaces. Slowly changing texture patterns give a strong perception for surface normals. Julesz studied texture segmentation intensively. He concludes that textured regions cannot be segregated if their first and second order statistics are identical. In Julesz and Bergen (1983) the notion of textons is introduced. The authors claim that they play a

complementary role in human texture segregation.

2.3 Marr's Theory about Vision

The physiological approach to vision answered the question: what happens where? How something happens cannot be fully explained unless the cell's behavior can be described by a complete wiring diagram. For answering the question why single cells respond they way they do, a broader view must be adopted. As Marr put it

> ... trying to understand perception by studying only neurons is like trying to understand bird flight by studying only feathers: It just cannot be done. In order to understand bird flight, we have to understand aerodynamics; only then do the structure of feathers and the different shapes of bird's wings make sense. (Marr, 1982, p. 27).

Marr's theory about vision has a strong information processing underpinning. He argues for understanding an information process – vision – at three different levels.

computational theory specifies what the visual system must do. It answers the question about the purpose of the computation and the strategy for solutions.

representation and algorithm investigates the representation of input and output and the algorithm that transform one into the other.

hardware implementation answers the question how the representation and the algorithm can be physically implemented by neurons.

The tenet of Marr's theory is that the shapes and positions of things can be made explicit from images without knowing what these things are and what role they play. However, this cannot be accomplished in one step, rather in a sequence of representations designed to facilitate the subsequent construction of physical properties of objects. The three main steps are briefly discussed.

Primal sketch

The purpose of the primal sketch is to make intensity changes in the image explicit. Intensity changes, or edges for short, are an important physical property of objects. In the real world edges occur over a wide range of spatial extents. A sharp edge, for example, is manifest within a small area, comprising a few pixels only. On the other hand, a fuzzy edge can only be detected by looking at a much larger area. Marr and Hildreth (1980) propose a sequence of LoG operators to detect edges at various scales. The LoG operator (Laplacian of a Gaussian) is obtained by taking the second derivative of a Gaussian filter. The Laplacian (∇^2) is particularly suited because it is direction independent. By varying the standard deviation σ of the Gaussian, the desired sequence, also called multi channel implementation, is obtained. Obviously, the parameter σ determines the spatial extent within which an edge is detected. Edges are identical with the zero-crossing contours that result from intersecting the convolution surface with a plane, whose convolution value is zero. Thus, a sharp edge is obtained by convolving the image with a small σ (fine channel), and fuzzy edges result from coarser channels.

There is much evidence that the human visual system performs the same operations. Cells exist in the cortex that respond to different spatial frequencies. Spatial information is processed in each part of the visual field by five independent channels (Wilson and Bergen, 1978). Actually, the LoG operator is approximated by the difference of two Gaussians of slightly different σ. The two coarser channels have transient properties, responding to fluctuating patterns, while the finer channels respond to stationary objects. The finest channel is related to acute vision.

The primal sketch is more than just an agglomeration of zero-crossings. Perceptual processes operate on the image as well as on the edges, resulting in a curvilinear organization, virtual lines and groupings. Zero-crossings from different channels are combined, governed by the rule that edges in different channels are localized in space.

2.5-D sketch

Its purpose is to make explicit the orientation and depth of visible surfaces as well as discontinuities. The name of this sketch derives from the assumption that it captures a great deal about the relative depths and surface orientations, and local changes and discontinuities, but some aspects are more accurately represented than others.

> Very locally we can easily say from motion or stereopsis information whether one point is in front of another. But if we try to compare the distances to two surfaces that lie in different parts of the visual field, we do very poorly and can do this much less accurately than we can compare their surface orientations. (Marr, 1982, p. 282)

The 2.5-D sketch is built up from the primal sketch, augmented with information from stereopsis, texture, analysis of motion, and shading. The surface orientation is much more accurate than depth. Only local changes in depth have a comparable accuracy. Discontinuities in depth may arise from stereopsis and occlusion. Occlusion may be specified by the presence of occluded edges in the primal sketch, or by analyzing motion patterns.

The 2.5-D sketch is represented as a set of primitives, depicted as "needles". The length of each needle describes the degree of tilt of that part of the surface, while the orientation of a needle reflects the direction of slant. The distance from the viewer is represented by a scalar quantity.

Interpolation procedures are invoked in areas of insufficient information. In areas of low contrast, no edges are present and therefore no depth information. The missing depth information is interpolated from surrounding areas where contrast is present. Another example for an interpolation process are illusory contours (see Fig. 2b).

The 2.5-D sketch is the end product of early vision processes, solely derived from images, without support from late vision or knowledge of the scene. The early vision processes are modular, they work parallel and independent from one another. The segmentation problem is implicitly solved by making explicit the discontinuities between different surfaces.

3-D Model representation

The purpose of this last step is to describe shapes and their spatial organization in object centered coordinate system. Marr and Nishihara (1978) suggest a modular organization of shape descriptions in a coordinate frame which is determined by the shape itself (canonical coordinate frame). The modular organization allows a description that is independent on the degree of details an object is described.

The theory is restricted to a set of generalized cones. A generalized cone is obtained by moving a cross section of constant shape but variable size along an axis. A vase is a good example of a generalized cone. An object may consist of several generalized cones, each with its own axis. All axes of one object form the component axes of that object.

A library of 3-D model descriptions at different levels of specificity is generated for objects that may possibly appear in a scene. The same 3-D model description must be derived from the image. Object recognition then entails to compare these descriptions with the library.

Occluding contours of an image provide strong clues for finding the axes of generalized cones. Occluding contours are the silhouettes of objects. Even though most silhouettes are ambiguous, humans interpret them in a particular way. Marr hypothesizes that additional information is used to constrain the perception of 3-D shapes to silhouettes as we see them. These constraints are general and do not require a priori knowledge of the scene.

3 MACHINE VISION

3.1 Introduction

From time immemorial people dreamed of creating machines that would exhibit mental abilities. With the invention of computers, researchers in the field of artificial intelligence

(AI) pursue this dream to endow computers with information processing capabilities similar to those of humans. Richie (1985) defines AI as " the study of how to make computers do things at which, at the moment, people are better". Vision is not only our most impressive sense but also the most intensively studied sense in AI.

By and large, machine vision pursues the same goal as human vision: generate descriptions about the scene from images. The descriptions must be explicit and meaningful so as to allow other system components to carry out a task. In that aspect, machine vision is part of an entire system that interacts with the environment, say a robot. Consequently, tasks such as decision making, planning, executing decisions, are not part of machine vision. By the way, the terms *computer vision* and *machine vision* are used interchangeably.

Machine vision is a relatively new and rapidly changing field. Many of the essential concepts have only evolved during the last ten years. The purpose of this chapter is to elucidate the most important concepts and to elaborate on the major issues. Even though machine vision is now a field in its own right it is related to other areas, such as psychology, computer graphics, pattern recognition and image processing. In fact, significant progress has been made, and will be made, when an interdisciplinary approach is adopted. Take Marr's theory of vision as an example. It is actually the combination of research results in neurophysiology, psychophysics, perception, computer science and signal processing.

Even though our knowledge of the human visual system is only fragmentary, we know that it is very complex. Machine vision, therefore, is a non trivial task. Not surprisingly then, no general purpose vision system exists today and will not exist in the foreseeable future. The lack of rapid success, as enthusiastically predicted thirty years ago, led some AI researchers to a rather pessimistic assessment. In their view, machine vision is so ill-defined and underconstrained that no general solution exists. As Barrow put it:

> Despite considerable progress in recent years, our understanding of the principles underlying visual perception remains primitive. Attempts to construct computer models for the interpretation of arbitrary scenes have resulted in such poor performance, limited range of abilities, and inflexibility that, were it not for the human existence proof, we might have been tempted long ago to conclude that high-performance, general-purpose vision is impossible. (Barrow, 1978).

Nevertheless, progress has been made, mainly in industrial applications, where the environment, such as lighting conditions, can be better controlled.

3.2 Machine Vision Paradigm

Marr's theory of vision gave rise to the most advanced and widely accepted paradigm of machine vision. Fig. 3 depicts the building blocks.

Usually, at the outset is a raw image. We also include image formation, a point forcefully advocated by Horn (see Horn, 1986) and now accepted by many vision researchers. After all, machine vision may be viewed as the inverse process of image formation. Thus it makes only sense to obtain a thorough understanding of image formation.

The primal sketch is the result of edge detection. Edges are likely to have been caused by structures in the scene, such as object boundaries, markings and surface discontinuities. The unorganized edge fragments, bars and blobs are grouped into higher-level tokens, which are now processed by the independent modules *stereopsis, shading, motion, texture* to yield the 2.5-D sketch.

The 2.5-D sketch contains fewer data than the raw image, but more important, it is more explicit. An edge could be an object boundary or a shadow; a single pixel can be everything. Depth and 3-D shape information is particularly important. Shape and depth information is obtained independently from stereo, shading, motion and texture processes, also called shape-from-X processes. Note that the 2.5-D sketch is purely obtained from the raw images. It is the result of bottom-up processes, also referred to as *early vision.*

The 2.5-D sketch is the transition from image space to object space. Subsequent processes, termed *late vision*, are scene oriented rather than image oriented. Extracted features are grouped together, segmented and eventually parameterized. If the application of the vision system is object recognition then a data base with models of objects is generated. The parameterized features are now matched with the object library.

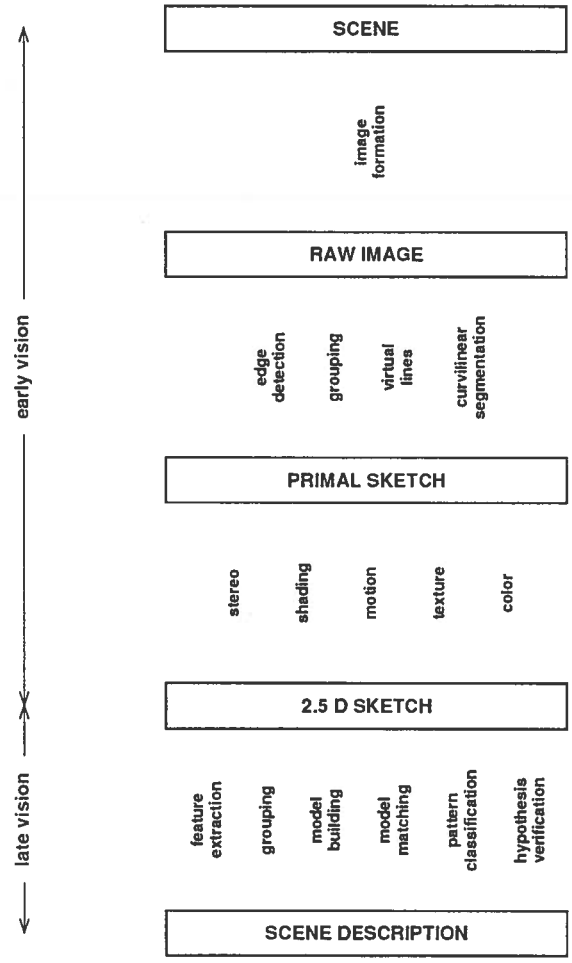


Fig. 3: Overview of machine vision paradigm.

3.3 Key concepts, key issues

In this section we summarize the most important concepts of machine vision that have been proven particularly useful.

Hierarchical approach

The main goal of machine vision is to derive meaningful descriptions of objects from images. Useful information that may lead to scene descriptions is not explicitly available in raw images, however. Physical phenomena, such as discontinuities of surfaces, depth, reflectance and illumination, are confound in the intensity values of the pixels by the imaging process. The raw image cannot be used directly to interpret the image function. The problem is to find intermediate descriptions. Local extrema and its derivatives are appropriate primitives because they are quite often related to object space events. For example, an edge may correspond to an object boundary.

Physical phenomena vastly differ in size and scale. Therefore, scale is one of the problems of obtaining discontinuities in gray levels of an image. For one, it may not be desirable to find all discontinuities. Noise and unnecessary details should be suppressed. This can be achieved by smoothing the raw image, for example by convolving it with a Gaussian filter. The size σ of the Gaussian filter determines the scale in which the intensity discontinuities are represented. We obtain an infinite number of representations if σ is varied continuously. This representation is known as *scale space image*, a term introduced by Witkin (1982). An attractive property of the Gaussian over other convolution kernels is that no new object space events occur when σ is increased. Descriptions found at a coarse scale also appear on finer scales, though not exactly at the same location.

The concept of representing the descriptive primitives in a continuum of scale levels is appealing because it circumvents the problem of specifying at what scale a particular object space event occurs. However, most practical implementations are discrete. That is, the scale space is discretized. The result of smoothing a raw image with a finite set of smoothing operators, e.g. Gaussian with discrete σ values, is called *image pyramid*. Every level of the pyramid corresponds to a different resolution of the raw image, hence the term *multi resolution*. A fundamental problem of the discrete scale space is tracking the descriptive primitives from one level to the next. The smoothing operation displaces the primitives. Tracking features through the discrete scale space is difficult.

The hierarchical approach, also termed coarse-to-fine approach, is a result of Marr's theory about vision. This concept is now also employed in digital photogrammetry (see, e.g. Li, 1989).

Vision is an ill-posed problem

As is evident from the paradigm, early vision processes are aimed toward reconstructing the scene from images. It is the inverse problem of optics. Like many inverse problems, early vision is ill-posed. Take edge detection, for example. It requires numerical differentiation and that is clearly ill-posed. Hadamard defined a mathematical problem to be well-posed when its solution exists, is unique, and is robust against noise.

Regularization theories have been proposed to solve ill-posed problems (e.g. Tichonov and Arsenin, 1977). Poggio *et al.* (1985) and Terzopoulos (1986) employ regularization to vision problems. The basic notion is to restrict the space of acceptable solutions by choosing a function that minimizes an appropriate functional. One method for finding x from $y = Ax$ is to minimize $\|Ax-y\|^2+\lambda\|Px\|^2$ where Px is a stabilizing functional and λ is the regularization parameter. In essence, an ill-posed vision process is regularized by imposing physically plausible constraints, for example, smoothness.

Despite the mathematical elegance of regularization theory, it remains a partial solution only. There are problems of applying the theory if the quantity to be computed is a discontinuous function. The world we want to reconstruct from images is full of discontinuities. Another problem is the degree of smoothness for the unknown function to be recovered, for example, when interpolating surfaces. This and other problems must be solved before regularization theory can serve as a general method for solving ill-posed vision processes.

Of course, vision processes have been solved before the framework of regularization theory became available. This was accomplished by introducing assumptions and constraints, sometimes in a rather ad hoc fashion. The secret is to strike a fine balance between constraining the problem just enough to obtain a solution and at the same time remain as general as possible. Quite frequently, assumptions are too restrictive and algorithms fail when applied to slightly different scenes.

Integrating different visual processes

The motivation for combining different visual processes is to convert early vision to a well-posed problem. In that regard it is an alternative approach to regularization.

The individual shape-from-X processes are computationally underconstrained and in themselves not robust. By combining different processes the parameters will be more constrained and the solution becomes more robust and unique. It is important to realize that it is the mere combination of processes which makes the solution more stable and *not* the introduction of additional constraints imposed on individual processes.

The combination (or integration) can be realized on two levels. The results of individual shape-from-X processes may be considered as contributions (clues) toward the 3-D shape description of surfaces. The integration can be performed within the framework of testing hypothesis.

Another approach is to integrate information from different cues in one process. Grimson (1984) proposes to combine stereo and shading; Brooks and Horn (1985) put forward a theory to solve the shape from shading problem by simultaneously computing the illuminant direction. The concept of adding the radiometric model to the geometric model in least-squares matching, suggested by Helava (1988), Wrobel (1987), and Ebner and Heipke (1988) falls into the category of combining stereo with shading.

Transition from early vision to late vision

The 2.5-D sketch in the machine vision paradigm (see Fig. 3) is the interface between early and late vision. Early vision processes, such as edge detection, grouping edge segments, stereopsis, texture, motion and color, are data-driven processes with the goal to derive physical properties of the object space (scene) from implicit information in the images. The most important information in the

2.5-D sketch is shape, depth and boundaries of surfaces, where boundaries are manifest in discontinuities of depth or surface normals.

The tenet of Marr's theory about vision is that this information can be captured from images only, without a priori knowledge about objects in the scene. Surface reconstruction is the result of early vision. Random dot stereograms prove that the human visual system reconstructs surfaces from parallax information only. If Fig. 4 is viewed under a stereoscope, the center is clearly perceived as a surface floating above the surrounding background.

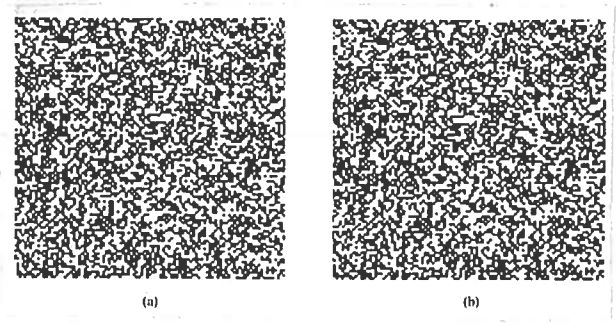


Fig. 4: Random dot stereogram. When viewed stereoscopically the center square is floating above the background.

While early vision is associated with reconstructing visible surfaces, late vision is primarily concerned with interpreting this information by way of symbol processing. The notion is to segment (group) the 2.5-D sketch into higher-level tokens and associate a meaning to it. The vague term "meaning" is chosen purposely for it strongly indicates that late vision is application dependent. The further we move from the 2.5-D sketch the more application dependent late vision becomes. An example may demonstrate the case. Suppose you are seeing a tiger ready to jump in your direction. The 2.5-D sketch, containing depth, shape and textural information, will be interpreted in a way that depends on your situation. If you are in a secured position and interested in studying wildlife, the interpretation will be radically different from the case where you are not protected and perhaps caught by surprise with the tiger's appearance. Thus, late vision is goal-driven.

Late vision processes depend on knowledge about the scene (domain knowledge) and general knowledge (world knowledge). An example for general, high-level knowledge is the fact that a shadow cast on a smooth surface does not cause people to perceive two different surfaces, despite the fact that the shadow is a very strong edge. Knowledge of this kind is readily accessible to people and part of the reason why human vision is so flexible and capable to deal with a whole variety of different scenes. Machine vision systems on the other hand lack this flexibility and versatility.

A crucial issue in any vision system is the proper interaction between early and late vision. Where exactly should data-driven processes end and goal-driven modules begin? How far should scene independency be preserved? How can knowledge be brought in without limiting the scope of applications?

Representations and data structures

A typical input to a machine vision system are images; a typical output is a symbolic description of the scene. There is a tremendous difference in the way the input and output are represented. It is not conceivable to bridge this wide gap in one step. Thus, machine vision is confronted with the design of suitable intermediate representations and algorithms that allow to derive one representation from another. Fig. 5 illustrates some of these intermediate representations. The figure is from Ballard and Brown (1982). The authors give an excellent exposition of representations, data structures and algorithms.

The raw image, the primal sketch and 2.5-D sketch are all iconic (image-like) representations. Image processing is characterized by taking an image as input and produce another image as output. Many early vision processes are basic image processing operations. For example, edge detection works on the raw image and produces an edge image which may be represented as a binary image.

An important characteristic of the human visual system is its remarkable ability to organize the image into meaningful regions to facilitate subsequent steps of image understanding. This ability is called perceptual organization. Its counterpart in the machine vision paradigm are the grouping processes (or segmentation).

Grouping occurs at all stages. Edge fragments in the primal sketch are grouped to edge segments in the 2.5-D sketch. Grouping processes that operate on the 2.5-D sketch are likely to be guided by general and domain knowledge. It is helpful to know if there are man-made objects in a scene which may give rise to straight lines, curves and rectangular polygons. Contrast this with an X-ray image of a chest, where segmentation follows other principles.

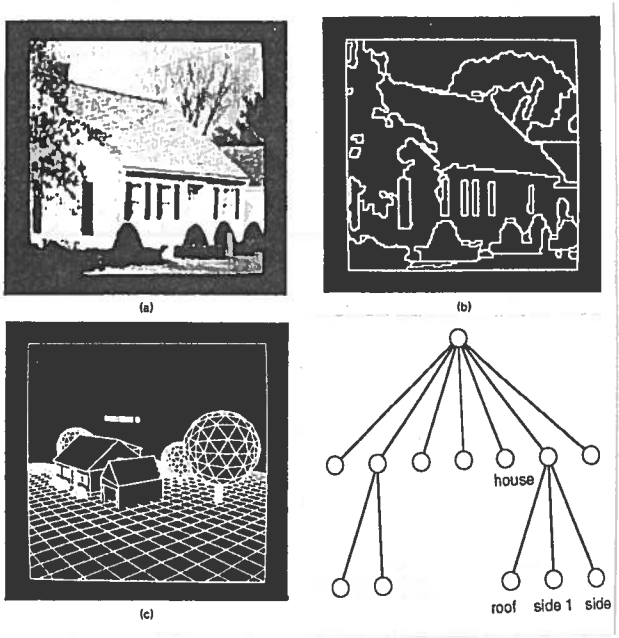


Fig. 5: Examples of different representations used in computer vision. (a) Iconic; (b) Segmented; (c) Geometric; (d) Relational. The figure is from Ballard and Brown (1982).

Another important facility of the human visual system is shape recognition. We observed what crucial role shape information plays in Marr's theory about vision. But how do we represent shape? How is it that with a few strokes of the pen we recognize a person's face? Shape recognition and representation in machine vision is still in its infancy. Algorithms and data structures are often derived in an ad hoc fashion. Not surprisingly, the scope of their applications is severely limited. One way to approach the problem is to describe shape as 2-D and 3-D geometrical structures. This is quite successful when dealing with scenes consisting of objects with regular shapes.

The success of late vision greatly depends on how well general and domain knowledge is represented. As mentioned earlier, general knowledge serves the purpose of reducing the solution space to a few (ideally one) solution that conforms with how we perceive the world. Most machine vision systems deal with general knowledge in form of constraints. Our expectation that surfaces appear smooth almost everywhere is accounted for by various smoothness constraints.

Ideally, domain knowledge is brought in only in late vision. From the rapidly developing field of knowledge engineering, methods can be employed to capture (encapsulate), represent and manipulate knowledge. There are many ways to represent knowledge, for example semantic nets, frames, schemas, predicate calculus, neural nets etc. Probably the most popular representation is the rule-based method. Here, knowledge is expressed in the "if-then" format.

4 MACHINE VISION AND CLOSE-RANGE PHOTOGRAMMETRY

In this section we briefly sketch what machine vision and close-range photogrammetry have in common and what separates them. Suggestions are made on how both disciplines could benefit from each other.

Close-range photogrammetry evolved as a special subfield of photogrammetry over the last thirty years. Its main application areas are architecture, industry and medicine. Only a few practicing photogrammetrists employ close-range photogrammetry. This is a clear indication that the market is rather small and that special know-how and equipment is required.

The increasing availability of digital cameras and digital image processing systems brings a radical change. This is particularly true for applications where the elapsed time from measurements to results is critical. Such is the case in many industrial applications. Here, we would ideally want to process the measurements in real-time. Much progress has been made in digital close-range photogrammetry. Systems are no longer only in use in research laboratories but also in production en-

vironment (see, e.g. Beyer, 1992; Haggrén and Pekkinen, 1992).

Despite of the lack of a general vision theory, tremendous progress has been made in machine vision. Though no general-purpose vision system is available (and will most likely not become available in the foreseeable future), systems which successfully deal with special applications have been reported. For example, a successful application is a vision system that feeds a robot with essential information so it can pick up a part out of a bin or from a conveyor belt.

4.1 Common Characteristics

Image formation

For both disciplines, a thorough understanding of image formation is essential. The automatic measurement, interpretation and analysis of images is greatly facilitated by the knowledge on how the images are formed.

Traditionally, photogrammetrists are particularly concerned with the geometric relationship between image and object space. The camera geometry is understood and deviations from the perspective projection are well handled, be it in form of calibration procedures or by extending the mathematical model to include additional parameters.

Since in conventional photogrammetry human operators interpret the image, not much effort has been spent to understand the radiometric aspect of image formation. This subject has been intensively investigated by Horn (see e.g. Horn, 1978; Horn, 1986). Of course, remote sensing can greatly contribute to a better understanding of image formation in aerial applications.

Early vision

Many early vision processes are identical in both disciplines. Examples include basic image processing tasks, such as image enhancement, image storage, image compression, etc. A fundamental problem in both disciplines is finding conjugate points in stereo images (correspondence problem). Related to the correspondence problem is the determination of surfaces.

4.2 Differences

Among the many differences which separate the two fields of close-range photogrammetry and machine vision, we elaborate only on the differences in applications and background.

Applications

The main areas of close-range photogrammetry applications are architecture, biostereometrics, and industrial engineering. Industrial applications are mainly in metrology. Digital close-range photogrammetry systems are most successful when measuring well targeted points in a controlled environment. Emphasis is placed on accuracy and reliability; the points are typically determined to sub-pixel accuracy.

Most applications require results in a 3-D object coordinate system. The predominant method for obtaining 3-D information is stereopsis. Depending on the application, the points in object space are now analyzed and geometrical properties, such as distances, diameters, slopes, surfaces, are derived. The determination of 3-D points (matching) is often combined with the computation of geometrical properties. Yet another step may include the comparison between measured quantities and design data (e.g. from a CAD system). An example are deformation analysis.

Most machine vision applications require various degrees of recognition and interpretation capabilities. Such is the case in process control, parts identification, robot guidance, planning, navigation, and obstacle avoidance for autonomous vehicles. Here, the emphasis is not on utmost accuracy but on extracting and grouping features, on image segmentation, on associating meaning to groups of symbols, on geometric modeling, matching and hypothesis verification. In contrast to close-range photogrammetry systems, machine vision systems deal with late vision problems.

Background

Close-range photogrammetry is a rather small community of specialists from photogrammetry, geodesy, and surveying. As pointed out

earlier, machine vision is a subfield of artificial intelligence with strong ties to computer science, electrical engineering, and cognitive science. The differences in background, experiences and preferences are quite obvious when it comes to model a vision process, to decide which tools should be used to implement the model, or what quality control criteria should be applied.

Not only is the background of scientists in close-range photogrammetry and machine vision quite different, but also the terminology. Due to the rapid development, both fields are far from having a unified, accepted terminology. Even worse, often times a complete different expression is used for the same thing, leading not only to amusement but to considerable confusion.

5 CONCLUDING REMARKS

In this paper I have described the goal, the basic concepts and issues of machine vision. A strong point was made that someone who is confronted to program a vision task should have a good understanding of the human visual system for it is a measure beyond all bounds.

I have then compared machine vision with digital close-range photogrammetry and indicated what both fields share and where they are different. In conclusion, we realize that both fields are confronted with the same massive problems. Despite many claims that pushed expectations high, few systems are successful and only in narrow applications. Much more research is needed to broaden the applications and to increase the robustness of vision systems.

In order to concentrate efforts and resources, it seems only natural that well proven methods and solutions should be shared. A prerequisite for reaching this desirable goal is a better understanding of each others problems, terminology, methodology and tool set. A way to facilitate this is by joint conferences, workshops and joint projects.

6 REFERENCES

Ballard, D.H., and C.M. Brown, 1982. *Computer vision.* Prentice-Hall, Englewood Cliffs, New Jersey.

Barrow, H., and J.M. Tenenbaum, 1978. Recovering intrinsic scene characteristics from images. In *Computer vision systems.* Academic Press, New York.

Beyer, H., 1992. Digital close-range photogrammetry in industrial measurement. *Int. Archives of Photogrammetry and Remote Sensing*, Congress Washington D.C.

Brooks, M.J., and B.K.P. Horn, 1985. Shape and source from shading. *Proc. Intern. Joint Conf. on Artificial Intelligence*, Los Angeles, pp. 932–936.

Campbell, F.W., and J.G. Robson, 1968. Application of Fourier analysis to the visibility of gratings. *Journal of Physiology*, 197, pp. 551–566.

Ebner, H., and Ch. Heipke, 1988. Integration of digital image matching and object surface reconstruction. *Int. Archives of Photogrammetry and Remote Sensing*, Congress Kyoto, Comm. III, Vol. 27, part B-11, pp. 534–545.

Grimson, W.E.L., 1984. Binocular shading and visual surface reconstruction. *CVGIP*, pp. 18–44.

Haggrén, H., and P. Pekkinen. Stability control of photogrammetric stations. *Int. Archives of Photogrammetry and Remote Sensing*, Congress Washington, D.C.

Helava, U.V., 1988. Object-space least-squares correlation. *Photogrammetric Engineering & Remote Sensing*, Vol.54, no. 6, pp. 711–714.

Horn, B.K.P., 1986. *Robot vision.* The MIT Press, Cambridge, Massachusetts.

Horn, B.K.P., 1977. Understanding image intensities. *Artificial Intelligence*, 8, pp. 201–231.

Hubel, D.H., 1988. *Eye, brain, and vision.* Scientific American Library, New York.

Julesz, B., and J.R. Bergen, 1983. *Textons: the fundamental elements in preattentive vision and perception of texture.* The Bell System Technical Journal, vol. 62, no. 6, pp. 1619–

1645.

Kanizsa, G., 1979. *Organization in vision: Essays on Gestalt Psychology.* Praeger, New York.

Li, M., 1989. *Detection and location of breaklines and discontinuities in stereo image matching.* Phot. Reports No. 54, Royal Institute of Technology, Stockholm.

McCafferty, J.D., and R.J. Fryer, 1987. *Perceptual organization and low level vision.* University of Strathclyde, Department of Computer Science, Technical Report.

Marr, D., 1982. *Vision.* W.H. Freeman and Company, New York.

Marr, D., and E. Hildreth, 1980. Theory of edge detection. *Proc. of the Royal Society of London.* Series B, 207, pp. 187–217.

Marr, D., and H.K. Nishihara, 1978. Representation and recognition of the spatial organisation of three-dimensional shapes. *Proc. of the Royal Society of London.* Series B, 207, pp. 269–294.

Poggio, T., V. Torre and C. Koch, 1985. Computational vision and regularization theory. *Nature*, 317, pp. 214–319.

Ramachandran, V.S., 1992. Blind Spots. *Scientific American*, May 1992, pp. 86–91.

Rich, E., 1983. *Artificial Intelligence.* McGraw-Hill, New York.

Rock, I., 1975. *An introduction to perception.* Macmillan Publishing Company, New York.

Rosenholm, D., 1987. Multi-point matching using the least-squares technique for evaluation of three-dimensional models. *International Archives for Photogrammetry and Remote Sensing*, Vol. 26, Comm. III, Rovaniemi, pp. 573–587.

Terzopoulos, D., 1986. Regularization of inverse problems involving discontinuities. *IEEE, Trans. PAMI*, 8, pp. 413–425.

Tikhonov, A.N., and V.Y. Arsenin, 1977. *Solution of ill-posed problems.* Winston.

Uttal, W.R., 1973. *The psychobiology of sensory coding.* Harper & Row, New York.

Wilson, H.R., and J.R. Bergen, 1979. A four mechanism model for threshold spatial vision. *Vision Research*, 19, pp. 19–32.

Witkin, A.P., 1982. Scale-space filtering. *Proc. 7th Intern. Joint Conference on Artificial Intelligence*, Karlsruhe, pp. 1019–1022.

Wrobel, B., 1987. Facets stereo vision (FAST VISION) - A new approach to computer stereo vision and to digital photogrammetry. *Proc. Fast Processing of Photogrammetric Data*, Interlaken, pp. 231–258.

ALGORITHMS AND SOFTWARE CONCEPTS FOR DIGITAL PHOTOGRAMMETRIC WORKSTATIONS

Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

ABSTRACT

Despite all the progress in digital photogrammetry, there is still a considerable lack of understanding of theories and methods which would allow a substantial increase of automation of softcopy workstations. This paper does not provide such a theory. Its purpose is to raise awareness that the automation problem is one that cannot be solved in a bottom-up fashion by a trial and error approach. We argue that more researchers should investigate the nature of the problem and what has been done in other fields to tackle it. We present a short overview of computer vision followed by a discussion of the surface reconstruction problem and how it is solved in photogrammetry and computer vision. We then touch on the important subjects of perceptual organization and object recognition.

1 INTRODUCTION

Considerable progress has been made in digital photogrammetry during the intercongress period. Several products are operational and available to photogrammetrists: softcopy workstations, programs to produce digital orthophotos and DEMs, systems to precisely determine points in industrial applications, just to mention a few. By operational we mean that the products do not work only in laboratory environments under tight supervision of the research staff.

That digital photogrammetry products become available to users is remarkable in several respects. For one it provides a loop between research, development (production) and users which will result in valuable feedback to research. Another important aspect to consider is that of responsibility. Users expect, quite rightfully, that the products deliver what has been promised so that their investments will eventually pay off. Most, if not all, commercially available systems are first-generation products and thus incomplete and perhaps not yet robust. Apart from the obligation to remedy deficiencies, there will also be a demand for more functionality and a higher degree of automation.

As far as automating photogrammetric processes is concerned we cannot paint an equally rosy picture for only little progress has been made. With automation we mean that the computer performs tasks usually assigned to operators, such as orientation or map compilation. With automation we do not mean that the computer has to perform tasks autonomously, that is, without human intervention. Automatic processes such as feature extraction execute in an interactive environment like softcopy workstations.

What humans solve without conscious effort poses almost unsurmountable problems to computers. What has been solved so far are the easier problems, what remains, in terms of automation, are the hard problems. It is not just a matter of assigning a large enough group of programmers to a project—the crux is that the problem is not sufficiently understood. It is quite a paradoxical situation indeed: on the one hand we have successfully produced maps for decades, but we do not seem to understand how; otherwise it would be possible to instruct a machine to it. In short, we lack a detailed theory about making maps (see also Förstner, 1990).

This paper does not provide such a theory nor does it make any attempt to fill gaps. Its purpose is to raise awareness that the automation problem is one that cannot be solved in a bottom-up fashion by a trial and error approach. We argue that more researchers should investigate the nature of the problem and what has been done in other fields to tackle it. Computer vision is confronted with similar problems and it seems reasonable to analyze its concepts, methods and algorithms. That is by no means to say we should blindly copy whatever has been done in computer vision. But to judge what concepts may apply for solving photogrammetric problems we must develop a reasonable understanding of what problems are solved in computer vision, what approaches have proven successful and on what theoretical underpinnings solutions are explored.

A short overview of computer vision, its paradigm, basic concepts and major issues is provided in the next section. In fact, it is a very concise summary of another paper (Schenk, 1992) which serves as an introduction to the present paper. Rather than touching loosely on all sorts of aspects we have opted to focus on three subjects. *Surface reconstruction* is an interesting topic because it is successfully approached in computer vision and photogrammetry, though by different methods and with quite different motivations. We contrast the approaches and provide some explanations why they are different.

Perceptual organization is an important but neglected subject. It refers to the ability of humans and animals to detect relevant groupings and structures in images which are the input for object recognition and image understanding. Section 4 explains the basic notion and provides examples of computational modules. The third subject in section 5 is *object recognition*, presently a hot topic in computer vision. Model-based object recognition may not seem tremendously important for our applications, except for certain close-range problems. However, a deeper understanding of how to overcome the combinatorial explosion and how to

form and verify hypotheses is useful.

2 COMPUTER VISION PARADIGM

In this section we provide a concise summary of the computer vision paradigm. More detailed explanations including an exposure to the human visual system is provided in (Schenk, 1992).

The most advanced and widely accepted paradigm of computer vision is based on Marr's theory about vision (Marr, 1982). His theory has a strong information processing underpinning. He argues for understanding an information process—vision—at three different levels.

computational theory specifies what the visual system must do. It answers the question about the purpose of the computation and the strategy for solutions.

representation and algorithm investigates the representation of input and output and the algorithm that transform one into the other.

hardware implementation answers the question of how the representation and the algorithm can be physically implemented by neurons.

The tenet of Marr's theory is that the shapes and positions of things can be made explicit from images without knowing what these things are and what role they play. However, this cannot be accomplished in one step, rather in a sequence of representations designed to facilitate the subsequent construction of physical properties of objects. The three main steps are generating the primal sketch, the 2.5-D sketch and representing 3-D models.

By and large, computer vision pursues the same goal as human vision: generate descriptions about the scene from images. The descriptions must be explicit and meaningful so as to allow other system components to carry out a task. In that aspect, computer vision is part of an entire system that interacts with the environment, say a robot. Consequently, tasks such as decision making, planning, executing decisions, are not part of computer vision.

Usually, the paradigm (see Fig. 1) begins with a raw image. We also include image formation, a point forcefully advocated by Horn (1986) and now accepted by many vision researchers. After all, machine vision may be viewed as the inverse process of image formation. Thus it only makes sense to obtain a thorough understanding of image formation.

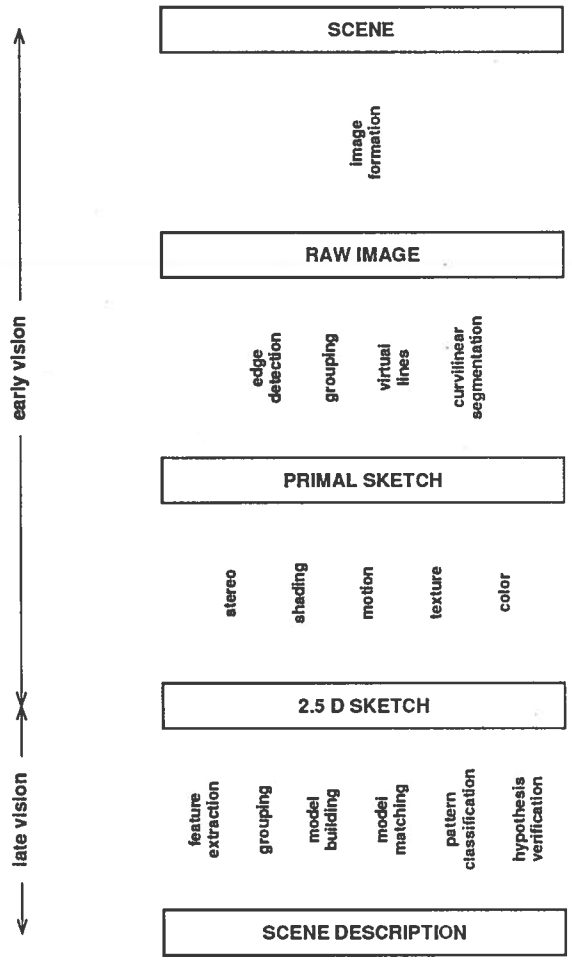


Fig. 1: Paradigm overview of computer vision.

The primal sketch is the result of edge detection. Edges are likely to have been caused by structures in the scene, such as object boundaries, markings and surface discontinuities. The unorganized edge fragments, bars and blobs are grouped into higher-level tokens, which are now processed by the independent modules *stereopsis, shading, motion, texture* to yield the 2.5-D sketch.

The 2.5-D sketch contains less data than the raw image, but more important, it is more explicit. An edge could be an object boundary or a shadow; a single pixel can be everything. Depth and 3-D shape information is particularly important. Shape and depth information is obtained independently from stereo, shading, motion and texture processes, also called shape-from-X processes. Note that the 2.5-D sketch is obtained purely from the raw images. It is the result of bottom-up processes, also referred to as *early vision.*

The 2.5-D sketch is the transition from image space to object space. Subsequent processes, termed *late vision*, are scene oriented rather than image oriented. Extracted features are grouped together, segmented and eventually parameterized. If the application of the vision system is object recognition then a data base with models of objects is generated. The parameterized features are now matched with the object library.

3 SURFACE RECONSTRUCTION

Surface reconstruction is an interesting example to point out how differently photogrammetry and computer vision have approached the problem. The subject has been intensively researched in both fields, ever since computers became available. There is a fundamental difference in the purpose of reconstructing surfaces which may, at least partially, explain the different approaches.

In photogrammetry surface reconstruction is equated with the problem of generating DEMs automatically. Here, the goal is to compute as many conjugate points as accurately as possible. Usually, the DEM, or the surface for that purpose, is a product in its own right. In computer vision, however, surfaces are primarily reconstructed for the purpose of guiding subsequent vision processes, such as object recognition. The goal is not to represent the surface as accurately and densely as possible, but as explicitly as possible. Rather than dealing with a cloud of unrelated 3-D points, it is preferred to segment the surface into piecewise continuous patches, which are related to surfaces of objects.

The surface reconstruction problem comprises the following four tasks.

1. Select a point in one of the two images of a stereopair.
2. Find the conjugate point in the other image.
3. Compute the 3-D position of the conjugate points.
4. Interpolate (densify) the surface.

The second and fourth tasks are hard problems. Task 1 may appear trivial at first glance. However, should any arbitrary pixel be selected, perhaps all of them, or are certain locations more interesting than others? The answer depends largely on the application (purpose) of surface reconstruction. Task 3 is trivial and not discussed any further.

3.1 Finding conjugate points

A few remarks about terminology seem appropriate here. Since correlation was the method of choice for more than twenty years, the term *correlation* was synonymous with finding conjugate points. More recently the term *image matching* has come to stay in photogrammetry but in computer vision the same process is sometimes called the *correspondence problem.*

Strategic considerations

In this paper we use "strategy" for the general approach to solve a problem while "method" refers more specifically to an algorithm. The *matching strategy* of a surface reconstruction system explains for example how the ill-posed character is dealt with, how the combinatorial explosion is kept under control, what quality control measures are taken, etc. Two different *matching methods* are available. *Area-based* matching is the preferred method in photogrammetry, *feature-based* matching in computer vision. It is worth pointing out that the success of a surface reconstruction system depends on more than the matching method. Other strategic considerations are equally important.

A crucial question is how the ill-posed nature of the surface reconstruction problem is addressed. The approaches range from utter ignorance to sophisticated regularization theoretic approaches. Whichever approach is chosen, the problem remains unsolved: no universal system exists that reconstructs surfaces under all kinds of different conditions (e.g., surface properties, lighting conditions, camera positions). Of course our applications in photogrammetry inherently limit the range of conditions with which we have to deal. Take aerial applications, for example, where digital imagery is obtained by scanning aerial photographs. These are of high quality and within a small pose range.

Various assumptions, restrictions and constraints are traditional methods to make surface reconstruction well-posed. This works well but the dilemma is to find just the right balance between a general solution and a very restricted one. The formulation of the matching process as a least-squares problem offers an elegant possibility to include various constraints and treat them simultaneously with the matching. The computer vision paradigm strongly suggests deriving surfaces from different depth cues by integrating different shapes from X-processes. How to combine the different depth cues is largely an unsolved problem.

The *hierarchical approach* most systems now adopt can also be seen as a measure to make surface reconstruction a well-posed problem. Its chief advantage is to reduce the search space, however. Consider pixel i,j with intensity value $g(i,j)$ in one image. There are many pixels with the same intensity in the other image. Two ways exist to solve this problem unambiguously. One is to take an area with unique properties of pixels which is equally unique in the other image. The other way is to restrict the search area. This is accomplished by exploiting the epipolar geometry of stereopairs and by the hierarchical approach which renders approximations for the surface and thus restricts the search space.

Area-based matching

In area-based matching the unique properties of pixels used to compare two image patches are the gray levels. The size of the image patches depends on their image functions: they must be sufficiently unique so as to allow discrimination with neighboring patches. The more unique the gray level distribution the more reliable is the match. The upper bound depends on the topography of the associated surface patch, at least in systems where the surface is approximated by a plane. Computational complexity is another factor that influences the patch size. Unfortunately, the "optimal" patch size is a heuristic parameter.

Let $n \times n$ be the size of the templet and $m \times n$ the size of the search window. The templet is centered on pixel i'_t, j'_t as defined in task 1. The search window in the other image is centered on location i''_t, j''_t with $i''_t = i'_t$ and $j_t{}'' = j'_t - px$ where px is the average x-parallax. The size of the search window is $m \times n$ pixels, $m - n = 2px''$, with px'' the expected elevation difference. The templet is moved within the search window and for every position a similarity measure is computed. The position which yields the maximum similarity is considered the conjugate point.

Several methods exist to measure the similarity. Probably the most popular method is correlation. Here, the similarity between the gray level distribution in the templet and the correlation window is conveniently expressed by the correlation factor ρ. The correlation window is moved pixel by pixel within the search window, and for every position the correlation factor ρ is computed. The position that yields the maximum correlation factor is chosen as the conjugate point. Obviously, this method depends on the assumption that there will be a uniquely defined maximum in the correlation function.

A variety of operational systems are based on this approach or on modified versions. The chief advantage of this method is the rather simple implementation. Among the disadvantages must be listed the arbitrary definition of the window sizes, the correlation function which may not render a unique maximum, and other problems associated with area-based matching.

In least-squares matching, the gray levels of the templet and the correlation window are considered as observed. Consequently, the following generic observation equation can be formed

$$\mathbf{r} = g(x_t, y_t) - \mathbf{T^r}\left[g\left(\mathbf{T^g}\left(x_c, y_c\right)\right)\right] \qquad (1)$$

with $\mathbf{r}$ the residual vector, dimension n^2, and $g(x_t, y_t)$ the image function of the templet. The gray levels $g(x_c, y_c)$ of the correlation window are modified by a radiometric transformation $\mathbf{T^r}$ and a geometric transformation $\mathbf{T^g}$. The geometric transformation accounts for relief distortions which cause conjugate image patches to appear differently in shape. Usually, an affine transformation is chosen. The radiometric transformation adjusts for differences in the gray levels between the two images.

The popularity of this method stems from the application of well-known principles (least-squares), from the straightforward implementation, and from the potential for high accuracy. On the negative side we note the requirement for good approximations (two pixels), and the neglect of the correlation between observations (gray levels) which results in too favorable error estimations.

A note on terminology. Task 1 of the surface reconstruction problem is concerned with selecting the location of the templet. The problems with choosing any arbitrary position have been recognized. Some systems select interest points as more suitable locations. However, area-based matching is still used. In my opinion it is misleading to call these approaches feature-based as they do not capture the notion of feature-based matching.

Feature-based matching

In signal processing, the raw signals are hardly ever used for interpretation because meaningful events are not directly available. Such is the case with images. The image formation process confounds much useful information into the gray levels of the pixels. An intermediate, more stable description must be found which makes object space events more explicit and thus more robust. Local extrema and their derivatives are appropriate descriptive features because they are often directly related to object space events. For example, an edge in the image may correspond to an object boundary.

Two different types of features are considered useful for the surface reconstruction problem: points and edges.

For satisfying the uniqueness requirement of task 2, points with distinct properties have been proposed (*interest points*). For example, the Förstner interest operator detects points, corners and centers of circular features (Förstner, 1986). Each pixel within a window of, say, 5 × 5 pixels is regarded as an edge element with an orientation derived from the gradient of the gray levels $\nabla^T g(x,y) = [g_x(x,y), g_y(x,y)]$. For example, a corner can be detected at location x_0, y_0 by intersecting all edge elements. Likewise, the center of a circular feature is obtained by intersecting the normals of the edge elements.

Edges are the predominant features used in feature-based matching. Edges in the image refer to discontinuities in the gray levels. Numerous edge operators have been proposed. They may be classified into two types: first derivative and second derivative operators. First derivative operators detect edges as the local maxima of the first derivative, which is approximated by the gradient of the gray values. Second derivative operators detect edges as the *zero-crossings* of the second derivative which is most commonly approximated by the Laplacian of the gray levels.

Because edge operators enhance noise, it is important to smooth the images prior to edge detection. The *Laplacian of the Gaussian (LoG)* operator combines smoothing (with a Gaussian) with the second derivative. The definition is given in equation (2):

$$\nabla^2 G(x,y) = \left[\frac{x^2+y^2}{\sigma^2} - 2\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (2)$$

The size w of the LoG operator refers to width of the central lobe and is related to the parameter σ of the Gaussian by $w = 2\sqrt{2}\sigma$. The smoothing operation has a negative effect in that edges are dislocalized. In general, edge matching does not render the same high accuracy as area-based methods.

The advantage of using edges (zero-crossings) is increased reliability and less stringent approximations. Above all, edges are much more likely to correspond to physical properties of surfaces of objects. We remember the goal of computer vision is to make information explicit. Matched

edges have a much higher level of explicitness compared to arbitrary chosen points. If you are not yet convinced compare Fig. 2. Here the same images have been matched with both matching methods. The figure shows the distribution of matched points.

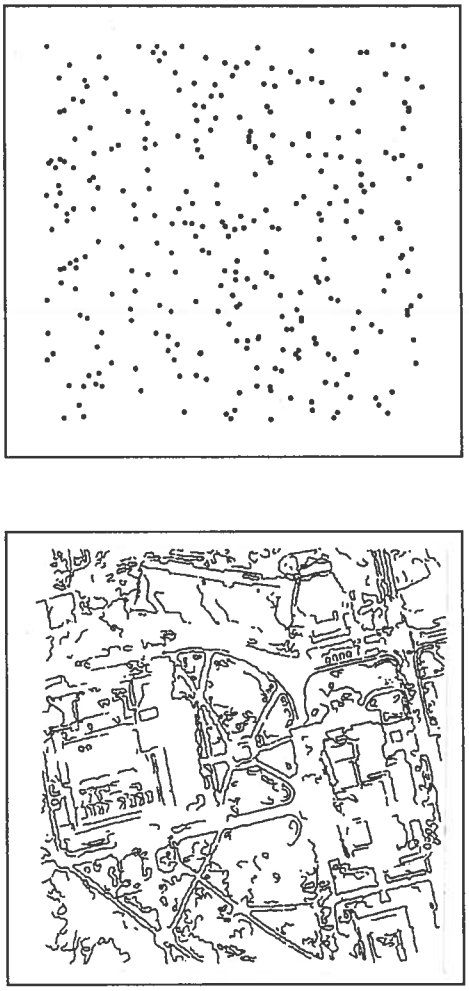


Fig. 2: Distribution of matched points selected with interest operator (top) and matched edges (bottom).

3.2 Surface fitting

The points obtained by task 3 are not evenly distributed and do not completely represent the surface. Even if all pixels were selected in task 1 there would be holes because matching is not always successful. Thus, the 3-D points must be interpolated. We use the term surface fitting as it refers to interpolation as well as to approximation methods.

Lancaster and Salkauskas (1986) define surface fitting as finding a function that agrees with the data points to some extent and behaves reasonably between data points. Surface fitting methods can be classified according to criteria such as goodness of fit, extent of support (local versus global methods) or type of mathematical model (weighted average, polynomials, splines). For a more detailed account we refer the reader to Al-Tahir and Schenk (1992).

In the weighted average method the elevation at a random position is obtained as the weighted average of all data points with the weight inversely proportional to the distance. To reduce the computational complexity local average methods do not consider all data points.

Polynomials have long been used for interpolating surfaces. Their notorious oscillation tendency is overcome by spline functions. A spline is a piecewise polynomial function defined on continuous segments. By imposing the existence of derivatives spline functions become continuous and smooth between segments. A nice property is the predictability of their location: splines lie within the convex hull of data points.

An interesting class of splines are the thin plate splines, derived from nodal basis functions. They are obtained by minimizing the total curvature of a cubic spline. The relationship with thin plates stems from the fact that minimizing the curvature is equal to determining the deflection of a plate due to an external force. This forms the basis for casting the surface fitting problem as an energy minimization problem.

The energy function E to be minimized is constructed from two functionals. The first functional measures the smoothness of the solution S. D is the second functional and it relates to the closeness of the solution. The energy function is the combination, or $E = S + D$. This concept is taken a step further by including discontinuities.

If the surface is broken along discontinuities then a lower state of energy is reached. In the extreme case of breaking up the surface into

small pieces around every data point, the lowest energy state is reached. This would certainly not correspond to visible surfaces in everyday scenes where we observe a great degree of continuity and smoothness. Therefore, breaking the surface into patches must be penalized. Formally, a third term P is introduced and we obtain $E = S + D + P$ where P is a cost function.

The energy function E is nonconvex and minimization is nontrivial. Blake and Zissermann (1987) suggest merging the interpolation function S with the line process P. This solution allows occasional discontinuities in an otherwise continuous environment.

We use this approach in the OSU surface reconstruction system (see Schenk and Toth, 1991) to test hypotheses about breaklines. Breaklines are related to discontinuities in the image function. Since gray level changes are caused by changes in illumination, edges and breaklines are related. By and large breaklines show up as edges. Thus matched edges are hypothetical breaklines. This hypothesis is confirmed or rejected by comparing the breaklines that are detected during the surface fitting process.

4 PERCEPTUAL ORGANIZATION

4.1 Background

Perceptual organization is the ability to derive relevant groupings and structures from an image without prior knowledge of its content. The human visual system is remarkably adept at detecting many different patterns and significant groupings of elements in an otherwise random set of elements. Such grouping processes include clustering, detecting single and parallel lines and curves, and segmenting the image into regions which share similar image characteristics.

Perceptual organization is a prerequisite for object recognition. It must be largely independent of lighting conditions, viewer point and scale. Thus a perceived grouping is an intrinsic property of the image. It must be derived by examining regions since a single gray level of an individual pixel contains almost no perceptual information.

Perceptual organization is not only a late vision or cognitive process. Preattentive vision experiments (response of human visual system to stimuli without focus of attention) indicate that perceptual organization occurs very early on in the visual pathway. Quite often, grouping processes are recursive. The output of one process becomes input to another grouping process, perhaps together with other image information. An example is edge fragments which are clustered to edge elements which in turn are grouped to lines.

Gestalt laws of organization

The school of Gestalt psychology emerged in the 1920s. It emphasized form and structure and their interrelationships. The Gestalt psychologists formulated a number of principles of perceptual organization known as the Gestalt laws of organization. They introduced the notion of "the whole is bigger than the sum of its parts." For example, two dots exhibit a sense of orientation which is not present in a single dot. The Gestalt laws of organization are listed below.

proximity groups local features together which are close together. In Fig. 3a we perceive columns because the vertical spacing of dots is smaller. Depth is a very strong factor for proximity. Things with similar disparity values are grouped together and perceived as belonging to the same surface.

similarity groups similar features together. In Fig. 3b the horizontal spacing of the dots is smaller than the vertical spacing, yet we perceive columns. This is an example where similarity overrides proximity. Fig. 3c is perceived as two separate regions because the orientation of local features is distinctly different in both regions.

common fate groups things together which appear to move together. Fig. 3d demonstrates the case. A copy of randomly generated dots is superimposed on the original, but only after rotating it. We clearly perceive the rotation because points of similar motions are grouped together.

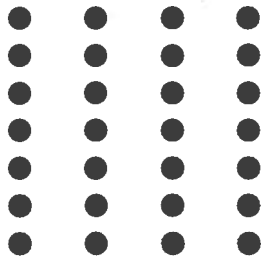


Fig. 3a: Proximity

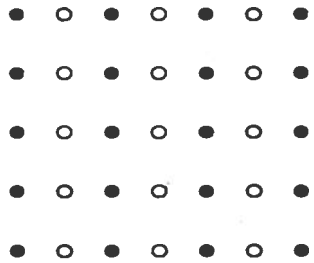


Fig. 3b: Similarity

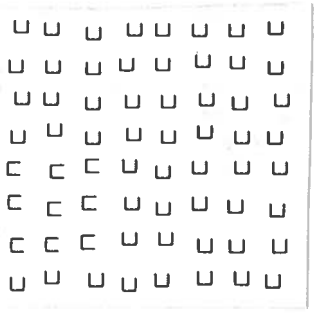


Fig. 3c: Similarity

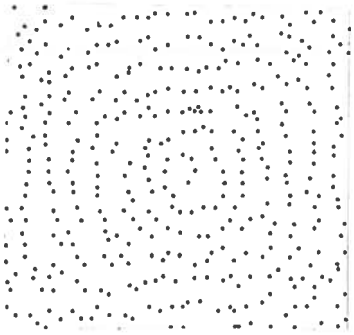


Fig. 3d: Common fate

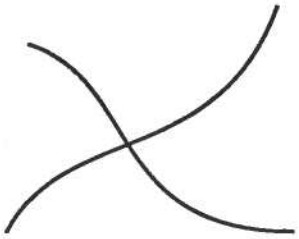


Fig. 3e: Good continuation

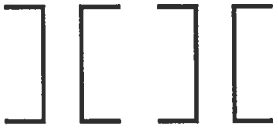


Fig. 3f: Closure

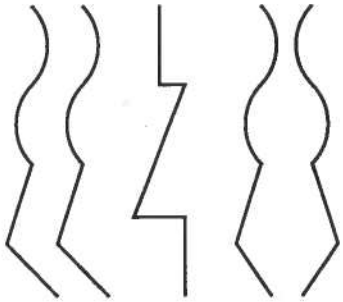


Fig. 3g: Symmetry

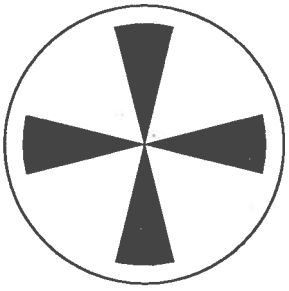


Fig. 3h: Figure/Ground separation

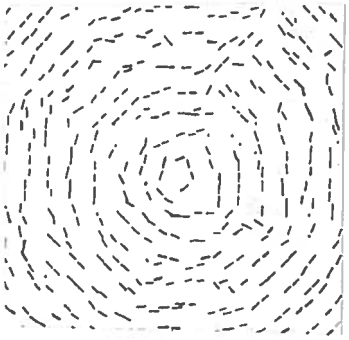


Fig. 4: Virtual lines

good continuation is demonstrated in Fig. 3e. We perceive two smooth lines which cross rather than two V-shaped forms. Smooth continuity is preferred over abrupt changes.

closure emphasizes a preference for closed figures. Fig. 3f is seen as a square and not as a cross.

symmetry groups features together that are similar. In Fig. 3g curves 1 and 2 are grouped together because of symmetry. However, the perception of curves 4 and 5 is stronger because of reflection around the vertical axis. Clearly, curve 3 is not grouped with any other curve.

figure ground separation . The smaller of two areas is seen as figure against background. In Fig. 3h a black propeller is seen against a white background.

Gestalt psychology came into disrepute mainly because it failed to explain how the grouping processes work. No evidence was found for their theory about brain field waves. However, there is no doubt that the Gestalt laws work.

As pointed out earlier, vision as a reconstruction process is ill-posed. One image may have many different interpretations. In order to reduce the solution space, assumptions about the world we see have to be made. Perceptual organization and in particular the Gestalt principles of organization provide the human visual system with sensible assumptions about physical and biological objects. Generally, the shapes of natural objects vary smoothly and quite often they are symmetrical. Matter is cohesive so adjacent regions are likely to belong to the same structure.

4.2 Perceptual organization in computer vision

In computer vision perceptual organization is known as grouping, or more generally, as segmentation.

Analysis of blockworld scenes

More than twenty years ago many AI researchers tried to solve the segmentation problem by dividing a visual scene into a number of distinct objects. The complexity of natural images was avoided and the experiments were restricted to what is known as "blockworld" in AI: white, prismatic solids which are evenly illuminated.

The best known program is probably from Waltz (1975). He set out to recognize objects from a collection of lines. A crucial element in the analysis are junctions (where two or more lines meet). There is a direct relationship between the junction type and the possible arrangement of surfaces. Another important relationship is how the different types of edges (convex, concave, occluding) are combined.

Despite the success of some of the scene analysis programs their practical importance is limited. The principles embodied in the segmentation of blockworld scenes fail to reveal significant structures in natural images.

Curvilinear segmentation

Boundaries of objects are one of the most important structures derived from images. Naturally, many computer vision programs group edge elements into boundaries. They usually begin with linking edge points (edge following, edge linking) followed by approximating the contour by straight lines and curves. A number of methods have been proposed for accomplishing this task, e.g., Ballard and Brown (1982), Pavlidis (1982), Duda and Hart (1973). Techniques for solving the edge following problem include graph search and dynamic programming.

Hough transform

One of the most popular methods for line and curve detection is the Hough transform, initially proposed by Hough (1962) and later modified by several authors, e.g., Duda and Hart (1972), Kimme *et al.* (1975) and Ballard (1981). The method is an interesting example of seeking a representation which is much better suited for solving the segmentation problem. This representation is called parameter space. Hough used the slope-intercept equation $y = mx + q$ for creating the parameter space m, q. That is, a line in the x, y coordinate system is represented as a point in the m, q space.

Likewise, all possible lines through a point in x, y are represented in the parameter space by a straight line. The strategy is now to compute for every edge element the corresponding line in the m, q space. Straight lines in x, y are found at intersecting positions in m, q.

A better parameterization of the line is the polar equation $x \cos \phi + y \sin \phi = r$, because m may be infinite in the slope-intercept representation. With the polar form the parameter space is r, ϕ. This form is suitable for other curves and the Hough method can be generalized. For example, $(x-a)^2+(y-b)^2 = r^2$ would allow finding circles in the three-dimensional parameter space a, b, r.

As elegant and straightforward as the Hough transform is, it has a number of disadvantages. Apart from the requirement for huge accumulator array sizes (for n-parameter curves each quantized to m levels, m^n accumulator cells are required), the distance between points is not considered. It will group points from far regions if they happen to be on a straight line but will fail at the same time to consider points in close proximity if they are only slightly off the line. This is in strong contrast to the way humans perform collinearity grouping.

Virtual lines

Fig. 4 shows a random-dot flow pattern, created by superimposing on a pattern of random dots an identical pattern which was rotated slightly. The human visual system groups the dots such that a rotational flow is perceived. A similar effect is obtained when the copy is expanded. The grouping process results in a radial flow pattern.

The virtual line algorithm links points based on proximity and orientation. Take a point and determine for all points closer than d_{min} the orientation. Repeat this procedure for every point and determine the distribution of all orientation. In a typical flow pattern certain directions are dominant. The preferred directions are emerging from the orientation distribution as peaks. A simple thresholding operation determines what orientations are allowed. Now the procedure is repeated by assigning a virtual line to the closest point which satisfies the orientation criterion. A modification of this general schema is necessary if all directions are equally likely. Such is the case in the rotational flow pattern of Fig. 4. However, preferred directions are clearly obtained on a local basis, for example, in one quadrant.

Region segmentation

Region segmentation divides the image into units that are homogeneous with respect to one or more characteristics. From a perceptual point of view those regions should be meaningful as far the interpretation is concerned. Thus, region segmentation is to a certain degree application dependent and an issue is how to influence the process by a priori knowledge about the scene.

One approach to region segmentation is region growing. Properties of local groups of pixels are aggregated. In the simplest case, pixels are grouped to regions based on their properties and neighborhood. Global techniques consider the properties of a much larger population of pixels, for example, by thresholding the histogram. This amounts to a foreground/background separation (see, e.g., Hanson and Riseman, 1978).

Another approach is split and merge (see Horowitz and Pavlidis, 1974). Brice and Fennema (1970) and Feldman and Yakimovsky (1974) applied the state-space approach of AI to region growing. Here, the image is regarded as a discrete state and every pixel is initially a region. The state changes when a boundary between regions is removed (merging) or introduced (splitting).

No doubt, texture plays an important role in perceptual organization. Texture segregation divides an image into regions of similar texture primitives, sometimes referred to as texels. A texel has certain invariant properties which occur repeatedly in different positions and orientations. Julesz performed extensive studies of human texture perception (see Julesz, 1965, 1975; Julesz and Bergen, 1983). He claimed that two regions cannot be discriminated if their first- and second-order statistics are identical. First-order statistics refer to overall brightness while second-order statistics express differences in granularity (spatial distribution of texels) and slope (orientation).

While Julesz tried to arrive at a universal mathematical formula to explain the perception of texture boundaries, Marr developed an information processing theory which explained how a set of descriptive principles can be used to recover texture (Marr, 1976). We recall that the primitives in the primal sketch comprise edges, bars and blobs with associated orientations, contrast, dimensions and positions. Structures are recovered by clustering place tokens into small structures which in turn are grouped to larger units on the basis of proximity and changes in spatial density, a process Marr called theta aggregation.

For a more detailed summary and experiments with aerial scenes the reader is referred to Lee and Schenk (1992).

4.2 More general approaches

Although many computer vision programs have incorporated some aspects of perceptual organization the approaches remain rather rudimentary. Most methods are ad hoc and treat grouping aspects in isolation. Obviously, perceptual organization as an information processing system is not understood on the theoretical level (referring to Marr's approach of understanding visual processes on the three levels of theory, algorithms, implementation). Another likely reason for the lack of an integrated approach is the fact that the output of some perceptual processes do not immediately lend themselves to a physical interpretation of the scene.

McCafferty (1990) describes a computational approach to structuring and grouping. He sets out to formulate a mathematical model for including the Gestalt laws of perceptual organization in a vision system. In his work grouping is posed as an energy minimization problem.

As described earlier the solution space of the ill-posed visual reconstruction problem is reduced by introducing stabilizing functionals (regularization theory). Since the Gestalt laws of organization constrain the human visual system to physically plausible solutions, they provide the basis for the stabilizing functionals. To enforce the grouping due to proximity, similarity, closure, continuation, symmetry, figure-ground separation and common-fate, the following minimization problem is formulated.

$$E(g) = \lambda_1\|E_1(g)\|^2 + \lambda_2\|E_2(g)\|^2 + \cdots \lambda_7\|E_7(g)\|^2 \quad (3)$$

where $\lambda_1, \ldots, \lambda_7$ are regularization parameters; where $E_1(g), \ldots, E_7(g)$ are stabilizing functions, and where $E(g)$ is the total energy to be minimized.

The grouping problem, cast as an energy minimization problem, involves finding the global minimum energy state. The energy distribution across the state space is multimodal, thus the global minimum cannot be found by a simple threshold. It is difficult if not impossible to know if the lowest energy state found is in fact identical with the global minimum.

For computing the energy of a grouping ways must be found to compute the individual energy terms. For example, proximity is a measure of the spatial separation between place tokens. It can be computed as:

$$\frac{1}{i}\sum_{p=1}^{i}\frac{1}{n_i}\sum_{q=1}n_i d_{pq} \quad (4)$$

where i is the number of tokens; n_i is the number of neighbors of tokens; and where d_{pq} is the distance between token p and q. That is, proximity is calculated as the average distance between each token and its neighbor.

Finally, proper values for the regularization factors λ must be found. They essentially determine the weight of an individual grouping in the combination of the Gestalt laws of organization. The weighting factors are either constants in which case the relative importance of groupings never changes. They could also be functions of some other information, such as raw image data, or a priori knowledge.

5 OBJECT RECOGNITION

For a person to respond properly to the environment he must analyze, interpret and understand visual stimuli. Ideally, the same feat must be accomplished by a robot or an autonomous vehicle. To pick a part from a bin or to navigate through a cluttered environment the robot must understand its environment from sensor data and stored knowledge. The net result of image understanding is a fully interpreted scene. Image understanding or image interpretation is application dependent.

Object recognition is an important subtopic of image understanding. While image understanding is essential for many machine vision applications its role in digital photogrammetry is less important. Here, the emphasis is on recognizing and locating objects. Subsequent tasks, such as analyzing objects and their interrelationships, are typically performed by GIS.

Object recognition is a broad and very active research area. Many different methods and aspects have been reported. It is of course impossible to describe in one section the different approaches or to make proper reference to the huge body of literature. The purpose here is to provide an overview and to touch on some of the most critical issues, such as reducing the search space, modelling the objects and matching data features with object models.

5.1 Global model-based recognition

The traditional approach to object recognition is to search for a correspondence (match) between object models and sensory data (e.g., image). Matching establishes an interpretation of image data consistent with objects and phenomena of the world. The single most critical factor in this endeavor is to find suitable representations, both for the objects and input data. Most recognition systems describe objects and data as geometric or relational structures.

Matching geometric structures

A common solution is to parameterize object models and data. Matching then is to find the best fit between two parameter representations. The fit is measured with a merit function $M(\mathbf{p},\mathbf{q})$ where $\mathbf{p}$ is the parameter vector describing the object, and where $\mathbf{q}$ is the parameter vector derived from the data. There are several techniques available to optimize the parameters including direct analytical solution and hill-climbing (gradient) techniques.

The parameters may refer to shape such as curvilinear boundary description, or to other global properties of the object, such as area, elongateness, Euler number, etc. Useful properties are those that remain invariant under transformation or projection.

Matching $\psi - s$ curves

The $\psi - s$ curve represents a line as a function of $\psi(s)$ where s is the parameter of the tangent ψ. Fig. 5a shows a line in the x, y domain and Fig. 5b depicts the same line in the parameter space s, ψ. Horizontal straight lines transform to horizontal straight lines on the s - axis ($\psi = 0$) and straight lines with a slope α are represented as horizontal lines, $\psi = \alpha$. Thus, vertices cause discontinuities in the $\psi - s$ domain and the amount of the discontinuity is proportional to the angle of the vertex. Since the tangent of a circle is a constant, a circle will be represented by a straight line with the slope proportional to the radius.

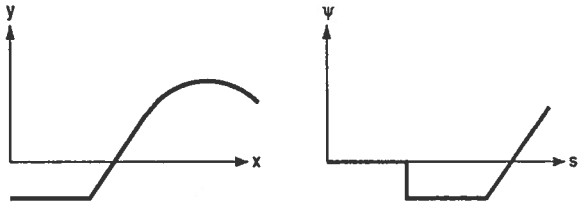


Fig. 5: A line consisting of straight line segments and a partial circle are represented in the original x, y space and in the parameter space ψ, s.

The $\psi - s$ representation for describing the shape of object models and data models is advantageous because it is rotation and translation independent. We use this representation successfully for matching edges in stereopairs (see e.g., Schenk *et al.*, 1991, Lee and Schenk,

1989).

Matching Fourier descriptors

Fourier descriptors have long been used for the representation of boundaries. The boundary of a region is considered a periodic function, and it can be expanded in a Fourier series. The low-order terms describe the general shape. The more coefficients are added the better the approximation of the original curve. In order to utilize Fourier descriptors in a matching scheme, Granlund (1972) and Lin and Hwang (1987) propose using invariants. However, two problems are observed. For one the success of a match is difficult to measure (merit function is ambiguous). Another disadvantage is that the matching procedure does not provide clues for the transformation between object model and data.

We remedied these disadvantages by putting Fourier descriptors into a least-squares matching scheme (see Tseng and Schenk, 1992). Here, the transformation parameters (similarity or affine transformation) are included in the matching process and are therefore explicitly determined. The standard deviation provides a quantitative measure for the goodness of the match.

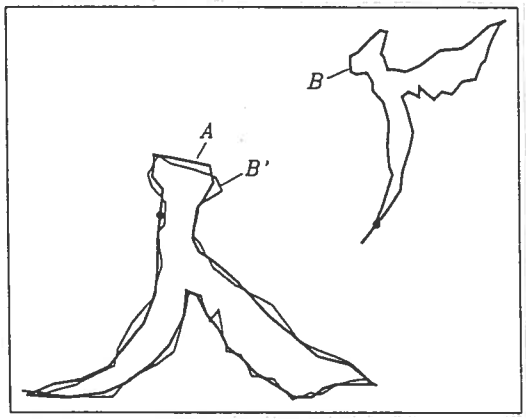


Fig. 6: Example of matching an object model (curve B) with the data (curve A). The matching procedure renders the transformation parameters as well as a quantitative measure for the goodness of the fit.

The reader is referred to Tseng and Schenk (1992) for a detailed description.

Matching moment invariants

Smith and Wright (1971) report about recognizing ships with moment invariants. The authors used up to fifth order moments. An image is represented by the spatial moments of its intensity function. The ij^{th} generalized moment is

$$m_{ij} = \int f_{ij} I(x,y) dx dy \qquad (5)$$

where $I(x,y)$ is the image intensity function. The function $f_{ij}(x,y)$ may represent the density in which case m_{ij} corresponds to the standard definition of moments; or it may involve sine and cosine functions where m_{ij} would represent the coefficients of a Fourier series expansion of $I(x,y)$.

Linear, quadratic and cubic polynomials of the moments are used as ship descriptors. Linear regression is used to distinguish these nonlinear functions by treating the powers of the moments as new linear variables.

A simple pattern recognition model with two moment invariants was chosen by Dudani (1979) to recognize aircrafts. For each aircraft the two moment invariants

$$\begin{aligned} X &= \mu_{20} + \mu_{02} \\ Y &= \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \end{aligned} \qquad (6)$$

are computed. The pattern is represented as a point in the X,Y parameter space. The system outperformed human observers in a test of identifying airplanes in a library of 132 different types.

5.2 Feature matching with geometric constraints

In contrast to global matching methods, feature-based matching approaches the object recognition problem in several steps. The basic idea is to extract local data features which are matched with spatially localized object features. Such features include distinct points, edges, curves and surface patches. The following tasks must be solved:

1. Build a library of objects (model base). The geometric description of every object depicts shape characteristics in a local object coordinate system.
2. Extract features from the images. Group and segment them such that data features correspond to one object *(segmentation problem)*.
3. From the model base select those objects which are likely to correspond to a set of given data features *(indexing problem)*.
4. Find instances of objects in the data by establishing a correspondence between data features and object features *(correspondence problem)*.
5. Find instances of objects in the data by transforming the object to the image for checking global consistency *(hypothesize-and-test problem)*.

Extracting features, organize and represent them in a suitable form is a typical perceptual organization problem. In the most simple form, it would involve edge detection, edge formation and edge segmentation. There is a plethora of edge detecting operators available. Edge forming depends on the edge detector used. Curvilinear edge segmentation results in straight line segments and curved parts of an edge. Yet another processing stage may analyze consecutive straight line segments for rectangular polygons.

A straightforward but naive solution to the indexing problem is simply to take every object from the model base and perform steps 3 and 4. A more intelligent approach would be to consider additional attributes such as color. Also the number of data features may be used to select only those objects that have more features, assuming that the data features are only a subset of all features describing an object. Finally, domain-specific knowledge may be used. For example, it may well be that certain objects do not occur in certain parts of the image or in the neighborhood of other objects.

Correspondence problem

We follow Grimson (1990) to explain the correspondence problem and the significance of geometric constraints by way of a simple 2-D example. Suppose the feature extraction process (e.g. edge detection, grouping and segmenting) generated the three data features f_1, f_2, f_3 (see Fig. 7a) which we assume correspond to the boundaries of an object. Also shown in Fig. 7b is an object selected from the object library. The problem is to establish a correspondence between data features and object features without performing the costly transformation object to image.

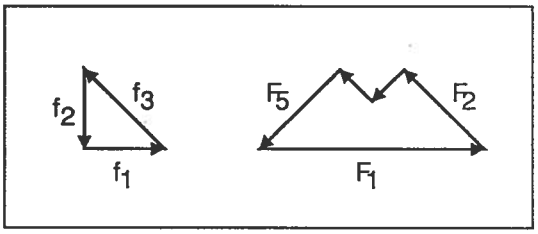


Fig. 7: Data features extracted from image (left) and object selected from the object library (right).

In a brute force approach we could pair each data feature with all object features and check all possible combinations for consistency. This would amount to an exhaustive search of the correspondence space (see Fig. 9). The $n-$dimensional correspondence space (n = number of data features) is tesselated by the number of object features. Referring to our example we obtain $5^3 = 125$ assignments most of which do not make any sense. Thus, the goal is to determine only likely pairings.

Table 1: Angles in grads between features of object shown in Fig. 7.

	F_1	F_2	F_3	F_4	F_5
F_1	0	150	250	150	250
F_2	250	0	100	0	100
F_3	150	300	0	300	0
F_4	250	0	100	0	100
F_5	150	300	0	300	0

Instead of exploring all pairings, geometrical constraints are used to eliminate those which make no sense. However, the method should require much less computing time than the rigorous solution of transforming the object to the image and checking if it matches. In our example angles are used as geometrical constraints. In Table 1 all angles between object features

are listed. Table 2 contains the same for the data features.

Table 2: Angles in grads between data features shown in Fig. 7.

	f_1	f_2	f_3
f_1	0	300	150
f_2	300	0	250
f_3	250	150	0

As indicated in Fig. 8 data feature f_1 is assigned to every object feature F_i. Next, data feature f_2 is assigned to those object features which satisfy the angle constraint. From Table 2 we take the angle between f_1 and f_2 (300^g) and check in the appropriate rows of Table 1 for the same angle. For the assignment f_1 to F_3 two pairings satisfy the constraint. The same holds for f_1 to F_5. Now, only these pairings need be pursued further. Again, the assignment for f_3 is performed by considering the angular constraint $\angle(f_2, f_3) = 250^g$. We end up with the four sets of possible pairings for f_1, f_2, f_3, namely, $(F_3, F_2, F_1), (F_3, F_4, F_1), (F_5, F_2, F_1), (F_5, F_4, F_1)$. With additional constraints, for example, distances, the potential pairings could be further pruned.

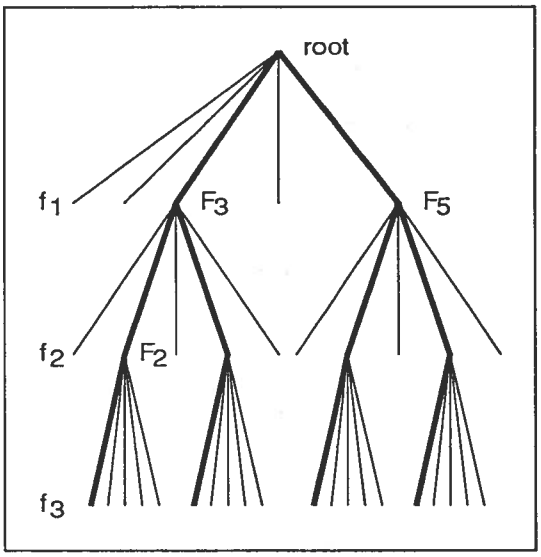


Fig. 8: Correspondence problem cast as a tree search problem.

We note that the four solutions found are not globally consistent. The geometric constraints only assure local consistency. The angle constraint is a binary constraint. If enforced two consecutive nodes in the tree are consistent. A unary constraint would ensure a single node to be consistent. With unary and binary constraints the consistency between three or more nodes is not guaranteed. Therefore, the transformation of the object to image is necessary.

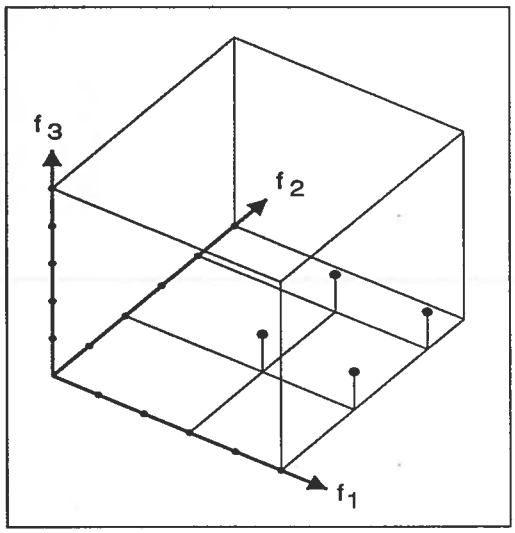


Fig. 9: Correspondence space. Solid dots indicate possible correspondences found by applying geometric constraints.

Every point in the discrete correspondence space constitutes a hypothesis for an assignment of data features to object features. Obviously, most assignments can be ruled out immediately. For example, two distinctly different data features cannot be assigned to one and the same object feature (assuming rigid objects). Unary and binary constraints further reduce the assignments to a feasible set.

As indicated in Fig. 8 the search for feasible assignments is cast as a tree search problem. On the first level of the tree, data feature f_1 is paired with all object features $F_i, i = 1, ...n$, but only if unary constraints are satisfied. An example of a unary constraint is the length. The same procedure is repeated on the next levels of the tree. Only those nodes are further expanded which satisfy the constraints. Every valid leaf node defines a path in the tree indicating feature assignments which must now be examined for global consistency.

Verifying hypotheses

Every leaf of the tree is a hypothesis about the correspondence of data features to object features. These hypotheses must be tested by transforming the object into the image. Global consistency is reached if all data features match the corresponding object features. Usually, there are more object features than data features. That is, there are some unpaired object features. They are now used to examine the gray levels for any evidence of data features which remained undetected in the feature extraction process.

In the general case of a 3-D transformation seven parameters must be determined. This transformation should be formulated in a general fashion; for example, it should not be restricted to points only, rather it should allow including straight line segments, curves or surface patches.

A hypothesis can be tested before the leaf of a tree is reached. In fact, as soon as enough data features are assigned to perform a 3-D transformation, it may well be advisable to verify the hypothesis immediately. If successful, the search in the correspondence space can be terminated. This variation of the general approach may be justified if the tree is very deep (e.g., many object features) or if a particularly salient matching is found. This method, termed local feature focus, was proposed by Bolles and Horaud (1986).

5.3 Nonmodel-based object recognition

The approaches to object recognition described so far are based on the assumption that objects can be described geometrically, for example, by defining their boundaries or surfaces. This may be impossible, however, if it comes to recognizing objects with complex shapes. Such is the case with aerial scenes. Even rather simple structures, such as buildings, come in such a variety of different sizes and shapes that it is a fruitless attempt to precisely describe and store all of them in a model library. For outdoor scenes other solutions must be found.

Context-based object recognition

Strat and Fischler (1991) describe a system, called Condor (context-driven object recognition), that deviates radically from the model-based paradigm. The authors argue that oftentimes objects of outdoor scenes can only be identified by taking their surroundings into account, because relationships among objects provide important clues. The notion of Condor is to embed objects in contexts rather than treating them as independent entities.

Condor is in an experimental stage. The goal is to recognize objects in natural scenes. The knowledge base is tailored for solving this task in a two-square mile area near the Stanford University campus with ground level images. As the authors put it, the recognition capability of Condor should equal that of a rabbit inhabiting the same environment.

At the heart of Condor is an object-oriented knowledge base, called core knowledge structure, that contains knowledge about the visual world. Spatial knowledge is represented as a multiresolution octree that allows recognizing objects at various scales. A semantic network is used for representing semantic knowledge at various levels of resolution.

Condor works like a production system. The main process types generate candidates, compare candidates, form and select cliques. The processes act as daemons and invoke themselves depending on the contextual environment. The output of the system is a labeled 3-D model of the scene where a label is an object class. Examples are sky, foliage, raised object and ground.

Each class has an associated set of simple recognition procedures designed to work in specific contexts. An example is recognizing foliage which is very difficult considering the different situations and conditions. Thus, this general recognition task is divided into rather specialized subtasks like finding the silhouette of foliage against the sky, or finding foliage of one type of tree.

A context set is basically a rule. If the context is satisfied (e.g., sky is clear, camera is horizontal, color is available) an operator is invoked (e.g., segment blue regions) and the result forms a candidate hypothesis. Candidates are checked for global consistency in a

process called clique formation. A clique is a set of mutually consistent candidate hypotheses. Inconsistencies of hypotheses are detected by context-specific procedures, again expressed as context sets. If the system labeled a region "ground", then it must satisfy the rule that it cannot extend above the skyline.

Function-based object recognition

Winston et al. (1984) point out that there are many different physical descriptions for objects, say, a cup. However, a single functional description can be used to represent all possible cups. Many man-made objects serve a specific purpose, and it may be possible to describe that purpose in a concise manner as functional descriptions which can be used together with other knowledge to recognize objects.

Stark and Bowyer (1991) describe a system to recognize chairs. The system takes 3-D polyhedral objects and recognizes whether the object belongs to category chair and, if yes, into which subcategory it falls. A first decision is made based on the size of the object.

First, all functional elements of an object are analyzed. Functional elements consists of (i) single surfaces, for example the seat of a chair, (ii) groups of surfaces serving one function, (iii) 3-D module of the structure. A function label (name of the functional property, e.g., *sittable surface*) is assigned to the functional elements.

Function labels are defined by procedural knowledge primitives. Stark and Bowyer use relative orientation, dimensions, stability, proximity and clearance as primitives. The dimension checks the size of surfaces if they are within reasonable sizes. The stability primitive checks for stable support by examining the contact points of the object with the ground plane.

A hierarchical tree represents the function-based descriptions. Associated to the nodes of the graph are frames with information about the name, type and functional plan. All function labels have specified constraint values.

6 CONCLUDING REMARKS

Digital photogrammetry is the most intensively researched area of photogrammetry. Research efforts are beginning to bear fruit; systems and software products are becoming operational, most notably digital photogrammetric workstations—the pendant of the analytical plotter.

Despite the progress achieved, we should not forget that most if not all research effort has been directed toward solving the easier problems. In order to respond to the increased demand for more automation we need to address the harder problems on a much broader base. First, the photogrammetric research community needs to realize the full scope of the problem. Automating photogrammetric tasks requires a shift from our traditional thinking of determining points as accurately as possible towards a more global view of information processing.

Even though we have successfully produced maps for decades, we do not seem to precisely understand how we accomplish it. The crux of the automation problem is that we only have a very sketchy theory of how photogrammetric products are generated. Take recognizing and measuring buildings for example. No one knows how human operators identify buildings, what knowledge they bring to bear and what experience is necessary. We realize that before an object can be measured it must be recognized. Recognizing objects involves image interpretation—a subject photogrammetrists hardly consider a photogrammetric problem (Helava, 1988).

Digital photogrammetry ultimately encompasses all levels of the computer vision paradigm. The main quest is to make information explicit, that is, from gray levels we must derive a geometric and semantic representation of the scene. So far, photogrammetrists have successfully tackled some of the early vision problems, mostly concentrating on geometric aspects, neglecting semantic information processing.

Digital photogrammetry poses not only a new challenge to researchers but also to educators. In order for users to more effectively use digital photogrammetric products and to articulate more precisely their requirements they must have an adequate background. Likewise, future researchers are expected to be sufficiently familiar with cognitive science, visual percep-

tion, computer vision, knowledge engineering, digital signal processing, just to mention a few related fields, to further advance digital photogrammetry. How many photogrammetry curricula have been revised to include subjects from artificial intelligence, computer vision, visual perception and knowledge engineering?

7 REFERENCES

Al-Tahir, R., and T. Schenk, 1992. On the interpolation problem of automated surface reconstruction. *Int. Archives of Photogrammetry and Remote Sensing*, Congress Washington D.C., Comm. III.

Ballard, D.H., and C.M. Brown, 1982. *Computer vision.* Prentice-Hall, Englewood Cliffs, New Jersey.

Ballard, D.H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, Vol. 13, No. 2, pp. 111-122.

Blake, A., and A. Zissermann, 1987. *Visual reconstruction.* The MIT Press, Cambridge, Massachusetts.

Bolles, R.C., and P. Horaud, 1986. 3DPO: A three-dimensional part orientation system. *Int. J. Robotics Res.*, Vol. 5, no. 3, pp. 3-26.

Brice, C. and C. Fennema, 1970. Scene analysis. *Artificial Intelligence*, Vol. 1, No. 3, pp. 205-226.

Duda, R.O., and P.E. Hart, 1973. *Pattern recognition and scene analysis.* Wiley, New York.

Duda, R.O., and P.E. Hart, 1972. Use of the Hough transform to detect line and curves in pictures. *Commun. ACM, Graphics and image processing*, Vol. 15, No. 1, pp. 11-15.

Dudani, S.A., 1979. Aircraft identification by moment invariants. *IEEE Trans. on Computers*, Vol. C-26, No. 1, pp.39-45.

Feldman, J., and Y. Yakimovsky, 1974. Decision theory and artificial intelligence. *Artificial Intelligence*, Vol. 5, No. 4, pp. 349-371.

Förstner, W., 1990. Concepts and algorithms for digital photogrammetric sytems. *Int. Archives of Photogrammetry and Remote Sensing*, Symp. Wuhan, Comm. III, Vol. 28, pp. 144-171.

Förstner, W., 1986. A feature based correspondence algorithm for image matching.*Int. Archives of Photogrammetry and Remote Sensing*, Symposium Helsinki, Comm. III, Vol. 19, pp. 534–545.

Granlund, G.H., 1972. Fourier preprocessing for hand character recognition. *IEEE Trans. on Computer*, Vol. 21, No. 2, 195-201.

Grimson, W.E.L., 1990. *Object recognition by computer.* The MIT Press, Cambridge, Massachusetts.

Hanson, A.R., and E.M. Riseman, 1978. Segmentation of natural scenes. In Hanson and Riseman (Ed) *Computer vision systems*, Academic Press, New York.

Helava, U.V., 1988. On system concepts for digital automation. *Int. Archives of Photogrammetry and Remote Sensing*, Congress Kyoto, 27/B2, pp. 171-190.

Horowitz, S.L., and T. Pavlidis, 1974. Picture segmentation by a directed split-and-merge procedure. *Proc. 2nd Intern. Joint Conf. Pattern Recogn.*, pp. 424-433.

Hough, P.V.C., 1962. Methods and means for recognizing complex patterns. U.S. Patent 3,069,654.

Julesz, B., 1965. Texture and visual perception. *Scientific American*, Vol. 212, pp. 38-48.

Julesz, B., 1975. Experiments in the visual perception of texture. *Scientific American*, Vol. 232, pp. 34-43.

Julesz, B., and J.R. Bergen, 1983. *Textons: the fundamental elements in preattentive vision and perception of texture.* The Bell System Technical Journal, vol. 62, no. 6, pp. 1619–1645.

Kimme, C., D. Ballard, and J. Sklansky, 1975. Finding circles by an array of accumulators. *Commun. ACM*, Vol. 18, No. 2, pp. 120-122.

Lancaster, P., and K. Salkauskas, 1986. *Curve and surface fitting: An introduction.* Academic Press, London.

Lee, D.C., and T. Schenk, 1992. Image segmentation from texture measurement. *Int. Archives of Photogrammetry and Remote Sens-*

ing, Congress Washington D.C., Comm. III.

Li, J.C., and T. Schenk, 1989. Aerial image matching using $\psi - s$ representation. *Techn. Notes in Photogrammetry*, No. 4, Departm. of Geodetic Science and Surveying, OSU.

Lin, C.S., and C.L. Hwang, 1987. New forms of shape invariants from elliptic Fourier descriptors. *Pattern Recognition*, vol. 20, No.5, pp. 535-545.

McCafferty, J.D., 1990. *Human and machine vision.* Ellis Horwood Ltd., Chichester, West Sussex, England.

Marr, D., 1982. *Vision.* W.H. Freeman and Company, New York.

Marr, D., 1976. Early processing of visual information.*Proc. of the Royal Society of London.* Series B, 275, pp. 483-524.

Pavlidis, T., 1982. *Algorithms for graphics and image processing.* Computer Science Press, Rockville, MD.

Schenk, T., 1992. Machine vision and close-range photogrammetry. *Techn. Notes in Photogrammetry*, No. 13, Departm. of Geodetic Science and Surveying, OSU.

Schenk, T., and Ch. Toth, 1991. Reconstructing visible surfaces. *Proc. Industrial Vision Metrology, SPIE*, Vol. 1526, pp. 78-89.

Schenk, T., J.C. Li, and Ch. Toth, 1991. Towards an autonomous system for orienting digital stereopairs. *Photogrammetric Engineering and Remote Sensing, PERS*, Vol. 57, No. 8, 1057-1064.

Smith, F.W., and M.H. Wright, 1971. Automatic ship photo interpretation by the method of moments. *IEEE Trans. on Computers*, pp. 1089-1095.

Stark, L., and K. Bowyer, 1991. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Trans. Pattern Recognition and Machine Intelligence, PAMI*, Vol. 13, No. 10, pp. 1097-1104.

Strat, T.M., and M.A. Fischler, 1991. Context-based vision: recognizing objects using information from both 2-D and 3-D imagery. *IEEE Trans. Pattern Recognition and Machine Intelligence, PAMI*, Vol. 13, No. 10, pp. 1050-1059.

Winston, P.H., T.O. Binford, B. Katz, and M. Lowry, 1984. Learning physical decription from functional definitions, examples, and precedents. In *Proc. Int. Symp. Robotics Research*, The MIT Press, vol. 1.

Tseng, Y., and T. Schenk, 1992. A least-squares approach to matching lines with Fourier descriptors. *Int. Archives of Photogrammetry and Remote Sensing*, Congress Washington D.C., Comm. III.

Waltz, D.L., 1975. Generating semantic descriptions from scenes with shadows. In P.H. Winston (Ed.), *The psychology of computer vision.* McGraw-Hill, New York.

RESAMPLING DIGITAL IMAGERY TO EPIPOLAR GEOMETRY

Woosug Cho
Toni Schenk
Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247

Mustafa Madani
Intergraph Corporation, Huntsville, Alabama
USA

Commission III

ABSTRACT

Most algorithms in computer vision and digital photogrammetry assume that digital stereo pairs are registered in epipolar geometry (normalized images) in order to confine the search of conjugate features along the same scan lines. In this paper we describe the procedure of computing normalized images of aerial photographs with respect to the object space. Based on the exterior orientation of the stereo pair the rotation matrix for the normalized images is computed. The focal length of the new images may be determined according to different criteria. We show the case of minimizing the decrease in resolution. During the same process systematic errors of the scanning device can be considered. Several examples demonstrate the feasibility of our approach.

1. INTRODUCTION

Most algorithms in computer vision and digital photogrammetry are based on the assumption that digital stereo pair is registered in epipolar geometry. That is, the scan lines of stereo pairs are epipolar lines. This condition is satisfied when the two camera axes of a stereo vision system are parallel to each other and perpendicular to the camera base. In conventional aerial photogrammetry, imagery is obtained directly by scanners, such as Landsat or SPOT, or indirectly by digitizing aerial photographs. Thus an aerial stereo pair is not likely to be in epipolar geometry since the attitude of the camera at the instant of exposure is different at every exposure station.

[Kreiling 1976] described a method for recovering the epipolar geometry from the parameters of an independent relative orientation. The epipolar geometry is only recovered with respect to the model space. In many instances it is desirable to establish epipolar geometry with respect to object space. The procedure to obtain resampled epipolar images with exterior orientation elements after absolute orientation was developed by [Schenk 90]. In this paper we call the resampled epipolar image reconstructed with respect to object space the *normalized image*. The original photograph obtained at the instant of exposure is referred to as the *real image*. The image which is parallel to the XY-plane of the object space system is called the *true vertical image*.

In this paper we describe the procedure to compute normalized images of aerial images with respect to the object space and the method to minimize the decrease in resolution. By considering systematic errors of the scanning device, we show that the normalized image is free of geometric distortion of the scanning device. The next section provides some background information followed by a detailed description of how to determine normalized images.

2. EPIPOLAR GEOMETRY

Fig. 1 shows a stereo pair in epipolar geometry with C', C'' the projection centers. The epipolar plane is defined by the two projection centers and object point P. The epipolar lines e', e'' are the intersection of the epipolar plane with the image planes. The epipoles are the centers of bundles of epipolar lines which result from intersecting the photographs with all possible epipolar planes.

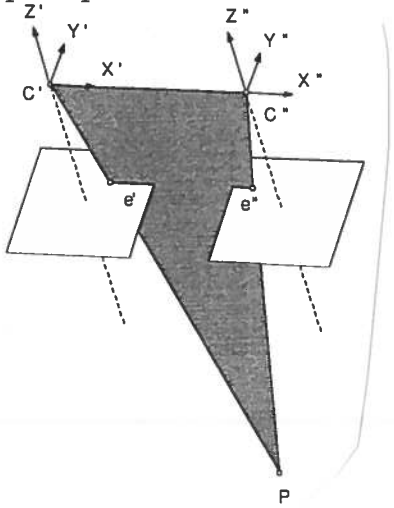


Figure 1: Epipolar geometry

The conjugate epipolar lines in Fig. 1 are parallel and identical to scan lines. The epipoles are in infinity because of vertical photographs. However, in most cases, two camera axes are neither parallel nor perpendicular to the air base ($C'C''$). We transform images into a position that conjugate epipolar lines are parallel to the x-axis of the image coordinate system such that they have the same y-coordinate. The transformed images satisfying the epipolar condition are called *normalized images* in this paper. The normalized images must be parallel to the air base and must have the same focal length. Having chosen a focal length, there is still an infinite number of possible normal image positions (by rotating around the air base).

3. COMPUTATION OF NORMALIZED IMAGE

3.1 Camera Calibration

Digital imagery can be obtained either directly by using digital cameras, or indirectly by scanning existing aerial photographs. In both cases, the digitizing devices (digital camera or scanner) must be calibrated to assure correct geometry. For our applications we use the rigorous calibration method suggested by [Chen and Schenk 92]. The method is a sequential

adjustment procedure to circumvent the high correlation between DLT parameters and camera distortion parameters. The distortion consists of two parts: lens distortion and digital camera error. Lens distortion is composed by radial and tangential distortion. Digital camera error is scan line movement distortion since EIKONIX camera used in our applications is a linear array camera. For more details about digital camera calibration, refer to [Chen and Schenk 92]. With the camera calibration, we can obtain a digital image free of systematic distortion. The image is called *pixel image* in this paper.

3.2 Transforming pixel image to normalized image

The normalized image is a pixel image in epipolar geometry with reference to the object space. Thus, exterior orientation elements after absolute orientation are to be used for transforming the pixel image to a normalized image. The exterior orientation elements consist of three rotation angles and the location of perspective center in the object space system. The relationship between pixel image and object space is expressed by the collinearity equation

$$x_p = -f_p \frac{r_{11}(X - X_C) + r_{12}(Y - Y_C) + r_{13}(Z - Z_C)}{r_{31}(X - X_C) + r_{32}(Y - Y_C) + r_{33}(Z - Z_C)}$$

$$y_p = -f_p \frac{r_{21}(X - X_C) + r_{22}(Y - Y_C) + r_{23}(Z - Z_C)}{r_{31}(X - X_C) + r_{32}(Y - Y_C) + r_{33}(Z - Z_C)}$$

where x_p, y_p are image coordinates and $r_{11} \cdots r_{33}$ are elements of an orthogonal rotation matrix R that rotates the object space to the image coordinate system. X_C, Y_C, Z_C are the coordinates of the projection center; X, Y, Z, the coordinates of object points.

There are two steps involved in the transformation of the pixel images (P', P'') to normalized images(N', N''). First, pixel images are transformed to *true vertical images* and from there to normalized images. Fig. 2 shows the relationship between pixel images and normalized images.

The first transformation from pixel image to true vertical position simply involves a rotation with R^T, where R is an orthogonal rotation matrix from the object space to image space.

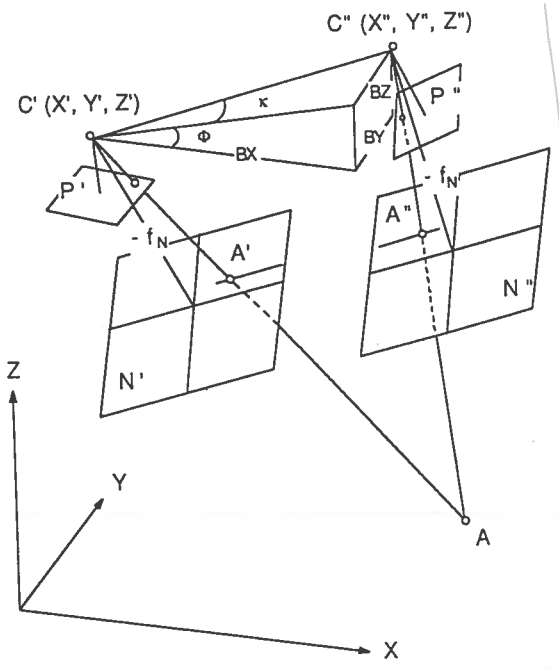


Figure 2: Relationship between pixel image and normalized image

Next, a transformation from true vertical to the normalized position is applied. The first angle of the rotation matrix R_B transforming from true vertical to the normalized position is K about the Z-axis, then Φ about the Y-axis, Ω about the X-axis. The rotation angles K, Φ can be computed from the base elements BX, BY, BZ, and Ω from the exterior orientation angles:

$$K = tan^{-1} \frac{BY}{BX} \tag{1}$$

$$\Phi = -tan^{-1} \frac{BZ}{(BX^2 + BY^2)^{1/2}} \tag{2}$$

$$\Omega = \frac{\omega' + \omega''}{2}, \tag{3}$$

where $BX = X'' - X', BY = Y'' - Y'$, and $BZ = Z'' - Z'$.

The base rotation matrix R_B will be the following.

$$R_B = R_\Omega R_\Phi R_K, \tag{4}$$

where

$$R_K = \begin{bmatrix} cosK & sinK & 0 \\ -sinK & cosK & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\Phi = \begin{bmatrix} cos\Phi & 0 & -sin\Phi \\ 0 & 1 & 0 \\ sin\Phi & 0 & cos\Phi \end{bmatrix}$$

$$R_\Omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Omega & sin\Omega \\ 0 & -sin\Omega & cos\Omega \end{bmatrix}.$$

The base rotation matrix R_B is a combined matrix in which the primary rotation axis is about the Z-axis, followed by a rotation about the Y-axis and X-axis. Depending on the Ω (X-axis rotation), there are many different normalized images. The rotation Ω about the X-axis influences the nonquadratic shape when computing the normalized images.

The normalized rotation matrix R_N is a multiplication of two rotation matrices: the rotation from pixel image to true vertical position and the rotation from true vertical to normalized position.

$$R_N = R_B R^T \tag{5}$$

The R_N is an orthogonal rotation matrix which transforms the pixel image to the normalized image. Since in Eq.(5) R^T is the transposed rotation matrix of exterior orientation elements, the R_N matrix must be computed for both images in stereo. We may use one of two transformations from pixel image to normalized image: transformation using collinearity condition or projective transformation.

3.2.1 Transformation using collinearity The collinearity condition equations can be used for the transformation of the pixel image to normalized image. The transformation is represented in the following equation and is illustrated in Fig. 3.

$$x_N = -f_N \frac{r_{11}x_P + r_{12}y_P - r_{13}f_P}{r_{31}x_P + r_{32}y_P - r_{33}f_P} \tag{6}$$
$$y_N = -f_N \frac{r_{21}x_P + r_{22}y_P - r_{23}f_P}{r_{31}x_P + r_{32}y_P - r_{33}f_P},$$

where $r_{11} \cdots r_{33}$ are the elements of the R_N rotation matrix.

3.2.2 Projective transformation The projective transformation can be applied since

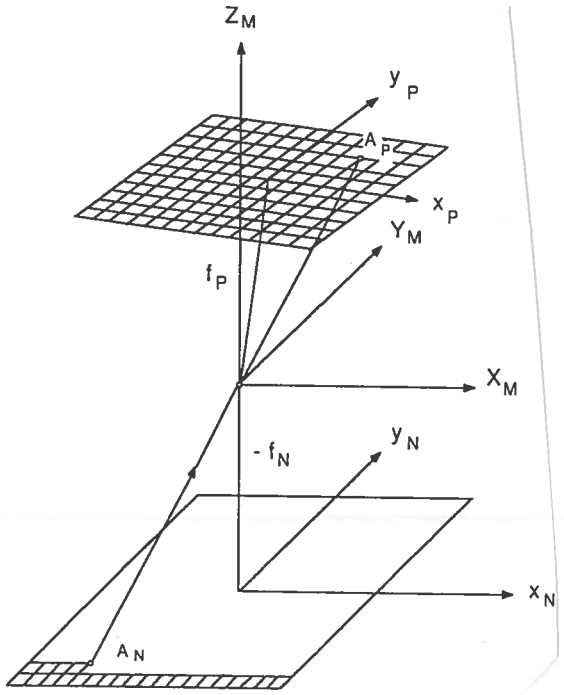


Figure 3: Transformation pixel image to normalized image

both pixel image and normalized image are planar.

$$x_N = \frac{c_{11}x_P + c_{12}y_P + c_{13}}{c_{31}x_P + c_{32}y_P + 1} \tag{7}$$
$$y_N = \frac{c_{21}x_P + c_{22}y_P + c_{23}}{c_{31}x_P + c_{32}y_P + 1}$$

By comparing the coefficients in the projective transformation with those in the collinearity equations, we find the following identities:

$$C_{11} = \frac{f_N r_{11}}{f_P r_{33}} \qquad C_{21} = \frac{f_N r_{21}}{f_P r_{33}} \tag{8}$$
$$C_{12} = \frac{f_N r_{12}}{f_P r_{33}} \qquad C_{22} = \frac{f_N r_{22}}{f_P r_{33}}$$
$$C_{13} = -\frac{f_N r_{13}}{r_{33}} \qquad C_{23} = -\frac{f_N r_{23}}{r_{33}}$$
$$C_{31} = -\frac{r_{31}}{f_P r_{33}} \qquad C_{32} = -\frac{r_{32}}{f_P r_{33}}$$

When performing the transformation pixel image to normalized image, the quadratic tesselation of the pixel image results in nonquadratic tesselation of the normalized image. In order to

avoid interpolation into quadratic tesselation, it is recommended to project the tesselation of the normalized image back to the pixel image (see also Fig. 3). The coefficients for backward projection are obtained in the same fashion by R_N^T if the focal lengths of the pixel and normalized image are the same ($f_P = f_N$).

$$\begin{aligned} C'_{11} &= C_{11} & C'_{21} &= C_{12} \\ C'_{12} &= C_{21} & C'_{22} &= C_{22} \\ C'_{13} &= C_{31} f_P f_N & C'_{23} &= C_{32} f_P f_N \\ C'_{31} &= C_{13} \frac{1}{f_P f_N} & C'_{32} &= C_{23} \frac{1}{f_P f_N} \end{aligned} \quad (9)$$

For the more general case of different focal lengths ($f_P \neq f_N$), the backward projection is obtained by inverting R_N because $R_N^{-1} \neq R_N^T$.

3.2.3 Resampling After applying a geometric transformation from the normalized image to pixel image, the problem now is to determine the gray value of the new pixel location in the normalized image, because the projected position in the pixel image is not identical to the center of the pixel. Therefore, gray values must be interpolated. This procedure is usually referred to as resampling. Several interpolation methods may be used.

- zero-order interpolation: the gray value of the nearest neighbor is chosen. This is identical to rounding the projected position to the integer, corresponding to the tesselation of the pixel image system. This simplest process may lead to unacceptable blurring effects.
- bilinear interpolation: the gray values of the four surrounding pixels contribute to the gray value of the projected point depending on the distance between the projected and four neighboring pixels.

3.3 Normalized image

The procedure discussed in the previous section establishes the transformation between pixel image and normalized image. The distortion parameters are determined during camera calibration. When resampling the gray values for the normalized image, we also apply the correction. Thus, the computation of the normalized image proceeds in four steps (see Fig. 4).

T_1: Transformation between pixel image and original photograph (diapositive). The transformation parameters are determined during camera calibration. Common references for these transformation parameters are fiducial marks, reseau points, and distinct ground features.

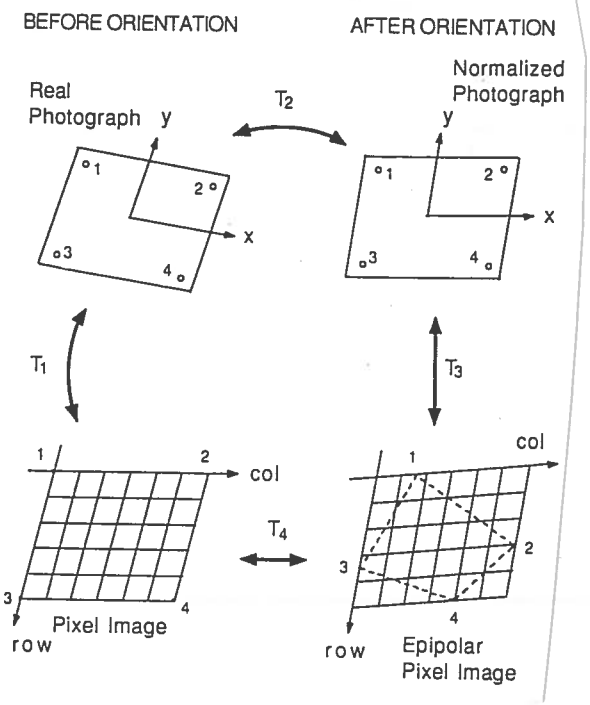


Figure 4: Relationship between photograph and pixel image, both in real and epipolar position

T_2: Projective transformation between original photograph and normalized photograph. The detailed procedure is described in Section 3.2.

T_3: Definition of coordinate system for the pixel image in epipolar geometry (normalized image). In order to minimize the decrease in resolution (or to optimize the size), first the four corners of the pixel image((0,0), (0,N), (N,0), (N,N)) are transformed to real photographs and then to normalized photo coordinates through T_1, T_2. The following procedure defines the normalized image coordinate system.

1. Determine maximum y coordinate of four corner points in both images. This defines row 0 in both normalized images.
2. Determine x and y differences of corner points in both photos and compute the maximum distance d_{max} in either x or

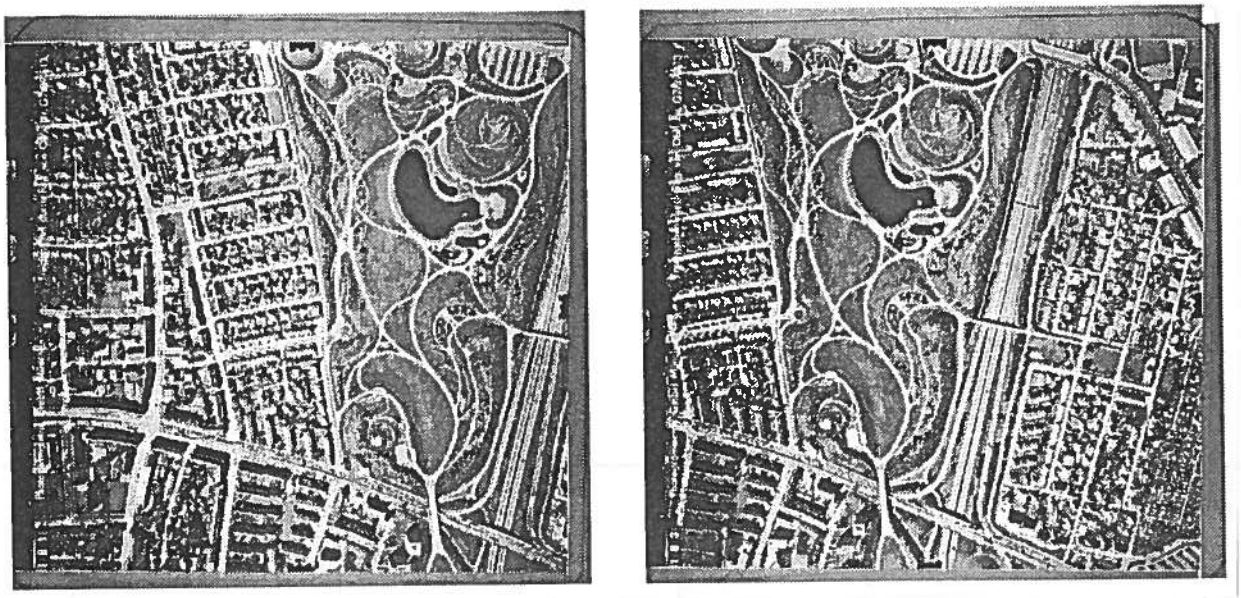


Figure 5: Stereo real images of Munich model (resolution: 512×512)

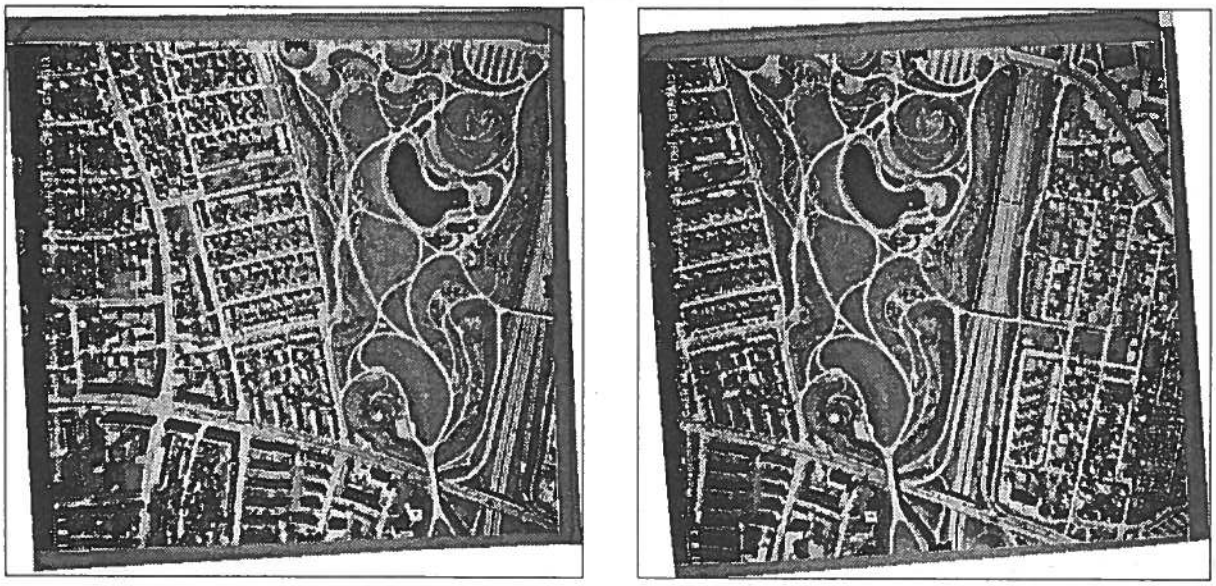


Figure 6: Stereo normalized images of Munich model (resolution: 512×512)

y direction (both photos). This determines the size of the epipolar pixel image in photo coordinates.

3. Change from photo coordinates to pixel coordinates by using the relationship d_{max} = resolution pixel image.

T_4: Transformation from normalized image to pixel image in order to perform resampling. This is accomplished by using T_3, T_2 and T_1.

4. EXPERIMENTAL RESULTS

The procedure discussed in section 3 to compute normalized images, has been implemented and tested with several pairs of aerial images. Some of our images are digitized by the PhotoScan scanner from Zeiss/Intergraph Corp. and some others by the EIKONIX camera (EC850). Here, we present the "Munich" model, scanned with the EIKONIX camera (see Fig. 5).

The real images have a resolution of 4096 by 4096, corresponding to $\approx 60\mu m$ and 256 gray values. As explained in detail in [Chen and Schenk 92], the EIKONIX camera introduces distortion to the scanned image. We remove this distortion during the procedure of computing normalized images. Fig. 6 shows the images normalized with respect to the object space. Note the curved margins of the normalized images. This is the effect of the camera distortion (now removed!). The transformation (T_1) discussed in section 3.3 must be well known in order to assure the correct geometry in normalized images. In our example, its accuracy is less than a half pixel in 4K resolution.

The normalized image coordinate system is established by transforming the four corner points of the pixel image so that the loss of information of the pixel image is minimized. By applying the rotation of the base by common omega (Ω) about the X-axis, we optimize the nonquadratic shape of normalized images. For resampling, the bilinear interpolation method is employed, which may introduce blurring effects into the normalized images.

5. CONCLUSIONS

We describe the procedure for obtaining the normalized images from exterior orientation after absolute orientation. We also present a direct solution to compute the coefficients of the projective transformation, and show a way to compute the inverse transformation parameters directly, without repeating the transformation backward.

The procedure of computing normalized images is successful and operational. The normalized images, with removed distortion caused by the scanning device, are in epipolar geometry with respect to the object space. Since scan lines are epipolar lines in normalized images, the automatic matching procedure for conjugate points will be performed on the same scan lines. The 3-D surface in object space can be reconstructed directly by using matched conjugate points.

6. ACKNOWLEDGMENTS

Funding for this research was provided in part by the NASA Center for the Commercial Development of Space Component of the Center for Mapping at The Ohio State University.

7. REFERENCES

Chen, Y., and Schenk, T., 1992. A rigorous calibration method for digital camera. Proc. 17th ISPRS Commission III., Washington, D.C.

Cho, W., 1989. Transformation of the pixel system to the vertical position and resampling. Technical Notes in Photogrammetry, No. 3., Dept. of Geodetic Science and Surveying, The Ohio State University.

Kreiling, W., 1976. Automatische Herstellung von Höhenmodellen und Orthophotos aus Stereobildern durch digitale Korrelation. Dissertation TU Karlsruhe.

Li, J., and Schenk, T., 1990. An accurate camera calibration for the aerial image analysis. Proc. 10th International Conference on Pattern Recognition., NJ, Vol 1, pp. 207-209.

Schenk, T., 1990. Computation of epipolar geometry. Technical Notes in Photogrammetry, No. 5., Dept. of Geodetic Science and Surveying, The Ohio State University.

AERIAL IMAGE MATCHING BASED ON ZERO-CROSSINGS

Jia Zong
Jin-Cheng Li
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

One of the basic tasks in digital photogrammetry is to find conjugate points in a stereo pair and to reconstruct the 3-D object space (DEM). Edges play an important role in that they may indicate breaklines in the surface. We use the LoG operator to extract edges (zero-crossings). In this paper the problem of matching zero-crossings is addressed. Zero-crossings computed from one image are matched with area-based method. A hierarchical matching approach is adopted by the use of both, interpolated disparity maps at each level of the image pyramid, and knowledge from image analysis at very high level of image pyramid. The method is particularly suited for matching aerial images for the purpose of restructing surfaces of urban areas.

1. INTRODUCTION

One of the major research areas in digital photogrammetry is image matching for reconstructing the three-dimensional surface of the object space. This process involves a fundamental problem of stereo vision: to find corresponding points in an stereo-pair. Once correspoinding points are determined their three-dimensional positions can be easily computed, and the surface is obtained from matched points by interpolation.

Two methods are commonly used in image matching: area-based image matching and feature-based image matching. Aera-based matching is predominantly used for the object space (DEM). Here, the corresponding points are found by comparing the gray levels of correponding areas (image patches) in a image stereo-pair. This approach is favored in photogrammetry because of its high accuracy potential. However, there are several critical factors that need special consideration in area-based matching. For example,

- good approximations for the corresponding image patches are required
- matching in flat area or of sharp relief changes is extremely hard and it produces bad results. Both cases usually occur in urban aerial images
- recovering the surface, especially in urban areas, from randomly distributed matched points is difficult
- the reliablity control of the matching is low
- computations are intensive

Some of these problems are avoided in feature-based matching. Here, properties (features) derived from the gray levels are matched, rather than gray levels themselves. This method usually proceeds in two steps, the first being a local similarity matching such as comparing the parameters of detected features, and the second being a global matching such as checking continuity constraints. Features detected monocularly may differ and may include spurious data due to differences in reflectance which are not caused by the surface shape. This problem is quite acute in large-scale aerial images of urban areas. Another point to bear in mind is that matched features (e.g. edges) do not necessarily consist of conjugate points. In general, feature-based matching is more robust and less computationally intensive. But most important, matched features are more meaningful than randomly matched points if it comes to automatically analyzing image.

The motivation for this research is to combine the merits of both area-based and feature-based matching methods. First, edges or zero-crossings (ZC) are detected as features. The edges are more likely to represent prominent features of the surface, such as breaklines. Instead of matching edges as entities as described in [Schenk *et. al.* 1991], here we match every point of an edge by correlation. A match is accepted if it satisfies epipolar geometry and figural continuity constraints. This strategy proved to be quite successful [Li *et. al.* 1990]. In order to cope with urban areas where correlation must be applied with caution, we have modified the strategy by including a surface analysis step in the hierarchical matching scheme. At each level of the image pyramid an interpolated disparity constraint map is generated which provides the necessary approximations for the next level of matching. Knowledge gained from previous levels is used to guide matching in the subsquent level of image pyramid. With this new strategy the success rate of matching aerial images of complex urban scenes is greatly improved.

2. FEATURE EXTRACTION

Detecting zero-crossings as features for matching was first proposed by Marr and Poggio [Marr and Poggio, 1979] on the basis of a computational theory on the human stereo vision. Mathematically, zero-crossings are obtained by applying the convolution operator $\nabla^2 G$ over the image $f(x, y)$ as

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp(-\frac{r^2}{2\pi\sigma^2})$$

$$\nabla^2 G(x,y) = (\frac{r^2 - 2\sigma^2}{2\pi\sigma^2}) exp(\frac{-r^2}{2\sigma^2})$$

$$f^{'}(x,y) = \nabla^2 G(x,y) * f(x,y)$$

where $G(x,y)$ is a Gaussian filter, $\nabla^2 G$ is the Laplacian of a Gaussian (called LoG), and $f(x,y)$ is the image gray level function. Convolution us denoted by $*$, and $r = (x^2 + y^2)^{1/2}$ implies that the operator is rotationally symmetric. The advantage of the LoG operator is that it combines smoothing and differentiating into one operator. Moreover, it is localized in space and frequency domains. The filtered image $f^{'}(x,y)$ is divided into positive and negative regions with average frequency of $\sqrt{2}/\sigma$. The boundaries of these regions are the zero-crossings. Zero-crossings occur wherever the gray levels change sharply. The degree of change can be described by the first-derivative of the gray level function, or the gradient of the gray levels. Zero-crossings are separated by an average distance which is equal to the window size of LoG operator, the diameter of positive central region of LoG curve $\omega = 2\sqrt{2}\sigma$. The larger the window size, the larger the dislocalization of detected zero-crossings from the real boundaries.

Edges in aerial images represent object boundaries or markings (e.g. shadows). Many object boundaries correspond to surface breaklines. The LoG operator is applied to both left and right image to obtain the zero-crossings. Several parameters are chosen to control feature detection. The window size ω of LoG operator is selected according to the quality and the scale of the images to ensure surface feature detection. In order to supress noise or less important features, a threshold value t is chosen according to the distinctness of the zero-crossing. The result of applying LoG operator are two binary images. Zero-crossings as feature entities are obtained in the left images as following:

- The location of zero-crossings is obtained by an edge following algorithm. The connected zero-crossing points form the zero-crossing curve as feature entity.

- Then each zero-crosssing curve is segmented using local curvature maxima as end points of each segment.

As a result, edges are detected as individual zero-crossing curves connected by several possible segments.

3. CORRELATION MATCHING

The flow chart of the matching scheme is shown in Fig. 1. Like most area-based matching algorithms, epipolar geometry is employed to constrain the searching to one dimension [Cho *et. al. 1992*]. At each level of the image pyramid, the image patches are first enhanced since area-based matching methods require good image quality. Next, zero-crossings are determined in both images. For each zero-crossing point in the left image the corresponding point on the right image is found along the epipolar (scan) line by area correlation. right image zero-crossings only help to define the searching window for the correlation matching. The matching is performed in two steps: initial point to point correlation, and figural continuity checking acceptance criterion. During the initial matching, points with maximum correlation values larger than the preset threshold value are selected as matched points. The key point here is to find a good approximation of the search window in the right image. This is accomplished by using the disparity constraint map at each level of the image pyramid. Once matching is completed, an interpolated disparity image is generated, providing the approximations needed for the next level matching . In the highest level of the image pyramid, knowledge gained from surface analysis is also fed back to the matching process through the use of the disparity map. After the initial matching, all matched points must satisfy the figural continuity constraint for final acceptance as conjugate points.

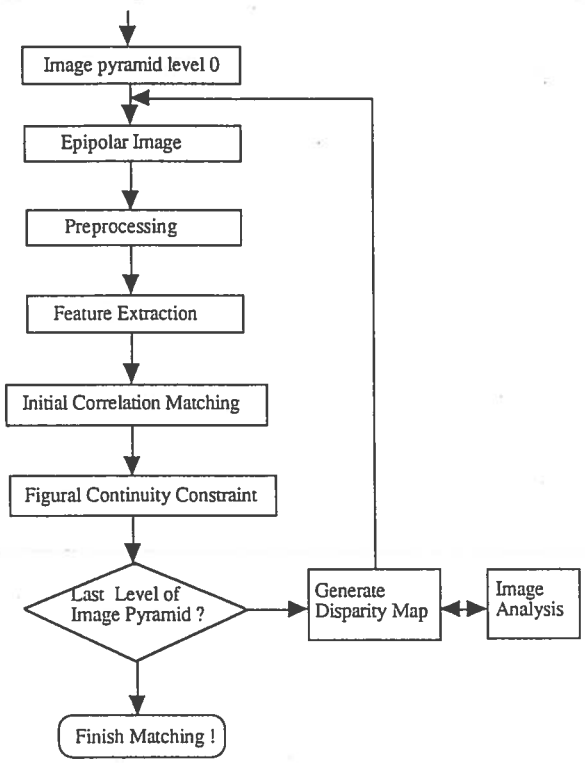


Fig. 1. Flowchart of matching scheme

3.1. Hierarchical Disparity Constraint

The search window in correlation matching is defined by two parameters: location and size. Obviously, the window size depends on the goodness of the approximations. We determine the window size (search range) dynamically based on the disparity map. If the search window is close to a zero-crossing contour detected in the right image, it is adjusted accordingly, because this zero-crossing is likely to be the conjugate point.

A crucial step in any matching system is the approximation of the matching location (center of search window). At the top of the image pyramid we have two options. One is using an average disparity value for all matching positions. This average approximation is computed from the matched points generated during automatic orientation [Schenk *et al.* 1992, and Zong *et al.* 1991]. The second option is to convert the matched points from the automatic orientation to disparity values. These disparity values are then interpolated to generate a disparity map to determine the location of the search window. After the matching process is completed in one level of the image pyramid, the disparity map is updated to ensure that better approximations in the next level are available. This is quite important, particularly in urban areas where the disparity values may abruptly change. It should be noted that the disparity map always corresponds to the resolution of current level in the image pyramid.

3.2. Surface Analysis in High Level Matching

The use of the disparity map provides not only good approximations for correlation matching but also a closer surface approximation after each level of matching. In higher levels of the image pyramid such a surface can even provide a lot of three-dimensional object information. This information can then be used for real object surface analysis, as discussed in [Wang *et. al.* 1992]. On the other hand, the information gained from the 3-dimensional analysis can be fed back to guide the matching.

One of the most difficult matching cases are urban aerial images in which there exist man-made features with extreme height, such as tall buildings or chimneys. The deformations and disparities of such features in stereo images can be very large causing the matching to be either incomplete or unsuccessful. The solution here is to analyze the disparity map. As one application of 3D feature analysis discussed in [Wng *et. al.* 1992], a contour map can be generated after segmenting a disparity map. The following rules are implemented to superimpose knowledge to the existing disparity map and used to guide the program to find potential high features.

- A cluster of close-centered contours indicates a potential hump
- If the inner disparity values are much larger than the outer ones, a potential high hump is indicated

- For a potential high hump, information is fed back to guide a future matching
- The boundary of a potential high hump is the second closed outer contour since the first one may indicate boundary of the environment
- The potential disparity values inside the selected boundary are the average disparity values of the matched points inside the boundary
- The obtained disparity values of the potential hump are appended to the matched data and a new disparity map is interpolated

3.3. Figural Continuity Constraint

The figural continuity criterion implies that the disparity values along zero-crossings must be continuous. We implemented the figural continuity constraint by performing a Hough transformation of all the matched points belonging to one segment of a zero-crossing contour. Continuous disparity values show up as clusters in the Hough space. If fewer than 15 points fall into the cluster a flag is set to indicate that there is no corresponding zero-crossing segment. Finally, the location of the corresponding segment in the right image is determined by the Hough transformation and the correlation threshold.

4. EXPERIMENTS

The matching algorithm was tested with several pairs of aerial photographs. In this paper, we present the results from stereo-images (193, 195) taken over the campus of the Ohio State University. This model represents a very typical urban area of all the different models tested, it was the most difficult one. The photo scale here is approximately 1 : 4000. The diapositives were scanned to a resolution of 30μ pixel size by Intergraph Corporation using the PhotoScan. However, we only used a resolution of 60μ which yielded a 4096×4096 pixel image. The ground coverage of a pixel is approximately 25×25cm.

Fig. 2 and 3 show the original aerial images at the coarsest resolution of 512×512. Zero-crossings were first detected with the LoG operator ($\omega = 5$), and then matched with a single average disparity approximation. The range of the search window was set to 10 pixels in order to avoid wrong matching. The matched zero-crossings are shown in Fig. 4 and 5. A disparity map was interpolated by using Modular function on Intergraph workstation. The result is shown in Fig. 14. which outlines the surface of the whole overlapping area of the model. Some humps are clearly visible.

The interpolated disparity map was then converted into an image of 512×512 resolution and the disparity values were treated as graylevels. Fig. 6 and 7 show matched zero-crossings of a 512×512 image patch selected from stereo images of $1K \times 1K$ resolution. Fig. 15 shows the interpolated 3D disparity map. The humps are now more prominent. Fig. 8 and 9 depict matched zero-crossings of a 512×512 image patch from $2K \times 2K$ resolution images. The interpolated disparity values are shown in Fig. 16. The surface is fairly well approximated at this level.

The procedure is repeated at the finest resolution, again with an image patch size of 512×512 pixels. In this example the disparity values range from 0 to 118. The segmented disparity image resulting from matching is shown in Fig. 12 where the hump is clearly indicated. Fig. 10 and 11 show the matching results superimposed to the resampled images, while Fig. 17 and 13 show the interpolation of the final matching results in the disparity map and the three-dimensional object space, respectively.

5. CONCLUSION

The presented matching scheme combines the merits of both area-based and feature-based matching methods and proved succussful in the aerial image matching. The use of a hierarchical approach and surface approximation makes this approach particularly suited for urban area image matching. It is found that the precise detection of prominent features is helpful for

recovering the object surface. The reliability of the correlation matching is improved by the employing the figural continuity constraint. Finally, this matching scheme shows a great potential for object surface analysis and reconstruction.

6. REFERENCES

Cho, W., M. Madani and T. Schenk, 1992. Resampling Digital Image to Epipolar Geometry. *Proceedings of ISPRS*

Li, J.C., and T. Schenk, 1990. Aerial Image Matching Based on Zero-crossings and Least-squares Correlation, *Technical Notes in Photogrammetry No. 9, Department of Geodetic Science and Surveying, The Ohio State University*

Marr, D. and T. Poggio, 1979. A Computational Theory of Human Stereo Vision, *Proc. Royal Society*, B 204 301 – 328

Schenk, T., J.C. Li and C. Toth, 1990. Hierarchical Approach to Reconstruct Surfaces by Using Iteratively Rectified Imagery , *Proceedings of the ISPRS*, Commission V, 28(5/1) : 464 – 470

Schenk, T. and C. Toth, 1992. Conceptual Issues of Softcopy Photogrammetric Workstations. *Photogrammetirc Engineering and Remote sensing*, Vol. 58, No. 1, Jan. 1992, pp. 101-110

Wang, Z. and T. Schenk, 1992. 3D Urban Aera Surface Anlysis *Proceedings of ISPRS*

Zong, J., J.C. Li and T. Schenk, 1991. Application of Forster Interest Operator in Automatic Orientation System, *Proceedings of the ASPRS-ACSM Annual Convention*, 5:440-448

Fig.2. Original left epipolar image (512×512)

Fig.3. Original right epipolar image (512×512)

Fig.4. Matched ZCs of L-image (512×512)

Fig.5. Matched ZCs of R-image (512×512)

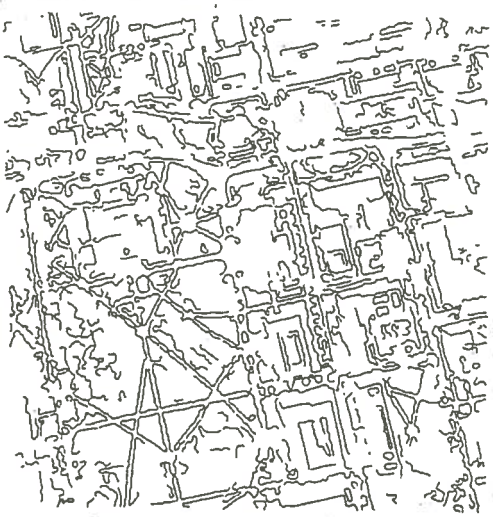


Fig.6. Matched ZCs of L-image ($1K \times 1K$)

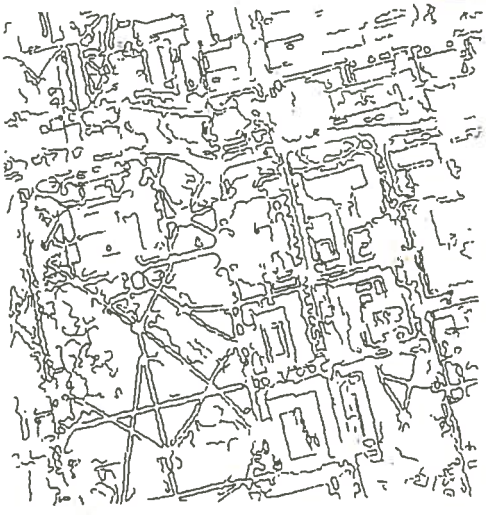


Fig.7. Matched ZCs of R-image ($1K \times 1K$)

Fig. 8. Matched ZCs of L-image ($2K \times 2K$)

Fig. 9. Matched ZCs of R-image ($2K \times 2K$)

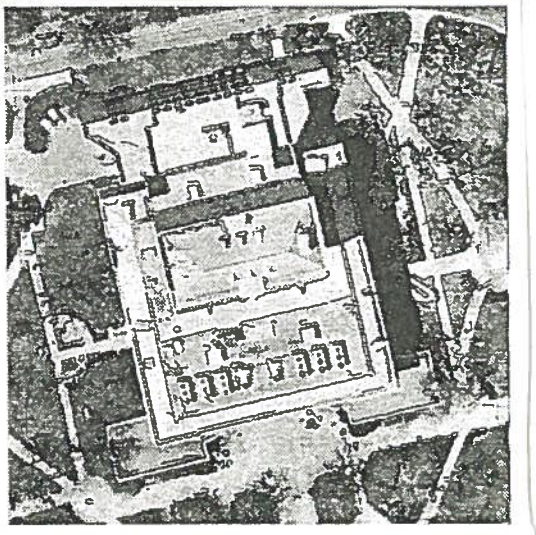


Fig. 10. Matched ZCs overlapping in L-image ($4K \times 4K$)

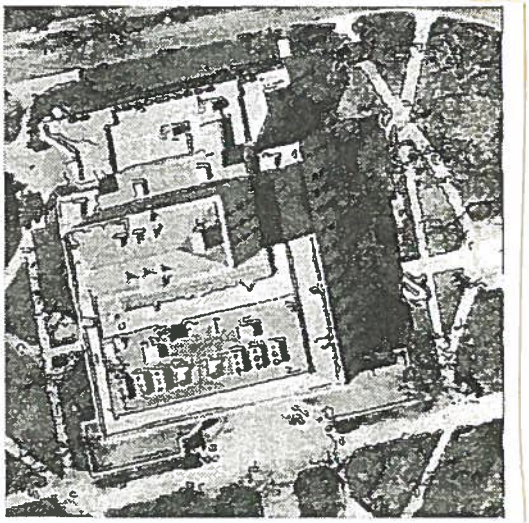


Fig. 11. Matched ZCs overlapping in R-image ($4K \times 4K$)

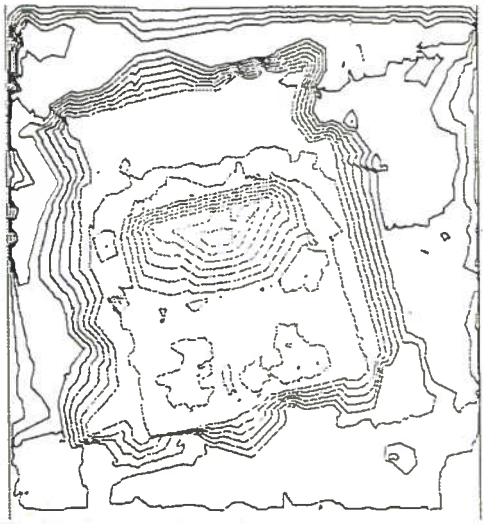


Fig. 12. Segmented disparity map from $4K$ matching

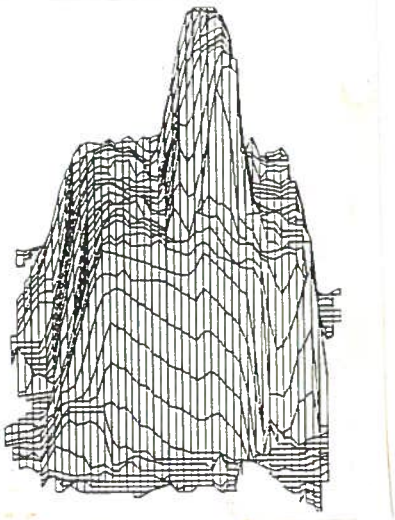


Fig. 13. DEM in object space from the final matching ($4K$)

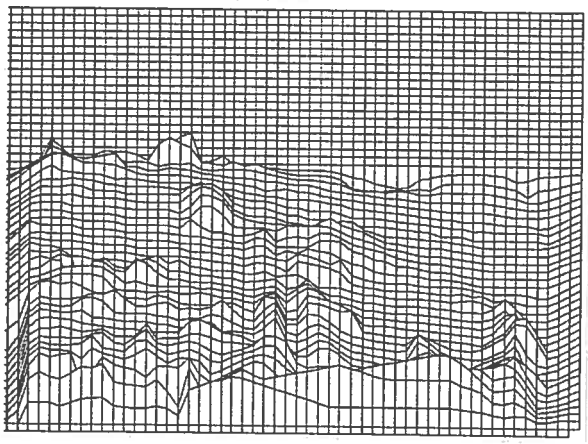


Fig. 14. Interpolated disparity map from 512 res. matching

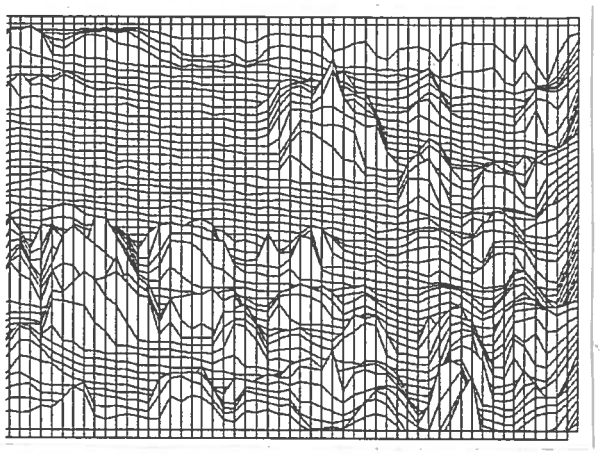


Fig. 15. Interpolated disparity map from 1k res. matching

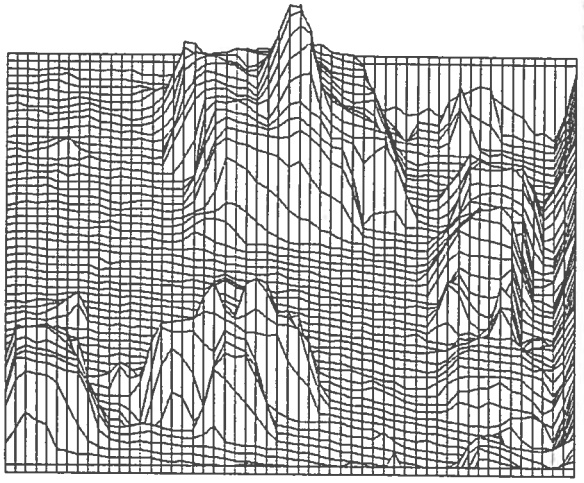


Fig. 16. Interpolated disparity map from 2k res. matching

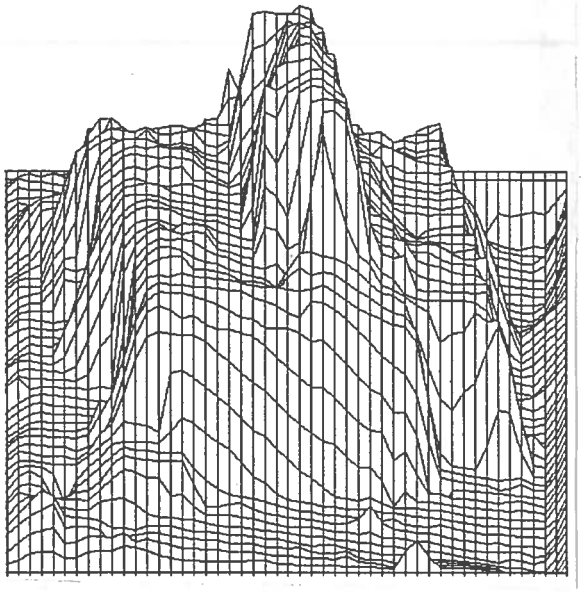


Fig. 17. Interpolated disparity map from 4k res. matching

ON THE INTERPOLATION PROBLEM OF AUTOMATED SURFACE RECONSTRUCTION

Raid Al-Tahir
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

Automatic surface reconstruction entails two major problems: Determining conjugate points or features (matching) and densifying the matched points in object space (interpolation). The two tasks are usually performed sequentially in a hierarchical approach, without interacting with one another. In order to improve the success rate and the reliability of automated surface reconstruction, particularly in large-scale urban areas, the matching on subsequent levels must take into account the results from densifying and analyzing the surface. In this paper we focus on a surface interpolator that produces as realistic surface representation as possible. The interpolation and surface analysis may give clues about surface discontinuities and occlusions – a vital feedback for the matching process on the next level in the hierarchical approach.

1. INTRODUCTION

The main objective of digital photogrammetry is to collect enough information to model the portion of the real world that has been photographed. Two kinds of information are of major interest to accomplish that goal; surface topography, represented by Digital Elevation Model (DEM), and objects on the surface (natural or man-made) which are characterized as discontinuities in the surface. Besides being an essential intermediate step for object recognition, reconstruction of a portion of the earth's surface is the end product for digital photogrammetry.

Automatic surface reconstruction entails two major problems: determining conjugate points or features in the images (matching), and densifying the matched points in object space (interpolation). The two tasks are usually performed sequentially in a hierarchical approach, without interacting with one another. In order to improve the success rate and the reliability of automated surface reconstruction, particularly in large-scale urban areas, the matching on subsequent levels must take into account the results from densifying and analyzing the surface.

This paper is a part of ongoing research focusing on the process of surface interpolation and analysis. The purpose of this paper is to define the tasks for such a process. The paper reviews previous works that have been done in the related fields. The emphasis is on the applicability of suitable for an automated surface interpolation.

2. OSU SURFACE RECONSTRUCTION SYSTEM

Due to the large amount and variety of information in the aerial images, the success of any image processing operation cannot be guaranteed. This is especially the case of large-scale urban scenes because occlusion is more frequent, and the visible surface is less smooth. The only alternative to constrain the processes is to adopt a scale-space approach that proceeds hierarchically from the lowest resolution for a stereo pair to the finest. OSU surface reconstruction (Schenk & Toth, 1992) is such hierarchical approach. It consists of several modules that are executed in an iterative fashion (Figure 1). Each level of the process aims at refining the geometry of the images and improving the surface representation.

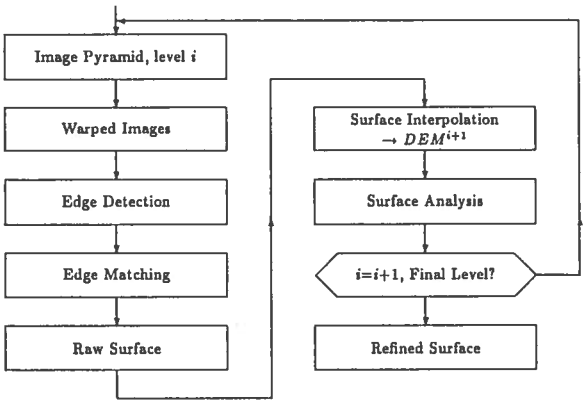


Figure 1: Outline of OSU surface reconstruction system.

In the OSU surface reconstruction system, the process starts by having two conjugate images sampled at the lowest level of resolution. The orientation of these images is obtained through edge detection and matching. The results of this step are the orientation parameters, as well as a set of highly reliable matched points. The raw surface is then constructed by computing the 3-D object space coordinate for the set of points. These points are sparsely and irregularly distributed. Thus, a dense surface representation (DEM) must be interpolated for. A DEM, tesselated at the next higher level of resolution, is essential for surface analysis, and for the subsequent cycles. The final step is surface analysis for hypothesis generation and verification concerning potential break lines and surface segmentation.

A new cycle starts with sampling the original stereo pair at the subsequent level of resolution, and warping the left and right images with respect to the interpolated surface. The whole process is repeated until the final refined surface is reached. At each level, images are rectified, the matching accuracy and reliability are improved, and a better surface representation is obtained. At the last level, the matching vectors vanish, the warped images become orthophotos, and the true surface is re-

constructed.

From this overview, it is clear that one of the objectives of surface interpolation is to construct as a realistic surface representation as possible. This task is crucial for the success of matching on subsequent levels. The search for a match is performed by centering a correlation window over a point of a zero-crossing contour in one image. On the other image, the search window is placed and shaped according to the expected depth range in that area (Schenk & Toth, 1991).

The other goal of the surface interpolation is to provide information for the surface analysis. It is important that the interpolator does not introduce new characteristics to the surface other than what is derived from the observations. Creating new maxima or minima in the surface is an example for undesired side effects of interpolation. Additionally, the surface interpolator should not smear essential surface shape characteristics. Such a situation may occur when a smooth surface is interpolated over observations on break lines.

3. SURFACE INTERPOLATION

The problem of surface fitting consists of taking a region containing a list of function values, and finding a function on this region that agrees with the data to some extent and behaves reasonably between data points (Lancaster & Salkauskas, 1986). The accuracy that can be obtained from a fitting process depends on the density and the distribution of the reference points, and the method. Data points are arranged in various distribution patterns and densities. Accordingly, surface fitting methods designed for one case differ from those designed for dealing with other distribution patterns.

There are several criteria for classifying surface fitting methods. The first criterion is the closeness of fit of the resulting representation to the original data. Thereby, a fitting method can be either an interpolation or an approximation. Interpolation methods fit a surface that passes through all data points. Approximation methods construct a surface that passes near data points and minimizes, at the same time, the difference between the observed and the interpolated values.

Another criterion is the extent of support of the surface fitting method; a method is classified as a global or a local one. In the global approach, the resulting surface representation incorporates all data points to derive the unknown coefficients of the function. By doing so, some of the local details submerge in the overall surface, and editing one point affects all distinct points. With local methods, the value of the constructed surface at a point considers only data at relatively nearby points. Thus, the resulting surface emphasizes the small-scale trends in the data (Watson, 1992). Many global schemes can be made local by partitioning the original domain into subdomains.

Yet another criterion for classifying interpolation methods is their mathematical models. Surface interpolation methods are divided into three main classes; weighted average methods, interpolation by polynomials, and interpolation by splines.

3.1 Weighted average methods

These methods use a direct summation of the data at each interpolation point. The value of the surface at a non-data point is obtained as a weighted average of all data points. The weight is inversely proportional to the distance r_i. Shepard's method may serve as an example. Here, the value of a point is evaluated as

$$f(x,y) = \begin{cases} \dfrac{\sum_{i=1}^{N} F_i/r_i^{\mu}}{\sum_{i=1}^{N} 1/r_i^{\mu}}, & \text{when } r_i \neq 0, \\ F_i, & \text{when } r_i = 0. \end{cases} \quad (1)$$

Weighted average methods are suitable for interpolating a surface from arbitrarily distributed data. However, one drawback is the large amount of calculations, especially for many data points. To overcome this problem, the method is modified into a local version. A smaller subset of data is selected for each non-data point based on a fixed number of points, or a fixed area. The problem now is to define proper parameters (e.g. the variable μ in equation (1)).

3.2 Interpolation by polynomials

A polynomial p is a function defined in one dimension for all real numbers x by

$$p(x) = a_0 + a_1 x + \cdots + a_{N-1} x^{N-1} + a_N x^N, \quad (2)$$

where N is a non-negative integer and $a_0, \ldots, a_N$ are fixed real numbers. Generally, fitting a surface by polynomials proceeds in two steps. The first one is the determination of the coefficients of the polynomial based on the set of data points and the criteria controlling the fit of the polynomial function. Then, using the computed parameters, the second phase evaluates the polynomial to obtain values of the fitted surface at given locations.

Piecewise polynomials are the local version for surface fitting with polynomials. This approach works well with irregularly spaced data. The general procedure for surface fitting with piecewise polynomials consists of the following operations:

1. partitioning the surface into patches of triangular or rectangular shape, the vertices of which are the reference points.
2. fitting locally a leveled, tilted, or second-degree plane at each patch, using one or more terms of the polynomial.
3. solving the unknown parameters of the polynomial. To enforce continuity (and smoothness) along the joining sides of neighboring patches, partial derivatives must have been estimated at each reference point.

Least squares fitting by polynomials performs well if many points are available and the surface has fairly simple form (Hayes, 1987). On the other hand, interpolation by polynomials with scattered data causes serious difficulties, one of which is a singular system of equations due to data distribution (e.g. data lie on a line). Another problem is an ill-conditioned normal equation system as is the case of consecutive intervals that contain no data. Yet another problem in using polynomials is their tendency to oscillate, resulting in a considerably undulating surface.

3.3 Interpolation by spline functions

A spline is a piecewise polynomial function defined on contiguous segments. In defining a spline function, the continuity and smoothness between two segments are constrained at the interior knots by demanding the existence of certain derivatives. For example, a spline of degree n has n-1 derivatives at the knots, denoted by C^{n-1}.

Bicubic splines, which have continuous second derivatives (i.e. C^2), are commonly used for surface fitting. The solution is obtained by a least-squares approach or the tensor product of orthogonal functions. With increasing number of data points, problems with computing efficiency and accuracy may occur. B-splines are also frequently used for surface fitting. They are characterized by their finite support, which is the interval over which the function is non-zero. Limiting the support of a spline changes the normal equation into a band form. Thereafter, the amount of computations is reduced by a factor of (number of knots/4)^2 (Hayes, 1987).

Bicubic splines and B-splines work best in the case of gridded or uniformly-distributed dense data (Hayes, 1987). However, rank-deficiency in the system of equations becomes a serious problem when applying these approaches to scattered data. Because of data distribution, data points may not lie in the support region of splines. Another situation rises when the data are clustered in one region creating a set of linear equations of marginal differences, thereby producing near singularity.

Nodal basis-functions are another sub-group of methods for surface fitting with splines. The general procedure in this approach consists of defining a set of basis functions and the corresponding data points. Each basis function is centered over a data point (node). The interpolation spline function then is a linear combination of the basis functions. The advantage in using such an approach is that knowledge about spline locations (knots) is not required. Another advantage is that values at the nodes of a regular grid are found directly instead of the two step approach mentioned earlier (Briggs, 1974).

Thin plate splines are derived from the nodal basis-functions. These splines are also called "minimum curvature splines" since they are obtained by minimizing the total curvature of cubic spline s

$$\iint \left(\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} \right)^2 dx\, dy. \qquad (3)$$

The same form can be obtained by solving the small deflection equation of an infinite plate that deforms by bending it only. The displacement u due to a force f_i acting at N points is represented by the differential equation (Briggs, 1974)

$$\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = f_i \quad \text{at observations,}$$
$$= 0 \quad \text{otherwise.} \quad (4)$$

Adopting the physical analogy, depth data is represented by a set of vertical pins scattered within the region; the height of an individual pin is related to the elevation of the point. Fitting a surface is then analogous to constraining a thin (elastic) plate to pass over the tips of the pins (Figure 2).

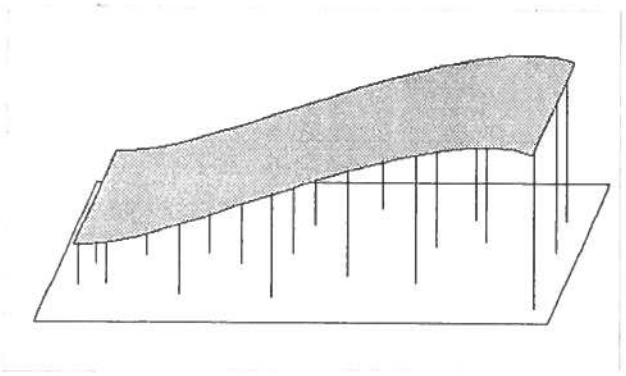


Figure 2: Fitting thin plate over pins.

One method for solving the differential equation is by finite differences or finite elements. Following this approach, the discrete interpolation becomes a repeated passage of a set of simple masks, such as the following mask for elements within a grid:

$$\begin{bmatrix} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{bmatrix} \quad (5)$$

3.4 Surface interpolation by regularization.

A problem is well-posed if a solution exists, is unique, and depends continuously on the initial data. It must also be well conditioned to ensure numerical stability (robust against noise) (Poggio et al., 1985). Shorter than these conditions, the problem is considered ill-posed. Reconstruction of the visible three-dimensional surfaces from two-dimensional images is an ill-posed problem because some information is lost during the imaging process (projecting 3-D into 2-D) (Poggio et al., 1985). Other reasons are the noise and erroneous, inconsistent, and sparse measurements (Terzopoulos, 1985).

Regularization is the frame within which an ill-posed problem is changed into a well-posed one (Poggio et al., 1985). The class of possible solutions is restricted by introducing suitable a priori knowledge, which in the case of surface interpolation is the continuity of the surface. The problem is then reformulated, based on the variational principle, so as to minimize an energy function E constructed from two functionals. The first one measures the smoothness of the solution S, while the second one, D, provides a measure of the closeness of the solution to the observations. The two measures are combined to form the energy function $E = S + D$. Applied to the surface reconstruction problem, the energy function can be written as

$$\int\int [f''(x,y)]^2 dx dy + \lambda \sum [f(x_i, y_i) - d_i]^2. \quad (6)$$

In practice, the function in the integration is either a thin-plate spline ($f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2$), a membrane ($f_{xx}^2 + f_{yy}^2$), or a combination of both. The variable λ is the regularization parameter which controls the influence of the two functionals. If λ is very large, the first term in the integral heavily affects the solution, turning it into interpolation (close to data). On the other hand, if λ is small, the solution emphasizes the smoothness of the surface.

4. DISCONTINUITY DETECTION

There are only a few methods which try to detect discontinuities in the surface. Grimson and Pavlidis propose detecting discontinuities before interpolating the surface to overcome the problem of oscillations in the fitted surface (Grimson & Pavlidis, 1985). The main idea for this approach is to fit locally a simple surface (plane) to the data and examine the distribution of the residual error. If it appears to be

"random", then the hypothesis of no discontinuity is accepted. If there is a systematic trend, then a discontinuity of a certain type is hypothesized. Discontinuities are subdivided into various types, each of which is characterized by a certain combination of change in magnitude and sign of the residual. Once a discontinuity is detected, the surface is broken down into smaller regions, and the surface reconstructor is passed over each of them.

The second approach, proposed by Terzopoulos (Terzopoulos, 1985), is related to the energy function of a thin plate. The thin plate surface over-shoots constraints near the discontinuity causing a sign change of the bending moments at surface inflections. Depth discontinuities are detected and localized by examining the bending moments in the interpolated surface. Changing control parameters within the energy function allows the surface to crease and fracture at the detected discontinuities and reduce the total energy.

Another approach we investigated for detecting discontinuities is based on the concept of a "line process" introduced in (Geman & Geman, 1984. A line process is a set of variables located at the lines which connect the original lattice (pixels or grid cells) (Figure 3). The purpose of a line process is to decouple adjacent pixels and reduce the total energy if the values of these pixels are different. In such a case, the variable of the line process associated with these pixels is set to one, otherwise it is set to zero.

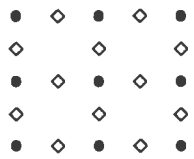


Figure 3: Dual lattice of depth (•) and line (◇) elements.

Eventually, breaking the surface into small pieces around each data point will result in the lowest energy state. To avoid this, a penalty α should be paid (in terms of energy) when a break line is introduced. Thus, a break line will only be introduced when paying the penalty is less expensive than not having the break line at all. The penalty function takes the form $P = \alpha l_i$, where l_i is the line process. This function is added to the original energy function, changing the problem into minimizing

$$E = S + D + P. \tag{7}$$

The result is a combination of a continuous function for the surface and a discrete one for the lines. This combination allows surface reconstruction and discontinuity detection at the same time. However, E is a non-convex function that has many local minima.

One proposal to solve the non-convex function is to adopt a deterministic approach. The line process P is merged with the interpolation function S (Blake & Zisserman, 1987). The modified function is expressed in one dimension as

$$g(u_i - u_{i-1}) = \lambda^2(u_i - u_{i-1})^2(1 - l_i) + \alpha l_i. \tag{8}$$

The resulting function controls the interaction between neighboring grid cells. Such a function prefers continuity in the surface, but allows occasional discontinuities if that makes for a simpler overall description – a theme called "weak continuity constraints".

The modified configuration is then solved by the graduated non-convexity algorithm. The non-convex function E is gradually approximated by a convex one through a family of p intermediate functions. The parameter p represents a sequence of numbers ranging from one to zero. The function $E^{(1)}$ is a crude approximation to the non-convex function. However, as p goes to zero, $E^{(p)}$ becomes closer to the original non-convex one. The neighbour interaction function is also modified into a function of λ, α, and p.

5. EXPERIMENTS AND CONCLUSION

For experimental purposes, we designed synthetic data representing a set of irregular blocks in a small region. Depth information is arranged in a fashion that mimics the pattern of the results of the matching process in the surface reconstruction system. Thus, depth values were provided for some points on, and near by,

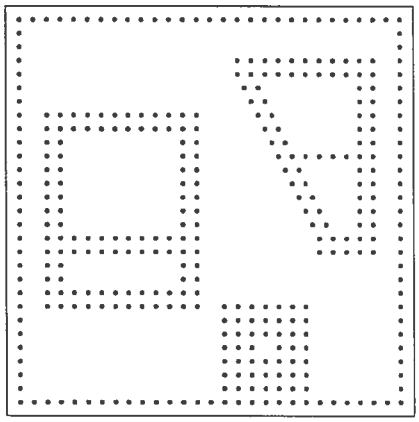


Figure 4: Distribution of synthetic data points.

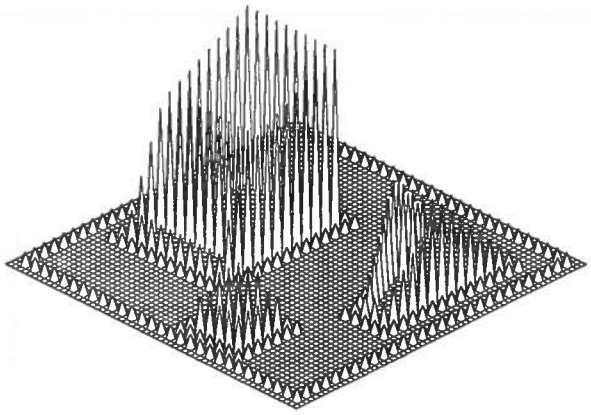


Figure 5: 3-D representation of synthetic data points.

the edges of the blocks and the edge of the region as shown in figure 4. Figure 5 is a 3-D representation of these points. The location and value of a data point is represented by a peak, while no data points are set to zero.

We evaluated the interpolation methods according to the following criteria:

1. Interpolated surface must be plausible compared to the visible surface in the real world.
2. The interpolation method must not jeopardize clues for surface analysis.
3. The method should be able to utilize a priori information on break lines.
4. The method must be suitable for automation. No human interaction should be necessary to correct parameters.
5. Reasonable demand on computer resources, i.e. time, memory, and storage.

Matching aerial images typically renders a large number of data points, especially at the finer resolutions. Therefore, we have excluded all methods of least square fitting by polynomials or splines because of computational considerations. These methods would lead to a huge system of equations (in the worst case is one equation per point). In addition, having sparse data increases the risk of deficiency in the normal equation. Fitting a surface by piecewise polynomials, furnished with proper triangulation algorithm, stands a better chance for more efficient and realistic surface interpolation. However, the user must identify the set of break lines prior to the interpolation. Otherwise, a peculiar surface representation would be obtained.

The methods of weighted average are better suited for handling sparse data. Besides, they do not introduce new global extrema in the surface. On the other hand, there is no established automatic strategy for defining the data subset for a point. Another concern is the fact that

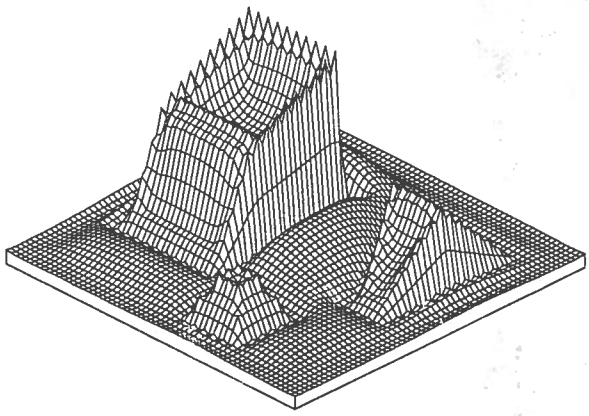


Figure 6: Surface interpolation by weighted average method.

no a priori information about break lines can be included. Therefore, the value of a point is computed based on data across break lines, creating undesired artifacts. Figure 6 shows the result of applying the weighted average method on the test data. The interpolated surface cannot be considered realistic.

None of these methods provides explicit information for surface analysis. This quite different for fitting a surface by a thin plate (or membrane). Adopting the analogy of a physical model allows exploring the mechanics of such model. Mechanical concepts, such as stress and bending moments of a plate provide the means for detecting break lines. Both models of thin plate and membrane are capable of achieving surface interpolation and break lines detection. Judging from figures 7 and 8, the membrane produces a more realistic surface that the thin plate model.

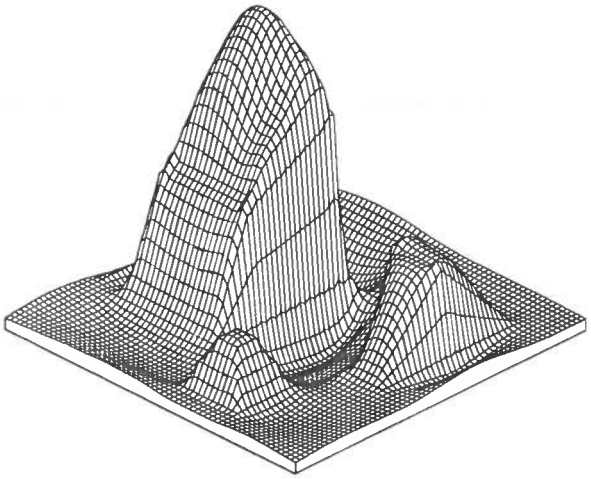


Figure 7: Surface interpolation by thin plate splines.

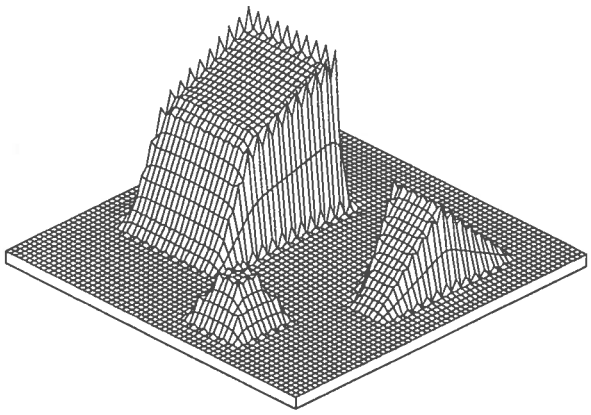


Figure 8: Surface interpolation by a membrane.

Figure 7 represents the interpolated test data by a thin plate. The problem of over-shooting between data points is clearly noticeable. Figure 8 shows the interpolation by a membrane. Here, the problem is interpolating between high frequency features. This is avoided by using the weak continuity constraints. Interpolation by a weak membrane is shown in figure 9.

The discontinuities are now detected during the surface interpolation. Figure 10 shows the detected break lines superimposed on the surface.

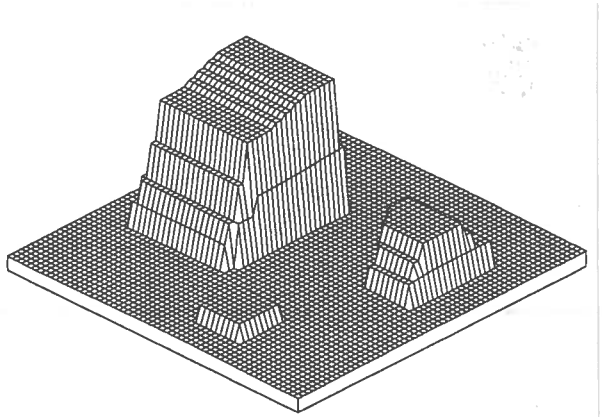


Figure 9: Surface interpolation by a weak membrane.

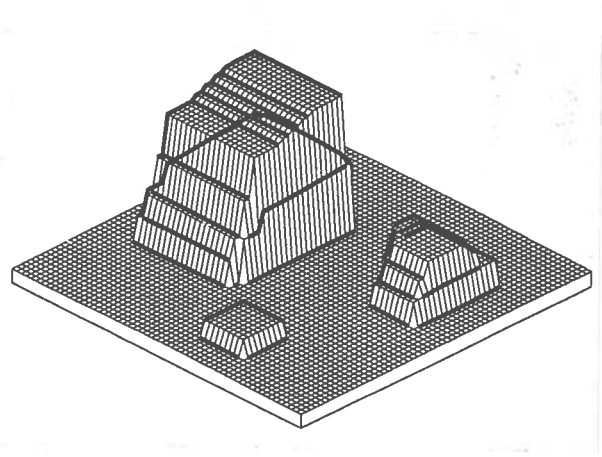


Figure 10: Detected break lines in the surface.

Ongoing research is addressing the following issues:

- Increasing the degree of automation of surface interpolation.
- Subpixel accuracy in the determination of a break line.

- Defining the means to convey discontinuity information to other modules and levels.
- Integration of other cues for discontinuity, such as the residuals between successive levels of surface representation.

References

[1] Blake, A. and Zisserman, A. 1987. Visual Reconstruction. MIT Press. MA.

[2] Briggs, I. 1974. Machine contouring using minimum curvature. Geophysics. 39 (1): 39-48.

[3] Geman, S. and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans on Pattern Analysis and Machine Intelligence. PAMI-6 (6).

[4] Grimson, W. 1981. From images to surface: A computational study of the human early visual system. MIT Press. MA.

[5] Grimson, W. and Pavlidis, T. 1985. Discontinuity detection for visual surface reconstruction. Computer Vision, Graphics, and Image Processing. 30: 316-330.

[6] Hayes, J. 1987. Fitting surfaces to data. The Mathematics of Surfaces II. Martin, R. R. (ed.). Claredon Press. Oxford.

[7] Lancaster, P. and Salkauskas, K. 1986. Curve and surface fitting: An introduction. Academic Press. London.

[8] Poggio, T., Torre, V. and Koch, C. 1985. Computational vision and regularization theory. Nature. 317: 314-319.

[9] Schenk, A. and Toth, C. 1991. Reconstructing visible surfaces. Industrial Vision Metrology. S. El-Hakim (ed.). Proc. SPIE 1526.

[10] Schenk, A. and Toth, C. 1992. Conceptual issues of softcopy photogrammetric workstations. PE&RS. 58 (1).

[11] Schumaker, L. 1976. Fitting surfaces to scattered data. Approximation Theory II. Lorentz et al. (eds.). Academic Press. New York.

[12] Terzopoulos, D. 1985. Computing visible-surface representation. Artificial Intelligence Laboratory Memo. No 800. M.I.T.

[13] Watson, D. 1992. Contouring: A Guide to the Analysis and Display of Spatial Data. Pergamon Press.

3D URBAN AREA SURFACE ANALYSIS

Zheng Wang
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

Surface reconstruction is a very important step towards the automation of mapping process. Surface analysis is a key part of the OSU surface reconstruction system. In this paper we introduce a surface analysis approach for the surface reconstruction of urban area. The approach consists of hump detection, grouping of 3D edges, and classification of 3D edges. The outputs of the surface analysis include locations and boundaries of humps, properties of 3D edges(e.g. horizontal or vertical, and on the topographic surface or above it), and occlusion prediction. Experimental results demonstrate this surface analysis approach can substantially improve the 2D edge matching and interpolation of surface.

1. INTRODUCTION

Surfaces, their properties and characteristics are probably the most important intermediate representation for extracting useful 3D information from images. As pointed out in Schenk et al., 1991, surface analysis is a key step towards reconstructing the topographic surface of urban area. The goal of 3D urban area surface analysis is to extract primitives with early vision processes(e.g. boundaries and depths), as well as symbolic primitives(e.g. properties of edges, such as breaklines and ridges, and occlusions) for the purpose of surface reconstruction and object recognition.

In digital photogrammetry, many successful examples of topographic surface reconstruction have been published, but mostly for small scales. Large-scale urban area are posing major problems, regardless of the matching method employed. Area based matching methods suffer from foreshortening problem which is very much a factor in urban areas. Feature based matching methods, on the other hand, are affected by dislocalization when using edge operators of large spatial extent.

In the OSU surface reconstruction system(see Schenk et al. 1991), we use a feature based matching approach(see Zong, 1992). The main goal is to reconstruct the surface by its breaklines. Breaklines are likely to correspond with edges in the image. Figure 1 depicts an 3D urban area surface analysis module which plays an important role in the OSU surface reconstruction system. The surface analysis serves two purposes: guiding the matching process and surface interpolation.

As can be seen in Figure 1, surface analysis consists of three parts: hump detection, grouping of 3D edges, and classification of 3D edges. In the following sections, we will explain the algorithms made for the three parts, report about experimental results, and conclude with suggestions for future research. The 2D edge matching and the interpolation are treated in the papers of Zong, 1992 and Al-Tahir, 1992, respectively.

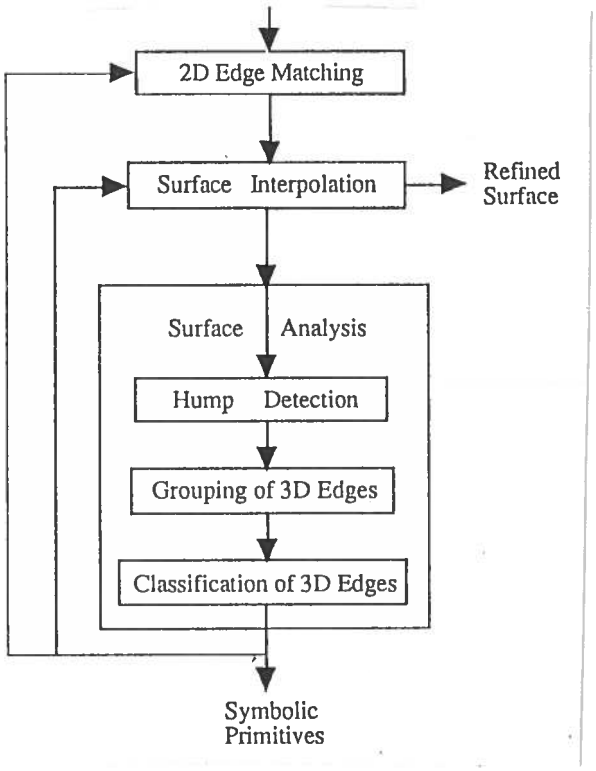


Figure 1: 3D urban area surface analysis module

2. BACKGROUND

A great deal of research in digital photogrammetry is devoted towards the automation of photogrammetric processes. It is a very difficult problem, far from being solved. The goal is to produce map as automatically as possible. Obviously, automation includes recognizing objects which then have to be digitized. Questions like how many objects there are and where they are in a given scene must be answered. To answer these questions, a reconstructed topographic surface is needed.

To reconstruct a topographic surface and recognize objects on the surface, edges are the main input. Physical boundaries of objects play a very important role in the human visual and recognition systems. Some psychological studies about the human visual and recognition process indicate that physical boundaries are the fundamental feature to graphically represent or describe objects[Attneave, 1954]. In Marr's paradigm for a machine vision system, which is the most advanced approach to date, edges (intensity changes) form the primal sketch[Marr, 1982]. Of course, surface reconstruction is a very complicated process, and to get a complete surface solely from edges is obviously not enough. To create a robust approach, we need to incorporate other information, such as texture and shape.

The edges detected in 2D images are intensity changes. They are caused by physical boundaries, but also by other phenomena like depth discontinuities between surfaces, shadow boundaries, changes in reflectivity, orientation, and texture of a surface. As known, the human visual system has an astounding perceptual classification and grouping ability to partition an image and to find associations among the various parts of the image. Grouping and classification make some property explicit in the whole process of object recognition, image understanding and image interpretation [McCafferty, 1990]. For surface reconstruction, we conclude that to find explicit properties of surface, grouping and classification must performed.

3. ALGORITHMS

3.1 Overview

For the surface reconstruction process, surface analysis should have the capacity of grouping 3D edges into humps or topographic surface, and further classifying them into horizontal and vertical edges. Horizontal edges are either on topographic surface or above it. Also surface analysis should be able to provide information about the boundaries and elevations of humps. The results of grouping and classification are used to complete the surface reconstruction process and later to aid object recognition, particularly the recognition of buildings.

Hump detection is the first step of the surface analysis. By a hump we mean something that clearly stands out from the topographic surface. Hump detection is important for several reasons:

- Humps may be the reasons for occlusion. Their detection can be used to determine occlusions in the image.
- Humps may cause problems for surface interpolation. Their known location can positively influence the surface interpolation.
- Hump detection is a requisite for the grouping and classification of 3D edge.
- Finally, humps may aid object recognition,particularly recognition of buildings.

Once humps are detected, their boundaries are known. Based on the hump information, all 3D edges are divided into groups, and then all the edges in each group are classified. In the process of classification, all edges are classified into horizontal and vertical edges. Further all horizontal edges are classified as edges on the topographic surface or above it.

After the classification, the results are used in the matching part and interpolation part. The information fed back to the two parts includes locations, boundaries, and elevations of humps and predicted occlusions. Additionally, information about the properties of 3D edges(e.g. horizontal, vertical, and on the topographic surface or not) is available for the interpolation

3.2 Hump detection

3.2.1 Generating DEM from matched 2D edges

The position of matched edges in object space is computed with exterior orientation parame-

ters. A DEM surface is generated by interpolating the 3D edges.

3.2.2 Transforming DEM surface to gray-value image

To detect humps, the DEM surface is transformed to a gray-value image. This gives us all the advantages of 2D image processing techniques. The formula used to transform a digital elevation value to a gray value is as follows:

$$g = 255 * \frac{Z - Z_{min}}{Z_{max} - Z_{min}} \qquad (1)$$

where g is the transformed gray value, Z_{max} and Z_{min} are maximum and minimum elevation values of DEM surface respectively, and Z is elevation value to be transformed. After the transformation, humps show up as bright clusters on the gray-value image.

3.2.3 Image segmentation and boundary formation

In Figure 3a we notice some bright clusters correspond to the humps of the DEM surface shown in Figure 2c. In order to find all the humps, we segment gray-value image to form contour lines. In this step, the interval between adjacent contours is a key parameter. In order to detect all humps, the interval should always be smaller than the lowest height of the humps in a given scene. In the contour image, humps are characterized by closed boundaries. See Figure 3b.

3.2.4 Eliminating non-hump boundaries and redundant hump boundaries

In Figure 3b some non-hump boundaries as well as redundant boundaries can be seen. To eliminate all non-hump boundaries, two generic properties are used. Closure property: a boundary for a hump is always closed. Length property: a hump boundary should not be too short or too long. By choosing the most outside boundary, redundant boundaries are eliminated.

3.2.5 Eliminating blunders

After all bright clusters in a gray-value DEM image are determined, they must be examined for blunders, such as some high peaks caused by wrong matching and bunkers. Shape operators may be useful to detect some blunders. An example for a simple shape operator is the ratio of length and width of a hump. For a complicated one, central moments may be used[Bian, 1988]. For instance, the second and third order central moments will tell the shape of an object and its symmetry. For bunkers, an elevation operator may be implemented to check all detected humps. If the gray value(elevation) inside a hump is lower than its surroundings, then it is not a hump, but a bunker. After all blunders have been eliminated(Figure 3c), the remaining humps are stored, together with shape information, such as average height, length, width, and volume.

3.3 Grouping of 3D edges

All 3D edges are now grouped into humps based on their locations under the condition that all edges in one group should belong to one hump. The number of groups is identical to the number of humps. Edges which do not belong to any hump are grouped into an extra class: topographic surface edge.

3.4 Segmentation and Classification of 3D edges

In this step hump edges are segmented into horizontal and vertical edges, and further horizontal edges are classified into edges on the topographic surface or above it.

3.4.1 Classifying 3D edges into horizontal and vertical edges

In the 3D space, an edge can be a 3D curve. For such an edge, some segment(s) of it may be horizontal and other segment(s) are vertical. Horizontal edges are composed of horizon-

tal edge segments, and vertical edges are from vertical edge segments. To get the segments, every point of a 3D edge is classified as horizontal point or vertical point based on an angle defined by the following formula:

$$angle = arctan(\frac{z_i - z_{i-1}}{d_{xy}}) \qquad (2)$$

where z_i and z_{i-1} are two elevation values of the two adjacent points, p_i and p_{i-1}, and d_{xy} is the distance between the two points on horizontal plane. If the angle is greater than a threshold, the point pi is classified as vertical. After all points of an edges have been classified, by simply connecting the adjacent points of the same class, horizontal and vertical edges are generated.

3.4.2 Classifying horizontal edges belonging to the topographic surface

To classify horizontal edges in a hump as edges on the topographic surface or above the surface, first it is necessary to find the minimum elevation of the edge points of a hump. Once the minimum elevation is found, according to the average elevation of a horizontal edge, the edge is classified as edge on the topographic surface or above it.

4. EXPERIMENTAL RESULTS

We tested our approach on several stereo pairs of urban area image patches.

4.1 Source Data

The image patches used in the experiments were selected from aerial images(model 193/195) of The Ohio State University campus, a good example of a typical urban scene. The scale of the photographs, from which the digital images were digitized, is about 1:4000. The experiment was performed on the images with a 2k x 2k resolution. Each pixel in the images represents a square 44cm x 44cm. For the experiment two image patches were selected with a size of 512 x 512.

Figure 2a shows the two image patches used in the experiment. The matched edges are shown in Figure 2b, and a DEM surface generated from the matched edges is shown in Figure 2c. The two figures in Figure 2c are two different view angles for same one DEM surface. The DEM surface was generated by using Intergraph's modeler software. We recognize from Figure 2c that the buildings are distorted by the interpolation process.

4.2 Experimental results

Figure 3a is the gray-value DEM image for the DEM in Figure 2c. In this image some bright clusters are recognizable, which indicate potential humps. Comparing this figure with Figure 2a, we see that areas with buildings are obviously brighter than their surroundings. Figure 3b shows a contour image of Figure 3b. The contour interval used was 4 meters. Figure 3c depicts all detected humps. At this stage the number of humps, the locations and boundaries of the humps become known. Additionally, the elevations and shapes of the humps are determined as well.

After the hump detection, all edges are associated to humps or topographic surface based on their geometrical locations. To test the results of this grouping process, a DEM was generated for every hump using only the edges belong to the hump. Figure 3d and 3e are two samples of them. One of the two humps is OSU library, and the other one is University Hall. The two humps have the same shape as they are in the DEM surface in Figure 2c, which indicates the result of grouping is correct. Finally the Figure 3f shows the DEM of the topographic surface after all the humps have been removed, with the exception of two incomplete humps(the contours of these two humps are not closed).

Figure 4 shows the results of classification. Based on the derived information of edge properties, we generated the top of OSU library in

Figure 4a by using all horizontal edges which are above the topographic surface in the hump "library". Figure 4b is a combination of vertical edges and horizontal edges which are above the topographic surface.

The derived hump information and edge properties are made available to the matching and interpolation processes. With this information, the matching improved considerably[Zong, 1992]. The improvement of the interpolation part is shown in Figure 5. Here we show the DEM after a new interpolation took place with hump information. The result in Figure 5 demonstrates that the building walls in Figure 5 are more vertical than those in Figure 2c.

5. CONCLUSION

Surface reconstruction of urban areas is a very important step towards the automation of mapping processes. A complete surface is essential in order to recognize man-made objects and interpret images. Surface analysis is a key part of the OSU surface reconstruction system.

The experimental results demonstrate that the surface analysis can substantially improve the matching and interpolation of the surface of urban area. Additionally, the results of the hump detection can be used to recognize buildings.

ACKNOWLEDGMENTS

Funding for this paper was provided in part by the NASA Center for the Commercial Development of Space Component of the Center for Mapping at The Ohio State University. The authors would like to thank Ms. Jia Zong for providing 2D edge matching results for the research, and Mr. W. Cho for providing some assistance.

REFERENCE

Al-Tahir, R., 1992. On the Interpolation Problem of Automated Surface Reconstruction. *Proceedings of ISPRS.*

Attneave, F., 1954. Some Informational Aspects of Visual Perception. *Psychological Review. vol. 61, No 3, pp183-193.*

Bian, Z., 1988. Pattern Recognition. *Qinghua University Publishing Press. Beijing, China pp272-274*

Marr, D., 1982. Vision: A computational Investigation into The Human Representation and Processing of Visual Information. *W.H. Freeman. San Francisco.*

McCafferty, J., 1990. Human and Machine Vision. *Ellis Horwood Limited. Chichester, England.*

Schenk, A. and Toth, C., 1991. Knowledge-Based Systems for Digital Photogarmmetric Workstations. *In Ebner H., Fritsch, D., and Heipke, C.(Ed.): Digital Photogrammetric Systems. Wichmann.pp123- 134.*

Zong, J., Li, J., and Schenk, T., 1992. Aerial Image matching Based on Zero-Crossings. *Proceedings of ISPRS.*

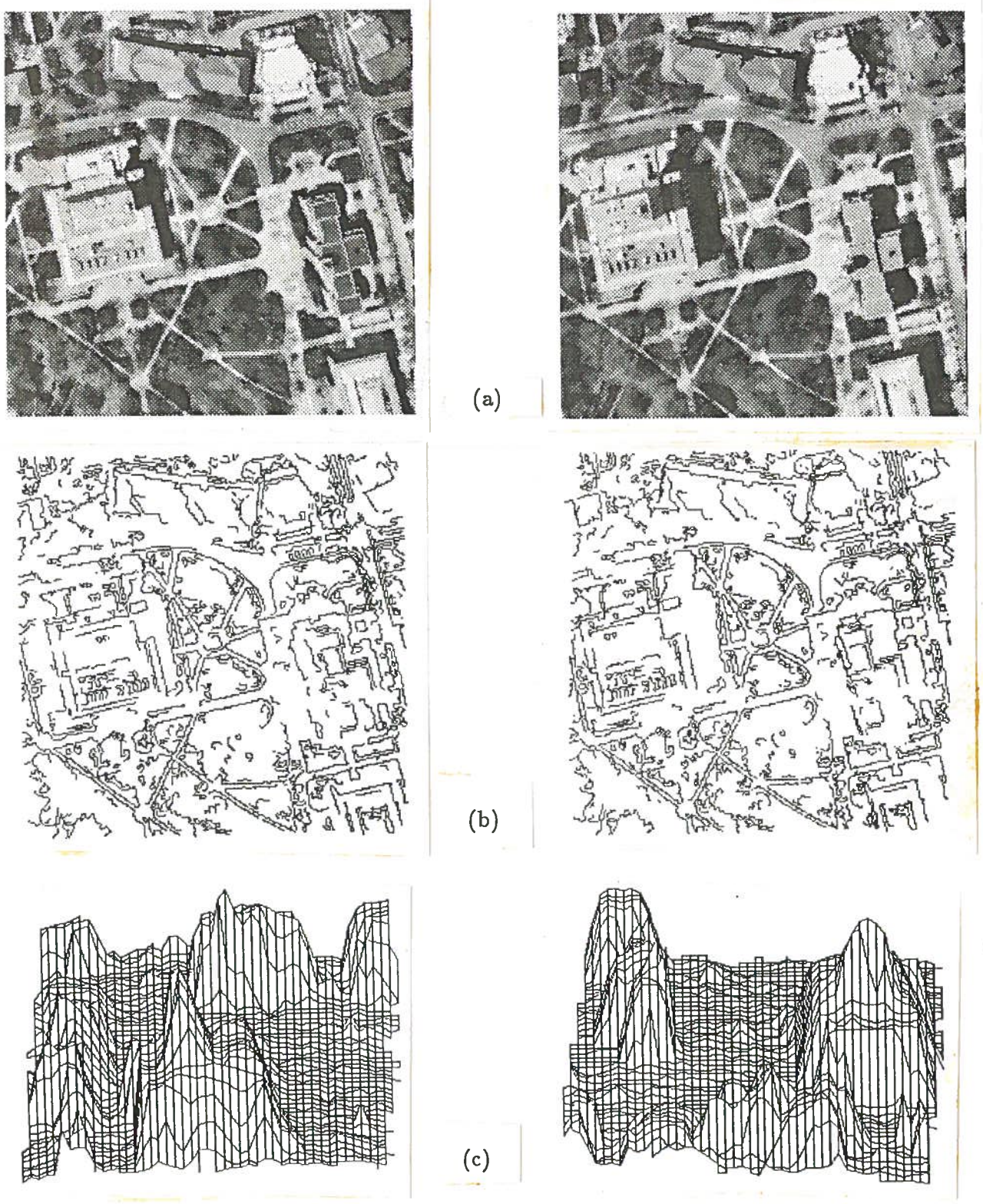


Figure 2: Source input data. (a) two stereo image patches selected from OSU campus images. (b) Matched 2D edges for the patches in (a). (c) Interpolated DEM surface for the scene in (a)(two view angles).

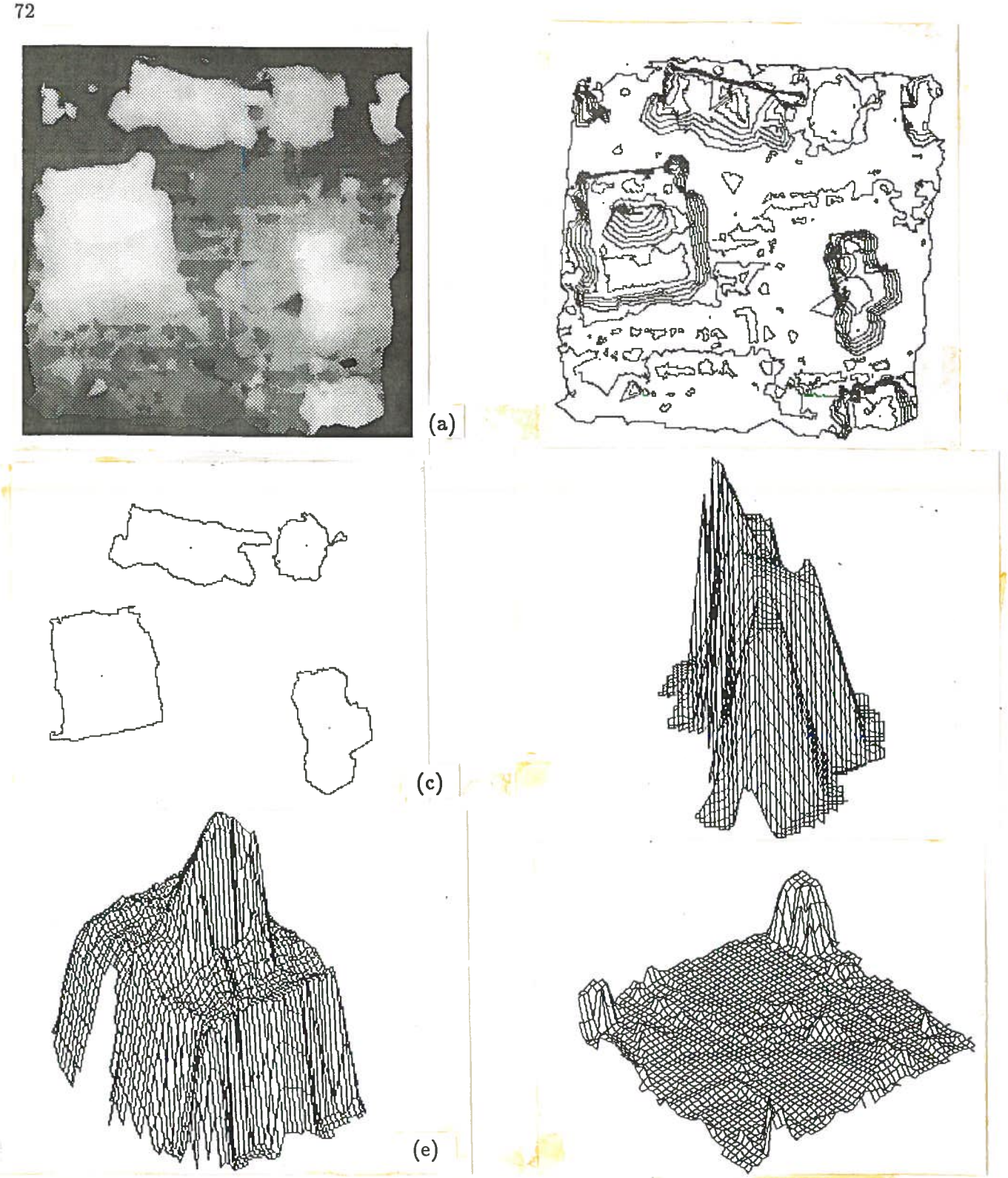


Figure 3: Detected humps and grouping examples. (a) Gray-value DEM image for the DEM surface in Figure 2(c). (b) Elevation contours for (a), the contour interval is 4 meters. (c) Detected humps for the scene in Figure 2(a). Each hump corresponds to one building in the scene. (d) and (e) Two hump DEM surfaces generated after grouping. The two humps are University Hall and library in OSU campus respectively. (f) The topographic surface after all humps were removed.

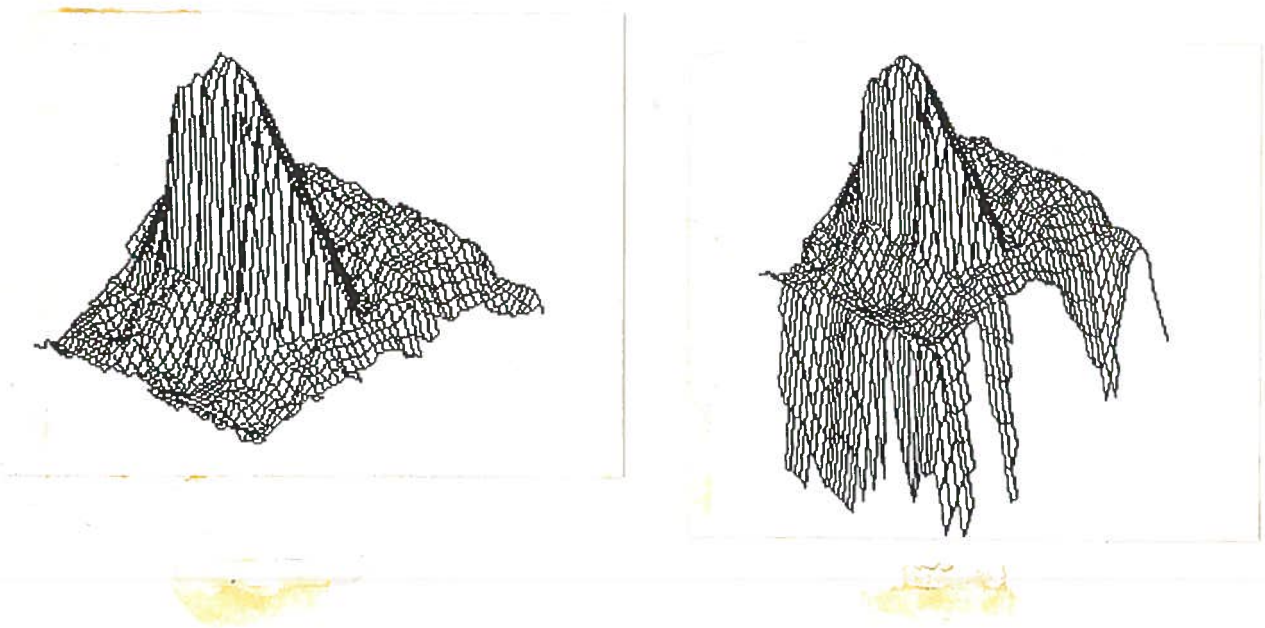


Figure 4: Classification examples. (a) Top of the library in Figure 3(e) generated by only the horizontal edges which are above the topographic surface. (b) An incomplete library generated by vertical edges and horizontal edges which are above the topographic surface.

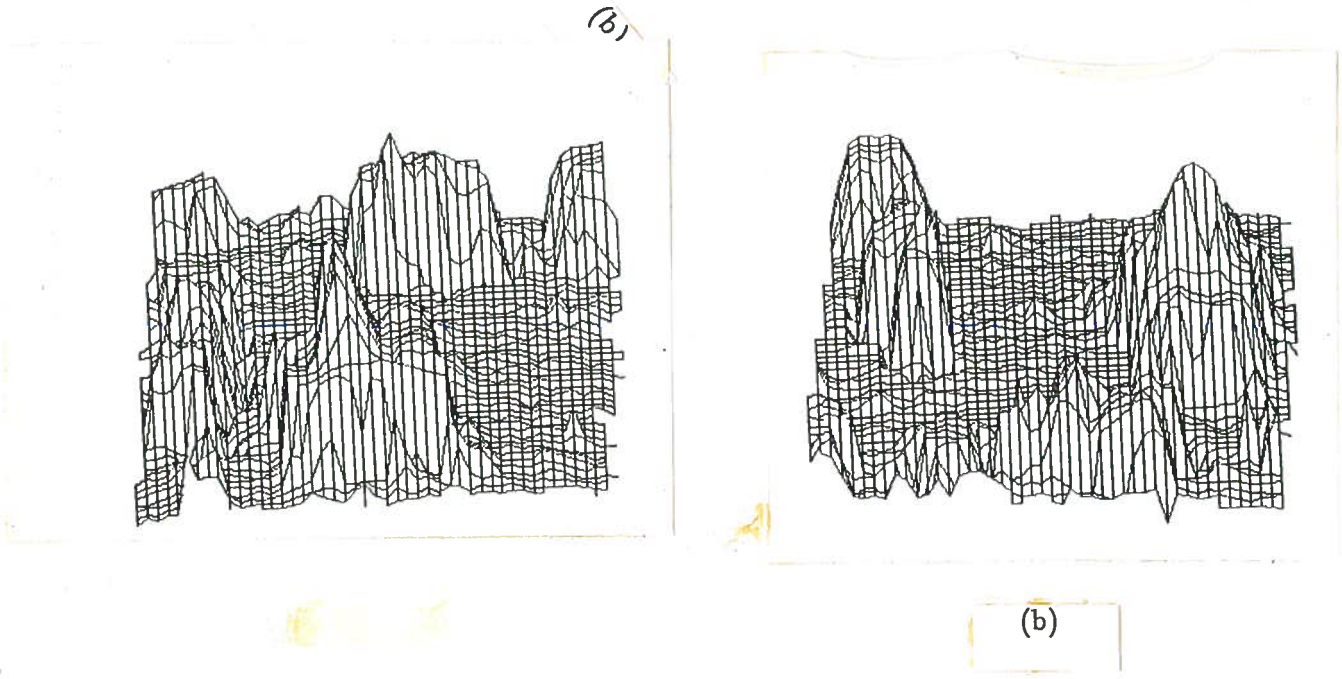


Figure 5: The re-interpolated DEM surface. (a) and (b) Two view angles of the re -interpolated DEM surface. The building walls in this figure are more vertical than those in Figure 2(c).

IMAGE SEGMENTATION FROM TEXTURE MEASUREMENT

Dong-Cheon Lee
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

It is known that the human visual system, unsurpassed in its ability to reconstruct surfaces, employs different cues to solve this difficult task. The prevailing method in digital photogrammetry is stereopsis. However, texture may provide valuable information about the shape of surfaces. In this paper we employ Laws' method of texture energy transforms to extract texture information from digital aerial imagery. The images are convolved with micro-texture filters to obtain local texture properties. Each micro-texture feature plane is transformed into an texture energy image by moving-window to render macro-texture features. Finally, the macro-texture feature planes are combined and then clustered into regions of similar texture pattern. The method is implemented in a scale-space approach, and the boundaries obtained from texture are compared with physical boundaries of the image.

1 INTRODUCTION

The goal of digital photogrammetry is to reconstruct surfaces automatically. Surface reconstruction from raw imagery is known as an ill-posed problem. To solve this difficult task, different cues which contribute to object recognition and scene interpretation are employed. One of the important cues is texture. Texture may provide information to estimate shape, surface orientation, depth changes, material of objects. Texture information aids image analysis and interpretation.

Many texture analysis methods have been developed during the last two decades. Among the great variety of available methods, Laws' approach of texture energy measures appears to be a suitable method (Ballard and Brown, 1982; Gool *et al.*, 1985; Gong and Huang, 1988; Unser and Eden, 1990). Furthermore, this method resembles human visual processing of texture according to Laws' dissertation. One of the advantages of this method is to provide several texture feature planes from an original image. This is a great benefit especially if only monochrome imagery is available because to extract useful texture information from raw monochrome images is a difficult task even for the human vision system. More useful information and segmentation results could be obtained by integrating the additional texture feature planes.

2 CHARACTERISTICS OF TEXTURE

Texture is qualitatively described by its coarseness under the same viewing condition, and related to the repetition of the local spatial patterns. In addition to coarseness, other textural dimensions or parameters are commonly proposed, namely, contrast, density, roughness, directionality, frequency, regularity, uniformity, orientation, and so on (Tamura *et al.*, 1978).

Texture is a sophisticate visual primitive since texture element (texel) is determined by contextual process and a different level of hierarchy. Texture primitives consists of micro-texture and macro-texture. Micro-texture is the smallest primitive while macro- texture is referred to larger primitive, i.e., macro-texture is homogeneous aggregation of micro-texture. These two primitives cannot be confused with fine texture and coarse texture. The coarseness of texture is related to the spatial repetition period of the local structure. Therefore, micro-texture and macro-texture are not related the coarseness. However, in fact there are not clear criteria to differentiate micro-texture from macro-texture primitives, rather it is related to somewhat psychological effect as well as image scale and resolution. Since texture is hierarchical, texture within texture primitives themselves is visible (Gool *et al.*, 1985). It is important to understand how the human visual system works for texture discrimination and grouping. To develop a computational texture analysis system is not an easy task due to the great complicatedness of properties of texture.

Texture has following characteristics;

- Texture is shift, orientation, moment, contrast, and illumination invariant.
- Human texture perception tends to be sensitive first-and second- order statistics, and does not respond to higher than second-order. Discriminable textures can be generated having a common mean, variance, and auto-correlation function. Thus, second-order moments are sufficient measures of texture.
- Texture is hierarchical, i.e., it corresponds to different resolutions and then global unitary impression is offered to the observer. Global features characterize the whole texture rather than texels.

The above characteristics are helpful guidelines in designing texture analysis system. The other important characteristic is that texture is both stochastic and deterministic, therefore, texture analysis methods are categorized by two major approaches; statistical and structural approaches.

3 TEXTURE MEASUREMENT

Texture energy transform developed by Laws is a class of spatial- statistical approach. The characteristic of this method is more matched to intuition about texture features, i.e., similar to human visual processing (Laws, 1980;

Ballard and Brown, 1982). This method was developed after he investigated and evaluated several existing methods including statistical, structural, co-occurrence, spatial frequency, and auto-correlation approaches.

3.1 Texture Energy

The original image or a patch of the original image (f) is convolved with micro-texture filters (h_k) to create micro-texture features (f'_k);

$$f'_k(i,j) = f(i,j) * h_k \qquad (1)$$

where the micro-texture filters can be formed from following four one-dimensional vector masks;

$$\begin{array}{lcl} L5 & = & (\ 1 \quad 4 \quad 6 \quad 4 \quad 1\ ) \\ E5 & = & (\ -1 \quad -2 \quad 0 \quad 2 \quad 1\ ) \\ S5 & = & (\ -1 \quad 0 \quad 2 \quad 0 \quad -1\ ) \\ R5 & = & (\ 1 \quad -4 \quad 6 \quad -4 \quad 1\ ) \end{array}$$

A total of sixteen two-dimensional micro-texture filters can be created. These are L5L5, L5E5, L5S5, L5R5, E5L5, ..., R5R5. However, L5L5 is not used because the sum of the filter elements is not zero.

In order to obtain macro-texture features (f''_k), each of the micro- texture images (f'_k) is transformed into an texture energy image by moving macro-texture window;

$$f''_k(i,j) = \frac{1}{w^2} \sum_{n=\frac{-w}{2}}^{\frac{w}{2}} \sum_{m=\frac{-w}{2}}^{\frac{w}{2}} |f'_k(n,m)| \qquad (2)$$

where w is size of a macro-window. The micro-texture feature values are replaced by average of absolute values in a macro- windows. The size of the optimal macro-window depends on texture coarseness or regularity, as well as the quality of the available micro-features.

Micro-texture filters are designed to measure local texture properties, while the macro-texture features measure properties of the texture field as a whole. The problem is there is no guarantee that any particular resolution or window size will be optimal for a given analysis (Laws, 1980).

3.2 Texture Classification

Texture segmentation can be performed by classification. Most of the classification algorithms are suitable for multispectral imagery. Since several different micro-texture filters provide many corresponding texture feature plates, to use a multispectral classification algorithm is a quite reasonable approach. Classification of imagery is one of the main tasks in remote sensing. However, the pure texture-based classification method does not seem to be successfully developed yet. The purpose of image segmentation, based on texture information, is to obtain useful surface information.

Unsupervised classification is more attractive than supervised classification methods, because sometimes *a priori* knowledge about the area of interest is not available. Furthermore, human operators' intervention will not be allowed in fully automatic mapping and surface reconstruction systems.

4 EXPERIMENTAL RESULTS

4.1 Selection of Imagery

Left image (photo scale: 1/3,800) of "Munich" model (Figure 1), which was digitized with an EIKONIX camera to a resolution of 4096 by 4096 pixels, was used to implement our task. The "Munich" image contains residential areas, major high-ways, small roads, different kinds of vegetation, and a water area (small pond). For the scale space approach, 512 by 512, 1024 by 1024, and 4096 by 4096 images were used.

4.2 Texture Energy Features

All fifteen micro-texture images were created in coarse level. Then, each image was evaluated to select suitable filters. In fact, computational methods were not involved to select filters. Based on visual evaluation of the micro- and macro-texture features, E5E5, E5S5, S5E5, and S5S5 filters were chosen. Any micro-filters which did not provide clear texture patterns were not used in further levels. Selected filters were used through the next two levels. Macro-texture features were obtained with different macro-window sizes for each level. Macro-filter

sizes selected for each level were 5, 15, and 31 for 512 by 512, 1024 by 1024, and 4096 by 4096 images, respectively.

The micro-filters selected in this study provided a similar pattern of texture energy features. The filters seems to detect horizontal, vertical and diagonal patterns of texture. It is obvious that micro-texture patterns will disappear by use of larger macro-window size. However, more homogeneous macro-textures will appear. Grouping of the micro-textures provided macro-texture.

4.3 Integration of Texture Feature Plates and Classification

Each texture feature plate is regarded as a spectral band to apply multispectral analysis. The texture feature plates are combined to one image file with BIL (band interleaved by line) format. Iterative self-organizing data analysis technique (ISODATA) in ERDAS was used for classification. The advantage of ISODATA is that the algorithm represents a fairly comprehensive set of additional heuristic procedures which have been incorporated into an interactive scheme (Tou and Gonzalez, 1974). Classification was performed through all three resolution levels. Figures 3, 4, and 5 are the results of classification for each level. The boundaries of both original and classified texture images were detected by using a Sobel edge operator (Figure 2 and 6).

The classification results were improved from coarse level to fine level. Classification result of fine level renders original image. However, the feature boundaries were not preserved due to the relatively large macro-window size. More boundaries were obtained from texture classification by comparing to the boundaries of the original image, especially in residential and vegetation areas. This result is possible, because in those areas different texture patterns are mixed. The result still shows lots of micro-textures which are reasonable to be grouped into a homogeneous area.

Optimal size of macro-window depends on scale, resolution, and objects in the image. To find optimal size of the window size is not easy. In addition, other very crucial factor for texture analysis is the classification method.

Figure 1: Original image of Munich model

Figure 2: Boundaries of original image

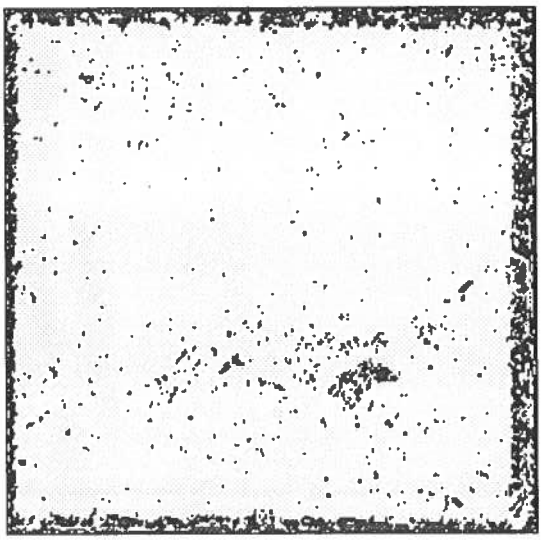


Figure 3: Texture classification of 512x512 image (macro-window size:5)

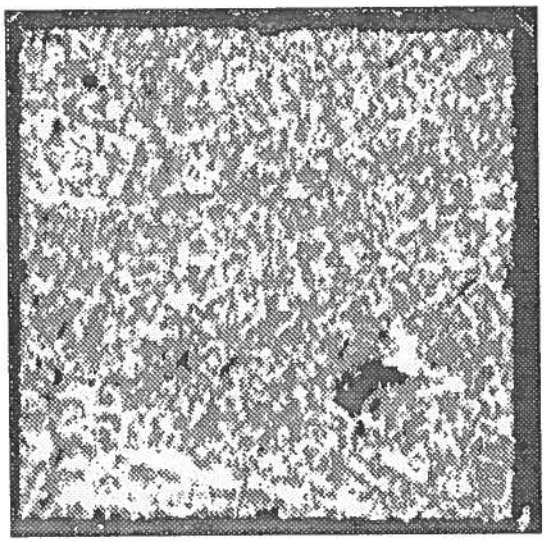


Figure 4: Texture classification of 1024x1024 image (macro-window size:15)

Figure 5: Texture classification of 4096x4096 image (macro-window size:31)

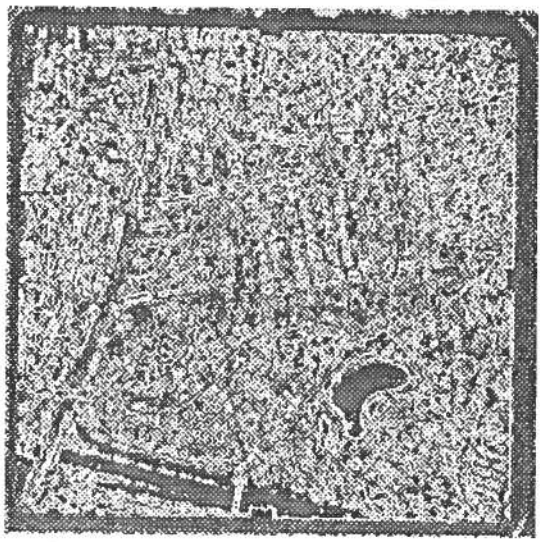


Figure 6: Boundaries of classified image (4096x4096)

5 CONCLUSIONS

Laws' texture energy transform provides information about texture patterns of the surface. His approach to detect micro-texture and then group into a macro-texture feature is very realistic and a proper approach. However, to determine the fixed macro-window size for entire image is a difficult task. It is not easy to develop the dynamic size of the window, i.e., the window size varies depending on the objects in the image.

So far, many of the texture analysis methods do not succeed for natural scene imagery. Most of authors have used synthetic image or geometrical composite (or mosaic) of natural texture image patches to develop and evaluate texture operators. However, these kinds of imagery do not provide enough texture properties of natural scene.

Since color imagery contains more information than a monochrome one, to use color image is one way to improve the texture analysis system.

Finally, the 3D object space approach of texture analysis is probably a more interesting and more powerful solution.

ACKNOWLEDGMENT

Funding for this research was provided in part by the NASA Center for Commercial Development of Space Component of the Center for Mapping at The Ohio State University.

REFERENCES

Ballard, D. and C. Brown, 1982. Computer Vision. Prentice-Hall Inc., Englewood Cliffs, N.J., pp. 166-194.

Laws, K., 1980. Textured image segmentation. Ph.D. dissertation, Dept. of Electrical Engineering, University of Southern California.

Gong, X. and N. Huang, 1988. Texture segmentation using iterative estimate of energy states. 9th Int. Conference on Pattern Recognition, Rome-Italy, Vol.1, pp. 51-55.

Gool, L., P. Dewaele, and A. Oosterlinck, 1985. Survey texture analysis anno 1983. Computer Vision, Graphics, and Image Processing, (29):336-357.

Tamura, H., S. Mori, and T. Yamawaki, 1978. Textural features corresponding to visual perception. IEEE Trans. SMC, 8(1):460-473.

Tou, T. and R. Gonzalez, 1974. Pattern Recognition Principles. Addison-Wesley, Reading, Mass. pp. 97-104.

Unser, M. and M. Eden, 1990. Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering. IEEE Trans. SMC, 20(4):804-815.

ON THE APPLICATION OF SCALE SPACE TECHNIQUES IN DIGITAL PHOTOGRAMMETRY

Anthony Stefanidis
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

Scale space techniques are widely used in digital photogrammetry. Typical implementations use the scale space as a discrete representation, thus inherently assuming that all features represented in images of similar resolutions belong to the same scale space level. However, this approach ignores differential scale variations that exist between conjugate features in multiple images, or even between different features in a single image. The subject of this paper is an investigation into theoretical and practical aspects associated with the use of scale space techniques in both the image and object space domains. The interrelationship between the scale space representations of these two domains and the effects of differential scale variations in digital photogrammetric operations, such as matching, object space reconstruction, and orthophoto production are also addressed.

Report No. 418 July 1992

1. INTRODUCTION

Physical phenomena in object space occur over a wide variety of spatial extents. Macro-variations of a surface express its major trend, while micro-variations correspond to trends of smaller extent. The concept of macro- and micro-variations is relative and depends on the specific application. What is considered a macro-variation in one application might very well be viewed as a micro-variation in another. In digital images, changes in gray values correspond to object space phenomena, which can also be perceived within areas of different sizes, ranging from few pixels to large regions. However, even region-wise changes occur over an extensive array of region sizes, ranging from as little as a few pixels to as much as a large part of the image. The identification of these changes is essential in decoding the information which inherently exists in an image.

The scale space representation of signals in general, or digital images in particular, is widely used to successfully produce several versions of the same image in which the information content is changing in a systematic and, therefore, easy to exploit fashion [Lindeberg, 1990], [Yuille & Poggio, 1983]. Physical phenomena of various extents can be easily identified through the behavior of their images in different levels of scale space [Lu & Jain, 1989], [Witkin, 1983].

In our paper, we present the basic axioms of scale space, and we analyze the corresponding mathematical aspects, together with the proper selection of scale-generating functions. The effect of differential scale variations on photogrammetric procedures is discussed, and we report how a continuous scale space can be used to bypass the shortcomings of this effect. Finally, the scale space representation of object space and its potential use in photogrammetry are explored.

2. SCALE SPACE

The scale space representation of a signal $f(x,y)$ is a set of signals $\{f_s^n(x,y;n)\}$, representing the original one in various scale levels as function of a scale parameter n. The set of signals $\{f_s^n(x,y;n)\}$ is called the scale space family of $f(x,y)$.

The objective of the scale space representation of any signal is to create a scale space family in a way that information conveyed by this signal will become more explicit. In order for this goal to be met, the generation of scale space family has to follow some basic guidelines [Lindeberg, 1990]:

- The scale space family has to be generated by the convolution of the original signal with a single scale-generating function $s(x,y;n)$

$$f_s^n(x,y;n) = s(x,y;n) * f(x,y) \qquad (1)$$

- The scale-generating function should be selected in a proper manner, such that larger values of n would create coarser versions of the original signal through elimination of the finer details which correspond to higher frequency phenomena. We want to be able to identify large trends in lower resolutions and include spatially limited details in finer levels. For $n = 0$, at the finest resolution of scale space, we have the original signal itself

$$f_s^0(x,y;0) = f(x,y) \qquad (2)$$

  which is obviously the upper limit as far as fine resolution is concerned.

A Gaussian filter is mathematically expressed as a function

$$g(x,y) = ke^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (3)$$

where σ is the associated standard deviation. In applications, the multiplicative factor k may receive various values, creating a large array of Gaussian filters which are essentially scaled variations of the core function, e.g.,

$$G(x_g,y_g;\sigma) = \frac{1}{2\pi\sigma}e^{-\frac{x_g^2+y_g^2}{2\sigma^2}} \qquad (4)$$

attempts to preserve the output within a prespecified range [Agouris et al., 1989]. The use of a Gaussian filter, with standard deviation σ as the associated scale parameter, as a scale-generating function satisfies the above set criteria [Babaud et al., 1986]. Therefore, the scale space family of a signal $f(x,y)$ can be created as

$$f_s^n(x,y;\sigma) = G(x_g,y_g;\sigma) * f(x,y) \qquad (5)$$

A digital image is a two-dimensional discrete signal $I(x,y)$. Its convolution with the Gaussian kernel

$$I^n_\sigma(x,y;\sigma) = \sum_{x_g=-\infty}^{\infty}\sum_{y_g=-\infty}^{\infty} \frac{1}{2\pi\sigma} e^{-\frac{x_g^2+y_g^2}{2\sigma^2}} \cdot \cdot \; I(x-x_g, y-y_g) \qquad (6)$$

can be used to construct its scale space family. Members of the scale space family may have the same dimensions as the original image, or, more commonly, their dimensions may decline in coarser resolutions. Assuming the original image $I(x,y)$ to have dimensions 4096 × 4096 pixels, we can form its scale space family by creating m versions of the image (all of dimensions 4096 × 4096 pixels), each one by convolving $I(x,y)$ with a Gaussian kernel of different scale parameter σ.

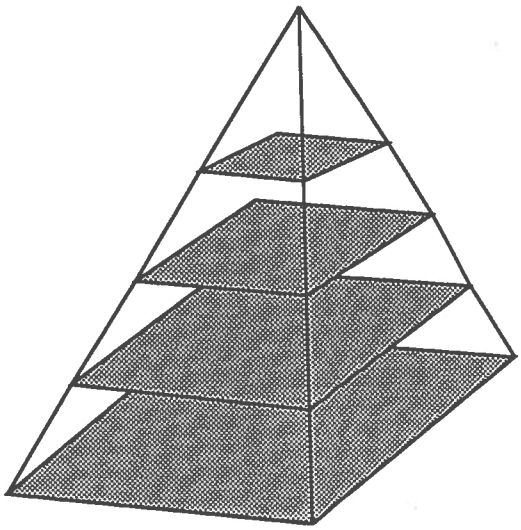


Figure 1: An image pyramid as a representation of discrete scale space

However, in most applications coarser levels of scale space are represented by images of smaller dimensions. By convolving the image with a Gaussian kernel and resampling every n^{th} pixel we can create a lower resolution copy of size $4096/n \times 4096/n$. A scale space family in which lower resolution members are represented by smaller size images is called an image pyramid [Fig. 1]. Various members of the image pyramid can be perceived as images of the same object scene in various geometric scales. For practical reasons the dimensions of the members of the scale space family are integer powers of two. Typically, the image pyramid of an original image of 4096 × 4096 pixels includes versions of the image in dimensions of 2048 × 2048, 1024 × 1024 and 512 × 512 pixels. Fig. 2 shows two windows of equal dimensions, one from the 512 × 512 pixel member of an image pyramid and the other from the 2048 × 2048 pixel version to demonstrate the associated differences in resolution. Both images were obtained by the convolution of the original 4096 × 4096 image with a Gaussian kernel, and by proper resampling.

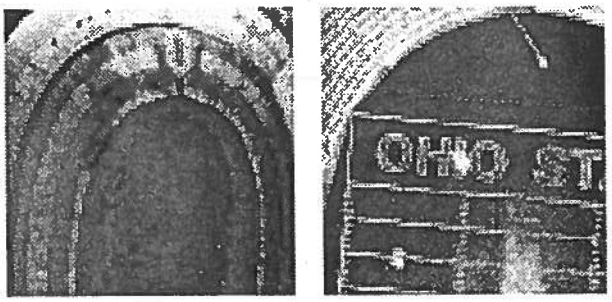


Figure 2: Two windows of equal size in pixels, one in 512 × 512 resolution (left) and the other in 2048 × 2048 resolution (right).

The use of a Gaussian kernel as a scale-generating function offers certain advantages, most notably exploited when combining smoothing with edge detection. Edges are identified as discontinuities in the image function, and therefore correspond to zero-crossings of the twice-differentiated image. The orientation independent second derivative of a two-dimensional function is obtained through a Laplacian operator

$$\nabla^2 = \frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} \qquad (7)$$

The associative property of convolution allows the combination of scale space generation with a Gaussian function $G(x_g, y_g)$ and differentiation with a Laplacian operator, thus substituting two convolutions by a single one

$$\nabla^2[G(x_g,y_g;\sigma) * I(x,y)] = [\nabla^2 G(x_g,y_g;\sigma)] * I(x,y) \qquad (8)$$

Instead of scaling the image with $G(x_g, y_g)$ and then looking for edges in the smoothed image, we simultaneously smooth the image and extract its orientation-independent second derivative in a single convolution by the Laplacian of Gaussian (LoG) function

$$\nabla^2 G(x_g, y_g; \sigma) = [2 - \frac{{x_g}^2 + {y_g}^2}{\sigma^2}] e^{-\frac{{x_g}^2 + {y_g}^2}{2\sigma^2}} \quad (9)$$

The size of the LoG operator is determined by the value of σ or altenatively, by the diameter w of its positive central region, which is related to σ through the equation

$$w = 2\sqrt{2}\sigma \quad (10)$$

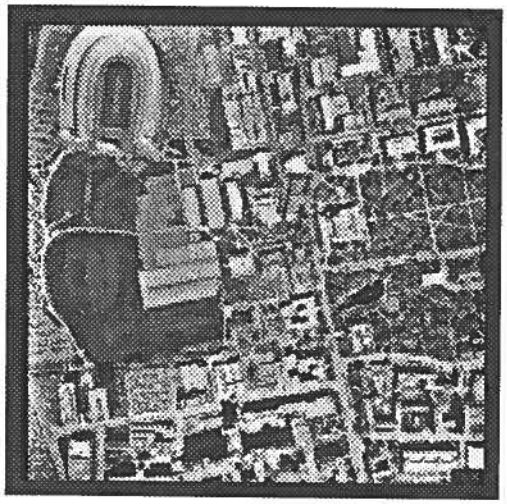


Figure 3: The original image

Scale space family generation and edge detection can thus be succesfully combined. By using the Gaussian kernel for scaling we ensure that in any scale level fewer edges occur than in finer resolutions and more than in coarser ones, thus performing proper scale space generation. This property has a qualitative aspect in addition to its obvious quantitative meaning. Edges detected in coarser levels using large σ (or w) values will also appear in finer levels. The same edge can be traced through various resolutions, since its images display a certain degree of geometric similarity, with the degree of localization (closeness to the true edge) increasing with resolution [Lu & Jain, 1989], [Witkin, 1983]. This is demonstrated in Fig. 4 and Fig. 5 which show edges of the original image (shown in Fig. 3) produced by its convolution with a fine ($w = 10$) and a coarse ($w = 30$) LoG operator respectively. In addition, the traces of edges in various resolutions offer a complete representation of the original signal, thus allowing its reconstruction [Yuille & Poggio, 1983].

Figure 4: Edges detected with a fine Log operator (w=10)

3. DIFFERENTIAL SCALE VARIATIONS

When representing the scale space family of a digital image as a pyramid, we create a number of discrete representations of the original image with each representation corresponding to a specific scale level. However, unless the image-generating projection is parallel, the exposure vertical and the object surface planar, features within the same image pyramid level will not have the same geometric scale, expressed as

$$S^A = \frac{A'}{A} \quad (11)$$

with A' the image of a feature A of the object space. For the projective transformation governing the image formation process, the scale factor S^i at a point (x^i, y^i) of the image, corresponding to a point (X^i, Y^i, Z^i) in the object space will be given through the formula

$$\begin{bmatrix} x^i \\ y^i \\ -c \end{bmatrix} = S^i R \begin{bmatrix} X^i - X_o \\ Y^i - Y_o \\ Z^i - Z_o \end{bmatrix} \quad (12)$$

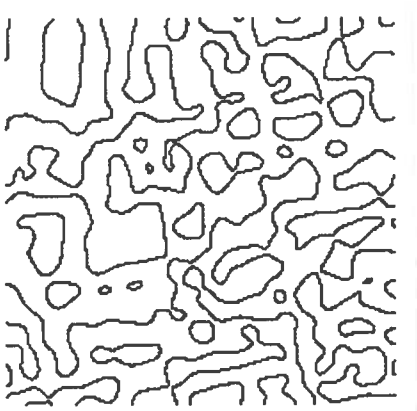


Figure 5: Edges detected with a coarse Log operator (w=30)

where R is the rotation matrix and (X_o, Y_o, Z_o) the exposure station coordinates of the photo. It is apparent that different features in the same image will have different scale factors. In addition, the images of the same object space feature in two or more different images will have different scales, particularly when the exposure conditions (rotations, exposure stations) differ significantly (e.g., converging photography) or the object space surface displays high variations. In the extreme case, the scale becomes 0 and occlusions occur.

Assuming each image pyramid level i to correspond to an average scale S_i, features within this image will thus appear in scales $S_i + dS_i$, where

$$dS_i = f(X, Y, Z, \omega, \phi, \kappa, \frac{\partial Z}{\partial X}, \frac{\partial Z}{\partial Y}) \qquad (13)$$

which in general will not coincide with any of the discrete scales represented by the image pyramid. Image pyramids though are discrete representations of the scale space which itself is continuous. While the discrete representation is obtained using only a number of values of the scale parameter σ of the Gaussian kernel used to convolve the image, a continuous representation is the outcome of the same convolution allowing σ to receive any allowable real value.

Scale variations between members of stereopairs become apparent in digital photogrammetric operations, with matching serving as a good example. In least squares matching, we attempt to match windows of pixels by minimizing their radiometric differences. This is achieved by forming one observation equation for every pair of conjugate pixels within a pair of approximately conjugate image windows $g_L(x_L, y_L)$ and $g_R(x_R, y_R)$ in the left and right image respectively

$$g_L(x_L, y_L) - g_R(x_R, y_R) = e(x, y) \qquad (14)$$

The solution is obtained by allowing one of the two windows to be geometrically reshaped according to an affine transformation and by resampling gray values for this newly defined window. Differences in scale are accommodated by the two scale factors assumed in the six-parameter affine transformation

$$x_R = a_1 + a_2 x_L + a_3 y_L \qquad (15)$$

and

$$y_R = b_1 + b_2 x_L + b_3 y_L \qquad (16)$$

Updating the above affine transformation parameters by the solution of the linearized observation equations

$$\begin{aligned} g_L(x_L, y_L) - e(x, y) &= g_R^o(x_R^o, y_R^o) + g_{R_x} da_1 \\ &+ g_{R_x} x_L da_2 + g_{R_x} y_L da_3 \\ &+ g_{R_y} db_1 + g_{R_y} x_L db_2 \\ &+ g_{R_x} y_L db_3 \qquad (17) \end{aligned}$$

we define a new window in the right image within which we resample the gray values.

Scale variations will affect this procedure in various stages. When two image patches are represented in two different scale levels in a stereopair, their scale difference will be both geometric and radiometric. When resampling the gray values $g_R(x_R, y_R)$ we use the original image, spreading or shrinking its gray values over a new area, according to the updated affine transformation parameters. As a result we produce a new window in the right image which might belong to the same geometric level of scale space as its conjugate left image template $g_L(x_L, y_L)$ but will still differ from it in the radiometric scale space. This will have obvious effects on the observation equations, since we use gray level differences as observations. The same problem occurs during digital image warping or rectification for orthophoto

production [Doorn, 1991],[Novak, 1992]. Conjugate patches in two overlapping orthophotos are brought to the same scale level geometrically, using as a reference a digital elevation model of the object space. Radiometrically though, these patches remain unequal to the same degree that the corresponding windows in the original stereopair were unequal. This causes conjugate patches in overlapping orthophotos to differ radiometrically, even when their gray level histograms are adjusted for average and standard deviation differences.

To accommodate for the problem of different scales, the scale concept has to be introduced into the matching process itself. This will be conceptually performed by the alteration of the observation equations to accommodate for scale as

$$g_L(x_L, y_L; s_L) - g_R(x_R, y_R; s_R) = e(x, y) \quad (18)$$

which would correspond to a matching process adapting itself into various scales. The above equation may be linearized with respect to x, y and s, essentially adding to the previously mentioned (eq. 17) linearized observation equations one term

$$\begin{aligned} g_L(x_L, y_L, s_L) - e(x, y) &= g_R^\circ(x_R^\circ, y_R^\circ, s_R^\circ) \\ &+ g_{R_x} dx_R + g_{R_y} dy_R \\ &+ g_{R_S} ds_R \qquad (19) \end{aligned}$$

The added term g_{R_S} expresses how gray levels change at a point whenever the scale level of the window within which this point is located changes within the continuous scale space. The term s has conceptual meaning and may be substituted by the σ of the Gaussian filter or any other quantity sufficiently describing scale.

The introduction of a scale parameter in least squares matching may introduce linear dependency. The terms g_{R_x} and g_{R_y} also express gray level gradients, but are different than the term g_{R_S} in that they are highly localized and obviously orientation dependent. Even in the case that high dependency exists, matching may be implemented in two distinct sets, properly constraining some of the parameters to realistic estimated values. To assure succesful implementation, matching has to be performed in the highest possible common resolution of the two conjugate patches. That will obviously be the resolution of the coarser patch, and therefore the finer patch has to be transferred into another scale level using a Gaussian filter.

4. SCALE SPACE REPRESENTATION OF OBJECT SPACE

Object space can be described by the combination of two two-dimensional continuous signals, one ($Z(X,Y)$) expressing its geometric and another ($R(X,Y)$) expressing its radiometric properties. Discretized, these signals are represented by a Digital Elevation Model and a Digital Radiometry Model which can be together referred to as DERM.

Each of the signals can be individually expressed in a scale space representation using the Gaussian kernel, thus preserving the scale space family properties that we presented in section 2. The scale space family of the DEM will consist of DEM of lower resolutions, with each lower resolution level representing a smoothed version of the original signal. Taking advantage of the self-reciprocity of the Gaussian function which states that the Fourier transform of a Gausian is another Gaussian

$$\mathcal{F}[G(x)] = G(w) \qquad (20)$$

we see that convolution with a Gaussian function in the space domain is equivalent to a filtering with a filter of the same shape in the frequency domain [Weaver, 1983]. Therefore, Gaussian convolution can be perceived as filtering with a low-pass filter, the cut-off frequency of which is determined by the scale parameter σ. Coarse scale representations of the DEM preserve the major geometric trends of the surface, corresponding to the lower frequencies of its frequency domain equivalent. In finer resolutions, frequencies of higher order are introduced. Edge detection, with the application of an LoG function to the DEM signal, will locate breaklines [Chakreyavanich, 1991]. Breakline detection can be applied hierarchically, similarly to edge detection in images. In coarse levels of scale space (large w parameter) we detect major breaklines in the topographic surface, while moving to finer resolutions we not only improve the spatial accuracy of these breaklines, but we also identify breaklines of smaller spatial extent.

In a similar fashion, the Digital Radiometry Model (DRM) of the surface can be processed with a Gaussian filter for the generation of its scale space family. Edges in the DRM will correspond to positions where the radiometric properties of the surface present discontinuities.

The recorded image gray values represent the DRM as altered due to the geometric properties of the object space. In the scale space family of DRM there will exist a member which most closely corresponds to the image depicting this DRM. For a DERM with no geometric variations, the edges detected in the image function would correspond to discontinuities in DRM. In realistic situations though, DEM is not flat and the image edges reflect the combined effect of geometric and radiometric discontinuities. Taking advantage of this we can distinguish edges created by geometric and radiometric discontinuities in the object space, by comparing the scale space of the image to the scale spaces of the object space.

5. COMMENTS

Scale space can be used to represent two dimensional signals in various resolutions. This representation can thus be used for images as well as for radiometric and/or geometric descriptions of the object space. It is structured and explorable and it can offer valuable assistance in various photogrammetric processes.

The concept of scale space provides the theoretical foundation for hierarchical implementation of digital photogrammetric tasks, allowing otherwise cumbersome and time consuming modules to be performed quickly and effectively. For instance, automatic stereopair orientation can be performed using digital image pyramids to effectively lead the results to continuously improving accuracies [Schenk et al., 1991].

However, besides implementing some modules in a hierarchical fashion, scale space theory can also be used to refine the performance of well-established processes, such as least squares matching and orthophoto production. By investigating the differential scale variations which exist between conjugate features in different images, we can deduce a scale-adapting matching process aiming at the optimization of least squares matching. In orthophoto production, we can bring features to the same radiometric and geometric level of scale space, thus eliminating discrepancies and improving its overall performance.

In general, the advantage of using scale space theory to represent the object space is twofold. Signals describing the object space can be stored in a compact yet efficient way by recording their discontinuities through scale space and in addition, image and object space can be directly compared and semantic information can be extracted from this comparison.

REFERENCES

Agouris P., Schenk A.F. & Stefanidis A. (1989) *Zero-Crossings for Edge Detection*, Proceedings 1989 ASPRS Fall Convention, Cleveland, pp. 91-99.

Babaud J., Witkin A., Baudin M. & Duda R.O. (1986) *Uniqueness of the Gaussian Kernel for Scale-Space Filtering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 1, pp. 26-33.

Bergholm F. (1987) *Edge Focusing*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 6, pp.726-741.

Chakreyavanich U. (1991) *Regular DEM Data Compression by Using Zero Crossings: The Automatic Breakline Detection Method*, Report No. 412, Dept. of Geodetic Science, The Ohio State University.

Doorn B.D. (1991) *Multi-Scale Surface Reconstruction in the Object Space*, Report No. 413, Dept. of Geodetic Science, The Ohio State University.

Lindeberg T. (1990) *Scale-Space for Discrete Signals*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 3, pp. 234-254.

Lu Y. & Jain R.C. (1989) *Behavior of Edges in Scale Space*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 4, pp. 337-356.

Novak K. (1992) *Rectification of Digital Imagery*, Photogrammetric Engineering & Remote Sensing, Vol. 58, No. 3, pp. 339-344.

Schenk A.F., Li J.C. & Toth C. (1991) *Towards an Autonomous System for Orienting Digital Stereopairs*, Photogrammetric Engineering & Remote Sensing, Vol. 57, No. 8, pp. 1057-1064.

Weaver J.H. (1983) *Applications of Discrete and Continuous Fourier Analysis*, John Wiley & Sons.

Witkin A.P. (1983) *Scale-Space Filtering*, Proceedings 7th International Joint Conference on Artificial Intelligence, Karlsruhe, pp. 1019-1022.

Yuille A.L. & Poggio T. (1983) *Fingerprints Theorems for Zero-Crossings*, A.I. Memo 730, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

SEGMENTATION OF EDGES IN 3-D OBJECT SPACE

Amnon Krupnik
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

In our feature-based matching approach, zero-crossings are matched and represented in 3-D object space by a sequence of densely spaced points. These spatial curves form the basis for reconstructing the surface. Since edges are likely to correspond to object boundaries, the 3-D curves also serve as an important input for object recognition. In this paper we address the problem of segmenting the contours in straight lines and regular curves. We compare different methods, such as split-and-merge and a 3-D version of the $\psi - S$ method.

1 INTRODUCTION

One of the goals of digital photogrammetry is to automatically recognize and extract man-made objects from aerial images. An essential step toward this goal is to extract features and to match them. We have adopted this approach and described it in several papers, e.g., [14,15,20]. A similar approach is also accepted by the computer vision community, e.g., [5].

The features detected in the images are discontinuities of gray values, or edges. In our current implementation, the images are convolved with the Laplacian of a Gaussian (LoG) operator [10]. The resulting zero-crossings are the edges. In the automatic orientation module, which is the first stage of our system, the zero-crossings are matched for determining conjugate points [15,18,20]. Once the orientation parameters are established, the images are resampled to epipolar geometry [2]. Now we begin to reconstruct surfaces where many edges are matched [21], resulting lists of densely spaced points in object space (3-D edges).

The feature-based matching approach offers two major advantages:

- Surface discontinuities are most likely to show up as edges in the image. By detecting these edges, breaklines can be found and the surface reconstruction process [16,17] becomes more robust.
- In many of the cases, edges correspond to object boundaries. Therefore, once the location of edges is known, they serve as building blocks for a symbolic description of the object space. Such descriptions can be matched with symbolic representations of "world" objects, stored in a library.

In order to use the edges for the symbolic description of the object space, we must segment and group them. In this paper, we focus on the segmentation aspect. The goal is to decompose the 3-D curves into primitives which are more explicit than a list of densely spaced points. Specifically, we want to segment the 3-D curves into straight lines, regular curves (circular arcs in our current implementation) and natural lines.

The curve segmentation problem has been addressed extensively in computer vision literature. A popular segmentation method is the Hough transform, proposed in [1]. This method tries to find straight lines from a sparse set of points. In our application points are already organized along edges; Thus, the Hough transform would unnecessarily increase the computational complexity. Ramer [13] presents a simple algorithm to approximate planar curves by polygons. He based his approximation on a maximum offset criterion. We have adopted this criterion in our approach. Pavlidis and Horowitz [11] use a least-squares algorithm to fit straight lines to portions of the curve, and then iterate a split-merge procedure to refine the initial segmentation. Grimson and Pavlidis [8] find the breakpoints of a curve by comparing the original and a smoothed version of the curve. Discontinuities are then easily detected, and regular curve fitting is performed only between discontinuities. Fischler and Bolles [3] describe two methods, one of them passes a "stick" of a certain width and length over the curve, and the other looks at the curve from different "views" followed by a selection of breakpoints according to the maximum votes obtained from these views. Both methods are based on segmenting the curve over different scales, and on perceptual organization. Grimson [7] suggests an approach which is a combination of split-and-merge and $\psi - s$ algorithms. Wuescher and Boyer [19] describe an algorithm based on a constant curvature criterion.

Except for the Hough transform, all the proposed methods consider a plane curve as the input for the process. Grimson [6] mentioned that the segmentation can also be performed in 3-D, but did not elaborate it any further. While curve segmentation in 2-D may be sufficient for many applications, it has some disadvantages:

- Features which appear in one image only are also segmented, although they do not lend themselves to edges in the object space.
- The segmentation of edges performed individually in both images, does not necessarily produce corresponding breakpoints. Therefore, the identification of the same feature in both images becomes

a nontrivial task.

- Although straight lines in the object space are also straight lines in all projections, the converse does not hold. Circular arcs appear as elliptic arcs, which are more difficult to detect.

To avoid these disadvantages we propose to segment the 3-D curves in the object space. In the next section we present two methods for segmenting a 3-D curve into its basic primitives.

2 METHODS

Here we describe two methods suitable for segmenting 3-D curves. These methods are not necessarily the best curve segmentation methods known, rather they demonstrate the concept of segmentation in 3-D space. The first method is a 3-D version of a split-and-merge concept, based on the offset from a straight line. In this method, the curve is segmented into straight lines only. The second method is an extension of the $\psi - s$ concept (see [9] for 2-D description) into 3-D. With this method, straight lines and circular arcs are detected.

The input for both methods are lists of densely spaced points of 3-D edges. It should be noted that the points are not evenly spaced. They are represented by real 3-D coordinates. We are presently investigating another representation, where the edge points are resampled into a 3-D discrete space (voxels).

2.1 Split-and-merge method

As the name indicates, the split-and-merge method consists of the two phases split and merge. In the split phase, the input data is segmented to assure that each segment fulfills a certain condition. In our case, the condition is that all the points contained in a segment are likely to correspond to a straight line. In the merge phase, redundant breakpoints that have been produced during the split phase are eliminated.

The criterion for deciding whether a group of small line fragments can be represented as a longer straight line is the maximum offset. We have chosen the maximum offset and not a least-squares criterion as suggested in [11]. The reason is that the computational cost for applying the least-squares criterion is much higher than the one for the maximum offset, especially in the 3-D case. In addition, we only perform one split and one merge phase because the initial segmentation criterion is strong. A refinement of the breakpoints and a fitting of a straight line to a list of points can be performed once the segmentation has been achieved. In general, the offset criterion is superior to other criteria, since it is not very noise sensitive. Other criteria, such as the orientation of a line, are quite sensitive to noise, as we will see in the next section.

Let us now define the maximum offset criterion. Consider a string of n small line fragments $l_1 \dots l_n$ which are formed by a corresponding set of densely spaced points $p_0 \dots p_n$. This string of fragments can be considered as one longer straight line if the distance from each point $p_1 \dots p_{n-1}$ to the straight line connecting p_0 and p_n does not exceed a predefined threshold value.

The following pseudo code describes the split phase of the split-and-merge method. It processes an edge $E = \{p_1 \dots p_n\}$ or a part of it. The input to the function in the first call is the indices 1 and n of the first and last points of the edge. The function works recursively, and in general its input is the indices of the first and last points of a sub-edge. The set s (which is a global parameter) is initialized to contain the numbers 1 and n.

Split(f, l)

1. $m := -1$
2. $\forall i,\ f \le i \le l$
   (a) $o :=$ the offset of point p_i from the line $\overline{p_f p_l}$
   (b) if $o > m$ then $m := o$; $t := i$
3. if $m >$ MAXOFFSET
   then $s := s \cup \{t\}$; Split(f, t); Split(t, l)

Once the above algorithm terminates, s contains an unsorted list of indices of the potential breakpoints. The merge algorithm, as specified in the following pseudo code, attempts to

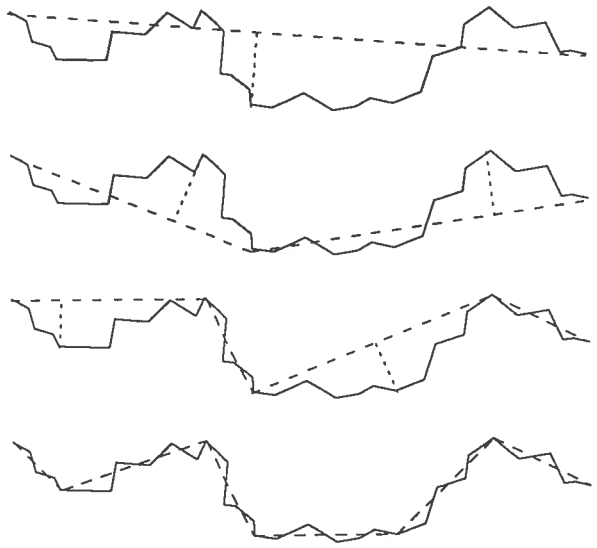


Figure 1: Splitting a curve according to the maximum offset criterion

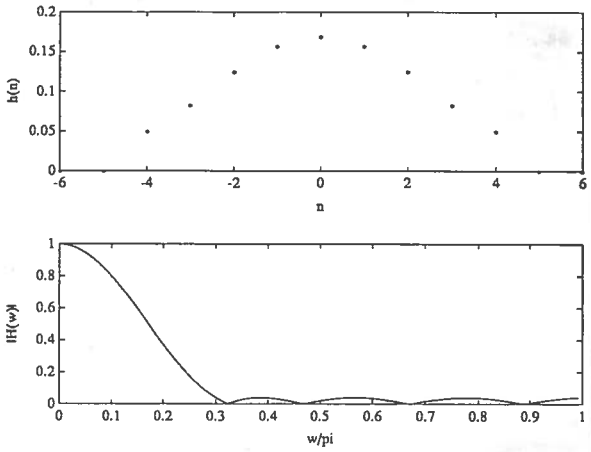


Figure 2: Low-pass filter for the $\psi - s$ curve: impulse response and magnitude of the frequency response

merge two neighboring segments, according to the same maximum offset criterion described earlier.

Merge()

1. Sort the s list by ascending order
2. $k :=$ the number of breakpoints in s
3. $\forall i,\ 2 \leq i \leq k-1$
   (a) $o :=$ the offset of point $p_{s[i]}$ from the line $\overline{p_{s[i-1]}p_{s[i+1]}}$
   (b) if $o >$ MAXOFFSET then $s := s - \{s[i]\}$; $k := k-1$

Figure 1 demonstrates a 2-D curve splitting. The 3-D case is similar, except for the calculation of the offsets. These are calculated in a 3-D coordinate system.

Another aspect is the analysis of the segmented line. Although the split-and-merge method aims at segmenting straight lines, some lines cannot be classified, be it because of noise, short segments or simply because no straight line segments are present.

2.2 Segmentation in $\psi - s$ domain

In order to easily detect circular arcs in addition to straight lines, the $\psi - s$ domain can be used. In this domain, straight lines appear as horizontal lines, and circular arcs as arbitrary straight lines. Since both straight lines and circular arcs appear in the $\psi - s$ domain as straight lines, we can use the split-and-merge algorithm described in section 2.1 to segment the $\psi - s$ curve and get as a result both the straight lines and the circular arcs.

As described earlier, the input is a list of points in a 3-D continuous coordinate system. Since this input is derived from a discrete 2-D representation, noise effects that were produced during the scanning of the original aerial photographs cannot be avoided. The $\psi - s$ method is very sensitive to noise. A point in the spatial domain that is displaced by approximately the distance between two neighboring points will cause $\sim 45^0$ "offset" in the $\psi - s$ curve. Therefore, the original data should be filtered by a low-pass filter. Since the breakpoints we try to detect are also high frequency phenomena, they will be affected too. In order to compromise between noise removal and information preservation, a filter with few coefficients should be used. We used the Parks-McClellan equirip-

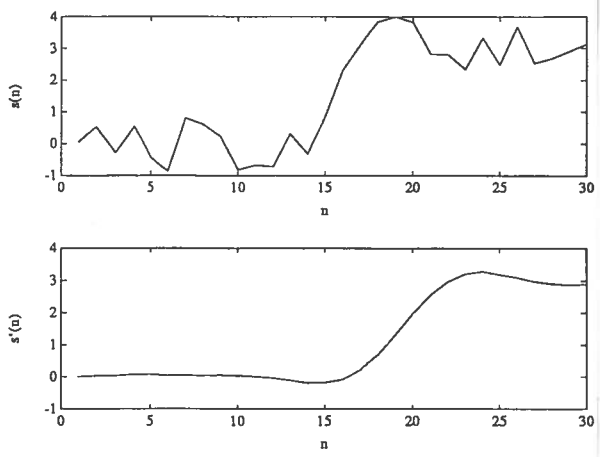


Figure 3: An example of filtering a 1-D sequence by the equiripple low-pass filter

ple algorithm [12] to design such a filter. The equiripple method minimizes the maximum error between an ideal (infinite length) low pass filter and a filter with a truncated number of coefficients. By this, an optimal filter can be achieved for a given set of specifications. This set includes the cutoff frequency, the transition band, a weighting function for the errors in the pass and the stop bands, and the number of coefficients. Recursive or nonlinear filters [19] are alternate solutions to the filtering problem. The impulse and frequency responses of the filter are shown in figure 2. Figure 3 shows a noisy 1-D sequence before and after filtering. In the case of 3-D edges, all three coordinates are convolved separately with this filter.

The $\psi - s$ domain in 2-D space consists basically of a plot of the orientation (ψ) versus length (s) of the original spatial curve. In this representation, the slope of the line corresponds to the curvature of the original curve. Therefore, it can be easily shown that a straight line in the spatial domain appears as a horizontal line (parallel to the s axis) in the $\psi - s$ domain, and a circular arc (which has a constant curvature) appears as an arbitrary straight line. The $\psi - s$ curve for a nonanalytical spatial curve is constructed by computing the directions between points. In order to overcome some residual noise effects, we calculate the direction at a certain point not between the point and its neighbor, but between its predecessor and successor. In cases of more extreme noise residuals, a larger interval can be used for calculations.

In order to segment the $\psi - s$ curve, discontinuities should appear only at breakpoints. An artificial discontinuity is present when the original curve orientation goes from 360^0 to 0^0 or vice versa. Hence, after representing the curve in the $\psi - s$ domain, this artificial discontinuity is eliminated. The procedure is described by the following pseudo code, where c is a parameter which compensates for the discontinuity:

Discontinuity_elimination1()

1. let $p_1 \ldots p_n$ be the list of points of the $\psi - s$ curve
2. $c := 0^0$
3. $\forall i,\ 2 \leq i \leq n$
   (a) $\psi_i := \psi_i + c$
   (b) if $|\psi_i - \psi_{i-1}| \geq 180^0$ then
   - if $\psi_i > \psi_{i-1}$ then
     $\psi_i := \psi_i - 360^0$
     $c := c - 360^0$
     else
     $\psi_i := \psi_i + 360^0$
     $c := c + 360^0$

With this procedure no changes in orientation of more than 180^0 will occur.

We have extended the $\psi - s$ approach to 3-D. A horizontal angle α and a vertical angle ϕ are used to express the spatial direction. Again, a straight line in the spatial domain appears as a line which is parallel to the distance axis of the $\psi - s$ domain. A circular arc, contained in an arbitrary plane in the 3-D space, appears as an arbitrary straight line in the 3-D $\psi - s$ space.

Special attention must be paid when the tangent of a circular arc at a certain point becomes vertical. This situation is described by the following:

$$|\alpha_i - \alpha_{i-1}| \approx 180^0$$

$$|\phi_i| \approx |\phi_{i-1}| \approx 90^0$$

$$\phi_i \approx \phi_{i-1}$$

Careful examination reveals the gradient of ϕ changes its sign leading to a discontinuity of the vertical angle. In order to eliminate this discontinuity problem, the following procedure is added to the transformation of a spatial curve into the $\psi - s$ domain:

Discontinuity_elimination2()

1. let $p_1 \ldots p_n$ be the points of the $\psi - s$ curve, c be a compensation factor for the horizontal angle, z be a zero elevation base for the vertical angle, and s a sign factor
2. $c := 0^0$; $z = 0^0$; $s = 1$
3. $\forall i,\ 2 \leq i \leq n$
   (a) $\alpha_i := \alpha_i + c$; $\phi_i := \phi_i + z * s$
   (b) if $(|\alpha_i - \alpha_{i-1}| \approx 180^0)$ & $(|\phi_i - z| \approx |\phi_{i-1} - z| \approx 90^0)$ & $(\phi_i \approx \phi_{i-1})$ then
      i. if $\alpha_i > \alpha_{i-1}$ then
         $c := c - 180^0$
         $\alpha_i := \alpha_i - 180^0$
         else
         $c := c + 180^0$
         $\alpha_i := \alpha_i + 180^0$
      ii. if $((\phi - z \approx 90^0)$ & $(s = 1))$ or $((\phi - z \approx -90^0)$ & $(s = -1))$ then
         $z := z + 180^0$
         $\phi_i := \phi_i + 180^0$
         else
         $z := z - 180^0$
         $\phi_i := \phi_i - 180^0$

Figure 4 shows a case where the compensation is necessary. α' and ϕ' are corrected angles.

The disadvantage of this approach is that the restoration of the original spatial curve from the $\psi - s$ curve is no longer possible. However, the conversion into the $\psi - s$ domain is done for approximating the location of the breakpoints on the curve. We can certainly store the indices of the found breakpoints, go back to the original spatial domain, and segment the original curve according to these breakpoints.

Once we have a $\psi - s$ curve which does not contain representation related discontinuities, the simplest way to segment it into straight lines

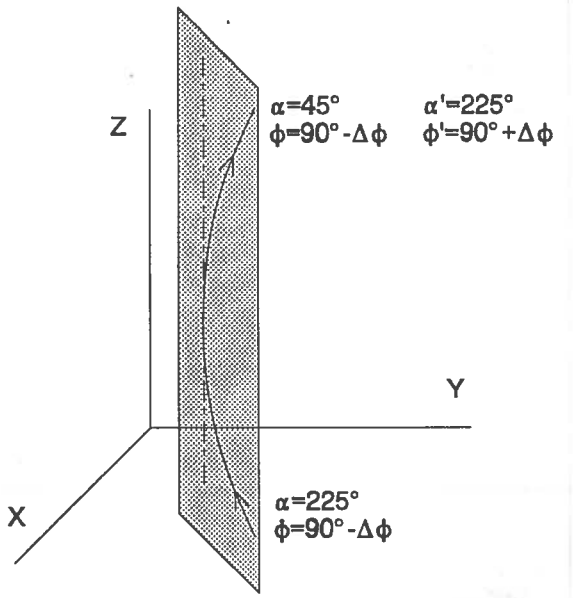
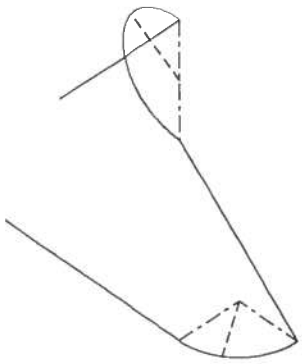


Figure 4: Compensation for an artificial discontinuity

is by the split-and-merge algorithm described in section 2.1. The result of this operation is a list of straight lines in the $\psi - s$ domain. Each of these straight lines is examined and classified into one of three spatial domain categories, namely, straight line, circular arc or "other," i.e., natural lines or noise effects, according to the following order of criteria:

- If the line is shorter than a predefined threshold value, it is classified as "other."
- If the slope of the line is less than a predefined threshold value, it is classified as straight line.
- The radius, arrow and angle of a circular arc are estimated from the slope and first and last points of a $\psi - s$ segment. If these parameters are within a predefined interval, the segment is classified as a circular arc.
- In other cases, the segment is classified as "other."

(a)



(b)

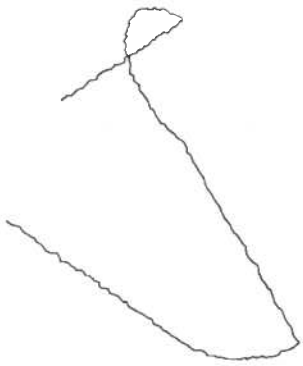


Figure 5: Synthetic data: (a) clean; (b) noisy

3 EXPERIMENTAL RESULTS

Both the split-and-merge and the $\psi - s$ methods were implemented and tested with synthetic and real data. Not all the experiments have been completed yet, leading to more results with real data.

The synthetic data were produced by combining a set of straight lines and circular arcs in 3-D space, which were then corrupted by noise that was produced by a pseudo-random number generator. The magnitude of the noise was chosen in a way that mimics the behavior of real data. Figure 5 shows the clean and noisy synthetic data as 3-D curves. The real data were taken from the results of the matching process, consisting of lists of 3-D points. Figure 6a shows the left image of the stereo pair which was used for the production of these edges. The 3-D edges are shown in figure 6b in an orthogonal projection.

(a)

(b)

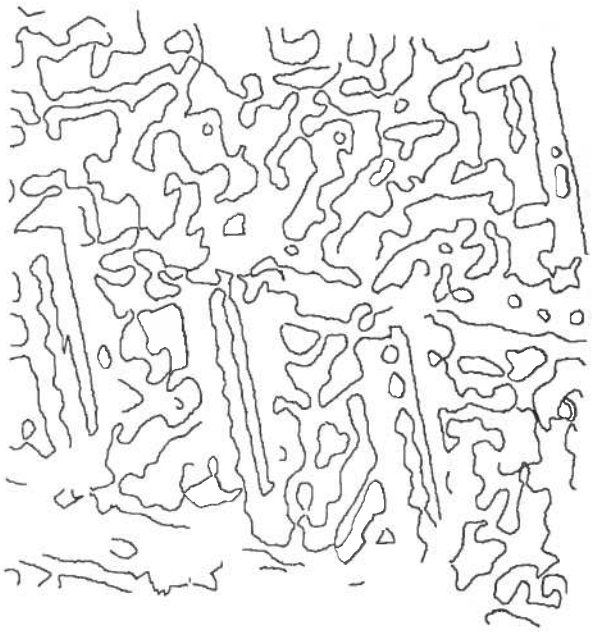


Figure 6: Real data: (a) left stereomate; (b) 3-D edges in orthogonal projection

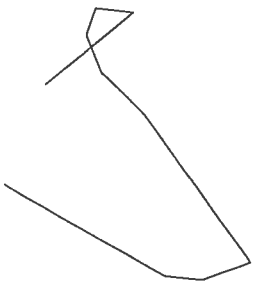


Figure 7: Results of applying the split-and-merge method to the synthetic data

3.1 Split-and-merge results

The split-and-merge algorithm was implemented according to the description in section 2.1. In general, the offset threshold can be derived directly from the scale and the scanning resolution of the aerial images, and it should be larger than the size of a pixel in object space. The aerial photographs we used have a scale of approximately 1/4000, and the scanning pixel size is approximately 60 μm. Therefore, a pixel size in object space is $\sim 0.25\ m$. We selected a value which is slightly higher, taking into account also other noise effects. The threshold was the same for both the split and the merge phases of the algorithm.

Synthetic data: Testing the split-and-merge procedure on the synthetic data did not present any troubles in the segmentation, just as we anticipated. The straight lines were extracted completely, and the circular arcs were segmented into small straight lines. The results are shown in figure 7.

Real data: The results of the split-and-merge segmentation for the real data are shown in figure 8 as an orthogonal projection of the 3-D segments received. These results are very encouraging. Many straight segments were detected. The noisy parts of the curves, which are interpreted as such also by a human observer, remained unchanged. A comparison between the results and the image shows correspondence between straight lines and man-made features.

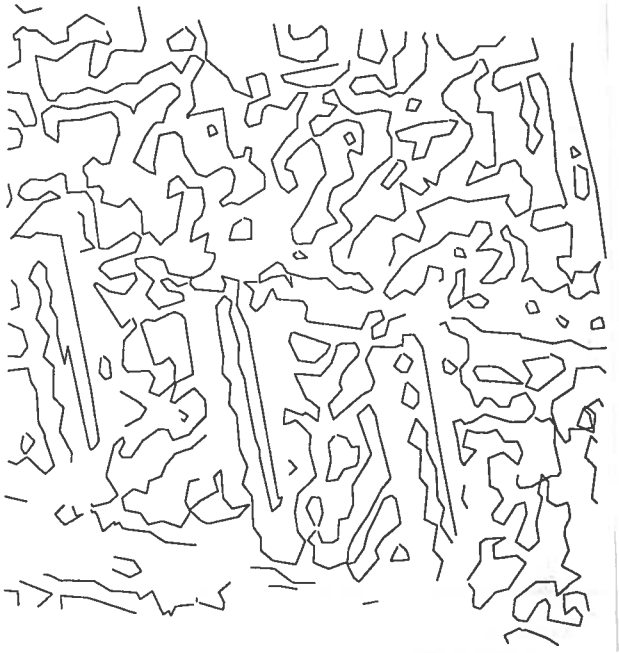


Figure 8: Results of applying the split-and-merge method to the real data (orthogonal projection)

3.2 $\psi - s$ results

The $\psi - s$ segmentation algorithm was implemented according to the description in section 2.2. The selection of threshold values is more crucial than it is for the split-and-merge case. The main reason for this problem is the fact that we deal with angular parameters, while the real physical perturbations are linear. Therefore, the threshold value for a certain line length will not necessarily be suitable for other lengths. Despite this, we used values which are acceptable for the synthetic data, as described below. The offset threshold for the $\psi - s$ curve was set to 10. We also limited the accepted circular arcs radii to the interval $2 - 200\ m$. We have not limited the arc angle and arrow at this stage.

Synthetic data: The results of executing the $\psi - s$ algorithm with the synthetic data are presented in a 3-D view in figure 9. The results need some explanations.

1. Longer segments (either straight lines or circular arcs) were segmented into shorter ones. However, it can be seen that most

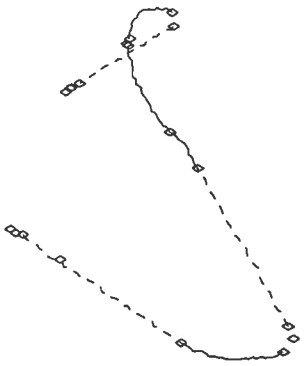


Figure 9: Results of applying the $\psi - s$ method to the synthetic data: straight segments are represented by dashed lines, circular arcs by solid lines, and noise effects are not presented. The breakpoints are represented by squares.

of the segments were classified correctly.

2. Small segments, which were characterized as noise effects were created near the discontinuity points.

The phenomenon of breaking an expected segment into a small number of shorter segments can be resolved in the spatial domain. For example, the second arc in the synthetic example was detected as two smaller arcs and a noise segment. In the spatial domain, these shorter arcs can be combined into a larger arc by applying a least-squares adjustment, and eliminating possible blunders. The noise effects near the breakpoints can be resolved as well. If we eliminate any "short" phenomena, we can intersect neighboring longer phenomena, and by that close the gaps produced by the elimination of the short segments.

Real data: Experiments with the $\psi - s$ method were also performed with real data. We found that the limitations of the $\psi - s$ method, in terms of predefined thresholds, are quite critical. The selection of the threshold values is application dependent, i.e., the approximate size of features should be known.

4 SUMMARY AND CONCLUSIONS

The paper describes curve segmentation in 3-D object space. Although the two methods described for that purpose are not necessarily the best available segmentation methods, the results are encouraging and show that 3-D segmentation is possible.

The split-and-merge method segments the data into straight lines only. Circular arcs are segmented into a list of short straight line segments. The offset criterion used reduces the sensitivity to noise. In other words, the split-and-merge method is quite robust and detects line segments even if they are very noisy.

The $\psi - s$ method offers the advantage of representing circular arcs as straight lines. This property allows detection of circular arcs by using the split-and-merge approach. However, determining threshold values becomes a crucial issue. Due to noise effects, it is dependent on the lengths of lines to be classified. The noise is reduced significantly by a proper filtering of the original data. However, filtering also blurs the breakpoints. Current research focuses on a 3-D Freeman code [4] representation. That is, the object space is discretized, thus reducing some of the noise caused by the scanning process.

The experience gained leads to the following conclusions:

1. Since the $\psi - s$ method allows easy detection of circular arcs, it can be used for a rough segmentation of the 3-D curve into straight lines and circular arcs. Once such approximations exist, other methods can be used to refine the segmentation.

2. Other segmentation methods should be investigated and eventually extended to 3-D.

The segmentation of the 3-D curves is an important clue for man-made features, which are usually composed of 3-D straight lines and other regular curves that provide information which is much more explicit than the original densely spaced points resulting from stereo matching.

ACKNOWLEDGEMENT

Funding for this research was provided by the NASA Center for the Commercial Develop-

ment of Space Component of the Center for Mapping at The Ohio State University.

REFERENCES

[1] Ballard, D. H. and C. M. Brown, 1987. *Computer Vision.* Prentice-Hall, Inc., Englewood, NJ.

[2] Cho, W., M. Madani, and T. Schenk, 1992. Resampling digital imagery to epipolar geometry. In *International Archives of Photogrammetry and Remote Sensing.*

[3] Fischler, M. A. and R. C. Bolles, 1986. Perceptual organization and curve partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):100–105.

[4] Freeman, H., 1974. Computer processing of line-drawing images. *Computing surveys*, 6(1).

[5] Grimson, W. E. L., 1985. Computational experiments with a feature-based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34.

[6] Grimson, W. E. L., 1989b. *Object Recognition by Romputer: The Role of Geometrical Constraints.* MIT Press, Cambridge, Massachusetts; London, England.

[7] Grimson, W. E. L., 1989a. On the recognition of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):632–642.

[8] Grimson, W. E. L. and T. Pavlidis, 1985. Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 30:316–330.

[9] Li, J. C. and T. Schenk, 1991. Stereo image matching with sub-pixel accuracy. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Photogrammetry and Primary Data Acquisition*, pp. 228–236.

[10] Marr, D. and E. C. Hildreth, 1980. Theory of edge detection. In *Proceedings, Royal Society London, B 207*, pp. 187–217.

[11] Pavlidis, T. and S. L. Horowitz, 1974. Segmentation of plane curves. *IEEE Transactions on Computers*, 23(8):860–870.

[12] Rabiner, L. R., J. H. McClellan, and T. W. Parks, 1975. FIR digital filter design techniques using weighted Chebyshev approximations. In *Proceedings, IEEE 63*, pp. 595–610.

[13] Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1:244–256.

[14] Schenk, T., 1989. Application of the zero-crossings method to digital mapping. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Surveying and Cartography*, pp. 325–333.

[15] Schenk, T., Li J. C., and C. Toth, 1991b. Towards an autonomous system for orienting digital stereo-pairs. *Photogrammetric Engineering and Remote Sensing*, 57(8):1057–1064.

[16] Schenk, T., J. C. Li, and C. Toth, 1991a. On using warped images in a hierarchical approach to reconstruct visible surfaces from aerial imagery. In *Proceedings, IEEE Computer Vision and Pattern Recognition Conference.*

[17] Schenk, T. and C. Toth, 1992. Conceptual issues of softcopy photogrammetric work stations. *Photogrammetric Engineering and Remote Sensing*, 58(1):101–110.

[18] Stefanidis, A., P. Agouris, and T. Schenk, 1991. Aspects of accuracy in automatic orientation. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Photogrammetry and Primary Data Acquisition*, pp. 334–343.

[19] Wuescher, D. M. and K. L. Boyer, 1991. Robust contour decomposition using a constant curvature criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):41–51.

[20] Zong, J., Li J. C., and T. Schenk, 1991. Application of förstner interest operator in automatic orientation systems. In *Proceedings, ACSM/ASPRS Annual Convention, Vol. 5:Photogrammetry and Primary Data Acquisition*, pp. 440–448.

[21] Zong, J. and T. Schenk, 1992. Aerial image matching based on zero-crossings. In *International Archives of Photogrammetry and Remote Sensing.*

A LEAST-SQUARES APPROACH TO MATCHING LINES WITH FOURIER DESCRIPTORS

Yi-Hsing Tseng
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

A common problem in computer vision, digital photogrammetry and cartography is to find the best match between a given line and a set of candidate lines, based on characteristics of shape. Fourier descriptors have been used successfully to match lines. In this paper we show how the best geometric fit of two matched lines is determined. The translation, scaling and rotation parameters are found by matching the Fourier descriptors with a least-squares adjustment. A mean-square error can be calculated after matching. This offers the advantage of a quantitative measure of goodness of fit. Experimental results using synthetic data demonstrate the feasibility of the proposed algorithm.

Report No. 418 July 1992

1. INTRODUCTION

To classify a set of patterns or to match two sets of features are common problems in computer vision, digital photogrammetry and cartography. These tasks are broadly known as pattern recognition. The fundamental approach to these problems is to find the best match in shape between a given feature and a set of candidate features. Each candidate feature should be fitted against the given feature one by one, based on their characteristics of shape. This matching process is often conducted to come out with some quantities of intrinsic measure for checking the degree of similarity. The best match is then determined according to the intrinsic measure.

An ambiguity may emerge on the determination of the best match, if there are more than one candidate features having a similar shape to that of the given feature. Because features may be distorted in practice, it is not surprising that the best match in shape is not guaranteed to be a correct match. Under the circumstance, additional information is needed to make a better decision. An extrinsic measure, such as a measure of the relative location, orientation and dilation between features, is considered to be key information to resolve this ambiguity. We therefore suggest that a matching process should generate both of intrinsic and extrinsic measures.

Geometric information of features is often described by lines of the vector form. Image features, such as object boundaries, skeletons, edges or textures can be represented by lines. Although other geometric information such as area, perimeter, number of holes and moments is believed to be useful, the information content of lines is thought to be the most compact, accurate and useful. This paper, therefore, focuses on the matching of lines.

In the last two decades, many techniques, such as polygonal approximation [Pavlidis and Ali, 1975; Greenfeld and Schenk, 1989], $\psi - s$ curves [Ballard and Brown, 1982; Schenk, Li and Toth, 1991], and invariants of Fourier descriptors [Granlund, 1972; Lin and Hwang, 1987], have been proposed to tackle the problem of matching linear features. These techniques emphasize the use of shape invariants, a kind of intrinsic measure, to discriminate linear features. Although these techniques were reported efficient in some cases, two disadvantages were recognized. First, measuring similarity by comparing the shape invariants between features does not provide a clear statistical sense. Second, they cannot provide any extrinsic measure.

In this paper, a new matching process is proposed. It is designed to generate both of intrinsic and extrinsic measures. Our strategy is to transform each candidate line to be optimally matched, in the condition of the least-squares fit, with the given line. The transformation parameters are solved by means of least-squares adjustment. A statistical quantity, the mean-square error, can be calculated after the adjustment. This quantity presents an ideal intrinsic measure. And the estimated transformation parameters offer an extrinsic measure.

A conventional transformation is usually performed about the origin of the coordinate system. The spatial relationships between the original and the transformed features cannot be explicitly described by using the parameters of a conventional transformation. We, therefore, developed a *centroid-based transformation* which transforms a feature about its centroid - the mean position.

The quantities of intrinsic and extrinsic measures are needed to be referred to a spatial coordinate system. It seems necessary to perform the matching process in the spatial domain. However, it is required to pre-define corresponding points between the lines. Difficulties in finding the corresponding points are expected because of the differences of sampling density, scale and starting point. In order to remedy this problem, an algorithm to perform the matching process in the frequency domain is developed, where the Fourier descriptors of lines are matched. The results of matching in the frequency domain are also interpreted with respect to the quantities desired in the spatial domain.

This paper is composed of 6 sections including introduction and conclusion. Section 2 outlines the Fourier descriptors of closed and open lines. Section 3 describes the centroid-based transformation in the spatial and frequency domains. An algorithm of least-squares matching in frequency domain and interpretation of the

results from the matching algorithm are illustrated in section 4. Section 5 presents some experimental results using synthetic data.

2. FOURIER DESCRIPTORS

2.1 Closed lines

A two-dimensional closed line can be described by two periodic functions $x(t)$ and $y(t)$ (Fig. 1). The parameter t is defined as $2\pi l/L$, where L is the perimeter of the closed line and l denotes the arc length along the line from the starting point s to p.

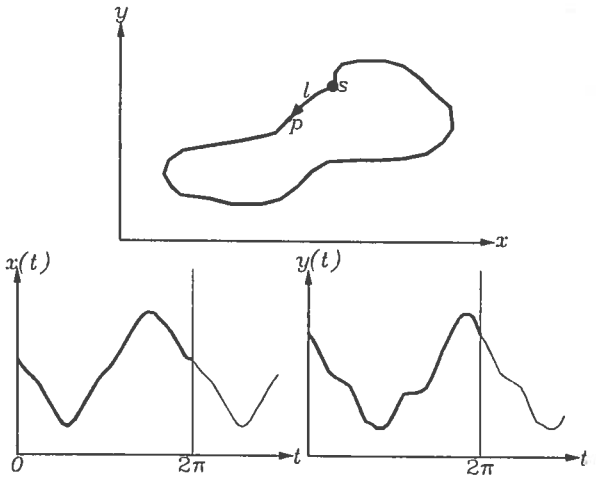


Fig. 1. A 2-D closed line and its periodic functions.

According to the theory of elliptic Fourier descriptors [Kuhl and Giardian, 1982; Lin and Hwang, 1987], these two periodic functions can be expressed by Fourier expansions in matrix form as

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix}, \tag{1}$$

where

$$\begin{aligned} a_0 &= \tfrac{1}{2\pi} \textstyle\int_0^{2\pi} x(t)\, dt; \\ c_0 &= \tfrac{1}{2\pi} \textstyle\int_0^{2\pi} y(t)\, dt; \\ a_k &= \tfrac{1}{\pi} \textstyle\int_0^{2\pi} x(t) \cos kt\, dt; \\ b_k &= \tfrac{1}{\pi} \textstyle\int_0^{2\pi} x(t) \sin kt\, dt; \\ c_k &= \tfrac{1}{\pi} \textstyle\int_0^{2\pi} y(t) \cos kt\, dt; \\ d_k &= \tfrac{1}{\pi} \textstyle\int_0^{2\pi} y(t) \sin kt\, dt. \end{aligned}$$

In Eq.(1), a_0 and c_0 are the mean values of $x(t)$ and $y(t)$ respectively, which indicate the geometric center of the closed line, or so called the *centroid.*

2.2 Open lines

An open line is traced once and then retraced backward so that a closed boundary is obtained (Fig. 2). The Fourier descriptors can then be applied. Let L denote the arc length of an open line and the parameter t is defined as $\pi l/L$. The functions of $x(t)$ and $y(t)$ can be expressed as periodic functions. A close examination of the periodic functions (Fig. 2) yields two important characteristics. First, they are even functions because $x(-t) = x(t)$ and $y(-t) = y(t)$. This implies that the coefficients of b_k and d_k are all zeros. Second, the integration $\int_{t_1}^{t_2} x(t) \cos kt\, dt$ is equal to that of $\int_{2\pi-t_2}^{2\pi-t_1} x(t) \cos kt\, dt$, and it is appropriate to $y(t)$ also. Therefore, an open line can be described with the Fourier expansions as

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k \cos kt \\ c_k \cos kt \end{bmatrix}, \tag{2}$$

where

$$\begin{aligned} a_0 &= \tfrac{1}{\pi} \textstyle\int_0^{\pi} x(t)\, dt; \\ c_0 &= \tfrac{1}{\pi} \textstyle\int_0^{\pi} y(t)\, dt; \\ a_k &= \tfrac{2}{\pi} \textstyle\int_0^{\pi} x(t) \cos kt\, dt; \\ c_k &= \tfrac{2}{\pi} \textstyle\int_0^{\pi} y(t) \cos kt\, dt. \end{aligned}$$

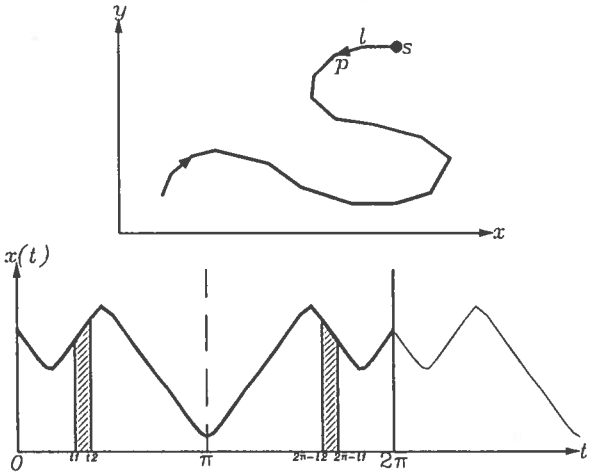


Fig. 2. A 2-D open line and periodic function of $x(t)$.

3. CENTROID-BASED TRANSFORMATION AND PHASE SHIFT

3.1 Transformation in Spatial Domain

If a linear feature consists of a list of (x, y) coordinate pairs of nodes, a transformation in spatial domain is implemented by transforming all coordinate pairs in the list. Conventionally, such transformation is operated about the origin of the coordinate system. For instance, let the list of (x', y') be the coordinate pairs after transformation. A similarity transformation about the origin is expressed as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = S \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \tag{3}$$

where

S	Scale factor;
θ	Rotation angle;
Δx,Δy	Translation.

With this transformation, one can easily discover that the positional change of the transformed feature does not correspond with the translation parameters Δx and Δy, because the centroid of the feature is changed by scaling and rotation. It is, therefore, appreciated that the parameters of this transformation do not explicitly represent the geometric relationships between the original and transformed features. In order to obtain an explicit form of transformation parameters, the change of the centroid should be isolated from scaling and rotation. This can be accomplished by means of transforming a feature about the centroid, which is called *centroid-based transformation*. For example, a centroid-based similarity transformation is expressed as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = S \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \tag{4}$$

where x_c and y_c are the coordinates of the centroid.

3.2 Transformation in Frequency Domain

In frequency domain, instead of transforming coordinate pairs, a transformation can directly operate on the Fourier coefficients. This can be seen mathematically, if the coordinate pairs (x, y) and (x', y') in Eq. (4) are substituted by Eq. (1). A notable fact is that it is natural to perform a centroid-based transformation in frequency domain, because coordinates of the centroid are represented by the coefficients of zero harmonic, a_0 and c_0, and the other coefficients of higher harmonics are independent of the centroid translation. Therefore, a centroid-based transformation in frequency domain can be divided into two parts. The first part is a translation involving just a_0 and c_0. The second part which deals with a transformation that does not affect the position of the centroid, such as scaling, rotation and shearing, involves the other coefficients. These two parts can be done separately.

For the first part and given that the coefficients a_0 and c_0 are coordinates of the centroid, a translation can be directly added to the coefficients of the zero harmonic. Let a_0' and c_0' represent the transformed coefficients, then a translation in frequency domain will be

$$\begin{bmatrix} a_0' \\ c_0' \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \tag{5}$$

For the second part, the Fourier coefficients of non-zero harmonics are pre-multiplied by a transformation matrix, which can be a matrix of similarity or affine transformation. The coefficients of each harmonic can be operated separately, because they are orthogonal. For a similarity transformation, the transformation matrix will be a combination of scale factor and rotation matrices. Let the coefficients with a prime be the transformed coefficients, then the transformation in frequency domain is expressed as

$$\begin{bmatrix} a_k' & b_k' \\ c_k' & d_k' \end{bmatrix} = S \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}, \tag{6}$$

where

$k = 1 \sim \infty$.

3.3 Phase Shift

If a linear feature is recorded by using a sequential list of (x, y) coordinate pairs along the feature, the first point to be recorded is defined as the starting point. A change of the starting point does not alter the geometric property of the feature. However, it does change the Fourier descriptors except for the coefficients of the zero harmonic. For a closed line, the starting point can be anywhere along the curve. If a change of the starting point is interpreted as a change of the phase t and denoted as a *phase shift* Δt, then Δt can be an arbitrary value between 0 and 2π. For an open line, the starting point is either one of the two end points. Its phase shift is therefore 0 or π.

According to the theory of Fourier series, a phase shift is accomplished by post-multiplying the coefficients of each harmonic by a phase shifting matrix, which is similar to a rotation matrix. Mathematically, it can be expressed as

$$\begin{bmatrix} a'_k & b'_k \\ c'_k & d'_k \end{bmatrix} = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos k\Delta t & -\sin k\Delta t \\ \sin k\Delta t & \cos k\Delta t \end{bmatrix}. \quad (7)$$

3.4 Combined Effect of Transformation and Phase Shift

The effects of a transformation and a phase shift can be combined in frequency domain. From Eqs. (6) and (7), a combined effect of a similarity transformation and a phase shift will be

$$\begin{bmatrix} a'_k & b'_k \\ c'_k & d'_k \end{bmatrix} = S \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos k\Delta t & -\sin k\Delta t \\ \sin k\Delta t & \cos k\Delta t \end{bmatrix}. \quad (8)$$

Other transformations can be derived in the same fashion as the similarity transformation. Eq. (5) can be used for all kinds of transformation. All what needs to be changed for another type of transformation is the transformation matrix in Eq. (8). For example, an affine transformation in frequency domain with a phase shift can be formulated as

$$\begin{bmatrix} a'_k & b'_k \\ c'_k & d'_k \end{bmatrix} = \begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos k\Delta t & -\sin k\Delta t \\ \sin k\Delta t & \cos k\Delta t \end{bmatrix}, \quad (9)$$

where

$\begin{bmatrix} e & f \\ g & h \end{bmatrix}$ is an affine transformation matrix.

4. LEAST-SQUARES MATCHING

4.1 Matching in the Spatial Domain

The matching process for two given lines has been defined in the first section. Let a list of (x, y) coordinate pairs represent a candidate line, which is to be transformed in order to match a given line pattern composed of a list of (x', y') coordinates. In the spatial domain, if corresponding points between the two lines can be defined, each pair of corresponding points can form two observation equations, which can be derived from Eq. (4) as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} v_{x'} \\ v_{y'} \end{bmatrix} = S \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad (10)$$

where

$v_{x'}$ and $v_{y'}$ are residuals.

Having the observation equations, a least-squares adjustment can be conducted to solve the unknowns of S, θ, Δx and Δy, by minimizing the summation of $v_{x'}^2$ and $v_{y'}^2$.

In practice, however, corresponding points between two lines are difficult to define, due to the differences of sampling density, scale and starting point. A possible solution is to model the

lines with some mathematical functions and resample the lines at equally spaced points. For instance, one could take the Fourier descriptors to model each line and resample the lines at every $2\pi/n$ interval, in which n is the number of points to be resampled for each line. Total number of $2n$ observation equations can be formed accordingly. Although this method is feasible, it seems not rigorous and efficient. First, the resampling space is difficult to determine. A complicated line requires a small sampling space, but a smaller sampling space increases computation time. Second, the computation is not straightforward. The lines are transformed into the frequency domain when modeling, and are transformed back to the spatial domain when resampling. Therefore, an idea of matching lines in frequency domain, matching Fourier descriptors, emerges to remedy these problems [Zhan and Roskies, 1972].

4.2 Matching in the Frequency Domain

Instead of matching spatial coordinates, Fourier descriptors of each harmonic are matched in the frequency domain. The problem of finding corresponding points no longer exists. The transformation parameters can be directly solved in the frequency domain and they naturally correspond to the idea of the centroid-based transformation, so that the computation becomes efficient and useful.

Since the Fourier descriptors of a closed line and an open line have different properties, the algorithms of closed-line matching and open-line matching are different. In general, open-line matching is a simplified case of the closed-line matching. Because a phase shift does not change the centroid, the translation parameters can be directly calculated for the both cases by using the following formula derived from Eq. (5):

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} - \begin{bmatrix} a'_0 \\ c'_0 \end{bmatrix}. \qquad (11)$$

For the case of matching closed lines, each harmonic (except the zero one) has 4 coefficients, so that 4 observation equations can be constructed for each harmonic. Let a_k, b_k, c_k and d_k be the Fourier coefficients of a candidate line, and the coefficients with a prime represent the given line. If a similarity transformation is applied, then the observation equations can be derived from Eq. (9) as follows:

$$\begin{bmatrix} a'_k \\ b'_k \\ c'_k \\ d'_k \end{bmatrix} + \begin{bmatrix} v_{a'_k} \\ v_{b'_k} \\ v_{c'_k} \\ v_{d'_k} \end{bmatrix} = S \begin{bmatrix} a_k & b_k & -c_k & -d_k \\ b_k & -a_k & -d_k & c_k \\ c_k & d_k & a_k & b_k \\ d_k & -c_k & b_k & -a_k \end{bmatrix} \begin{bmatrix} \cos\theta \cos k\Delta t \\ \cos\theta \sin k\Delta t \\ \sin\theta \cos k\Delta t \\ \sin\theta \sin k\Delta t \end{bmatrix}. \qquad (12)$$

In Eq. (12), the parameters, S, θ and Δt, are unknowns, and the equations are nonlinear. Combining the equations of all the harmonics from 1 to a maximum harmonic m, we obtain an redundant system of nonlinear equations. Least-squares adjustment can be used to solve the unknowns by minimizing the summation of squared residuals. An iterative approach of least-squares adjustment can be applied to solve such nonlinear equations with the given approximations of the unknowns.

For open-line matching, Eq. (12) can be simplified as

$$\begin{bmatrix} a'_k \\ c'_k \end{bmatrix} = S \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_k \cos k\Delta t \\ c_k \cos k\Delta t \end{bmatrix}. \qquad (13)$$

In order to linearize the equations, we let $e = S\cos\theta$ and $f = S\sin\theta$. In addition, the starting point is assumed to be at either end of the open line, so that $\Delta t = 0$ or π. When $\Delta t = 0$, the observation equations will be

$$\begin{bmatrix} a'_k \\ c'_k \end{bmatrix} + \begin{bmatrix} v_{a'_k} \\ v_{c'_k} \end{bmatrix} = \begin{bmatrix} a_k & -c_k \\ c_k & a_k \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix}. \qquad (14)$$

When $\Delta t = \pi$, they become

$$\begin{bmatrix} a'_k \\ c'_k \end{bmatrix} + \begin{bmatrix} v_{a'_k} \\ v_{c'_k} \end{bmatrix} = \begin{bmatrix} a_k(-1)^k & -c_k(-1)^k \\ c_k(-1)^k & a_k(-1)^k \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix}. \qquad (15)$$

The observation equations become linear in this case, so that unknowns can be solved without iteration. However, in order to know whether Eqs. (14) or (15) should be used, the parameter Δt should be determined in advance. The method to approach this will be described in section 4.4. The parameters S and θ can be derived from the solution of e and f by using

$$\begin{aligned} S &= \sqrt{e^2+f^2}\,, \\ \theta &= \arctan(f/e)\,. \end{aligned} \tag{16}$$

4.3 Weight Matrix and Mean-Square Error

Using least-squares adjustment, the mean value of the coordinate differences between two matched lines should be 0, and a mean square error can be calculated from the differences. If we treat the coordinates x' and y' as observations with a variance σ^2, the mean-square error of the match is the best estimate of σ^2. It is obvious that those properties of least-squares adjustment are defined in the spatial domain. Because the matching is performed in the frequency domain, two questions arise. First, what should the weight matrix for the observations a'_k, b'_k, c'_k and d'_k be? Second, how is the mean square error from the residuals $v_{a'_k}$, $v_{b'_k}$, $v_{c'_k}$ and $v_{d'_k}$ calculated?

In order to answer the first question, variances and covariances between the Fourier coefficients should be analyzed. Let the covariance matrix of the x and y coordinates be an identity matrix multiplied by a unit weight variance σ_0^2. From Eq. (1) we have the relationship between the coordinates and the coefficients as follows:

$$\begin{bmatrix} a_k \\ b_k \\ c_k \\ d_k \end{bmatrix} = \begin{bmatrix} \frac{1}{\pi}\int_0^{2\pi} x(t)\cos kt\,dt \\ \frac{1}{\pi}\int_0^{2\pi} x(t)\sin kt\,dt \\ \frac{1}{\pi}\int_0^{2\pi} y(t)\cos kt\,dt \\ \frac{1}{\pi}\int_0^{2\pi} y(t)\sin kt\,dt \end{bmatrix}. \tag{17}$$

One notable fact is that the off diagonal terms of the covariance matrix of the Fourier coefficients should all be zeros, because the functions used to calculate the coefficients are orthogonal. Also according to the error propagation law and Eq. (17), the diagonal terms (variances of the coefficients) can be calculated as follows

$$\begin{aligned} \sigma_{a_k}^2 &= \sigma_{c_k}^2 = \frac{\sigma_0^2}{\pi}\int_0^{2\pi}\cos^2 kt\,dt = \sigma_0^2\,, \\ \sigma_{b_k}^2 &= \sigma_{d_k}^2 = \frac{\sigma_0^2}{\pi}\int_0^{2\pi}\sin^2 kt\,dt = \sigma_0^2\,. \end{aligned} \tag{18}$$

It can, therefore, be concluded that the weight matrix is an identity matrix multiplied by σ_0^2. This conclusion is also appropriate to the case of open-line matching.

The answer to the second question is that the mean-square error (MSE) is equal to the summation of the squared residuals in the frequency domain divided by 2. If the maximum harmonic is m, it can be expressed as

$$MSE = \frac{1}{2}\sum_{k=1}^{m}\left(v_{a'_k}^2 + v_{b'_k}^2 + v_{c'_k}^2 + v_{d'_k}^2\right). \tag{19}$$

Proof. The mean-square error is defined in the spatial domain as

$$MSE = \frac{1}{2\pi}\left(\int_0^{2\pi} v_x^2(t)dt + \int_0^{2\pi} v_y^2(t)dt\right).$$

By substituting Eq. (1) into above equation, the MSE can be expressed in the frequency domain as

$$\begin{aligned} MSE = \frac{1}{2\pi}\Bigg[&\int_0^{2\pi}\sum_{k=1}^{m}\left(v_{a_k}\cos kt + v_{b_k}\sin kt\right)^2 dt + \\ &\int_0^{2\pi}\sum_{k=1}^{m}\left(v_{c_k}\cos kt + v_{d_k}\sin kt\right)^2 dt\Bigg]. \end{aligned}$$

Because of the orthogonal property, the formula becomes

$$\begin{aligned} MSE = \frac{1}{2\pi}\sum_{k=1}^{m}\Bigg(&\int_0^{2\pi} v_{a_k}^2\cos^2 ktdt + \\ &\int_0^{2\pi} v_{b_k}^2\sin^2 ktdt + \int_0^{2\pi} v_{c_k}^2\cos^2 ktdt + \\ &\int_0^{2\pi} v_{d_k}^2\sin^2 ktdt\Bigg). \end{aligned}$$

Because

$$\int_0^{2\pi} \cos^2 ktdt = \int_0^{2\pi} \sin^2 ktdt = \pi\,,$$

we obtain

$$MSE = \frac{1}{2}\sum_{k=1}^{m}(v_{a'_k}^2 + v_{b'_k}^2 + v_{c'_k}^2 + v_{d'_k}^2).$$

The theory and proof are also appropriate to the case of open-line matching, except there are no $v_{b'_k}$ and $v_{d'_k}$ terms.

4.4 First Approximations

The first approximations may be crucial for solving a set of nonlinear equations. With poor approximations, the computation may converge to a wrong solution or even be divergent. It is, therefore, important to provide good approximations for the adjustment computations.

In Eq. (12), let $k = 1$, then it seems possible to solve the approximations, S_0, θ_0 and Δt_0, from the four equations. Unfortunately, the parameters θ and Δt are dependent in each harmonic, so that θ_0 and Δt_0 solved from the first harmonic may be correct or incorrect with a difference of π. In order to assure the approximations are correct, the equations of the first two harmonics should be used.

We firstly linearize Eq. (12) by letting

$$\begin{array}{ll} cc1 = S\cos\theta\cos\Delta t; & cc2 = S\cos\theta\cos 2\Delta t; \\ cs1 = S\cos\theta\sin\Delta t; & cs2 = S\cos\theta\sin 2\Delta t; \\ sc1 = S\sin\theta\cos\Delta t; & sc2 = S\sin\theta\cos 2\Delta t; \\ ss1 = S\sin\theta\sin\Delta t; & ss2 = S\sin\theta\sin 2\Delta t. \end{array} \tag{20}$$

Then they can be solved by using the following formulas:

$$\begin{bmatrix} cc1 \\ cs1 \\ sc1 \\ ss1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & -c_1 & -d_1 \\ b_1 & -a_1 & -d_1 & c_1 \\ c_1 & d_1 & a_1 & b_1 \\ d_1 & -c_1 & b_1 & -a_1 \end{bmatrix}^{-1} \begin{bmatrix} a'_1 \\ b'_1 \\ c'_1 \\ d'_1 \end{bmatrix};$$
$$\begin{bmatrix} cc2 \\ cs2 \\ sc2 \\ ss2 \end{bmatrix} = \begin{bmatrix} a_2 & b_2 & -c_2 & -d_2 \\ b_2 & -a_2 & -d_2 & c_2 \\ c_2 & d_2 & a_2 & b_2 \\ d_2 & -c_2 & b_2 & -a_2 \end{bmatrix}^{-1} \begin{bmatrix} a'_2 \\ b'_2 \\ c'_2 \\ d'_2 \end{bmatrix}. \tag{21}$$

According to Eq. (21), we define

$$\begin{array}{lllll} C1 & = & cc1 - ss1 & = & S_0\cos(\theta_0 + \Delta t_0); \\ S1 & = & cs1 + sc1 & = & S_0\sin(\theta_0 + \Delta t_0); \\ C2 & = & cc2 - ss2 & = & S_0\cos(\theta_0 + 2\Delta t_0); \\ S2 & = & cs2 + sc2 & = & S_0\sin(\theta_0 + 2\Delta t_0). \end{array} \tag{22}$$

Then the approximations can be calculated as

$$\begin{array}{lll} S_0 & = & \sqrt{C1^2 + S1^2}, \\ \theta_0 & = & 2\arctan(S1/C1) - \arctan(S2/C2), \\ \Delta t_0 & = & \arctan(S2/C2) - \arctan(S1/C1). \end{array} \tag{23}$$

In fact, there are other combinations of Eq. (20) to solve the approximations. However, in practice, it is not necessary to elaborate the computation of the approximations.

For open-line matching, although the system is linear, the parameter Δt should be determined in advance. The observation equations of the first two harmonics are also required to solve the problem. According to Eq. (14), there are 2 unknowns and 2 equations, so that one can solve 2 approximate rotation angles from the equations of each harmonic. Let θ_1 and θ_2 denote the rotation angles solved when $k = 1$ and $k = 2$ respectively, then Δt can be determined by using the following algorithm:

$$\begin{array}{ll} \text{if } |\theta_1 - \theta_2| \text{ close to } 0, & \Delta t = 0; \\ \text{if } |\theta_1 - \theta_2| \text{ close to } \pi, & \Delta t = \pi. \end{array}$$

5. EXPERIMENTS

5.1 Matching Closed Lines

Fig. 3 shows a digitized, closed line (A). The line B is a candidate to match line A. The starting points are indicated by solid circles. In order to check the computed transformation parameters we copied line A to line B by the following transformation:

translation	Δx	$= 150$
	Δy	$= 100$
scale	S	$= 0.6$
rotation	θ	$= 45°$

After the transformation, we shifted the starting point to the 10th node and added Gaussian noise ($\mu = 0, \sigma = 3$).

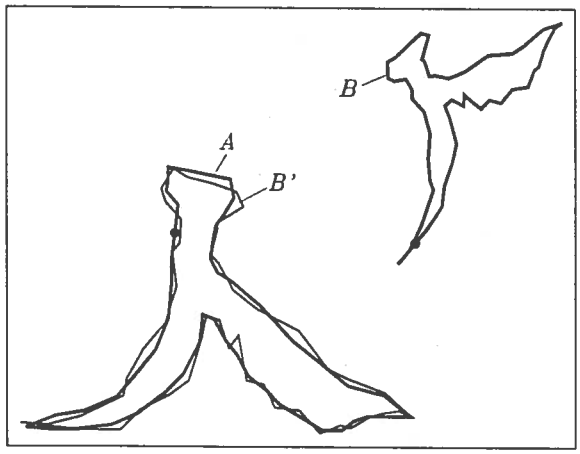


Fig. 3. Example of matching closed lines.

According to the matching process described in section 4, line B is transformed to fit line A. Through the matching process, the mean-square error (intrinsic measure) as well as the transformation parameters and phase shift (extrinsic measure) are calculated. The mean-square error is expected to be about 3, and the calculated transformation parameters should be the inverse-transformation parameters used to copy line B.

The following results are obtained:

Intrinsic measure -mean-square error $= 3.44$
Extrinsic measure-translation $\Delta x = -149.3$

	Δy	$= -99.9$
scale	S	$= 1.65$
rotation	θ	$= -44.9°$
phase shift	Δt	$= -86.4°$

The intrinsic measure is close to the number we expected. Line B' in Fig. 3 is the transformed line B using the estimated transformation parameters. Note that the area between line A and B' is minimized by the proposed matching process.

5.2 Matching Open Lines

Here, we repeat the procedure for open lines. Line D in Fig. 4 is a copy of line C, obtained with the following transformation:

translation	Δx	$= -150$
	Δy	$= -100$
scale	S	$= 0.5$
rotation	θ	$= 180°$

Before the transformation we added Gaussian noise ($\mu = 0, \sigma = 3$) to C.

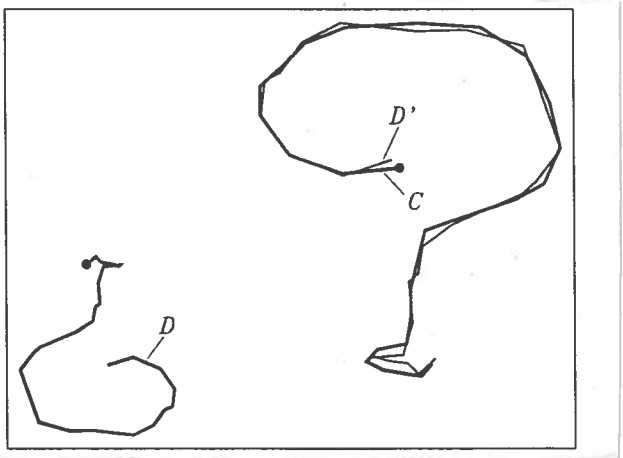


Fig. 4. Example of matching open lines.

We performed two experiments. First, the starting points of line C and D are at the same end. Second, we changed the starting point of line D to the other end. In both cases we obtained the same results. Line D' shown in Fig. 4 is the transformed version of line D. The matching results are listed as follows

Intrinsic measure -	mean-square error		= 2.6
Extrinsic measure-	translation	Δx	= 150.5
		Δy	= 99.8
	scale	S	= 1.976
	rotation	θ	$=-179.4°$

They correspondent with the transformation applied to generate line D.

CONCLUSION

The mean-square error obtained in our matching approach is an ideal intrinsic measure to the goodness of the match. This quantity is obviously more transparent than the use of a table of deviation on shape invariants [Granlund,1972; Lin and Hwang, 1989].

The quantities of extrinsic measure may be useful in some aspects. For example, in the case of recognizing hand-writing of digital numbers, the digits of "2" and "5" as well as "6" and "9" are similar in shape but different in orientation. With intrinsic measure alone, they can hardly be distinguished [Pavlidis, 1980]. Checking the rotation parameter of the extrinsic measure will be helpful in this case. Matching edges of a pair of stereo images [Schenk, Li and Toth, 1991], is another example. One of the edges in the left image may be similar in shape to more than one edge in the right image. In this case, the extrinsic measure provides a global criterion.

In this study we match lines globally, that is, as whole entities. Consequently, lines which only match in parts cannot be dealt with. Since in many applications lines only match in parts, we are presently extending our approach to cope with this situation.

REFERENCES

Ballard, D. H. and Brown,C. M., 1982. *Computer Vision.* Prantice Hall Inc, pp. 237-238.

Granlund, G.H.,1972. Fourier Preprocessing for Hand Character Recognition. *IEEE trans. on Computer*, 21(2):195-201.

Greenfeld, J. S. and Schenk, A. F., 1989. Experiments with Edge-Based Stereo Matching. *Photogrammetric Engineering and Remote Sensing*, 55(12):1771-1777.

Kuhl, F. P. and Giardian, C. R., 1982. Elliptic Fourier Features of a closed Contour. *Computer Graphics and Image Processing*, 18(3):236-258.

Lin, C. S. and Hwang, C. L., 1987. New Forms of Shape Invariants from Elliptic Fourier Descriptors. *Pattern Recognition*, 20(5):535-545.

Pavlidis, T., 1980. Algorithms for Shape Analysis of Contours and Waveforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI*, 2(4):301-312.

Pavlidis, T and Ali, F., 1975. Computer Recognition of Handwritten Numerals by Polygonal Approximations. *IEEE Trans. System, Man and Cybernetic SMC*, 5(6):610-614.

Schenk, A. F., Li, J. C. and Toth, C., 1991. Towards an Autonomous System for Orienting Digital Stereopairs. *Photogrammetric Engineering and Remote Sensing*, 57(8):1057-1064.

Zahn, C. T. and Roskies, R. Z., 1972. Fourier Descriptors for Plane Closed Curves. *IEEE Trans. on Computers*, 21(3):269-281.

CONTROL STRATEGIES FOR AN EXPERT SYSTEM TO INTERPRET LANDFORMS

Abdullah Al-garni
Toni Schenk
Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247

Douglas S. Way
Department of Architechture
The Ohio State University, Columbus, Ohio 43210-1368
USA

Commission VII

ABSTRACT

One of the problems of interpreting landforms by an expert system is the definition of an initial space state by reducing the infinite number of landform types to plausible candidates. In this paper we focus on the initial space states. In particular, we investigate the set size of plausible candidates, and the associated control strategy and search techniques. The comparison of different strategies and techniques in view of landform identification and terrain analysis lends itself into recommended specifications of an expert system with which the problem should be solved.

1 BACKGROUND

Expert systems and their roles in image interpretation receive great interest nowadays (Argialas, D., 1988; Mintezer, O., 1989; Bolstad, P. et al., 1991). Artificial Intelligence (AI) is defined as "*the study of how to make computers do things, which at the moment, people do better*" (Rich, E., et al., 1991). The problem of image interpretation is in quite compliance with this definition; and therefore, image interpretation is recognized as an AI problem.

Expert systems are considered as "*vigorous part of the burgeoning field of artificial intelligence*" (Edmunds, R., 1988). Many definitions of expert system exist today. Bowerman, R., et al., 1988 define it as follows: *"An expert system is a system of software or combined software and hardware capable of competently executing a specific task usually performed by a human expert."* One of the most important aspects of an AI system is the search strategy (Patten, J.,1991; Rich, E., et. al., 1991; Barr, A., et al., 1982). After the knowledge acquisition is completed, a suitable search method must be selected.

Today, two main types of limitations can be observed in the AI field. These limitations are technical limitations, such as storage problems and theoretical limitations, such as the general lack of understanding that characterizes the field of AI, vis-*à*-vis the way human minds process knowledge.

With the rapid advancement in the hardware technology, the technical limitations become less significant. The theoretical problem is improving slowly, and acceptable approximations to human reasoning are available. Scientific experiments are essential to provide suitable theoretical bases about how human minds process large knowledge bases in a matter of microseconds.

In the next section we analyze the problem of interpreting landforms based on terrain analysis. Then we investigate different search strategies followed by developing a control strategy that takes into account the technical and theoretical limitations of AI. Finally, we describe a rule-based program that combines the *establish-and-refine* and *ordered state space* search strategies.

2 STATE SPACE SEARCH AND CONTROL STRATEGIES FOR ITA

To provide an acceptable state-space search and control strategy for *Image Interpretation Using Terrain Analysis* (ITA), a conceptual view of the problem should be investigated. There are three general factors based on which a control strategy can be qualified for an ITA problem. The first factor is the nature of the problem, which can be revealed based on a careful task analysis. The second factor is the experts' methods of attacking the problem in the real world. The final factor is the intended capacity of the system (scalability).

2.1 A Real World Human Model for ITA

Before any strategy can be devised for an expert system, a proper task analysis must be performed (Chandrasekaran, B.,1992 and Patten, J., 1991). The following paragraphs discuss ITA for the purpose of identifying landforms and deducing their parent materials and characteristics for site analysis and evaluation.

First, the ITA task is properly accomplished by experts in the field but not by computers at the moment. Therefore, landform identification for site evaluation purposes is commonly acknowledged to be an AI problem. The other aspect of the problem is that while many facts are well documented in different sources, such as books, reports, and maps, the most important knowledge for ITA is written nowhere but in the minds of the experts. This knowledge contains the strategy of approaching the problem at different circumstances.

To the question "How did you do it?" an expert may reply "It is easy! Well....I know it, but I do not know how I know it". It is this part of the problem that points out the missing links in the chain of the theoretical aspects of AI (Patten, J., 1991). Also, this part of the puzzle calls for more research and exploration to uncover the high level of intelligence required for introducing AI systems into image interpretations in general.

ITA possesses two important AI properties. First, the problem consists of many concepts that can be decomposed, within a general domain, into many subconcepts according to certain criteria (Hoffman, R., 1989a; Mintzer, O., et al., 1984; Mintzer, O., 1988; Strahler,N.,

1981; Way, D., 1973; Zuidam, R., 1985). The other property of the problem is the way the solution is obtained by a human expert. At the beginning the solution is very general; then it is refined until specific conclusions are reached (Way, D., 1992). This property, called coarse-to-fine property, is more obvious in relatively hard and very hard (complex terrain) environments. The coarse-to-fine property is known in AI fields as *hierarchy classification* property. These two properties of the problem are indicative and to a large extent determinative of what control strategies should be devised in AI systems that are to be developed for the ITA problem.

Analysis of processing more than forty models in the field, processed by a recognized expert, indicated that a human expert analyzes the ITA problem in a consecutive logical way. Figure 1 shows a human analysis model for the problem. The model consists of five major phases or modules:
1. Adjustment module
2. Initial settings module
3. Transition phase module
4. Hypotheses module and
5. Verification module.

Many experts do not realize that they reason in this sequence . For instance, experts note the fourth and fifth phases but, often, not the first and third phases. This chain of logical analysis is very important to be realized by the Knowledge Engineer (KE) due to its essentiality in qualifying certain state-space search strategies over others for the ITA problem.

In the real world, an expert is sitting in his office and ready to provide interpretations and consultations for his customers. This is what an expert expects. However, he cannot predict what a customer's image will contain. That is, the expert might work on tens of stereo pairs, each containing different features, terrain, and characteristics. Analogously, an AI system for the ITA task should be ready for any type of tasks for ITA, but within the prespecified limits of the system. For instance, if the system was developed to identify thirty landforms on earth, it should be able to define any of these landforms at any time without an a priori expectation of which landform it will face with the next customer. This ability calls for an engineer to develop a systematic or methodological way of ITA that is general enough to cover the whole spectrum of the task.

This paper assumes a large system with definite number of goals. Figure 2 illustrates the properties of the task of ITA. The general configuration of the triangle indicates the coarse-to-fine property of the problem while the small squares inside the triangle portray the decomposability of the problem to smaller individual concepts. Depending on the granularity or resolution intended by the system, the reached and verified concept could be a single or several concepts. In fact, the ITA problem is methodological in nature (Avery, T. and G. Berlin, 1985) and modular in concept. The modularity of the problem is explained next as a set and subset concept.

2.2 ITA Decomposability Property

Using set theory, let the general concept of the above task be denoted by C_g, and let the first level of the decomposable concepts be a set L_1 where

$$L_1 = C_{11}, C_{12}, ...C_{1n}$$

such that:

$$C_g \supset C_{11}, C_{12}, ..., C_{1n}$$

Then, it is necessary and sufficient for the ITA problem to be decomposable if it has:
1. $C_g \neq \emptyset$.
2. $C_g \supset C_{11}, C_{12}...., C_{1n}$
3. $C_{ij} \cap C_{ik} = \{\emptyset\}$, where $j \neq k$

Now, let $C_{11}, C_{12}, ...C_{1n}$ which are denoted previously by L_1, presents the coarsest level of the concept C_g, then

$$L_1 \in C_g$$

By the same analogy, L_1 may be further decomposed. Let

$$L_2 = \{C_{21}, C_{22}, ..., C_{2k}\}$$

be the second level of the concept that is filtered from the first level. In a similar fashion:

$$C_g \supset C_{21}, C_{22}, ..., C_{2k}$$

Then the set relations

$$L_2 \in L_1$$

and

$$C_{21} \cap C_{22}... \cap C_{2k} = \{\emptyset\}, \text{ where } L_2 \neq \emptyset$$

are held.

The same decomposition continues for the concept until L_g is reached, where L_g denotes the resolution level which contains the goal node:

$$L_3 = \{C_{31}, C_{32}, ..., C_{3i}\}$$

$$\vdots$$

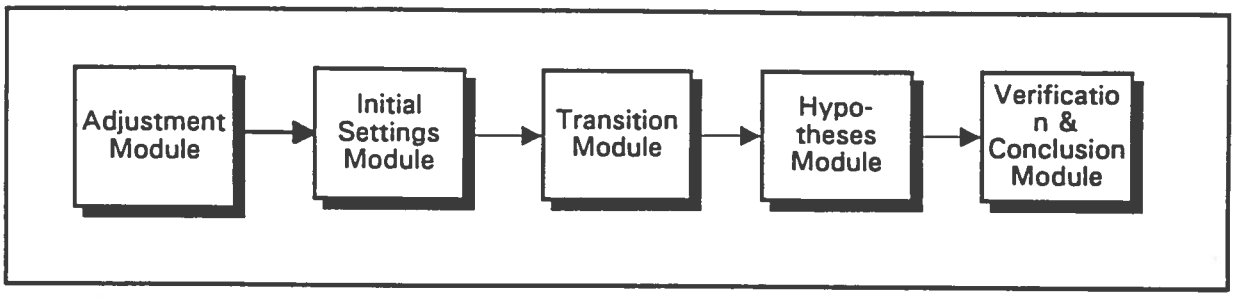


Figure 1: ITA modularity as processed by human experts in the real world

$$L_r = \{C_{r1}, C_{r2}, ..., C_{rm}\}$$
$$\vdots$$
$$L_g = \{C_{g1}, C_{g2}, ..., C_{gs}\}$$

where $g > r > ... > 1$. Then

$$L_g \in L_{g-1} \in ... \in L_1.$$

Denote the level qualification factors that an expert uses to move from one level to the other by $Q_1, Q_2, ..., Q_g$. These qualification factors are the criteria based on which subconcepts are derived until the solution is reached. Figure 3 illustrates the filtration concept and the notion of sets and subsets of the ITA problem. (The reader is referred to Childress, R., 1974; Kaplansky, I., 1972; Eisenberg, M., 1971; Reed, G., 1977 for more information about set theory.)

The explained sets and subsets portray the solution path and should not be confused with the general problem configuration, which may appear quite opposite in a diagram. To illustrate the difference, Figure 4 combines the whole concept. More intention should be paid to the setting of the large triangle as opposed to the settings of the interior, smaller, triangles. Conceptually, these two triangles are similar in that both have coarser knowledge up and finer knowledge down. The difference, however, is in the final outcome of each. The larger triangle presents the whole spectrum of the problem. That is, all landforms existing on earth that the system may identify are listed at the bottom of the large triangle. In contrast, the smaller triangle presents only those landforms that are of interest and appear on a particular image. Therefore, smaller triangles represent a solution while the larger one represents the whole problem (domain). The individual events represented by the small triangles are eventually summed up to constitute the whole population.

2.3 Search Flow of The Human Model

Based on the previously mentioned properties and theories of the nature of ITA problem, it is fair to say that in the real world the absolute initial states of the problem are unknown at the first few moments. This general statement immediately implies unknown goals at the initial state space. For instance, an analyst is told to define all existing landforms in a stereoscopic pair of images. Before looking at the pair, the analyst has no way of knowing where to start and what to expect. This momentary vagueness is soon adjusted according to the adjustment module based on certain criteria in the very few starting steps of the interpretation processes.

This part of the problem (an unknown hypotheses) calls for an immediate forward tracking of the solution by the expert system (initial setting module). Likewise, the human expert is unconsciously conducting a forward search or tracking at his initial settings and scanning of the problem. As soon as the human expert handles the images, looks at them, and reads them, he narrows the problem and defines his starting points or what is called initial state-space. As mentioned previously, control strategy should be in a close compliance with human search strategies. Accordingly, at this level of discussion, the first conclusion is that the initial search control strategy should be developed to work in a forward-tracking (knowledge-driven) manner.

The next step of search control strategy conducted by human experts is to do further careful analysis based on well established criteria to prune all irrelevant concepts from the whole space, sticking only the candidate concepts. This middle level of the search can be either

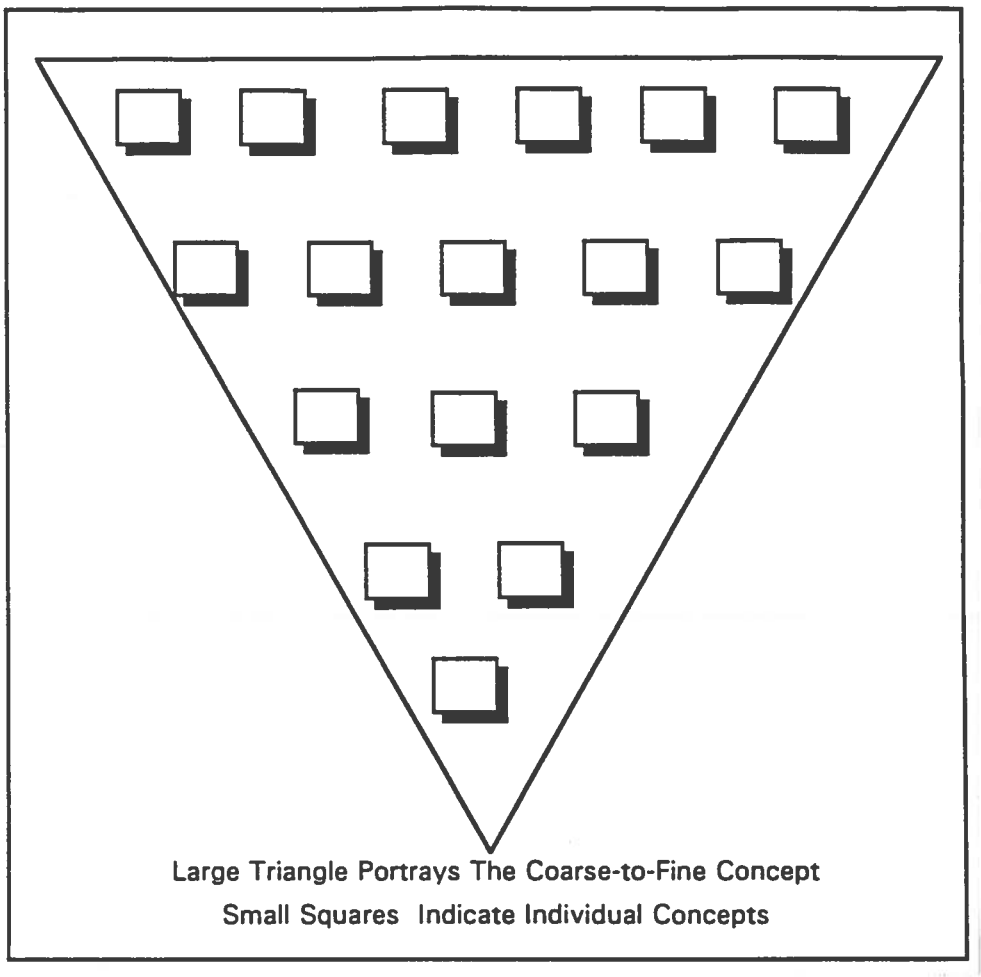


Figure 2: ITAI with coarse-to-fine concept and decomposition property

forward- or backward-tracking. The tracking method depends on how the expert attacks the problem to decompose it into subconcepts. If he has already developed a certain broad hypothesis about several subconcepts, then he is doing a temporary backward tracking of this hypothesis in his mind. But if the problem is still too vague, a forward tracking may continue because the expert has not yet developed any goal to verify (transition phase module).

The third and last step achieved by the expert is to rank the possible and most promising concepts (landforms) in the image and to start to verify them one (or several) at a time (hypotheses module). This implies that, at this level of image interpretation process, a very determined hypothesis (goal or concept) is clearly defined in the expert's mind. Until the goal is verified or disapproved, the whole process is a goal-oriented (or a goal-driven) process. The AI system must follow the human way of attacking the problem and act accordingly. From here on, the rest of the process of the control strategy should use backward tracking for the knowledge search since some goals are developed (verification module). Since there is no absolute forward tracking in AI, it is important to realize that there is a dummy or transitional parameter so that the data-driven search can progress (Chandrasekaran, B., 1992).

2.4 Qualifications and Implementations of Strategies

Like any other AI problem-solving system, the ITA expert system consists of three main components: a database, a set of operators, and a control strategy. Current research is carefully investigating all three components from the viewpoint of image interpretation using ter-

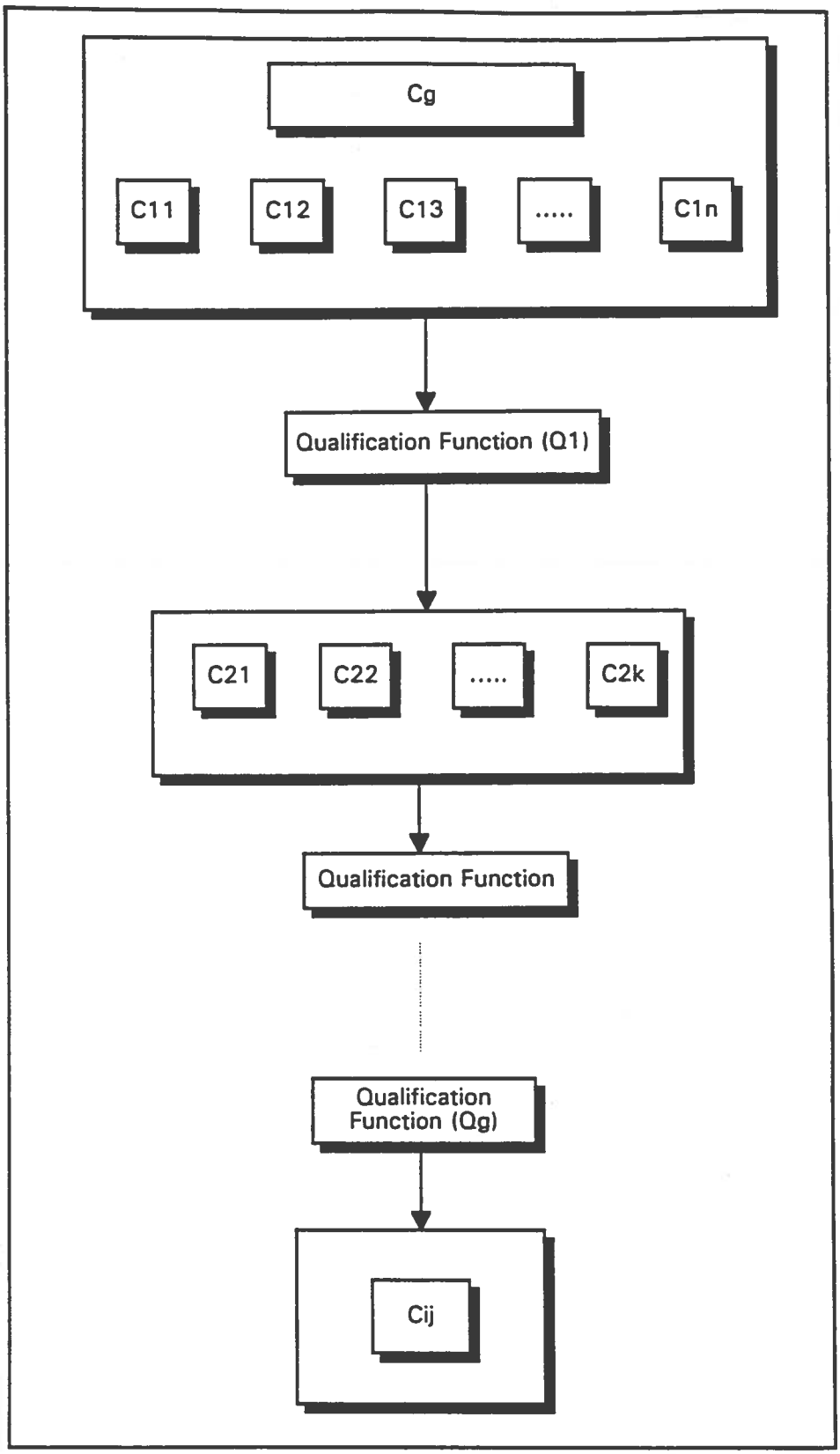


Figure 3: The concept of sets and subsets of the ITAI problem

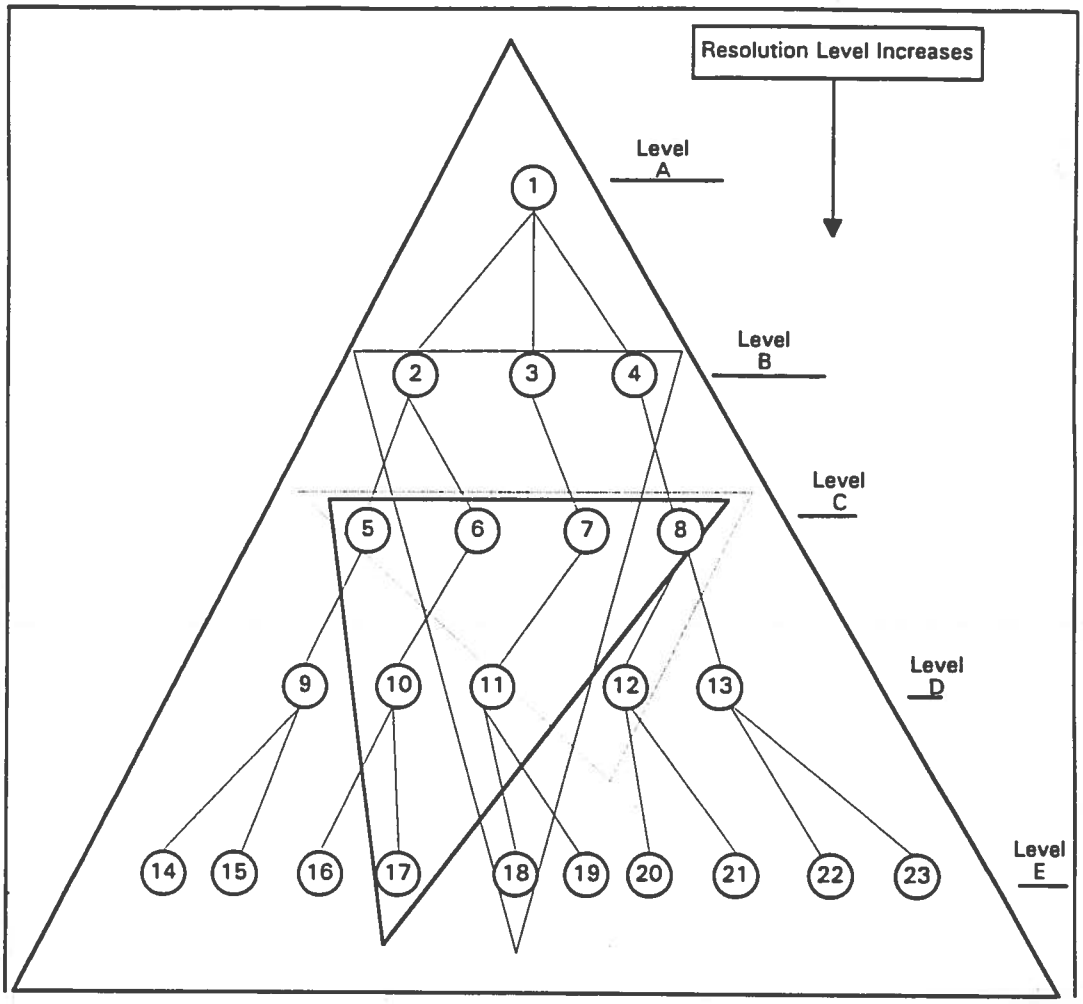


Figure 4: Problem domain setting vs. solution path settings

rain analysis. This paper concerns the third component.

The basic characteristics that any good control strategy should possess are the ability to maintain a dynamic character (motion) of the state-space and the ability to provide a systematic behavior to the whole space (Rich, E. and K. Knight, 1991; Chandrasekaran, B., 1990). The mobility property of any strategy provides the avenues to eventually reach the solutions to the problem under consideration. On the other hand, the systematic property of any strategy prevents the undesirable repeated exploration of useless state-space several times before the solution is reached (Patten, J., 1991).

The content and the organization of the system's knowledge base are influenced by the selected control strategy. The control strategy of a system becomes very obvious in tasks that use operators to modify the problem concepts in a multiple task-domain situation. The ITA problem needs several operator sequences at every level so that the next move is conducted intelligently. This property of the probem exposes two different types of search theories.

The first theory is called blind search theory or control strategy (e.g., breadth-first and depth-first search)(Barr, A., et al. 1982; Rich, E. and K. Knight, 1991). The second theory is called heuristic search theory or strategy (e.g. ordered state-space or best-first search) . These theories are illustrated by presenting three examples so that proper conclusions about the suitability of these theories to the ITA problem are reached.

2.5 Blind State-Space Search Strategies

2.5.1 Breadth-First and Depth-First Search Strategies

Breadth-first search strategy expands the concepts (nodes) according to their proximity to the starting node or concept. Arcs can be used as a measure for node proximity. Accordingly, all possible operator sequence of length n is considered before any sequence of length $(n+1)$. In the ITA problem this strategy declines in value as the system's scalability increases. If careful planning is not practiced before developing the expert system, this problem is dangerous for it may not be very obvious at the initial stages of developing the system.

As it should be understood, expert systems are developed incrementally (Jackson, P., 1986). That is, system development passes through three phases. The first phase is the prototype development of the system. Most often this phase can use the breadth-first search strategy, which can be of great advantage. The next phase is a transition phase. In this phase, the attributes, parameters, and number of landforms to be treated increase. At this phase the system's slowness becomes evident. The third phase of the development is the hybrid system phase. In this phase the problem spectrum is almost completely covered by the system.

Since the number of landforms on earth and their parameters and attributes are so large, a very big knowledge base can be foreseen. This fact makes the breadth-first search strategy unacceptable since its blind behavior causes time and space limitations. The limitations can be visualized by looking at the exponentially expanding nodes in Figure 4. In breadth-first search, if node 23 is an assumed hypothesis in the tree, then this hypothesis cannot be reached until the system searches the whole tree, starting at node 1 on level A through the last hypothesis just before hypothesis 23 on level E (For basic algorithms for this strategy the reader is referred to the references at the end of this paper).

The depth-first search strategy operates as another blind state-space strategy. This search strategy gives the starting node 0 depth, and from there all other nodes are numbered so that the depth of any node is 1 more than the depth of its predecessor. Depth-first strategy expands the most recently generated node by following a single path through the state space downward from the starting node until a goal is reached or a dead end is found. Figure 4 illustrates how depth-first search works. Notice here that the nodes 1, 2, 5, 9, and 14 are treated in the first processed single path, but in the next alternate path operations start at node 9. The process continues until hypothesis at node 23 (an assumed goal) is reached. Thus, after the initial settings of node 2 and its branches are explored, the search starts back at node 3 and explores initial settings of its branches.

Conceptually, these methods of state-space search are incompatible with human expert methods conducted for an ITA task in the real world. It should be realized, however, that this conclusion is based on pure blind search methods in which no criteria are developed to qualify the promising nodes to be explored amongst the list in every level in the state-space problem. When a set of qualifying criteria is developed for these methods, a new and more sophisticated state-space search and control strategies are obtained, which are closer to the human way of reasoning about the

2.5.2 Heuristic State-Space Search for ITA

Heuristic control strategy assesses various operator sequences and signalizes or instantiates the most promising sequence (Barr, A. and E. Feigenbaum, 1982). In fact, heuristic search strategies use certain criteria to direct the search in the state-space of the problem. Based on these criteria and based on the nature of the ITA problem, heuristic state-space search and a combination of forward and backward chain reasoning constitute a set of control strategies that meet the conceptual aspects of the ITA problem; and, therefore, this set is implemented by this study for this problem. This type of search is justified by many facts, some of which were described previously and some of which are discussed next.

Representing the knowledge in the expert system according to the logic of the human expert is a prerequisite in developing control and search strategies for the system. This prerequisite stems from two essential factors. First, the expert has to understand the KE's real attempts to model the expert's own expertise and, as a result, the expert gains the confi-

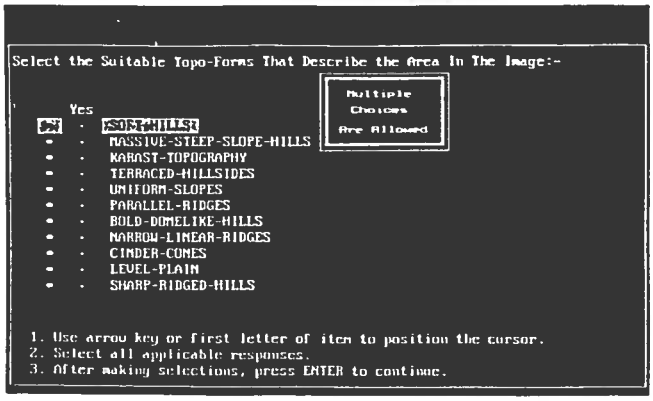
dence to test and evaluate the system's success based on his knowledge and familiarity with the main workings of the system. In relation to this issue, the end user's acceptance of and confidence in the system are more likely to be attained if the knowledge representation and control strategy schemes approximate the expert's knowledge.

Second, the expert's strategy of representing and controlling the knowledge is a whole package of expertise that any AI system should maintain. In a heuristic search the ingredients of the ITA problem are the initial state, the operators, and the goal states (I.O.G). The main objective of the KE is to model the control strategy and logic that the human expert uses when connecting the initial state-space with the goal state-space through appropriate operators.

As can be concluded by now, the blind search of a state-space expands a very large number of nodes before a solution is reached. The reason for that is the arbitrary behavior of expanding the nodes without controlling the search mobility according to the properties of the problem at hand. As a part of the control strategy the triple (I.O.G) is assumed to be established. The rest of the control strategy is, then, to develop heuristic information about the ITA problem and to implement a search method which uses this information to effectively search the given space.

3 ORDERED ESTABLISH AND REFINE SEARCH ALGORITHM (OERSA)

We develop a heuristic state-space search algorithm that well fit the ITA problem. This hybrid search strategy combines the properties of the well known *establish and refine* and *ordered state-space* heuristic search strategies. It is necessary to have a general understanding about what type of heuristic information can be used in searching the space of the ITA problem. This information includes heuristic strategy constraints and can be categorized according to its function into two different categories. The first category is a set of information that qualifies the most promising node to be expanded and which evaluates node successors to generate the best node amongst them. This




```
 ($AND
   (SAME FRAME WEATHER HUMID))
(DO-ALL
  (CONCLUDE FRAME REFERENCE "REFERENCE-MORPHOLOGY" TALLY 100)
  (CONCLUDE FRAME MH11 0 TALLY 100)
  (CONCLUDE FRAME MH22 0 TALLY 100)
  (CONCLUDE FRAME MH33 0 TALLY 100)
  (CONCLUDE FRAME MH44 0 TALLY 100)
  (CONCLUDE FRAME MH47 0 TALLY 100)
  (CONCLUDE FRAME MH54 0 TALLY 100)
  (CONCLUDE FRAME MH65 0 TALLY 100)
  (CONCLUDE FRAME MH76 0 TALLY 100)
  (CONCLUDE FRAME MH811 0 TALLY 100)
  (CONCLUDE FRAME MH99 0 TALLY 100)
  (CONCLUDE FRAME MH1010 0 TALLY 100)
  (CONCLUDE FRAME MH118 0 TALLY 100)
  (CONCLUDE FRAME REFERENCE-M-H "Referencing Morphology
  template Is Accomplished Correctly" TALLY 100))
 ($AND
   (SAME FRAME MORPHOLOGY-H MASSIVE-STEEP-SLOPE-HILLS))
(DO-ALL
  (SET-VALUE TEST 38.5)
  (CONCLUDE FRAME MH11
    (VALUE-OF TEST) TALLY 100)
  (CONCLUDE FRAME CELLS-H
    (VALUE-OF TEST) TALLY 100))
 ($AND
   (SAME FRAME DRAINAGE-H INTERNAL))
(DO-ALL
  (SET-VALUE TEST 23.4)
  (CONCLUDE FRAME CELLS-H
    (VALUE-OF TEST) TALLY 100)
  (CONCLUDE FRAME DH63
    (VALUE-OF TEST) TALLY 100))
 ($AND
   (KNOWN FRAME MORPHOLOGY-H)
   (KNOWN FRAME DRAINAGE-H)
   (KNOWN FRAME PHOTOGRAPHIC-TONE-H)
   (KNOWN FRAME LAND-USE-AND-LAND-COVER-H)
   (KNOWN FRAME EROSION-H))
(DO-ALL
  (CONCLUDE FRAME SS-H
    (PLUS
      (PLUS
        (PLUS
          (PLUS
            (VAL1 FRAME SS-HMA)
            (VAL1 FRAME SS-HDA))
          (VAL1 FRAME SS-HTA))
        (VAL1 FRAME SS-HLA))
      (VAL1 FRAME SS-HGA)) TALLY 100))
```

Figure 5: An expert system contains the OERSA

type of information is used by the heuristic search strategies to eliminate the blind expansions that characterize breadth-first and depth-first strategies. The second category is a set of information that eliminates irrelevant nodes from the whole space.

The implemented algorithm represents the general idea of the heuristic search methods as compared to the blind search methods. The general concept of the OERSA is that it works globally on the total set of nodes that are not yet expanded, and it evaluates them to expand the most promising successors or nodes only. The evaluation function Q is a problem dependent. In the ITA problem, the qualification function Q should be the similarity measure between the current space state node and the goal node instead of the distance or difficulty qualification measure that is used by some other problems. In some instances the Q function in the ITA problem is developed based on elimination criteria, where refinement is conducted for the established nodes. Figure 5 represents small portion of the rules and screens of the expert system which contains OERSA.
The OERSA, implemented by this study, is as follows:

- Start the adjustment module by applying global qualification function Q to the ITA space in order to establish the initial state node S.
- Prepare a list in the initial state node S and evaluate the individual elements in the list according to the Q function (an evaluation function).
- If the node S is empty, then report a failure as an indication that no solution exists.
- If S is not empty, then according to the Q function establish the most promising concept (concept i) in the node.
- Call a recognition agent and test concept i, if the concept is a goal node, then report the proper conclusions and exit with success.
- If concept i is not a goal node, then establish successors of concept i and refine each successor node, say concept k, using the Q function:
  1. If concept k is new, then list it among the other unexpended concepts and give it a pointer to its parent node to trace its path toward the goal concept if found later.
  2. If concept k is not new, then call the probability function, compare k's current value with the previously calculated one, and make proper substitutions. Refinement based on certainty factors are in effect at this stage of the inference process.
- Return to step number 2 and continue.

4 DISCUSSION

The human strategy for interpreting landforms is systematic and of clear conceptual blocks. That is, the process is coarse-to-fine, in general, and is knowledge-driven until the initial space states are set; then the rest of the process is a goal-driven verification of the hypothesis. An AI system for the same purpose must closely follow the same general guide lines. It is not impossible, however, to follow other strategies that could solve the problem but will be characterized by two properties:

- The AI system will not act according to the human methods. This will lead to two conceptual consequences:
  1. The problem may be regarded as not an AI problem, which contradicts the reality of the ITA problem;
  2. The system will lack the property adhering in the word "EXPERT"; or
- The efficiency of the system, both time-wise and storage-wise, may be questionable, especially for large tasks.

It has been explained how each type of control strategy behaves as viewed from an image-interpretation perspective. In reality, both breadth-first and depth-first search methods are characterized by the mobility property that a good strategy maintains. The drawbacks of both, however, are listed here from an AI viewpoint and from the ITA viewpoint as well:

1. Incompatible with the human logic of solving the ITA problem.
2. Depth-first method may be trapped in the state-space and goes through an endless loop.
3. The breadth-first search is characterized by time and space inefficiencies.
4. In both methods, the obtained solutions may be not the optimal solution.

These drawbacks are not necessarily disadvantages for some other types of AI problems. For instance, in some other situations the following are advantages of these methods:

1. Depth-first search is fast.
2. Breadth-first search guarantees a solution if one exists.
3. Breadth-first search finds the shortest path to a solution.

On the other hand, for our particular AI problem , the heuristic search is preferable, for this type of control strategy is applicable to the ITA problem. This is quite true since the number of concepts (concepts of landform identification and evaluation) to be treated is very large and since methods for pruning irrelevant nodes are essential for avoiding the probable "combinatorial explosion" property (exponentially growing nodes). The least important advantage of this search method is the ability to combine the advantages of depth-first search (exploring a minimum number of branches) and the advantages of breadth-first search (avoiding being trapped by a dead end). Most important are the compliance of the heuristic search methods with human reasoning and the intelligence that these methods can provide if proper knowledge about the problem domain is acquired.

5 CONCLUSIONS

Any control strategy must maintain the motion and systematic property for the state-space strategy. For the ITA problem a hybrid control strategy that fits the conceptual aspects of the problem and maintains the AI aspects is recommended and implemented by this study. The proposed control strategy consists of a heuristic state-space search strategy along with a combination of forward and backward chaining.

A close look at the brute-force state-space search strategies (called blind strategies in AI), heuristic state-space search strategies, and theories of chaining can provide great clues to the success of expert systems in the ITA problem. Since this paper concentrates on the conceptual aspects of the problem, there was no attempt to provide recommendations for particular algorithms. However, on the conceptual level we recommend heuristic state-space strategies for the ITA problem. For search algorithms, there are many different ways of selecting from a variety of algorithms for a heuristic state-space search. Popular algorithms, such as generate-and-test, hill climbing, best-first search, problem reduction, and means-ends analysis, can be investigated for the ITA tasks.

REFERENCES

Argialas, D., and R. Narasimhan, 1988. TAX: Prototype Expert System for Terrain Analysis. Journal of Aerospace Engineering, American Society of Civil Engineers, Vol. 1, No. 3, pp.151-170.

Argialas, D. and, C. Harlow, 1990. Computational Image Models: an Overview and a Perspective.Photogrammetric Engineering & Remote Sensing, Vol. 56, No. 6, pp. 871-886.

Avery, T. E. and G. L. Berlin, 1985. Introduction of Aerial Photographs, Fourth Edition. Burgess Publishing Company, Minneapolis, Minesota.

Bowerman, R. and D. Glover, 1988. Putting Expert Systems Into Practice. ISBN.

Barr, A., and E. Feigenbaum, 1982. The Handbook of Artificial Intelligence. Vol. I, William Kaufmann, Inc.

Childress, R., 1974. Sets, Matrices, and Linear Programming. Prentice-Hall, Inc.

Chandrasekaran, B., 1992. CIS 731: Knowledge-Based Systems. The Ohio State University

Chandrasekaran, B.,1990. Design Problem Solving: A Task Analysis. AI Magazine, Vol. 11,No.4,pp.59-71.

Chandrasekaran, B., M. Tanner, and J. Josephson, 1989. Explaining Control Strategies in Problem Solving. IEEE Expert, Vol. 4,No.1, pp. 9-24.

Eisenberg, M.,1971. Axiomatic Theory of Sets and Classes. Holt, Rinehart and Winston, Inc.

Edmunds, R., 1988. Guide to Expert Systems. Prentice-Hall Inc.

Fieschi, M., 1990. Artificial Intelligence In Medicine, Expert Systems. Chapman and Hall.

Hoffman, R., 1989a. What is a hill? Computing the Meanings of Topographic Terms. In A. Kunz & U. Schmitz (Eds.) Linguistic Approaches to Artificial Intelligence. Duisburg, West Germany: University of Duisburg Press.

Jackson, P., 1986.Introduction to Expert Systems. Addison-Wesley Publishing Company.

Kaplansky, I.,1972. Set Theory and Metric Spaces. Allyn and Bacon, Inc. Boston.

Bolstad, P. and T. Lillesand, 1991. Automated GIS Integration in Landcover Classification. ACSM-ASPRS Annual Convention; Technical Papers, Remote Sensing, Vol. 3., pp. 23-32.

Luger, G. F. and W. A. Stubblefield,1989. Artificial Intelligence and the Design of Expert Systems. The Benjamin/Cummings Publishing Company, Inc.

Mintzer, O., and J. A. Messmore, 1984. Terrain Analysis Procedural Guide For Surface Configuration. Report ETL-0352. Engineering Topographic Laboratories, Fort. Belvoir, VA.

Mintzer, O. W., 1989. Research In Terrain Knowledge Representation For Image interpretation And Terrain Analysis, U.S. Army Symposium On Artificial Intelligence Research For Exploitation Of Battlefield Environment, 1-16 Nov,1989 EI Paso, Texas, pp. 277-293.

Mintzer, O., and J. A. Messmore, 1984. Terrain Analysis Procedural Guide For Surface Configuration. Report ETL-0352. Engineering Topographic Laboratories, Fort. Belvoir, VA.

Patten, J.,1991. CIS 630 "Survey of Artificial Intelligence I: Basic Techniques and CIS 730 Survey of Artificial Intelligence II: Advanced Topics". The Ohio State University.

Strahler, N. A.,1981. Introduction to Physical Geography. John Wiley & Sons, Inc., N.Y.

Schenk, T., and O. Zilberstein, 1990. Experiments with a Rule-Based System for Interpreting Linear Map Features. Photogrammetric Engineering & Remote Sensing, Vol. 56, No. 6, pp. 911-917.

Quinlan, J. R.(editor), 1989. Applications of Expert Systems; Volume 2; Based on the Proceedings of the Third And Fourth Australian Conferences. Addison-Wesley Publishing Company

Reed, G. 1977. Set-Theoretic Topology. Academic Press, Inc.

Rich, E. and K. Knight, 1991. Artificial Intelligence. Second Edition. McGraw-Hill,Inc.

Way, D., 1992. Landform Interpretation Sessions For Expert System Implementation at The Ohio State University.

Way, D. S., 1973. Terrain Analysis, Second Edition. ISBN

Wyckoff, J.,1966. Rock, Time, And Landforms. Harper & Row, Publishers, New York.

Zuidam, R. A.,1985. Aerial Photo-Interpretation In Terrain Analysis and Geomorphologic Mapping. Smits Publishers, The Hague.

MULTIPLE IMAGE MATCHING

Peggy Agouris
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

Digital photogrammetry is concerned with the development of algorithms to automate photogrammetric tasks. The majority of efforts though are focused on single stereopairs. This paper addresses the task of simultaneously matching conjugate windows from multiple overlapping images. After establishing a theoretical understanding of the problem, we introduce several approaches and present the associated mathematical principles. We report on the advantages and disadvantages of each one, discuss various implementation issues and in conclusion, we examine potential applications in photogrammetric procedure.

Report No. 418 July 1992

1. INTRODUCTION

Digital photogrammetry has recently emerged as one of the most promising and multi-faceted photogrammetric subfields. A solid body of research work and a wide array of topics have laid the foundation for the evolution of the photogrammetric procedure. Among the research topics, automatic matching is one of the most challenging.

Digital image matching attempts to identify sets of conjugate entities from two or more overlapping images. From the diverse set of matching techniques [Lemmens, 1988], least squares matching is a popular choice [Ackerman, 1984]. Even though there already exists substantial work on this subject, most efforts have been focused on the stereomatching case, which involves a single pair of images. This paper deals with simultaneously matching windows from multiple overlapping images using least squares techniques. The significance of this issue lies in the impracticality of handling single models at the time when processing large blocks is common practice in the photogrammetric industry. Successful and efficient completion of multiple image matching is expected to contribute significantly in the transition of digital photogrammetry from an experimental to a production-oriented status.

Significant research in the area of multiple image matching can be found in [Grün & Baltsavias, 1988],[Heipke, 1992] and [Helava, 1988]. In this paper, we present alternative approaches to the subject by introducing geometric constraints and performing matching in the object space. The general least squares matching procedure is discussed in detail and is subsequently expanded to accommodate multiple image windows. We explore the theoretical issues of the proposed approaches and establish the corresponding mathematical principles. Then, we report on their advantages and disadvantages from a photogrammetric point of view and address several implementation issues.

2. LEAST SQUARES MATCHING

Least squares matching techniques attempt to match windows of pixels by establishing a correspondence between them which minimizes the differences of their gray values. Assuming $g_L(x_L, y_L)$ to be a window of $n_1 \times n_2$ pixels in the left image, and $g_R^o(x_R^o, y_R^o)$ an equal size approximation to its conjugate position in the right image, the objective is to estimate a new location of the right image window $g_R(x_R, y_R)$ such that the gray value differences

$$g_L(x_L, y_L) - g_R(x_R, y_R) = e(x, y) \qquad (1)$$

are minimized. The estimation is performed by the transformation of the coordinates (x_R^o, y_R^o) and resampling of the corresponding gray values. The coordinates of the two windows are related through a perspective transformation to a common surface patch in the object space. Taking into account the very small size of the windows to be matched, their coordinates are assumed to be related to each other by a 6-parameter affine transformation

$$x_R = a_1 + a_2 x_L + a_3 y_L \qquad (2)$$

and

$$y_R = b_1 + b_2 x_L + b_3 y_L \qquad (3)$$

With linearization, the equations

$$g_L(x_L, y_L) - e(x, y) = g_R^o(x_R^o, y_R^o) \qquad (4)$$

become

$$\begin{aligned} g_L(x_L, y_L) - e(x, y) &= g_R^o(x_R^o, y_R^o) + g_{R_x} dx_R \\ &+ g_{R_y} dy_R \end{aligned} \qquad (5)$$

with the terms g_{R_x} and g_{R_y} expressing the local gradient of the right image function in the x and y direction respectively as

$$g_{R_x} = \frac{\partial g_R^o(x_R^o, y_R^o)}{\partial x_R}, \quad g_{R_y} = \frac{\partial g_R^o(x_R^o, y_R^o)}{\partial y_R} \qquad (6)$$

By differentiating and substituting the affine transformation parameters, the observation equations become

$$\begin{aligned} g_L(x_L, y_L) - e(x, y) &= g_R^o(x_R^o, y_R^o) + g_{R_x} da_1 \\ &+ g_{R_x} x_L da_2 + g_{R_x} y_L da_3 \\ &+ g_{R_y} db_1 + g_{R_y} x_L db_2 \\ &+ g_{R_y} y_L db_3 \end{aligned} \qquad (7)$$

One observation equation is formed for every pair of pixels from the left and right image templates, resulting in a total of $n_1 \cdot n_2$ equations

for templates of size $n_1 \times n_2$. Using matrix notation we have

$$-e = Ax - l \tag{8}$$

where the vector of unknowns x is

$$x^T = [da_1, da_2, da_3, db_1, db_2, db_3] \tag{9}$$

and each element of the vector of observations l is of the form

$$l = g_L(x_L, y_L) - g_R^o(x_R^o, y_R^o) \tag{10}$$

while each line of the design matrix A is

$$A = [g_{R_x}, g_{R_x}x_L, g_{R_x}y_L, g_{R_y}, g_{R_y}x_L, g_{R_y}y_L] \tag{11}$$

The least squares solution is

$$x = (A^T P A)^{-1} A^T P l \tag{12}$$

with P the associated, typically diagonal, weight matrix. By using the transformation parameters obtained through the least squares solution to update the coordinates and resample gray values at integer grid coordinates, a new right image window $g_R^1(x_R^1, y_R^1)$ centered at

$$\begin{aligned} x_R^1 &= (a_1^o + da_1) + (a_2^o + da_2)x_L \\ &+ (a_3^o + da_3)y_L \end{aligned} \tag{13}$$

and

$$\begin{aligned} y_R^1 &= (b_1^o + db_1) + (b_2^o + db_2)x_L \\ &+ (b_3^o + db_3)y_L \end{aligned} \tag{14}$$

is selected as conjugate of the stationary left image template $g_L(x_L, y_L)$. A new set of observation equations is formed and solved. In this manner, the true conjugate window $g_R(x_R, y_R)$ is identified as the window $g_R^n(x_R^n, y_R^n)$ at which the least squares iterated solution is converging. It is common practice to use least squares matching as a means for identifying conjugate points rather than windows. Thus, we correspond the point (x_R^n, y_R^n), center of the right image window, to the point (x_L, y_L) of the left image. The maximum allowable pixel coordinate difference between the initial approximation and the final solution for which the technique can still converge is termed pull-in range.

The great advantage of least squares matching is its flexibility and the fact that it is a well-known and documented technique. The basic model which has been described here can easily be expanded to accommodate more than two images or to include various additional constraints. Radiometric parameters can also be included in an effort to compensate for differences in brightness and contrast between the two images, and are particularly helpful when using digitized images of analog diapositives [Pertl, 1985]. However, a radiometric adjustment is typically performed prior to the least squares solution, equalizing the average and the standard deviation of gray values of the two conjugate windows, thus accommodating for uneven radiometric properties of the two images.

3. MULTIPLE IMAGE LEAST SQUARES MATCHING

3.1 Mathematical Formulation for Multiple Images

Multiple image matching can be performed by simultaneously minimizing the gray value differences between all the possible pairs of conjugate image windows. One image window has to be kept constant and serves as the matching template. For every pair of conjugate image windows (w_i, w_j), depicting the same object-space area in the overlapping photos i and j, we form the observation equations

$$g_i(x_i, y_i) - g_j(x_j, y_j) = e_{ij}(x, y) \tag{15}$$

For windows of $n_1 \times n_2$ pixels appearing in n overlapping photographs we have a total of $(n-1) + (n-2) + \ldots + 2 + 1 = \frac{n(n-1)}{2}$ pairs of conjugate image windows and therefore $\frac{n(n-1)}{2} n_1 n_2$ observation equations. According to the general least squares matching approach, each pair of conjugate windows is geometrically related through a six-parameter affine transformation

$$x_j = a_1^{ij} + a_2^{ij}x_i + a_3^{ij}y_i \tag{16}$$

$$y_j = b_1^{ij} + b_2^{ij}x_i + b_3^{ij}y_i \tag{17}$$

or, conceptually

$$(x_j, y_j) = f^{ij}(x_i, y_i) \tag{18}$$

However, we cannot introduce a set of affine transformation parameters for every pair of image windows since that leads to dependency between transformation parameters. Instead, we can use the set of transformation parameters relating each window w_i to the template window w_1

$$(x_i, y_i) = f^{1i}(x_1, y_1) \text{ for } i = 2, 3 \ldots n \quad (19)$$

which uniquely and sufficiently describes the geometric relationships between all possible conjugate window pairs [Tsingas, 1991]. Indeed, the transformation between a window w_j in photo j and its conjugate window w_i in photo i is uniquely described through the parameters relating each window to the template window w_1 as

$$(x_j, y_j) = f^{1j}(f^{1i})^{-1}(x_i, y_i) \quad (20)$$

with the inverse affine transformation $f^{i1} = (f^{1i})^{-1}$ defined as

$$\begin{aligned} x_1 &= \frac{b_3^{1i}a_1^{1i} - b_1^{1i}a_3^{1i}}{a_3^{1i}b_2^{1i} - a_2^{1i}b_3^{1i}} + \frac{-b_3^{1i}}{a_3^{1i}b_2^{1i} - a_2^{1i}b_3^{1i}} x_i \\ &+ \frac{a_3^{1i}}{a_3^{1i}b_2^{1i} - a_2^{1i}b_3^{1i}} y_i \end{aligned} \quad (21)$$

$$\begin{aligned} y_1 &= \frac{a_2^{1i}b_1^{1i} - b_2^{1i}a_1^{1i}}{a_3^{1i}b_2^{1i} - 1_2^{1i}b_3^{1i}} + \frac{b_2^{1i}}{a_3^{1i}b_2^{1i} - a_2^{1i}b_3^{1i}} x_i \\ &+ \frac{-a_2^{1i}}{a_3^{1i}b_2^{1i} - a_2^{1i}b_3^{1i}} y_i \end{aligned} \quad (22)$$

Substituting in equations

$$x_j = a_1^{1j} + a_2^{1j}x_1 + a_3^{1j}y_1 \quad (23)$$

$$y_j = b_1^{1j} + b_2^{1j}x_1 + b_3^{1j}y_1 \quad (24)$$

x_1 and y_1 from equations 21 and 22 we can rewrite the affine transformation relating windows w_i and w_j (equations 16 and 17) as a function of the two sets of parameters which relate each window to the template.

Proceeding further according to conventional least squares approach, we have a total of $6(n-1)$ statistically independent transformation parameters, relating each image window w_i $(i = 2, 3, \ldots n)$ to the reference template. Therefore the dimensions of the associated vector of unknowns

$$x^T = [da_1^{12}, da_2^{12}, \ldots db_2^{1n}, db_3^{1n}] \quad (25)$$

will be $6(n-1) \times 1$. Each set of observation equations (equation 15) must be linearized as following

$$\begin{aligned} g_i(x_i, y_i) - e(x, y) &= g_j^o(x_j^o, y_j^o) + g_{j_x}\frac{\partial x_j}{\partial a_1^{1j}} da_1^{1j} \\ &+ \ldots + g_{j_y}\frac{\partial y_j}{\partial b_3^{1j}} db_3^{1j} \\ &+ g_{j_x}\frac{\partial x_j}{\partial a_1^{1i}} da_1^{1i} + \ldots \\ &+ \ldots + g_{j_y}\frac{\partial y_j}{\partial b_3^{1i}} db_3^{1i} \end{aligned} \quad (26)$$

Each pair of pixels from every pair of windows produces one observation equation. Among the $\frac{n(n-1)}{2}$ distinct pairs of conjugate windows, there exist $(n-1)$ pairs relating each window w_i $(i = 2, 3, \ldots n)$ to the reference template w_1. Observation equations formed by these pairs will only produce six nonzero elements for each line of the coefficient matrix A, at the columns which correspond to the parameters of the f^{1i} affine transformation. Observation equations relating two windows w_i and w_j $(i \neq j \neq 1)$ will produce twelve nonzero elements per line, at the columns corresponding to the parameters of both the f^{1i} and f^{1j} affine transformations. The sparsity pattern of the design matrix A for the case of five conjugate windows is shown in Fig. 1. The dimensions of each block of nonzero elements (gray square) are $(n_1 \cdot n_2) \times 6$, while the parameters are ordered as $f^{12}, \ldots f^{15}$ and the observations as $1-2, 1-3, \ldots, 1-5, 2-3, \ldots, 4-5$. The least squares solution is again

$$x = (A^T P A)^{-1} A^T P l \quad (27)$$

and the final solution is obtained after iterations. The normal matrix $(A^T P A)$ is full but the exploitation of the sparsity patterns of matrix A can facilitate computations and storage requirements.

3.2 Introduction of Geometric Constraints

The previously described technique attempts to match multiple images using solely the

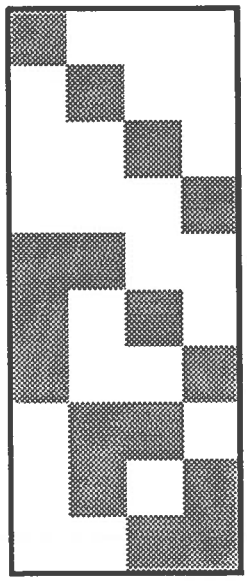


Figure 1: Sparsity pattern of the design matrix for multiple image least squares matching without additional constraints

recorded gray values, without imposing any geometric constraints on the relative position of overlapping images in the object space. By simply using the affine transformation as the geometric relationship between two or more conjugate windows, their geometric interdependence, as expressed by the satisfaction of the collinearity condition equations, is not taken into consideration. Therefore, this approach just minimizes gray value differences without enforcing a geometrically coherent solution. Windows displaying sufficient radiometric similarity can be matched even though their parallax values may be unacceptable. This problem can be overcome either by checking the resulting parallax values or, in a more robust fashion, by introducing geometric constraints within the solution process itself.

Geometric constraints can be introduced either as additional equations [Grün & Baltsavias, 1988], or by properly modifying the expression which relates the coordinate systems of conjugate windows. The image coordinates (x_P^j, y_P^j) (reduced to principal point) of a point $P(X_P, Y_P, Z_P)$ of the object space in photo j satisfy the collinearity condition

$$\begin{bmatrix} x_P^j \\ y_P^j \\ -c \end{bmatrix} = \frac{1}{\lambda_P^j} R^j \begin{bmatrix} X_P - X_o^j \\ Y_P - Y_o^j \\ Z_P - Z_o^j \end{bmatrix} \tag{28}$$

or, in matrix notation

$$\vec{x_P^j} = \frac{1}{\lambda_P^j} R^j (\vec{X_P} - \vec{X_o^j}) \tag{29}$$

where R_j the rotation matrix of image j, $\vec{X_o^j}$ the ground coordinates of the exposure center of photo j and λ_P^j the associated scale factor.

Backsolving the collinearity condition for the image of the same point P in photo i we obtain

$$\vec{X_P} = \lambda_P^i R_i^T \vec{x_P^i} + \vec{X_o^i} \tag{30}$$

and substituting this expression of $\vec{X_P}$ into equation (29) gives

$$\vec{x_P^j} = \frac{\lambda_P^i}{\lambda_P^j} R_j R_i^T \vec{x_P^i} + \frac{1}{\lambda_P^j} R_j (\vec{X_o^i} - \vec{X_o^j}) \tag{31}$$

in which we have two expressions (one for x and and one for y) relating the (x_P^j, y_P^j) image coordinates of point P in photo j to the (x_P^i, y_P^i) image coordinates of the same point in photo i, as a function of the exterior orientation parameters of both photos. Conceptually, in accordance to equation 18, by expanding the equation over pairs of window coordinates (w_i, w_j) and dropping the index P we have

$$(x_j, y_j) = \phi^{ij}(x_i, y_i) \tag{32}$$

with ϕ being the above described function. This function should be considered the object space equivalent of equation 20 rather than equation 18 since the relationship between a pair of windows is described through their relationship to a reference window, which in this case is the object space patch.

By using all potential unique permutations of photo pairs as observation equations, and using one window as the radiometric reference template, as previously described in section 3.1, we can form up to $\frac{n(n-1)}{2}$ distinct pairs of conjugate windows, or $\frac{n(n-1)}{2} n_1 n_2$ corresponding

observation equations. Each observation equation (equation 15) can be linearized with respect to any preselected set of m orientation parameters per photo $(o_1^i, \ldots, o_m^i, o_1^j, \ldots, o_m^j)$ as

$$\begin{aligned} g_i(x_i, y_i) - e(x, y) &= g_j^o(x_j^o, y_j^o) + g_{j_x}\frac{\partial x_j}{\partial o_1^i}do_1^i \\ &+ \ldots + g_{j_y}\frac{\partial y_j}{\partial o_m^i}do_m^i \\ &+ g_{j_x}\frac{\partial x_j}{\partial o_1^j}do_1^j + \ldots \\ &+ \ldots + g_{j_y}\frac{\partial y_j}{\partial o_m^j}do_m^j \quad (33) \end{aligned}$$

The reference template (in photo 1) has to be kept stable, therefore the exterior orientation parameters of photo 1 will be kept constant during the matching process. Thus, the solution can be considered the digital equivalent of dependent analog orientation. Since the original model is non-linear, the final solution is obtained through iterations. The design matrix for this case will have similar sparsity pattern to the one shown in Fig. 1, but the dimensions of each block of nonzero elements will be $(n_1 \cdot n_2) \times m$. After each iteration, the image coordinates of point P in photo j are updated due to changes in orientation parameters

$$x_j = x_j^o + \frac{\partial x_j}{\partial o_1^j}do_1^j + \ldots + \frac{\partial x_j}{\partial o_m^j}do_m^j \quad (34)$$

and

$$y_j = y_j^o + \frac{\partial y_j}{\partial o_1^j}do_1^j + \ldots + \frac{\partial y_j}{\partial o_m^j}do_m^j \quad (35)$$

By solving the above system we inherently ensure that conjugate image rays intersect at a point in the object space. While plain least squares matching is solely a radiometric adjustment, the use of object space constraints to express the relationship of two or more conjugate windows allows the combination of the radiometric and geometric solutions in a single adjustment procedure. The model can be expanded to include the object space coordinates of point P which can be introduced into the adjustment by properly expressing the scale factors as functions of them. In addition, the technique can be expanded to simultaneously adjust observations of more than one point in the object space. The images of all points in each photo will be related through a common set of exterior orientation parameters and the adjustment can thus proceed in a global manner.

3.3 Matching in the Object Space

By examining the image formation process we can extract some rules which can later be used in the matching process not only as constraints but also to expand the problem into the radiometric and/or geometric reconstruction of the object space itself.

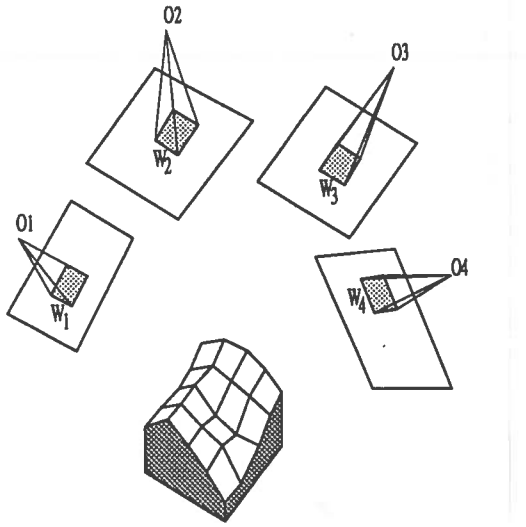


Figure 2: Overlapping image windows in the object space

Fig. 2 shows four image windows $w_1, \ldots w_4$ displaying approximately the same surface patch S in four overlapping images. The surface is described by two continuous functions, one geometric $Z(X,Y)$ (elevations) and another radiometric $G(X,Y)$ (gray values). Assuming a local tesselation, whereby the surface patch S is represented as a Digital Elevation and Radiometry Model ($DERM$, a term analogous to DEM) with a resolution of $n_1 \times n_2$ grid points, the patch is defined by $n_1 \cdot n_2$ elevations and by an equal number of gray values. The reconstruction of the patch would therefore involve the determination of these $2 \cdot n_1 \cdot n_2$ parameters. These parameters can be determined by defin-

ing the geometric and radiometric transformations which relate S to its images $w_1, \ldots w_4$. Each image window w_i corresponds to a gray value function $g_i(x_i, y_i)$, related to S through a geometric transformation

$$(x_i, y_i) = T_g^i(X, Y, Z) \tag{36}$$

and a radiometric one

$$g_i(x_i, y_i) = T_r^i[G(X, Y)] \tag{37}$$

Assuming the object space patch S to be a Lambertian surface, the recorded image irradiance $g(x, y)$ (image gray values) is directly related to the surface radiance $G(X, Y)$ (surface patch gray levels). Furthermore, taking into account the relatively small size of the surface patch, the rather complex radiometric relationship between image and object space can be effectively approximated by a linear transformation

$$g_i(x, y) = r_0^i + r_1^i G(X, Y) \tag{38}$$

Assuming a Lambertian light source, the values of the radiometric shift (r_0^i) and scale (r_1^i) parameters are unique for each image and they are functions of the surface albedo as well as of the angles formed between the image window w_i and the normal to the surface [Horn, 1986].

The radiometric adjustment is typically performed prior to the matching process, by forcing each window w_i to have the same average and standard deviation of gray values as the reference template w_1. Thus, we actually force

$$r_0^i = r_0^1 = r_0 \qquad \text{and} \qquad r_1^i = r_1^1 = r_1 \tag{39}$$

Subsequently, the gray values $g_1(x_1, y_1)$ of w_1 are assigned to the surface patch S. The assignment can be performed either directly:

$$G(X, Y) = g_1(x_1, y_1) \tag{40}$$

or through an inverse linear transformation

$$G(X, Y) = \frac{g(x, y) - r_0}{r_1} \tag{41}$$

In order for an inverse linear transformation to be used, the parameters r_0 and r_1 have to be determined using a priori knowledge on the surface radiance, while for the correct assignment of gray values we must have some approximations of image orientation.

Assuming a smooth surface and no extreme variations in exposure geometry, we can accept an one-to-one correspondence between the object space tesselation and the image windows. Therefore, the observation equations which were developed in the previous sections can now be formed using the object space as the reference template

$$G(X, Y) - g_i(x_i, y_i) = e(x, y) \tag{42}$$

thus transferring the matching procedure to the object space. The geometric relationship between the image coordinates (x_i, y_i) of a point in photo i and its object space coordinates (X, Y, Z) will be in general a seven-parameter transformation

$$(x_i, y_i) = \xi^i(X, Y, Z) \tag{43}$$

This transformation need not be the collinearity condition, as long as the seven parameters which describe the three translations, three rotations and one scale factor are linearly independent. Some of the transformation parameters may also be kept constant during the adjustment, if a priori information allows us to consider them known.

In order for the elevation values to be computed through the adjustment, they have to be introduced as adjustable quantities. This can be performed by proper selection of the other six transformation parameters $(p_1^i, \ldots p_6^i)$ to avoid dependencies which would lead to ill-conditioned systems. The linearized observation equations for this case are

$$\begin{aligned} G(X, Y) - e(x, y) &= g_i^o(x_i^o, y_i^o) + (g_{i_x} \frac{\partial x_i}{\partial p_1^i} \\ &+ g_{i_y} \frac{\partial y_i}{\partial p_1^i}) dp_1^i + \ldots \\ &+ (g_{i_x} \frac{\partial x_i}{\partial p_6^i} + g_{i_y} \frac{\partial y_i}{\partial p_6^i}) dp_6^i \\ &+ (g_{i_x} \frac{\partial x_i}{\partial Z} + g_{i_y} \frac{\partial y_i}{\partial Z}) dZ \end{aligned} \tag{44}$$

Taking into account that different pixels in the image window w_i correspond to different elements of the ground tesselation (groundels, [Helava, 1988]), we see that the dZ element of

the above equation is actually a vector of $n_1 \cdot n_2$ elements. Thus, the design matrix A of the adjustment solution will have the sparsity pattern shown in Fig. 3. In this figure, the large gray blocks have dimensions $(n_1 \cdot n_2) \times 6$, while the small black blocks indicate single entries. This pattern corresponds to the four overlapping images of Fig. 2, without using observations in between windows. Window w_1 has been projected to the object space during the radiometric adjustment and the observations relate the surface patch S to the windows w_2, w_3 and w_4. A more detailed description and in-depth analysis of this technique can be found in [Schenk & Toth, 1992].

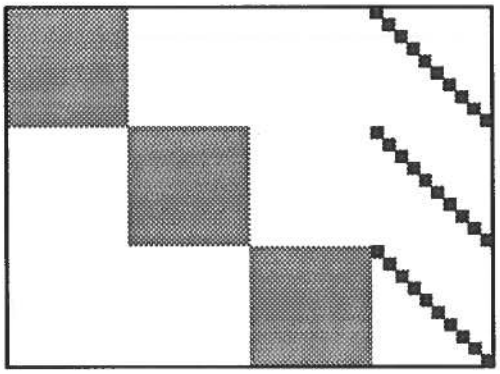


Figure 3: Sparsity pattern of the design matrix for object space matching

Conceptually, object space matching resembles matching with geometric constraints. Taking into account the fact that all images are created from the same object space patch, least squares matching is enforced to produce a geometrically acceptable solution. Simultaneously, we are able to reconstruct the object space $DERM$. Considering that one photo is used to create the object space patch, it is clear that this technique is equivalent to dependent orientation.

4. IMPLEMENTATION ISSUES

In the previous sections we presented and analyzed the formulation of least squares matching using multiple images. By introducing geometric constraints and performing matching in the object space, consistent matching results can be ensured and surface patches can be reconstructed geometrically as well as radiometrically.

Approximations are obviously necessary and they can be in the form of conjugate point image coordinates, orientation parameters and/or the object space surface, as expressed by, e.g., an initial DEM approximation. These approximations can be easily obtained through an automatic stereopair orientation module [Schenk et al., 1991]. Experiments have shown that accuracies of the order of $\frac{1}{10}$ to $\frac{1}{15}$ of a pixel (or $4 - 6\mu m$ in photo coordinates) are to be expected when the technique is applied as a combination of feature-based hierarchical matching and correlation methods with continuous updating of the results through scale space [Stefanidis et al., 1991].

Automatic stereopair orientation and least squares multiphoto matching can be ideally combined in automatic aerotriangulation of large blocks of images, the former providing valid initial approximations and the latter, being the core module of the procedure, performing precise point determination. This fusion of more than one module should be expected, since initial approximations are required in aerotriangulation. Stereopair orientation essentially performs automatically the task of selecting conjugate image windows located in the areas where conjugate points are desired, the equivalent of the preparation phase in the conventional aerotriangulation procedure. Using these initial matching approximations, the images are approximately brought in their correct relative positions in space. This can be visually materialized for operator inspection, if desired, through the generation and continuous updating of a photomosaic. Fig. 3 depicts a photomosaic of three images, an early product of the automatic aerotriangulation procedure. The simultaneous multiphoto matching technique can also be conceptually viewed as the digital equivalent of an n-stage comparator, allowing for the measurement of conjugate points in more than two images at a time. Several gross errors, associated with erroneous conjugate point identification, which limit the accuracy of conventional analytical aerotriangulation can thus be avoided, optimizing the potential accuracies of the technique.

Figure 4: A photomosaic of three photos

By using a feature-driven stereomatching method to obtain the initial approximations for multiphoto matching, we ensure the selection of areas of sufficient radiometric variation which inherently lead to better matching accuracy. In addition, these areas will most likely correspond to features of interest in the object space, since gray level variations are caused by markings on the ground, and changes in radiance and/or surface orientation. The use of least squares techniques for matching provides the additional advantage of producing results with objectively estimable accuracy, allowing for the proper assignment of weights. Observations in windows of low entropy, which are typically susceptible to erroneous matches can be assigned smaller weights, thus minimizing their effect in a global solution.

In conclusion, it is obvious that multiple image matching is an essential tool in digital photogrammetry. The introduction of geometric constraints and its performance in the object space can contribute to making it more rigorous in theory and consequently practically improved. Combined as discussed with already developed modules, such as stereomatching, it can fully automate the aerotriangulation procedure, and significantly assist in upgrading the mapping process.

REFERENCES

Ackermann, F. (1984) *Digital Image Correlation: Performance and Potential Application in Photogrammetry*, Photogrammetric Record, Vol. 11, No. 64, pp. 429-439.

Grün, A. & E. Baltsavias (1988) *Geometrically Constrained Multiphoto Matching*, Photogrammetric Engineering & Remote Sensing, Vol. 54, No. 5, pp. 633-641.

Heipke, C. (1992) *A Global Approach for Least-Squares Image Matching and Surface Reconstruction in Object Space*, Photogrammetric Engineering & Remote Sensing, Vol. 58, No. 3, pp. 317-323.

Helava, U.V. (1988) *Object-Space Least-Squares Correlation*, Photogrammetric Engineering & Remote Sensing, Vol. 54, No. 6, pp. 711-714.

Horn, B.K.P. (1986) *Robot Vision*, MIT Press, McGraw-Hill Book Co., 1986.

Lemmens, M.J.P.M. (1988) *A Survey on Stereo Matching Techniques*, International Archives of Photogrammetry and Remote Sensing, Kyoto, Japan, Vol. 27, Part B8, pp.V11-V23.

Pertl, A. (1985) *Digital Image Correlation with an Analytical Plotter*, Photogrammetria, Vol. 40, No. 1, pp. 9-19.

Schenk, A.F., J.C. Li & C. Toth (1991) *Towards an Autonomous System for Orienting Digital Stereopairs*, Photogrammetric Engineering & Remote Sensing, Vol. 57, No. 8, pp. 1057-1064.

Schenk, A.F. & C. Toth (1992) *Reconstructing Small Surface Patches from Multiple Images*, International Archives of Photogrammetry & Remote Sensing, ISPRS XVII Congress, Washington, D.C..

Stefanidis, A., P. Agouris & A.F. Schenk (1991) *Aspects of Accuracy in Automatic Orientation*, Proceedings 1991 ASPRS Annual Convention, Baltimore, Vol. 5, pp. 334-343.

Tsingas, V. (1991) *Automatische Aerotriangulation*, Proceedings of the 43rd Photogrammetric Week, Stuttgart, Heft 15, pp. 253-268.

RECONSTRUCTING SMALL SURFACE PATCHES FROM MULTIPLE IMAGES

Toni Schenk
Charles Toth

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

ABSTRACT

In this paper we formulate suitable mathematical models for rigorously solving the multiple image matching problem. We describe a general solution by introducing a geometric and radiometric relation between a surface patch and the corresponding image patches. We then investigate several geometric transformation models between object and image space, including linearized observation equations. We conclude with describing two approaches of using multiple image matching in aerotriangulation. The first approach is the most general one where the exterior orientation, the surface patches with elevations (DEM) and gray levels are all determined simultaneously. In the second approach conjugate points are determined independently from one another and without the exterior orientation parameters. This solution corresponds to the traditional method where all points are individually measured and then entered into a block adjustment.

1 INTRODUCTION

Determining conjugate points is a fundamental task that occurs in almost any photogrammetric application. In digital photogrammetry it has become customary to call this process *image matching*. It is fascinating to observe the development of image matching during the last decade. Much progress has been made since Ackermann (1984) and Förstner (1984) presented the first rigorous mathematical models for the image matching process. Apart from extending the basic mathematical model by introducing geometrical constraints (see, e.g., Grün & Baltsavias, 1988), a decisive step was to combine matching with reconstructing the surface (see e.g. Wrobel, 1987; Helava, 1987; Ebner *et al.*, 1987).

Ebner and Heipke (1988) propose a new approach where matching several images and surface reconstruction is treated as a simultaneous adjustment problem. Matching multiple image patches is of great practical importance, most notably in aerotriangulation where as many as nine photographs may partially overlap. Identifying and measuring tie points, particularly between strips, is a notorious problem, since only two photographs can be viewed stereoscopically at the same time. The reliability and accuracy of tie points is expected to significantly increase when multiple image matching methods are employed.

The purpose of this paper is to formulate suitable mathematical models for rigorously solving the multiple image matching problem. We describe a general solution by introducing a geometric and radiometric relation between a surface patch and the corresponding image patches. We then investigate several geometric transformation models between object and image space, including linearized observation equations. We conclude with describing two approaches of using multiple image matching in aerotriangulation. The first approach is the most general one where the exterior orientation, the surface patches with elevations (DEM) and gray levels are all determined simultaneously. In the second approach conjugate points are determined independently from one another and without the exterior orientation parameters. This solution corresponds to the traditional method where all points are individually measured and then entered into a block adjustment.

2 GENERAL APPROACH

Fig. 1 depicts four images $I_1, \ldots, I_4$ with image patches $p_1, \ldots, p_4$ covering the surface patch S. To generalize let p be the number of image patches, size $n \times n$ pixels. Associated with every image patch p_i is a gray level function $g_i(x, y)$.We may consider the gray levels as observed.

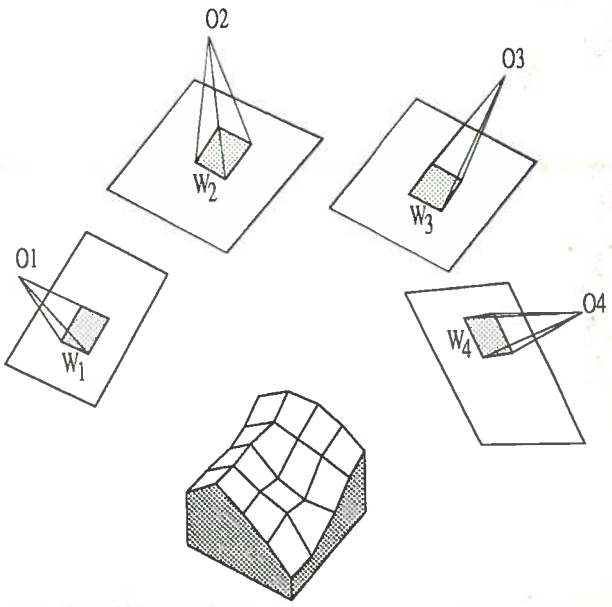


Fig. 1: Multiple image patches covering the same surface patch S.

The surface patch S is represented as a DEM with a resolution of $m \times m$ grid points, $m \ll n$. The task is now to reconstruct the surface patch S from the observed gray levels of the image patches. This involves both, geometric and radiometric reconstruction. Let $Z(X,Y)$ be the geometric function and $G(X,Y)$ the radiometric function for representing the surface patch S. Capital letters are used to better differentiate object space functions and variables from their counterparts in image space. Obviously, the image functions $g_i(x,y)$ correspond to $G(X,Y)$, the gray level distribution of the surface patch S. The discrete representation of $Z(X,Y)$ can be considered the DEM of S. The reconstruction of S involves $2m^2$ parameters (m^2 elevations, m^2 gray levels).

Since S is small we may approximate it by a Lambertian surface. The gray levels of the image patches are then directly related to the gray levels of the surface patch. Suppose the surface is flat and parallel to all image patches. In this (unrealistic) case, the gray levels of S would simply be the mean of the gray levels of all image patches.

Next, we need to define the geometrical relationship between the image patches and the surface patch. This is accomplished by a geometrical transformation $\mathbf{T^g}$. Combining the geometric and radiometric relationship leads to the following non linear observation equations:

$$\mathbf{r_i} = G(X,Y) - \mathbf{T_i^r}\left[g_i\left(\mathbf{T_i^g}(x,y)\right)\right] \qquad (1)$$

with $\mathbf{r_i}$, $i = 1,2,\ldots,p$ the residual vectors, dimension m^2, and $G(X,Y)$ the gray level function of the surface patch S. Each image patch contributes m^2 equations leading to a total of pm^2 observation equations. The parameters to be determined include the gray levels $G(X,Y)$ and the elevations $Z(X,Y)$ of the surface patch, as well as the transformation parameters $\mathbf{T_i^r}$ and $\mathbf{T_i^g}$ for every image patch.

With equation 1 the task of reconstructing S from multiple image patches is formulated as a least squares problem where the gray level differences between the image patches and the surface are minimized by varying the surface shape $Z(X,Y)$, the surface gray levels $G(X,Y)$ and the exterior orientation of the image patches. This takes the concept of matching in object space a step further to include the determination of the gray levels of the surface. Since the image patches p_i have to be resampled with the geometric transformation $\mathbf{T^g}$ (which in turn is a function of the unknown exterior orientation parameters!) the image patches must be larger $(n \times n)$ than the surface patch $(m \times m)$, hence $m \ll n$.

2.1 Geometric Transformation T^g

There are several possibilities to model the geometric transformation between the image patches p_i and the surface $Z(X,Y)$.

1. The collinearity equations are the most general transformation between surface and images. The transformation parameters comprise the exterior orientation elements of the images I_i.
2. Since the image patches are rather small the central projection may be approximated by a parallel projection. In that case the transformation parameters would include the spatial direction of the projection (3 angles), a translation of the image patch and a scale factor.
3. It is also conceivable to approximate the surface by an analytical function and determine its parameters.
4. As a further simplification of model 3 we approximate the surface S by a plane. The relationship between an image patch and S can now be expressed by a projective transformation which in turn may be approximated by an affine transformation. This would correspond to the classical case of least squares matching with shape parameters.

2.2 Radiometric Transformation T^r

Based on the assumption that S is a Lambertian surface a linear radiometric relationship of the form

$$\mathbf{T^r} = r_0 + r_1\left(g_i(x,y)\right) \qquad (2)$$

exists between the image patches. It may even be adviseable to perform the radiometric adjustment prior to the matching. Therefore, we exclude the radiometric transformation from the following considerations.

3 GEOMETRIC TRANSFORMATION MODELS

3.1 Central Projection

We linearize the observation equation 1 under the assumption that $\mathbf{T^g}$ is a central projection. Disregarding the radiometric transformation and dropping the indices i for denoting i^{th} image patch equation 1 reads

$$\mathbf{r} = G(X,Y) - g(T_x, T_y) \qquad (3)$$

where $g(T_x, T_y)$ needs to be linearized with respect to the exterior orientation parameters and Z.

$$\begin{aligned} \mathbf{r} &= G(X,Y) - g^0(T_x^0, T_y^0) - \\ &\sum_{i=1}^{7} \left(\frac{\vartheta g(T_x, T_y)}{\vartheta T_x} \frac{\vartheta T_x}{\vartheta a_i} + \right. \\ &\left. \frac{\vartheta g(T_x, T_y)}{\vartheta T_y} \frac{\vartheta T_y}{\vartheta a_i} \right) \Delta a_i \end{aligned} \qquad (4)$$

with

$$\begin{aligned} g_x &= \frac{\vartheta g(x,y)}{\vartheta T_x} \\ g_y &= \frac{\vartheta g(x,y)}{\vartheta T_y} \end{aligned} \qquad (5)$$

we obtain

$$\begin{aligned} \mathbf{r} &= G(X,Y) - g^0(T_x^0, T_y^0) - \\ &\sum_{i=1}^{7} \left(g_x \frac{\vartheta T_x}{\vartheta a_i} + g_y \frac{\vartheta T_y}{\vartheta a_i} \right) \Delta a_i \end{aligned} \qquad (6)$$

where g_x, g_y are the gradients in $x-$ and $y-$ directions, a_i the partial derivatives of the collinearity equations. The initial gray levels $g^0(T_x^0, T_y^0)$ of the image patch are determined by transforming the initial elevations $Z^0(X,Y)$ to the image with the initial collinearity equations T_x^0, T_y^0. The parameters Δa_i include the 6 exterior orientation elements and ΔZ, the unknown elevations of the surface patch. Finally, $G(X,Y)$ are the unknown gray levels of the surface.

3.2 Parallel Projection

If the exterior orientation in the collinearity model is determined from one image patch only the normal equation system will be ill-conditioned. The perspective center is only weakly determined by the small image patch since the intersecting bundle rays form very acute angles. Of course, if the same image is involved in several well distributed surface patches, the situation improves, but only if a simultaneous adjustment is performed.

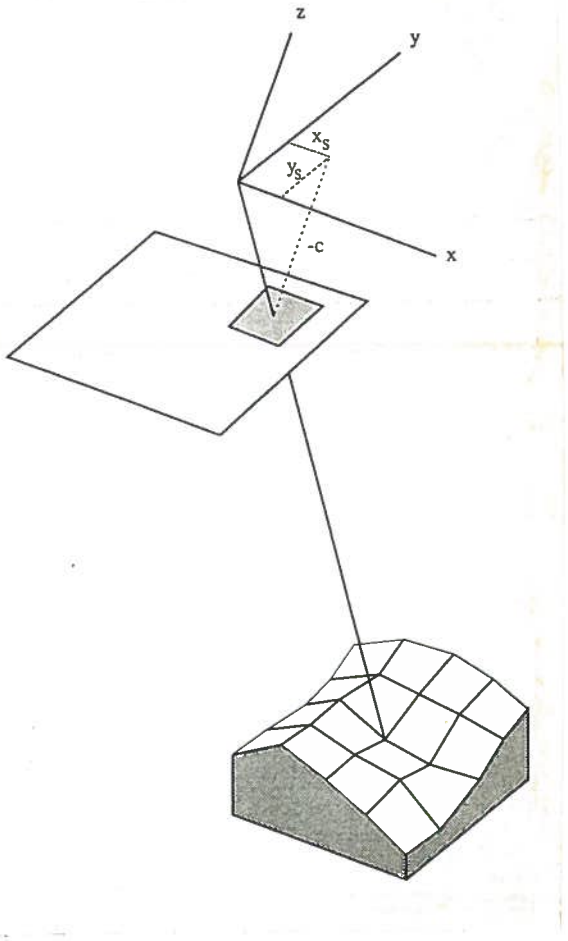


Fig. 2: Direction of parallel projection defined by bundle ray through center of surface patch S.

If the surface patches are determined individually then a parallel projection should be used instead of the central projection. The direction of the projection is defined by the bundle ray through the center of the surface patch (see Fig. 2). This direction can easily be determined in image space. Let $[x_s, y_s, -c]^T$ be the center of the image patch in the traditional photocoordinate system (origin at perspective center). The bundle ray through the center of the patch is

then defined by the following equations:

$$\begin{aligned} z &= nx \qquad & y &= mx \\ n &= \frac{c}{x_c} & m &= \frac{y_c}{x_c} \end{aligned} \tag{7}$$

The angles α, β, γ determine the spatial direction of the bundle ray that represents the image patch.

$$\begin{aligned} cos(\alpha) &= \frac{1}{\sqrt{1+m^2+n^2}} \\ cos(\beta) &= \frac{m}{\sqrt{1+m^2+n^2}} \\ cos(\gamma) &= \frac{n}{\sqrt{1+m^2+n^2}} \end{aligned} \tag{8}$$

With these three independent angles about the axis of the photo coordinate system the rotation matrix $\mathbf{R}$ is formed in the usual fashion. The corresponding rotation matrix $\mathbf{R_s}$ in the object space coordinate system is obtained by multiplying $\mathbf{R}$ with the rotation matrix from the exterior orientation $\mathbf{R_e}$

$$\mathbf{R_s} = \mathbf{R_e}\mathbf{R} \tag{9}$$

If we rotate the surface patch S by $\mathbf{R_s}$ then the projection becomes parallel to z in the photo coordinate system. Since we deal with a parallel projection the object/image space relationship is trivial, that is, $x = X', y = Y'$ where x, y are the photo coordinates and X', Y' the rotated surface coordinates.

Now we complete the geometric transformation by adding a translation vector $[x_t, y_t]^T$ and a scale factor s. This compensates for not including the perspective center in the transformation. Finally, the following transformation equations describe the parallel projection

$$\begin{aligned} x &= (r_{11}X + r_{12}Y + r_{13}Z)\,s + x_t \\ y &= (r_{21}X + r_{22}Y + r_{23}Z)\,s + y_t \end{aligned} \tag{10}$$

Linearizing the general observation equations 3 with respect to the geometric transformation of equation 10 we obtain equation 11 which corresponds to equation 6

$$\mathbf{r} = G(X,Y) - g^0(T_x^0, T_y^0) - \sum_{i=1}^{7}\left(g_x\frac{\vartheta T_x}{\vartheta a_i} + g_y\frac{\vartheta T_y}{\vartheta a_i}\right)\Delta a_i \tag{11}$$

with the partial derivatives

$$\begin{aligned} \frac{\vartheta T_x}{\vartheta \alpha} &= a_1^x = s\,[0,0,0]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_y}{\vartheta \alpha} &= a_1^y = s\,[r_{31}, r_{32}, r_{33}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_x}{\vartheta \beta} &= a_2^x = -s\cos(\gamma)\,[r_{31}, r_{32}, r_{33}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_y}{\vartheta \beta} &= a_2^y = s\sin(\gamma)\,[r_{31}, r_{32}, r_{33}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_x}{\vartheta \gamma} &= a_3^x = s\,[r_{21}, r_{22}, r_{23}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_y}{\vartheta \gamma} &= a_3^y = s\,[-r_{11}, -r_{12}, -r_{13}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_x}{\vartheta s} &= a_4^x = s\,[r_{11}, r_{12}, r_{13}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_y}{\vartheta s} &= a_4^y = [r_{21}, r_{22}, r_{23}]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ \frac{\vartheta T_x}{\vartheta x_t} &= a_5 = 1 \\ \frac{\vartheta T_y}{\vartheta y_t} &= a_6 = 1 \\ \frac{\vartheta T_x}{\vartheta Z} &= s r_{13} \\ \frac{\vartheta T_y}{\vartheta Z} &= s r_{23} \end{aligned} \tag{12}$$

where r_{ij} are the elements of the rotation matrix $\mathbf{R_s}$ in equation 9.

4 APPLICATIONS

There are several practical applications for multiple image matching. We focus on aero-

triangulation and describe briefly the simultaneous adjustment of surface patches within a block. Aerotriangulation consists of several tasks, for example point preparation, point transfer, measuring, adjustment, and analysis. Oviously, multiple image matching is related to transferring and measuring points. However, we may combine the matching process for individual points with the blockadjustment. For the following discussion we assume that good approximations for all tie points are available. How to determine good approximations automatically is a problem in its own right and is not treated here.

4.1 Automatic Aerotriangulation

The model derived in 3.1. is suitable for an automatic aerotriangulation. The parameters in the observation equations (6) comprise the exterior orientation elements of the images whose patches are involved in the reconstruction of the surface patch S, and the elevations and gray levels of S. Assuming a surface patch size of $m \times m$ grid cells that corresponds approximately to the pixel size of the images we obtain a total pm^2 observation equations for p image patches. Suppose we now move to the next surface patch and repeat the same procedure. Some of the images involved in the previous patch participate also in the new patch. We note that the images involved in both patches relate the new set of observation equations with the previous one. Obviously, adding more and more surface patches which partially share the same images is analogous to measuring points on photographs forming a block.

Suppose we have a block of three strips with four photographs per strip and regularly distributed surface patches such that on every image at least nine surface patches are visible. This would lead to approximately 40 surface patches. We further assume that the size of every surface patch is 13×13. The number of unknowns to be determined is 12×6 exterior orientation elements plus $40 \times 13 \times 13$ elevations and $40 \times 13 \times 13$ gray values for the 40 surface patches. Thus, the total number of unknowns is 13592 and the number of observation equations is $4 \times 40 \times 13 \times 13 = 27040$, assuming that a surface patch shows on four photographs.

As shown in Agouris and Schenk (1992) the structure of the normal equation matrix is such that the unknown gray values can easily be eliminated. Thus, the size of the reduced normal equations would be 6832 in our example. This is still a large system considering the small block. A reasonable alternative then is to determine the surface patches independently and to introduce them later in the aerotriangulation. In this case, the model described in 3.2. should be used.

5 REFERENCES

Ackermann, F. 1984. High Precision Digital Image Correlation. *Institute of Photogrammetry*, TU Stuttgart, Vol. 9, p. 231-243.

Agouris, P., T. Schenk 1992. Multiple Image Matching. *International Archives of Photogrammetry & Remote Sensing*, ISPRS XVII Congress, Washington, D.C..

Ebner, H., Ch. Heipke, 1988. Integration of Digital Image Matching and Object Surface Reconstruction. *International Archives of Photogrammetry & Remote Sensing*, ISPRS XVI Congress, Vol. 27, Part B 11, pp III-534 – III-545.

Förstner, W., 1984. Quality Assessment of Object Location and Point Transfer Using Digital Image Correlation Techniques. *International Archives of Photogrammetry & Remote Sensing*, ISPRS XV Congress, Vol. 25, Part A3a, pp 197-219.

Grün, A., E. Baltsavias, 1988. Geometrically Constrained Multiphoto Matching. *Photogrammetric Engineering & Remote Sensing*, Vol. 54, No. 5, pp. 633-641.

Helava, U., 1988. Object-Space Least-Squares Correlation. *Photogrammetric Engineering & Remote Sensing*, Vol. 54, No. 6, pp. 711-714.

Wrobel, B., 1988. Facets Stereo Vision (Fast Vision) - A new Approach to Computer Stereo Vision and to Digital Photogrammetry. *Proc. Intercomm. Conference of ISPRS on Fast Processing of Photogr. Data*, Interlaken, pp. 231-258.

ON MATCHING IMAGE PATCHES UNDER VARIOUS GEOMETRICAL CONSTRAINTS

Charles Toth
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission III

ABSTRACT

Image matching plays an important role in digital photogrammetry. Finding conjugate points occurs in different photogrammetric tasks. Image matching is usually performed in two steps: determining approximations and computing precise conjugate locations. In this paper we are concerned with the second aspect, that is, the image patches are already close to their final position. An image patch analyzer determines which matching primitives should be used first. Based on the results other primitives can be activated. The process terminates if a predefined level of confidence is reached or if no further improvements are to be expected. The matching process can be geometrically constrained, for example, along vertical lines, epipolar lines or by fixing one image patch in its location.

Report No. 418 July1992

1 INTRODUCTION

Image matching – finding conjugate points – plays an important role in digital photogrammetry. It is an essential operation of many basic photogrammetric procedures, like automatic orientation or DEM generation (Schenk *et al.*, 1990). Much research in digital photogrammetry has been devoted to matching, including theoretical as well as implementation issues. The results are reported in numerous publications and technical papers.

One of the first products of digital photogrammetry are digital photogrammetric workstations (or softcopy stations). They will have a great impact on how daily photogrammetric tasks are handled (Kaiser, 1991). For one, they provide the operator with all functionality of the analytical plotters. The major difference between softcopy stations and analytical plotters is the fact that the operator views the 3-D stereo model directly on the display screen. More important, softcopy stations offer an unprecedented opportunity for automation. This is the first time that digital photogrammetric methods are implemented a in real production environment. This is as encouraging as it is challenging for the research community to transfer and commercialize research results. The process of automating routine tasks has just started. Since it is a very complex problem, results will probably not meet the high expectations for quite some time (Schenk and Toth, 1992).

In this paper we report about the preliminary phase of a project with the objective to automatically keep the floating mark of a softcopy station on the ground (or dot on the ground, for short, DOG). The idea is that after a stereo model is oriented the operator is not forced to set the measuring mark (3-D cursor) precisely on the ground. If the function is evoked the system will do it automatically. To automatically place the floating mark on the surface is a problem of finding conjugate points in both image patches. In other words, when an operator slightly moves the floating mark in the XY plane, then the proposed algorithm should automatically find the corresponding Z coordinate and move the measuring mark accordingly (adjust Z). Consequently, these application-specific conditions immediately define the scope of the matching techniques feasible to solve the correspondence problem. We investigated the concurrent application of different matching methods with good localization results. In the following, conceptual issues and some major subtasks are discussed.

2 OVERVIEW OF THE PROPOSED ALGORITHM

2.1 Application Specific Conditions

A variety of different matching methods are now available, each with its specific advantages and disadvantages. In order to narrow down the set of possible techniques suitable for our project, the application characteristics must be considered:

- digital stereo model is oriented
- good approximations for conjugate points are given
- optional epipolar image geometry
- parameter adaptability
- relatively small size of the image patches

The exterior orientation is necessary to move between image and object space. It is used to constrain the movement of the floating mark. Also, it allows for employing object space matching methods (Helava, 1988).

Since the operator will keep the measuring mark fairly close to the ground, good approximations for the conjugate points can be assumed. Thus, image patches always sufficiently overlap.

As shown in (Kaiser, 1991) epipolar geometry can be easily achieved on softcopy stations, thus it is worth taking advantage of that special geometrical condition. Sometimes the images are not resampled but converted on the fly during display operation or data processing. Usually, the operator moves the floating mark quite slowly. Therefore, the image patches of a current matching operation may have very similar image characteristics like neighboring patches. This means that certain parameters of the previous (neighboring) matching operation can immediately be used for the processing

strategy of the current patches. For example, surface direction can be approximated at one side of the patch, or texture based segment data and basic statistics can reveal occlusion situations.

The patch size in our application is more an implementational than algorithmic issue. However, it is still important since most matching methods are very timeconsuming and our application needs a quasi real-time response (the processing time should not take longer than what an operator would need).

2.2 Structure of the Algorithm

Based on conditions imposed by our application the default matching method is cross correlation (Ackermann, 1984). If the current patch has enough texture information, the foreshortening is negligible and there is no occlusion or other artifact, then correlation performs well. Since these conditions are not always met, other methods must be used. A first key issue is to find and parametrize the image characteristics (called actual features in this paper) of the patches. This problem itself is as complex as the matching, because ideally it would address many high-level paradigms of scene analysis and image interpretation. Because of the lack of a robust, scene independent matching scheme, an iterative hierarchical strategy is proposed. Figure 1 shows the flowchart of the proposed DOG method. The suggested system has five processing levels:

- Patch preprocessing
- Feature extraction - image patch analysis
- Matching procedure
- Matching strategy
- Evaluation - result analysis

The patches may be subject to some image enhancement. Scaling the pixel intensities or histogram equalization can compensate for bad contrast. Spatial filtering, like median or Gauss operators can remove noise or unneccessary details which may be important for scale-space algorithms.

Feature extraction provides clues about the patch and may guide the selection of the most appropriate matching method. The basic statistical properties, like mean, median, minimum and maximum intensities, standard deviation, auto-correlation, etc. provide additional patch information.

Local features are grouped into three classes:

- Interest points can produce target matching locations (Zong *et al.*, 1991).
- Edges can be matched (Schenk *et al.*, 1991). Although their localization is weak, the matched edges are relatively reliable. Therefore they may render good approximations for other methods with precise localization properties.
- Texture values and texture based segmentation are useful for detecting shadows, water bodies, and the like. Matched segments provide good global matching constraints.

In the context of our DOG project, matching procedure refers to the matching method in general. We use four methods:

- Cross correlation
- Least squares matching
- $\Psi - S$ feature based matching
- Symbolic matching

Cross correlation and least squares matching methods are discussed in the next sections. The $\Psi - S$ feature-based matching is a very reliable technique to obtain numerous matching points with fair localization accuracy. In our implementation a scalable LoG operator generates the edges. The edges are sorted and transformed into the $\Psi - S$ domain where a global matching takes place. Symbolic matching is very useful for global matching (Zilberstein, 1991). Due to implementation issues its use in our project is limited to the 1-D case. We presently use it in profile matching.

Matching strategy refers to constraints how the extremes are sought. Based on the original DOG objective – automatically adjust Z (find conjugate points, compute Z coordinate and drive the floating mark) – the conjugate location is confined to the vertical line which translates into two lines in the image planes. Since

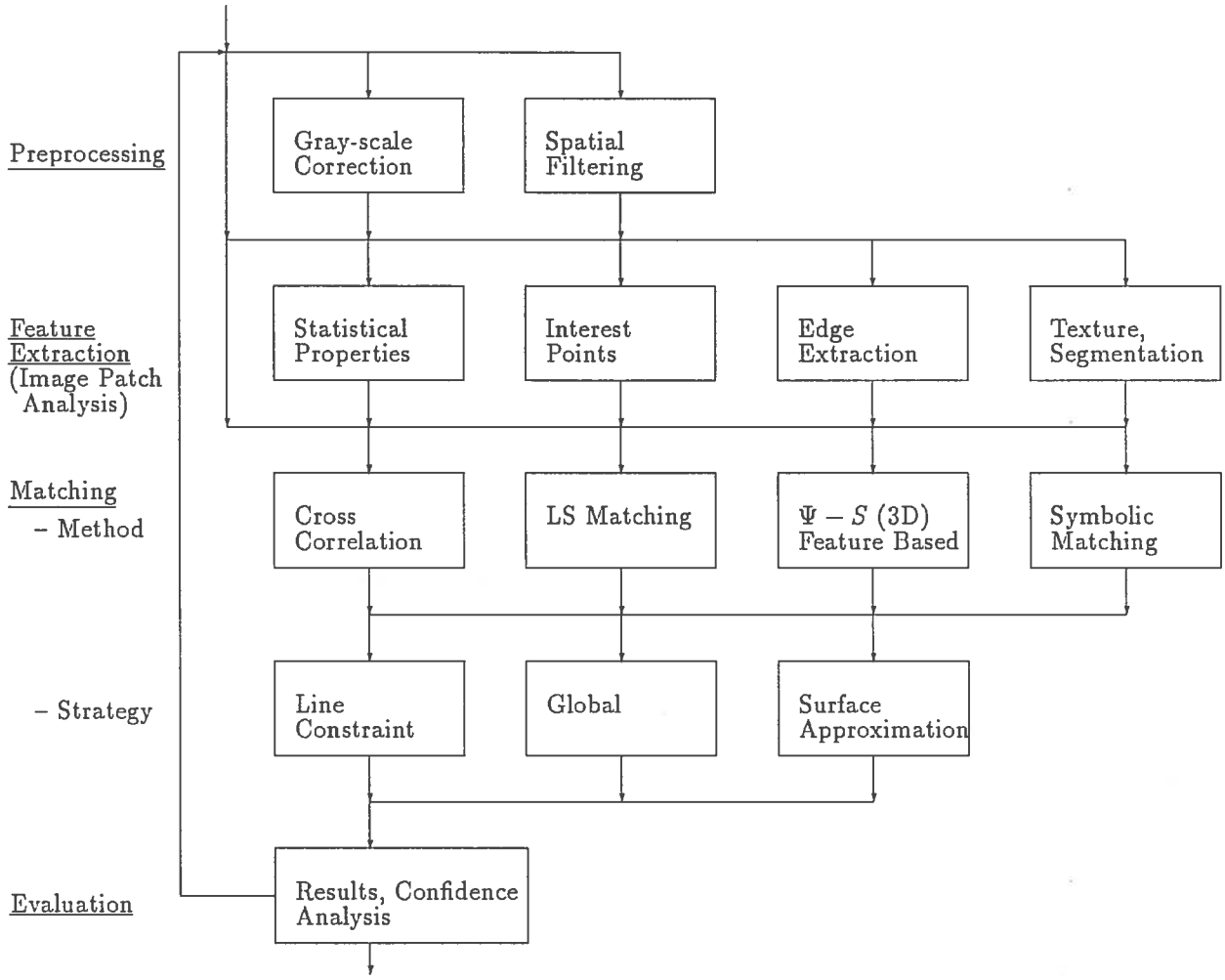


Figure 1: Flowchart of the DOG system

occlusions may block out certain segments of the constraint line, global methods are superior to cope with this case. Although the terrain is modelled in least squares matching by the shape parameters, a significantly better approach is to use a priori surface approximations (Schenk *et al.*, 1990). Based on the surface data the patches are warped and in this format, basically free from terrain relief distortion, normal cross correlation is used. Since the surface data are quite sparse it is very critical how the surface interpolation algorithm performs.

The most crucial component of our DOG project is the evaluation and performance analysis of the results. The evaluation module serves as a system controller to implement a data driven algorithm. Typical operation tasks are:

1. The two patches directly go to the statistical module (the preprocessing is skipped unless specified by the user).

2. The results of the image patch analysis are compared to data obtained from the neighboring patches; if significant differences are found then the other three feature extraction modules are activated, otherwise the same sequence of module processing is executed used for the neighboring patches (it may also include other feature extractions) .

3. After executing a matching function (a defined sequence of module operations)

the matching results are compared to the results of the neighboring patches and to predefined global parameter values. If everything is all right the new Z value is computed and the process terminates.

4. If in (3) the results are not satisfactory then based on built in rules the matching sequence – in whole or part – is modified and a new computation starts; this is repeated until a satisfactory solution is found; if all strategies are exhausted, the process terminates with a message that matching is impossible.

5. If in (2) the comparison fails, the system controller assigns the initial matching strategy according to the results of the three feature extraction modules, then point (4) is executed.

6. Upon termination, the parameter set and the last processing sequence are saved for the next application of the DOG function.

In summary, the system controller can be considered as a data driven, self-organizing, adaptive system which finds the optimal matching strategy to any data input from a list of prestored computation sequences. The performance of module computations is measured in the usual parameters, like the absolute and relative value of the correlation coefficient in cross correlation, or residuals and variances in least squares matching, or the relative number of matched edges in $\Psi - S$ feature based matching, etc. The system controller can be viewed as two tables, one containing matching sequences and the other consisting of rules on how to evaluate the results. Tables are extended to include experimental results, thus the system can learn. Under normal conditions the neighboring image patches are similar enough and the search process for optimal strategy is called only where there are significant scene changes in the images.

2.3 Cross Correlation

Cross correlation matching can be used in all three geometrical constraint strategies. Less important details, such as window size, which may be determined from statistical properties, are omitted, and the typical epipolar condition (correlation window becomes a line) is assumed in the following description.

Line Constrained Search

The search line can be easily determined from the XY coordinates of the floating mark, and from the assumption that the current Z value is quite close to the real surface value. Thus the Z coordinate of the possible P surface point should be in the range:

$$\begin{aligned} Z_p^u &= Z_c + dZ \\ Z_p^d &= Z_c - dZ \end{aligned}$$

where Z_c is the current elevation of the floating mark and dZ is the search range defined by global constraints. The two extreme points, Z_p^u and Z_p^d are projected to both left and right photo planes by collinearity equations and then transformed into image coordinates, yielding the two search lines. As a next step, the correlation windows are moved along the search lines and the coefficients are computed. Under ideal conditions there is only one pair of conjugate points pointing to the desired surface location, and for this point the correlation coefficient is likely to have a maximum value. The analysis of the correlation curve around the maximum gives some indication about the reliability of the maximum. Using a band of parallel search lines can further improve the reliability of the results. In this case a correlation ridge is obtained, and its analysis can lead to more reliable results. In general, independent global matching should be applied to increase the confidence level of the matching results.

Global Search

In this case the correlation window is moved within the entire patch area, and a 2-D correlation function is computed. The shape of the correlation function may help to confirm or drop our hypothesis about a location, although it is not necessarily feasible to determine directly the desired location.

Surface Warping

If true surface data are available, then the distortion of the image patches caused by terrain relief can be totally removed. In this case, cross correlation is concerned only with texture information, and reliable results are obtained. Surface data may be known for the

previous patches, but in general the true surface is never available, and therefore it must be approximated. Based on our experience (Schenk *et al.*, 1990), with $\Psi - S$ feature based matching, enough reliable surface points are obtained. Nevertheless, the surface data are still quite sparse and a surface interpolation algorithm must be used. Through hierarchical iterations the surface approximation usually improves. It is appreciated that in this process the interpolation algorithm itself plays an important role (Al-Tahir, 1992). A bad approximation strategy can slow down the convergence or even make the surface diverge.

2.4 Least Squares Matching

In general, least squares matching can be similarly constrained as cross correlation. The general approach is:

$$g_2(x,y) = h_0 + h_1 \cdot g_1(a_0 + a_1 x + a_2 y, b_0 + b_1 y + b_2 x) \quad (1)$$

Eq. 1. can be simplified if epipolar geometry exits:

$$g_2(x) = h_0 + h_1 \cdot g_1(a_0 + a_1 x) \quad (2)$$

The surface data can be used to set better initial values for the adjustment procedure.

3 EXPERIMENTS, RESULTS

A prototype version of the proposed solution in the DOG project has been implemented on Intergraph workstations. Most of the modules are operational. As of writing this paper extensive tests have been performed, and the system controller tables have been built.

The first observations are:

- the major modules perform as expected
- the basic cross correlation matching works well for reasonable test data (for example, with fixed X and Y increments it automatically collects DEM grid points)
- the automatic parameter tuning is difficult and needs good initial values
- the deterministic approach of the system controller is not optimal
- it is quite complex to define the rules

In summary, the preliminary test results are encouraging. Theoretical investigations are needed to analyze the results and to parametrization the confidence level. On the implementation side, the growing number of rules and module sequences justifies the use of an off-the-shelf expert system (Schenk and Toth, 1991). The accuracy tests will include large numbers of varying image data, and more independent operators are needed to provide the ground truth for performance evaluation.

4 REFERENCES

Ackermann, F., 1984. "Digital Image Correlation: Performance and Potential Application in Photogrammetry," *Photogrammetric Record*, Vol. 11, No. 64, pp. 429-439.

Al-Tahir, R., 1992. "On the Interpolation Problem of Automated Surface Reconstruction," *International Archives of Photogrammetry and Remote Sensing*, Vol. XXIX.

Helava, U.V., 1988. "Object-Space Least-Squares Correlation," *Photogrammetric Engineering and Remote Sensing*, Vol. 54, No. 6, pp. 711-714.

Kaiser, R., 1991. "ImageStation: Intergraph's Digital Photogrammetric Workstation," Digital Photogrammetric Systems, Wichmann, pp. 188–197.

Schenk, T., Li, J-C., and Toth, Ch., 1990. "Hierarchical Approach to Reconstruct Surfaces by Using Iteratively Rectified Images," *Proc. Int. Soc. of Photogr. and Remote Sensing ISPRS*, Symp. Comm. V, vol. 28, part 5/1, pp. 464–470.

Schenk, T., Li, J.C., and Toth, Ch., 1991. "Towards an Autonomous System for Orienting Digital Stereopairs," *Photogrammetric Engineering and Remote Sensing*, vol. 57, no. 8, pp.1057–1064.

Schenk, T., and Toth, Ch., 1991. "Knowledge-Based Systems for Digital Photogrammetric Workstations," Digital Photogrammetric Systems, Wichmann, pp. 123–134.

Schenk, T., and Toth, Ch., 1992. "Conceptual Issues of Softcopy Photogrammetric Workstations," *Photogrammetric Engineering and Remote Sensing*, Vol. 58, No. 1, pp. 101–110.

Zong, J., Schenk, T., and Li, J-C., 1991. "Application of Förstner Interest Operator in Automatic Orientation System," *Proc. ASPRS-ACSM Annual Convention*, Vol.5, pp. 440–448.

Zilberstein, O., 1991. "Solving the Correspondence Problem in Aerial Imagery Using Relational Matching," Phd dissertation, Dept. of Geodetic Science and Surveying, The Ohio State University, Columbus, OH.

A GIS WORKSTATION-BASED ANALYTICAL PLOTTER

Charles Toth
Toni Schenk

Department of Geodetic Science and Surveying
The Ohio State University, Columbus, Ohio 43210-1247
USA

Commission II

ABSTRACT

Today's high performance analytical plotters are typical stand-alone, single-user systems. Though they serve as intelligent 3-D data acquisition stations to geographical information systems, the link between them is usually limited to data transfer. Because their hardware and software platforms are different, no sophisiticated interactions with GIS workstations on the system level take place. In the future we expect a much closer integration of analytical plotters into GIS systems. This entails sharing similar design concepts, data structures, user interfaces, etc. The integration is greatly facilitated by the separation of the real-time loop processes from application software as well as by the rapidly decreasing cost of high-performance UNIX workstations. In this paper we describe the design concepts and the development of basic analytical plotter software, such as orientation, closely integrated on Intergraph's GIS workstations. A generic interface supports different analytical plotters without the need for hardware or software modifications.

1 INTRODUCTION

The introduction of the *plate processor* was a major milestone in the continuous development of the analytical plotters. Dedicating a special processor to instrument specific tasks significantly improves plotter operation and dramatically decreases the involvement of the host computer, thus freeing up computer power for the applications. Prior to this development the host computers had to run the mapping application software and at the same time maintain the plotter operation *real-time loop*. This inefficient structure required special interfaces and was demanding on the host (e.g., complex software). Separating the real-time loop from the application program brought new flexibility and opportunity to the design. Most important, the mutual dependency of the plotter hardware and the host computer disappeared. In the past the host computer was an integral part of the analytical plotter and so was the proprietary application software. Customers had little choice in using third party mapping programs. With the new design, the situation started to change. First, alternative hosts and an increased software selection were offered. Finally, analytical plotters became peripheral devices. Modern plotters can easily connect to any computer platform.

The never stopping, continuously advancing computer technology is the main driving force for development in many, quite different, application areas. It was the case with the plate processor of the analytical plotters. Currently, GIS technology is getting a big boost. On a theoretical level, the GIS is a database with spatial context. The compact definition can be detailed if technical aspects of the implementation are considered. The hardware components are off-the-shelf products: high capacity central mass storage with servers, fast local and wide-area networking and powerful user workstations. The software picture is more complex, including standard packages like networking, database management, 2- or 3-D graphic packages, etc. What makes a real GIS different from a big database system is the customization. The layered database is organized on the spatial – geographical – relation providing application optimal data access. A key issue is the user interface and data exchange, mainly importing data into the GIS. We believe that on the data input side analytical plotters should be considered as a main source among others. In this context, the analytical plotter is used as an intelligent front-end data acquisition system connected to an information processing system. It is important to note that plotters not only send but also receive data. The core information system can host not only GIS but other applications like CAD engineering. We think the key point of the above-outlined development path is the common user interface. The software engineering is undoubtedly the focus of this decade. Many signs of major standardization have already appeared, like windows systems with the same look and feel, networking protocols, etc. This standardization should reach the applications. The user working in a GIS should be able to use an analytical plotter with minimum extra effort. This can be easily achieved by using the same computer hardware platform – workstation – and providing an identical user interface for the plotter program. This way the need for learning the specifics of the analytical plotter is minimized.

The turning point in the evolution of analytical plotters was in 1987, when Carl Zeiss, Inc. introduced the P-Series Planicomp analytical plotters (Hobbie, 1987). All three members of the P-Series family had built-in plate processors with additional coprocessors to support real-time computations. The first systems comprised an HP 1000 workstation and the PHOCUS application software. These pioneering instruments had an IEEE-488 (HP-IB) interface for host communication which later very dramatically influenced the future of the instrument. Back then workstation and PC prices differed a lot. It was clear from the beginning that a low cost PC-based configuration should be developed. In 1987, Carl Zeiss Inc. donated a full fledged P1 instrument to the Department of Geodetic Science and Surveying at The Ohio State University. As a result of the good cooperation, a PC-based stereo orientation program, called PCAP was developed by the authors. The program was based on, at that time yet completely unknown, user interface, Microsoft Windows (Zilberstein *et al.*, 1989). Later DEM (Toth and Schenk, 1990, 1992) and aerotriangulation (Toth and Schenk, 1991) data acquisition application programs

were added. The wide selection of the mapping products offered by many vendors also contributed to the opening of a new market niche for the PC-based P3 analytical plotters.

2 FUTURE OUTLOOK

The introduction provided a short analysis about trends in the development of the analytical plotters. In this section some current development issues are addressed.

Hardware Changes

Probably the most important interface standardization of analytical plotters is the general acceptance of the serial line as a link to the host computer. For example, Zeiss P-Series instruments offer an alternative serial interface. The Leica SD 2000 plotter interface is exclusively based on the serial line. The SD 2000 real-time processor, unlike the Zeiss plate processor, is accessible by the user. Most of the planned upgrades for older analytical plotters will also use the serial line as the communication interface. The motivation for the serial line interface is simply because it is the only interface which is available on almost any computer platform. While a PC can easily be interfaced to IEEE-488 or other standards, this is not true for UNIX workstations (thereafter called workstation), where the serial line is basically the only economical option.

The internal operation of the analytical plotters is based on model and photo coordinate systems. Currently most of the driver programs, which connect the plotter to the application programs, have to transform the input data from model to ground coordinates on the host. With the increased processing capabilities of plate processors, this computation can take place on the plotter. On newer systems it is very likely that the absolute orientation can be downloaded to the plate processor.

The recently introduced softcopy workstations (Kaiser, 1991) create a very good alternative to the analytical plotters. We think the new systems will not compete directly with analytical plotters but instead expand the market (Schenk and Toth, 1992b).

Software Issues

Prices for workstations have dropped considerably. Entry level workstations and fully loaded PCs are now in the same price range. Many applications justify the use of the more powerful workstation for performance reasons. Workstations are expected to increase their market share, but PCs will dominate for quite some time.

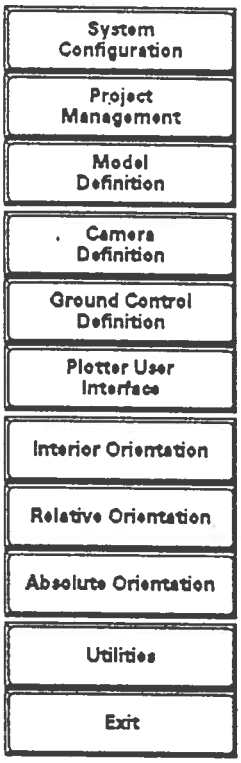


Figure 1: Control Bar

Much less but more expensive application software is available on workstations, compared to PCs, because the market is smaller and the hardware is quite different. This is the main reason why system developers try to work with platform independent software environment. Concerning the key component of the application programs – the graphical user interface – the above-mentioned demand led to the wide acceptance of the X Window System. This standardization is very likely in the mapping applications. A good illustration of this process is that the Intergraph Corporation, which is a one person workstation hard-

ware manufacturer and software vendor and has its own windowing system, has started to offer its products on the X11 platform. Concerning the applications, the X11 protocol is very generic and therefore toolkits have been developed and used to make program development easier and faster. Using the same toolkit for different applications results in a similar feel and look which is very good for the user. Based on recent surveys, the OSF/Motif environment is likely to dominate.

The standardization of the analytical plotter hardware interface is not likely to be followed by a similar move in the interface software protocol because such a standardization is probably not in the interest of plotter manufacturers. Luckily, this does not preclude the use of the same orientation or mapping application program on different analytical plotters. This trend offers the user increased flexibility.

3 INTERPRO ANALYTICAL STEREOPLOTTER INTERFACE

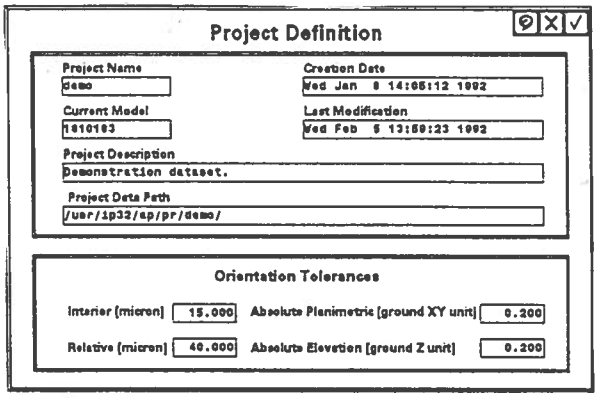


Figure 2: Project Definition Form

Several reasons led to the development of an Interpro workstation based orientation program:

- Our analytical and digital photogrammetry laboratories have gone through major changes during the last few years. Currently, we have more workstations than PCs. All the major workstation brands are represented, although almost half of them are from Intergraph Corporation.
- We do most of our research on Intergraph workstations and have acqiured the knowledge to develop application programs.
- For various reasons we needed to integrate our Zeiss P1 analytical plotter into the Intergraph MicroStation environment.
- We have had good experiences with the PC-based PCAP program.

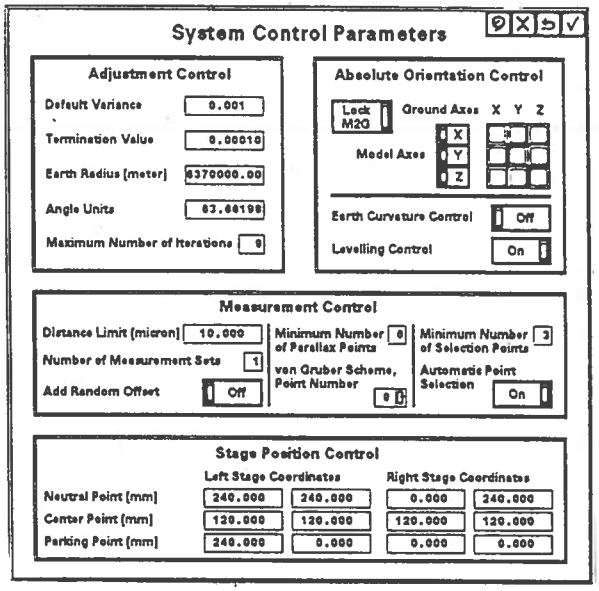


Figure 3: System Control Definition

A major design objective was to use the Intergraph I-form-based windows system to provide the user with the well-known look and feel of other Intergraph mapping applications. Another objective was to avoid unnecessary hardware dependency on the analytical plotter interface because we plan to connect our workstations to other analytical plotters. In the following, the major building blocks of the program are described. For more details, interested the reader is referred to (Stereoplotter Interface/Analytical (SPI/A), 1992).

Stereoplotter Interface/Analytical (SPI/A) Overview

The SPI/A software provides the user with the capability to measure points and to perform the interior, relative and absolute orientation of a stereopair on the Zeiss Planicomp P-Series analytical plotters. In addition, SPI/A provides forms for data entry and editing. The

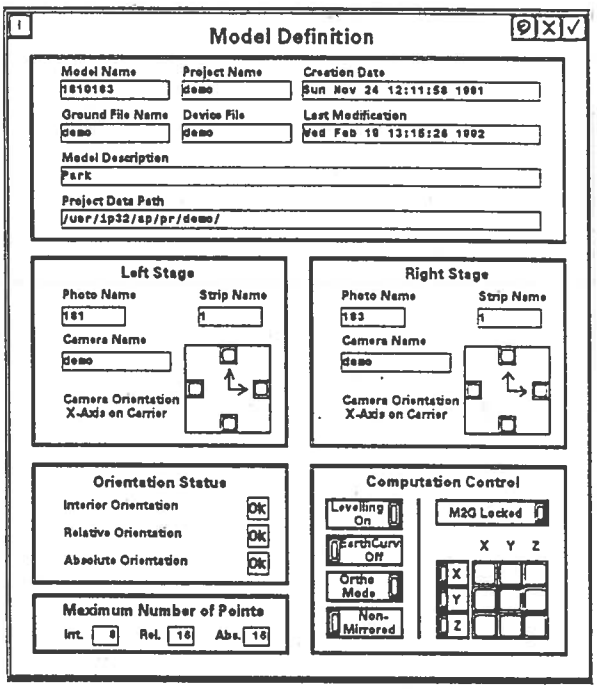


Figure 4: Model definition form

major functions include:

- Input and edit system configuration, project, model, camera, control point, and analytical plotter control data through forms.
- Perform interior, relative, and absolute orientation.
- Generate ASCII protocol files of project, model, camera, control point, and analytical plotter data.
- Generate detailed orientation reports.
- Generate ASCII files of measured points in various formats for later use in other applications.
- Dynamically display carrier, photo, model, or ground coordinates.
- Drive to points in the model.
- Create a standard orientation data interchange file to accept orientation results generated on systems other than SPI/A, or to pass SPI/A orientation data to other systems.
- Driver to connect the analytical plotter to the MicroStation.

SPI/A Workflow

Activating SPI/A, a control bar is displayed. The button functions of the vertical bar reflect the typical workflow in a top-down fashion. The control bar shown in Fig. 1. has four

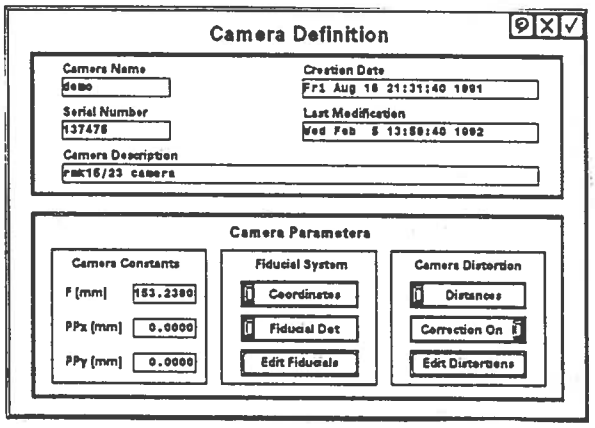


Figure 5: Camera Definition Form

groups:

- System, project, and model data management functions.
- Camera, control point, and plotter definition functions.
- Interior, relative, and absolute orientation functions.
- Other utility functions.

SPI/A Data Management

SPI/A has its own data management system. The orientation data is organized into projects.

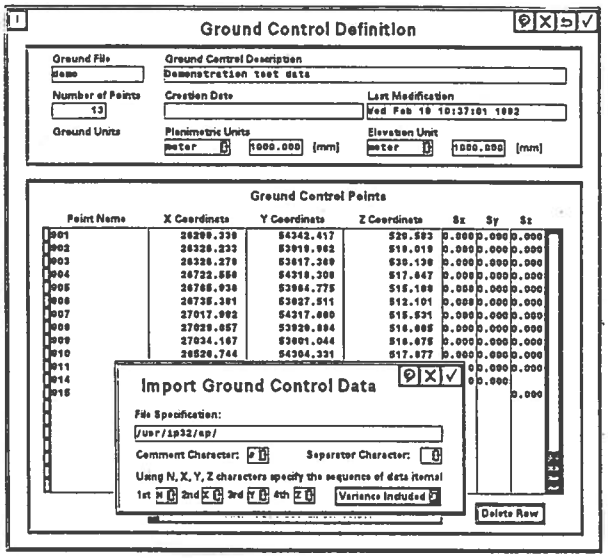


Figure 6: Import function

A project usually contains the stereo model of a flight mission. The project definition form (Fig. 2.) contains only very generic parameters. The most important part of the project definition is the so-called system control definition (Fig. 3.). It sets parameters for adjustment procedures and defines the measuring style. Finally, specific information of a stereo model is entered in the model definition file. The form (Fig. 4.) has a generic definition field to define the control point file and other verbal information, two identical fields for the two photographs, and two fields for the orientation and adjustment status.

SPI/A Definition Files

Definition files are organized independently from the projects since they are most likely used in other projects. The camera form (Fig. 5.) is used to enter data from the camera protocol. Different subforms are used for fiducial coordinate or distance entries and for camera distortion data. The ground control form is used to edit control point information. An import function lets the user transfer control data from ASCII files (Fig. 6.). Plotter definition files are used to customize the analytical plotter. The operator can freely configure the input devices and set automatic recording parameters.

SPI/A Orientations

The operation of the three orientations is based on the well-accepted concept introduced in the Zeiss P-Series PC orientation program (Schenk and Toth, 1989). The adjustment computa-

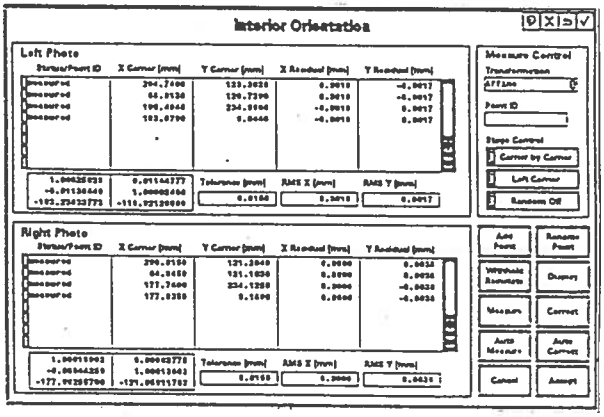


Figure 7: Interior orientation

tions automatically take place whenever there is any change in the measurement data. As a result, the operator's task is reduced to measuring the points and to analyzing the results. On the forms, the residuals and other transformation parameters always reflect the most recent adjustment results.

A very useful feature of the system is the "drive to" support. That is, the system drives automatically to points to be measured. For example, in the interior orientation the system drives to all fiducial marks. With every measurement the predicted positions become more accurate. Another example is the relative orientation, where the system drives to the von Gruber locations. In order to let the absolute orientation drive automatically to control points at least two XY control points must be measured (either in the relative or in the absolute orientation).

The functionality of the orientations can be seen in the forms shown in Fig. 7-8. Whenever the instrument is in point recording status, a small form is displayed which shows

the actual P-Cursor button assignment and the stereo model coordinates.

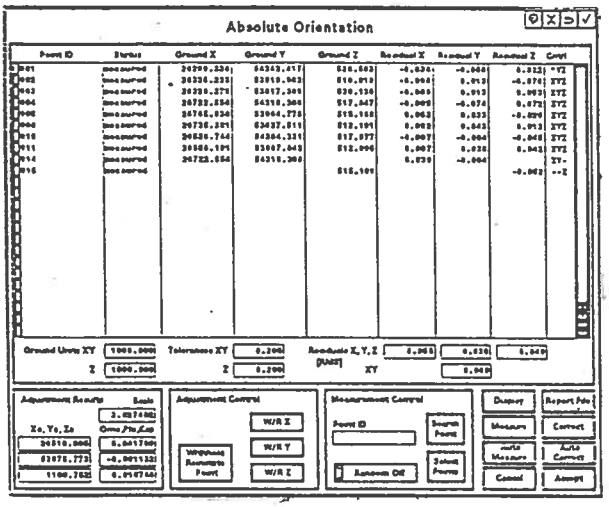


Figure 8: Absolute orientation

MicroStation Driver (SPI/D)

The SPI/D program can be loaded as an application when the MicroStation is invoked. During initialization, the program automatically downloads the current orientation into the analytical plotter and configures the input devices for MicroStation operation. In the preferred data acquisition mode – tracking mode – the position of the measuring mark drives the MicroStation mouse, and three of the P-Cursor buttons are programmed to the corresponding workstation mouse functions. Some small forms inform the user about the current operating mode, floating mark location, etc. Fig. 9. shows a map of The Ohio State University campus digitized by the Feature Collection module of the MicroStation.

4 DISCUSSION

The SPI/A and SPI/D programs have been tested in our laboratory. Our operators who are very experienced in using the PC-based orientation program found no reasonable difference in the user interface although the windows systems are quite different. Compared to PCAP, a noticeable performance difference was

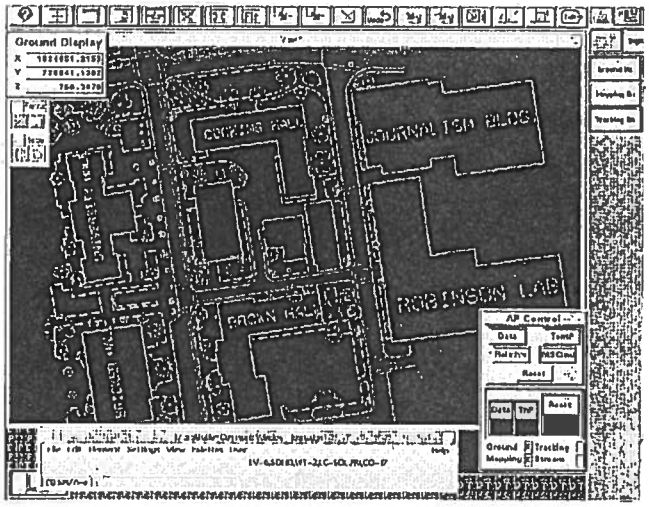


Figure 9: The Ohio State University campus

found mostly as a result of faster workstation operation. We also greatly benefit in our research (Schenk and Toth, 1992a). For example, the output data of the automatic surface reconstruction projects can be directly matched to the ground truth by driving the measuring mark to the locations and visually checking whether the point is on the surface or somewhere else.

We strongly believe that the development of the SPI/A program is an important step in the evolution toward the fully integrated geographical information systems. In the future we plan to port it to the X Window System and OSF/Motif environment.

5 REFERENCES

Hobbie, D., 1987. "Introduction Into the New Product Generation from Zeiss: P-Series Planicomp/Phocus: Vorträge 41." *Photogrammetrische Woche Stuttgart, Heft 12,* pp. 21–24.

Kaiser, R., 1991. "ImageStation: Intergraph's Digital Photogrammetric Workstation," Digital Photogrammetric Systems, Wichmann, pp. 188–197.

Schenk, T., and Toth, Ch., 1992a. "An Automatic System for DEM Data Collection," *Proc. ASPRS-ACSM Annual Convention*, Vol. 2, pp. 215–226.

Schenk, T., and Toth, Ch., 1992b. "Conceptual Issues of Softcopy Photogrammetric Workstations," *Photogrammetric Engineering and Remote Sensing*, Vol. 58, No. 1, pp. 101–110.

Schenk, T., and Toth, Ch., 1989. "A PC-Based Version of the Planicomp Analytical Plotter," *Proc. ASPRS-ACSM Annual Convention*, vol. 1, pp. 10–18.

"Stereoplotter Interface/Analytical (SPI/A)," *User's Guide*, 1992. Intergraph Corporation, DJA852910.

Toth, Ch., and Schenk, T., 1992. "A Map-Referenced Data Base System for Real-Time DEM Data Acquisition," *Proc. ASPRS-ACSM Annual Convention*, Vol. 2, pp. 230–236.

Toth, Ch., and Schenk, T., 1991. "Data Acquisition for Aerotriangulation on the Zeiss P-Series Analytical Plotters," *Proc. ASPRS-ACSM Annual Convention*, Vol. 5, pp. 353–360.

Toth, Ch., and Schenk, T., 1990. "A New Approach for DEM Measurement with the ZEISS P-Series Analytical Plotters," *Proc. ACSM/ASPRS Fall Convention*, pp. B-145—B-151.

Zilberstein, O., Greenfeld, J., Schenk, T., and Toth, Ch., 1989. "A Modern Man-Machine Interface for the Planicomp Analytical Plotter," *Proc. ASPRS-ACSM Annual Convention*, vol. 1, pp. 1–9.