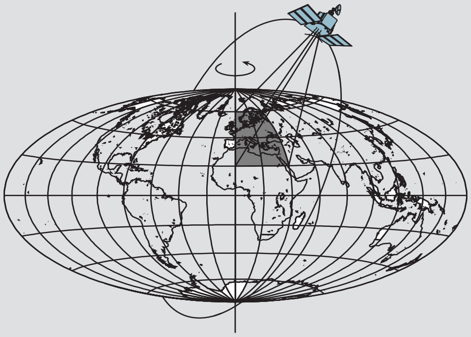
Automatic Recognition and Location of Civil Infrastructure Objects Using Mobile Mapping Technology, Neural Network and Markov Chain Monte Carlo

by

Ron Li
Zhuowen Tu



Report No. 457

Geodetic Science and Surveying
Department of Civil and Environmental Engineering and Geodetic Science
The Ohio State University
Columbus, Ohio 43210-1275

January 2000

Automatic Recognition and Location of Civil Infrastructure Objects Using Mobile Mapping Technology, Neural Network and Markov Chain Monte Carlo

Project Report
(November 1998 – December 1999)

Submitted to
The OSU Center For Mapping

By

Principal Investigator: Dr. Ron Li
Researchers: Z.W. Tu

Automatic Recognition and Location of Civil Infrastructure Objects Using Mobile Mapping Technology, Neural Network and Markov Chain Monte Carlo

Automatic Recognition and Location of Civil Infrastructure Objects Using Mobile Mapping Technology, Neural Network and Markov Chain Monte Carlo

1 Introduction

Information technology is increasingly used to support civil infrastructure systems that are large complex heterogeneous, distributed, dynamic systems including communication systems, roads, bridges, traffic control facilities, and distribution of water, gas and electricity. One of the most important data sources for such systems is updated spatial locations, physical conditions, and other attributes of infrastructure objects. The new technology of mobile mapping systems integrates GPS receivers, INS (Inertial Navigation System), and stereo CCD (Couple Charged Device) cameras on a mobile platform, such as a van, for rapid high quality spatial data acquisition. Infrastructure objects appearing in georeferenced mobile mapping image sequences can be measured on computer screen and their 3-D ground locations are calculated from measured 2-D image coordinates using a photogrammetric model.

The mobile mapping technology has been explored in Li (1997) and Tao (1997) and has been used in industrial to obtain spatial information about features in a much faster and easier way than traditional methods. The capture of stereo image sequences with georeference data is performed in a quite automatic fashion, the measurement of objects, however, is still far away to be full automation. This is because the 3D object recognition in intensity images, which has been studied in literature for many years, is yet unsolved and there are still many related researches going on.

In this article, a framework of 3D object recognition system is proposed and some existing 3D-Object recognition systems are discussed. We found out that most existing object recognition systems fit this framework. Under this framework, a new system using Multilayer Hopfield Neural Networks is proposed followed by our observation that this structure is a special case in Gibbs model that recognizes objects in stochastic relaxation. A novel system that integrates top-down and bottom-up methods by MCMC (Markov Chain Mote Carlo) to recognize traffic lights in color images is then developed.

2 Object Recognition Framework and a literature review

There are many 3D object recognition systems existing in literature already. We may characterize such systems with five aspects as the following:

1. Scene acquisition

There are many different kinds of sensors, e.g. acoustic, radar, laser, machine vision, available and most of them fall into two categories, active sensor or passive sensor. Active sensor like laser gives us depth information from the time interval between a sensor sends out a laser and it receives the bounded one. Passive sensor like CCD camera just records intensity value that objects show at every position.

2. Model acquisition

If very limited number and types of objects are going to be recognized in a system, we generally have the assumption that these types of objects are available. However, if we want to detect many different types of objects as to allow our system to learn how to recognize innovative objects then same acquisitions as scene acquisition are available.

3. Scene representation

The representations of scene are different in different systems. Range data gives us 3D coordinates of the world that an active sensor lives in and intensity image data gives us illuminant of the world that a passive sensor lives with.

4. Model representation

There are many different types of model representations that lead to either object-centered or view-centered approach, which is a long-term debate in literature. We will give a detailed discussion in the later section.

5. Matching strategy

In McCane, three 3 predominant matching approach are proposed by the author as:

- Graph matching approaches
- Feature indexing / hash tables
- Evidence / rule based approaches

The method we are proposing here tries to combine them all together as solve a MAP (maxim a posterior) problem.

The final goal of any object recognition system is to interpret every object that stands in the data set that sensors acquire. It's still a long way to go to finally reach this point. Since methods in scene acquisition and model acquisition are quite traditional and scene representation is determined by what kind of sensor is used, let's focus on model representation and matching strategy that tell apart different systems and control the quality of each ORS (object recognition system). We will discuss mainly on the recognition of objects in intensity image because the data captured by MMS (mobile mapping system) is color image sequences which are typical intensity images with georefenced information. Actually, most available systems in literature are based on intensity images as well.

We are trying to simulate the way that human being are using in interpreting the scene where they live in. The stereo system that human being are using runs very fast and accurate to some extent to allow people survive in struggling in environment. The question "How are 3D objects represented in human visual system?" (Bulthoff et al. 1994) is the major problem we should ask in the visual system. Different answers to this question lead to different model representations and thus lead to different approaches. There are two possible answers to this question: viewpoint invariant and viewpoint dependent, which yield object-centered and view-centered approaches respectively. Viewpoint invariant answer says that people actually store in brains with viewpoint invariant properties, which could be used to match with invariant properties extracted out from 2D image. Bergevin and Levine (1993)**,** Clemens (1991), Jacobs (1992), Lamdan et al. (1990), Lin et al. (1991), McCane (1996), Nagao and Grimson (1997), Shufelt (1996), Slater and Healey (1996), Slater and Healey (1997) and Wong et al. (1989) all tried to capture invariant information from 2D image and use them to match 3D object. In this approach a list of invariant properties, either photometric or geometric, are extracted to match those rooted in 3D object. Korn and Dyer (1987), Pontil and Verri (1998), Seibert and Waxman (1992) and Ullman and Basri (1991) instead use multiple views of 3D object following view-centered theory to match 3D objects. Template matching is an old and well-known technology that could be used in view-centered approach but it's impossible to compare 2D image with infinite number of views of object using simple template matching. Dickinson et al. (1991) gave a very nice framework of how to recognize objects through multiviews. In Bulthoff et al. (1994) the authors made a very good point saying that if an object-centered reference frame can recover object independently of its pose, then neither recognition time nor accuracy should be related to the viewpoint of the observer with respect to the object. If instead model is represented as viewpoint depend as long as complexity scales with normalization and transformation both recognition time and accuracy should be systematically related to the viewpoint of the sensor with respect to the object. The author also made the conclusions from psychophysical and computational studies that human encodes 3D objects as 2D multiple viewpoint representations and subordinate-level recognition is achieved by employing a time-consuming normalization process to match objects seen in unfamiliar view points to familiar stored viewpoints. Poggio and Edelman proposed a network that learns how to recognize objects from sets of 2D view through regularization network using this idea. Because the matching of 3D invariant properties between a 3D model and 2D scene is faster than the matching between number of 2D images of a 3D model viewed at different poses and 2D scene. We argue here that although view-centered approach is the final way human uses, 3D invariant properties are still used to guide visual system to tell how likely a model will be given a 2D scene.

Dickinson et al. (1991) proposed a model representation hierarchy that separates 3D models into finite number of primitives that are further decomposed into aspects, faces etc. We here expand this hierarchy into a more general framework that will be consistent with most existing ORS.

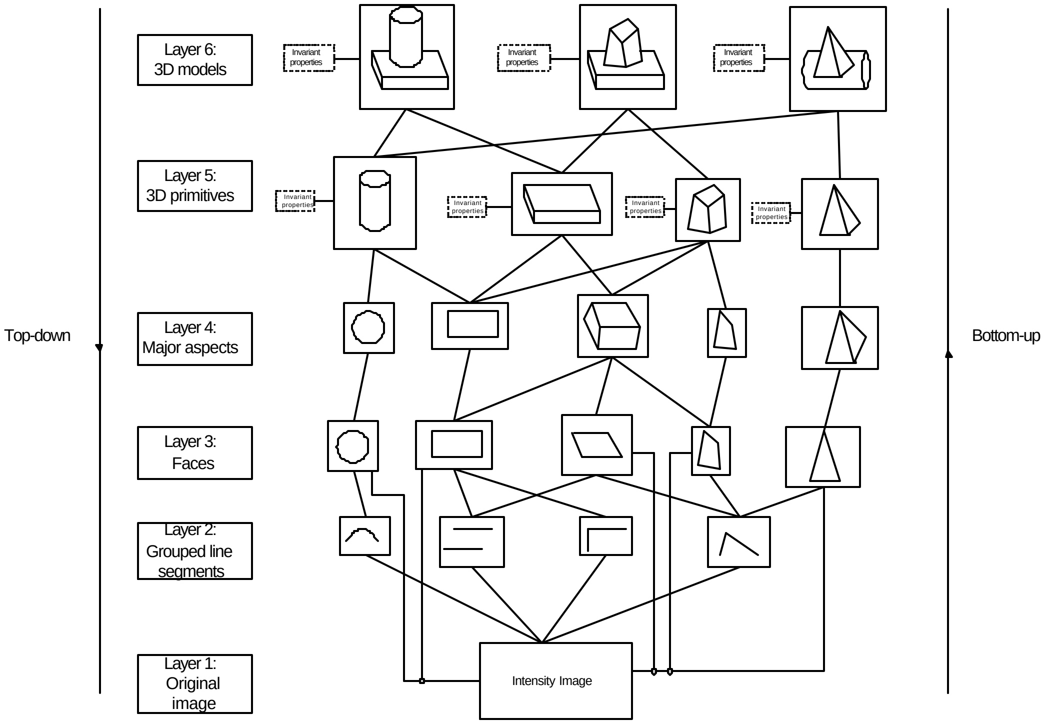


Fig. 1. The framework of a model representation. Layer 6 is the detailed 3D model descriptions from which 3D invariants, either photometric or geometric, could be extracted. Each model is thought to be made of several 3D primitives from which 3D invariants could be extracted in layer 5. Each 3D primitive can be further decomposed into many major aspects, which are 2D projections of 3D objects in layer 4. In layer 3 we show that combinations of different faces made different aspects. Faces in layer 3 are decomposed into grouped line segments in layer 2. In layer 1 intensity image may be made of both line segments and faces directly. Self-occlusion and occlusion within each layer are expressed implicitly. Going from layer 1 to layer 2 is called edge detection and perceptual organization. Going from layer 2 to layer 3 is called line-based segmentation while going directly from layer 1 to layer 3 is called region-based segmentation. The process starting from layer 1, original intensity image, followed by edge detection, segmentation,

perceptual organization and matching is called bottom-up approach. The process worked the other way around starting from layer 6, 3D model, followed by decompositions and verifications is called top-down approach.

Fig. 1 gives a general framework of model representation into which many existing 3D object recognition systems can be fitted. Different systems may have different jumps from one layer to another leading different complexity and flexibility. Dickinson et al. gave a detailed comparison, in primitive complexity, model complexity, search complexity, etc., among different systems. They showed us that 3D volumetric primitive representation method has the best overall performance.

Drew et al. (1997), Funt and Finlayson, (1995), Nagao and Grimson, (1997), Slater and Healey, (1996), Slater and Healey, (1997) and Stricker (1992) were typical bottom-up methods that mainly used photometric, specially color, invariant properties going from layer 1 directly to layer 6 to match objects. Some of them may use a little help of geometric invariant properties that may improve the accuracy and robustness of their systems. The use of photometric invariants as indexing greatly improves the speed of ORS because no time is spent in segmentation, organization and final matching. This strategy, however, suffered the problems that only one object should be present in the scene, 2D image of object can't change too much at different views and no further verification is applied.

Poggio and Edelman proposed a famous neural network system, GRBF (Generalized Radial Basis Function), which was trained how to recognize scene in neural networks. Although the learning method was very good it still suffered the same problems as those in the above.

Pontil and Verri, (1998) used a new technology called Support Vector Machine going from original image to 2D projections of 3D object at different poses to match 3D object. This is again a bottom-up method. Different images of many objects at different poses were stored in database and every pixel of input scene was feed to the Support Vector Machine as a property in one dimension. Given an image with the size $N \times M$, the recognition will find which is the one stored in database that is nearest to input scene in the $N \times M$ dimension space. Although this system ran very fast it suffered the same problems as stated in the above systems.

Lin et al. (1991) was a bottom-up method that used extracted regions and vertices to match with 3D model in Hopfield Neural Networks that considered 3D invariant properties as constraints among neurons. This system was processed in a hierarchical manner as bottom-up object-centered method. It, however, suffered many problems as: region and vertices correspondences are not processed at the same time, regions may not that easy to be segmented out from real images, global minimization can not be guaranteed to be approached and neuron connectivity matrix is extremely large when the number of model and scene regions is large. Suganthan et al. (1995) used Hopfield networks in recognizing 2D object as graph matching. Young et al. (1997) used a mutilayer Hopfield networks to recognize 2D object at different scales.

Ullman and Basri (1991) claimed that 2D coordinates of a 3D object under one view could be represented by the combination of two coordinates at other two different views. Alignment was used to match object within scene in their method.

Lamdan et al. (1990) used affine invariant properties as indexing to recognize objects. Wong et al. (1989) expressed 3D object model as attributed Hypergraphs and were trying to match extracted features with object model as labeling in attributed graphs. This was a bottom-up method as object-centered approach. Attributed information helps a lot in matching and we will see later this idea could be generalized in stochastic relaxation where attributes are defined on cliques.

Seibert and Waxman (1992) used multiview approach as a view-centered method that learns incoming novel views which made this system a typical view-centered method.

3 A Mutilayer Hopfield Neural Netowrks

Following the ideas of Lin et al. (1991), Suganthan et al. (1995) and Yong et al. (1997), we propose a new mutilayer Hopfield networks that recognize 3D object in 2D image as a bottom-up approach that compares 3D invariant properties of 3D model with those of extracted features.

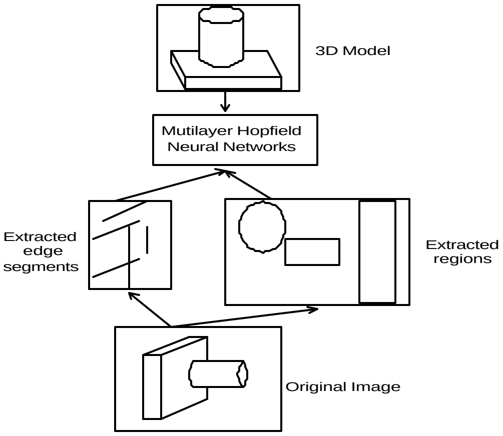


Fig. 2. The structure of a mutilayer Hopfield network. Edge detection and segmentation are applied to original image. 3D invariant properties of model and extracted features are compared simultaneously in the network.

3.1 Single Layer Hopfield Neural Network

Object recognition by graph matching, also referred to as morphism, is a mapping from a scene graph to a model graph. The morphism can be categorized on the basis of the constraints that are enforced during the mapping as follows: when the mapping is one-to-one and onto, it is an isomorphism; when it is one-to-one, it is a monomorphism; and when it is many-to-one, it is a homomorphism. Figure 3 gives a basic framework of the Hopfield neural network.

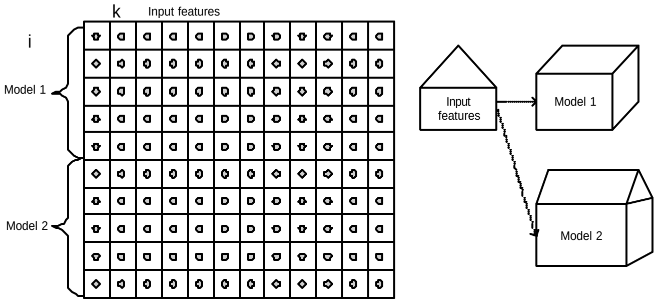


Fig. 3. Neuron states and candidate model-input features correspondence.

Each dot in the matrix represents a neuron that stands for similarity between one input feature and one feature of the candidate model. Its state (1 meaning absolutely similar and 0 meaning totally different,) can be determined when the minimization of the energy function is reached.

Energy function

We use a top-down strategy to achieve object recognition. The problem is treated as an optimization problem, where the correct answer is given when a global minimized energy state is reached. Let $C^1{}_{ik}$ and $C^2{}_{ikjl}$ be unary and binary similarity measure respectively. The energy function is

$$E = -A\sum_i\sum_k\sum_j\sum_l C_{ikjl}V_{ik}V_{jl} + B\sum_i(1-\sum_k V_{ik})^2 +$$

$$C\sum_i\sum_k\sum_{l\neq k}V_{ik}\times V_{il} + D\sum_k(1-\sum_i V_{ik})^2 + E\sum_k\sum_i\sum_{j\neq i}V_{ik}\times V_{jk}. \qquad (1)$$

The neuron state, V_{ik}, converges to 1.0 if the model feature i matches the input image feature k perfectly, otherwise, it is equal or close to 0. Thus, the first term measures similarity between the model and image features. The second term $\sum_i(1-\sum_k V_{ik})^2$ implies that the final states of neurons in the same row add up to 1, and the third term $\sum_i\sum_k\sum_{l\neq k}V_{ik}\times V_{il}$ confirms that there is at most one neuron that has a value greater than 0 in each row. This means that only one input image feature matches with each model feature. The forth term $\sum_k(1-\sum_i V_{ik})^2$ implies that the final states of neurons in the same column add up to 1, and the fifth term $\sum_k\sum_i\sum_{j\neq i}V_{ik}\times V_{jk}$ confirms that there is at most one neuron that has a value greater than 0 in each column. That means that each input image feature matches with only one model feature. Combining the second term $\sum_i(1-\sum_k V_{ik})^2$ with the third term $\sum_i\sum_k\sum_{l\neq k}V_{ik}\times V_{il}$ gives a solution that forces each model feature to match only one input image feature. Similarly, combining the forth term $\sum_k(1-\sum_i V_{ik})^2$ with the fifth term $\sum_k\sum_i\sum_{j\neq i}V_{ik}\times V_{jk}$ gives a solution that guarantees each input image feature will match only one model feature. The determination of coefficients A, B, C, D and E depends on how strictly the unique matching conditions should be implemented. Different values of in Equation (1) apply to various cases of our tasks. For monomorphism, coefficients B, C, D and E are assigned with high values based on the assumption that one model feature will uniquely match one input feature. The final solution yields a one-to-one mapping. In the case of homomorphism, coefficients B and C are assigned with low values (even zero) based on the assumption that one model feature will match several image input features.

The following is a detailed discussion on the single layer Hopfield neural network. Let C_{ikjl} denote similarity/disparity between a model feature pair (i, j) and an input image feature pair (k, l). It is then represented as:

$$C_{ikjl} = C^1_{ik} + C^1_{jl} + C^2_{ikjl} \ . \tag{2}$$

where

$$C^1_{ik} = \sum_{n=1}^{N_1} w^1{}_n \, f_n{}^1 (x_{in}, y_{kn}) \tag{3}$$

and $$C^2_{ikjl} = \sum_{n=1}^{N_2} w^2{}_n \, f_n{}^2 (x_{ijn}, y_{k\mathrm{ln}}) \ . \tag{4}$$

In the above equations C^1_{ik} and C^2_{ikjl} represent unary and binary similarity respectively. C^1_{ik} encodes compatibility between model feature i and input feature k, and C^2_{ikjl} encodes compatibility between the correspondence of the model feature pair (i, j) and that of the input feature pair (k, l). f is a similarity-measuring function and weighted by w that meets the condition

$$2\sum_{n=1}^{N_1} w^1{}_n + \sum_{n=1}^{N_2} w^2{}_n = 1 \ . \tag{5}$$

Output function

For neuron i, if its charge is u_i that is computed in the energy minimization, its neuron state output is represented as

$$v_i = g(u_i) = \frac{1}{1 + e^{-u_i / T}} \tag{6}$$

T is the "temperature" (an annealing term) that determines the speed and quality of the final solution. A very large value of T will cause neuron values to be 1, while a very small a value will drive the network to a local minimum state, or a slow convergence. An annealing process keeps the value of T large at the beginning and reduces the T value as iteration progresses. This is important for achieving a global minimum and a fast convergence.

Initialization

The initial values of neuron states can be chosen randomly as described in Lin et al. (1991). The network may converge to a local minimum state. As stated above, an annealing process may overcome this problem. However, C_{ik} may be calculated and used as a byproduct to set the initial neuron states as

$$V^0_{ik} = \begin{cases} C^1_{ik} / \sum_{j \in S_i} C^1_{ik}, & if \sum_{j \in S_i} C^1_{jk} > w \, and C^1_{ik} > w \\ C^1_{ik}, & if \sum_{j \in S_i} C^1_{jk} > w \, and \, C^1_{ik} > 0 \\ 0, & if C^1_{ik} < 0 \end{cases} \tag{7}$$

where S_i $is\{k \mid C^1_{ik} > 0\}$ and $w = \sum_{n=1}^{N_1} w_n$.

Combining matched features

After iterations using homomorphism, each neuron reached its final state V_{ik}. Those final states close to 1 yield matches between corresponding input image features with model features. However, there is still a need to put the matched features to form object(s). The following procedure combines the features under the assumption that there are N features forming an object.

a) Establish N sets of $S_i = \{k \mid V_{ik} \approx 1\}, i = 1, \ldots, N$. Each set contains all the input features that matched the corresponding model features.
b) Establish an empty set Q.
c) Set i to be 1.
d) For each $k \in S_i$ we get $m_{ik} = V_{ik}, k \in S_i$ if Q is empty, otherwise $m_{ik} = \sum_{(j,l) \in Q} (C_{ikjl} + C_{jlik})$. Find the feature k_n that satisfies $\forall k_p \in S_i,\ m_{ik_n} \geq m_{ik_p}$, add (i, k_n) to Q.
e) If $i = N$, one object is recognized and detected, go back to step a); otherwise, $i = i + 1$ and go back to step b).

3.2 **Multilayer Hopfield Neural Netowrk**

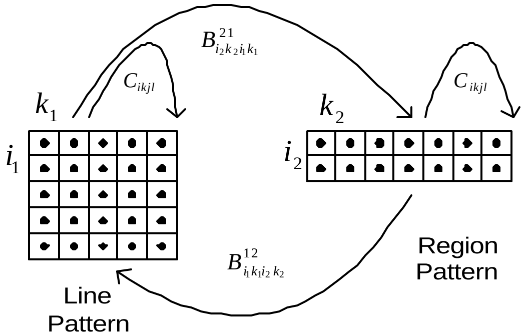


Fig. 4. Two-layer (line pattern and region pattern) Hopfield neural network

Fig. 4. shows the structure of a two layer Hopfield network that comparison 3D invariant properties of line segments and regions simultaneously. Matching of line segments will give supports to the layer of regions, and vice versa. This new algorithm changes the method that matches object in hierarchy ways into parallel approach with more robustness and parallelism. Connections among neurons in each single layer are fully dependent on geometric and photogrammetric constraints and are fixed before the initial iteration. During iterations the interconnections between the two layers vary. Let L_1 denote layer 1, which is a line pattern layer, and L_2 denote layer 2, which is a region pattern layer. We thus have an energy function

$$E = E_1(L_1) + E_2(L_2). \tag{8}$$

where $E_1(L_1) = E_{11}(L_1, L_1) + E_{12}(L_1, L_2)$ and $E_2(L_2) = E_{22}(L_2, L_2) + E_{21}(L_2, L_1)$. $E_{11}(L_1, L_1)$ and $E_{22}(L_2, L_2)$ are same as the terms in Equation (1). The energy relevant to interlayers are

$$E_{12} = a_1 \times \left(-\frac{1}{2}\right) \sum_{i_1} \sum_{k_1} \sum_{i_2} \sum_{k_2} B^{12}{}_{i_1 k_1 i_2 k_2} V_{i_1 k_1} V_{i_2 k_2} \tag{9}$$

$$E_{21} = a_2 \times \left(-\frac{1}{2}\right) \sum_{i_2} \sum_{k_2} \sum_{i_1} \sum_{k_1} B^{21}{}_{i_2 k_2 i_1 k_1} V_{i_2 k_2} V_{i_1 k_1} . \tag{10}$$

$\mathrm{B}^{12}{}_{\mathrm{i_1 k_1 i_2 k_2}}$ is a connectivity variable from neuron (i_1, k_1) in layer L_1 to neuron (i_2, k_2) in layer L_2. $B^{21}{}_{i_2 k_2 i_1 k_1}$ is a similar term. They change dynamically during iterations. We also have $B^{12}{}_{i_1 k_1 i_2 k_2} \neq B^{21}{}_{i_2 k_2 i_1 k_1}$ because contributions from one layer to another layer are non-symmetric. Using energy function (11), we can recognize the objects when a global minimized energy value is achieved.

$$B^{12}{}_{i_1 k_1 i_2 k_2} = \begin{cases} 2 \times \left(V2_{i_2 k_2} - \frac{1}{2}\right) & if\ Line\ k_1 \in Area\ k_2\ and i_2 = 0 \\ -2 \times \left(V2_{i_2 k_2} - \frac{1}{2}\right) & if\ Line\ k_1 \in Area\ k_2\ and i_2 = 1 \\ 0 & otherwise \end{cases} \tag{11}$$

The connectivity term contributes when a model region is a truck top and an input line belongs to an input region or when the model region is a truck shadow and the input line belongs to an input region. Similarly, $B^{21}{}_{i_2 k_2 i_1 k_1}$ is defined as

$$B^{21}{}_{i_2 k_2 i_1 k_1} = \begin{cases} 2 \times \left(V1_{i_1 k_1} - \frac{1}{2}\right) & if Line\ k_1 \in Area\ k_2 and i_2 = 0 \\ -2 \times \left(V1_{i_1 k_1} - \frac{1}{2}\right) & if\ Line\ k_1 \in Area\ k_2 and i_2 = 1 \\ 0 & otherwise \end{cases} \tag{12}$$

In this method, a Multilayer Hopfield Network is used to solve labeling problem which is actually thought as optimization problem. Because of the structure of Hopfield network, it suffers the following shortcomings:

1. Even with careful selection of initial values it's difficult for to the system to jump out of local minimal energy status. Since we can tell the similarity between input scene and model only by traveling to the global minimization, the fail of reach to global minimal energy status will lead to the fail of the system.

2. $\mathrm{C}_{\mathrm{ijkl}}$, $B^{12}_{i_1k_1i_2k_2}$ and $B^{21}_{i_2k_2i_1k_1}$, as four dimensional matrix, are very expensive to compute and store when the number of features extracted from scene and features in model goes to a large number.

4 Gibbs Distribution and Stochastic Relaxation Labeling

Gibbs distribution, MRF (Markov random field) equivalence, introduced by Geman and Geman (1984) receives enormous attentions in both low-level image analysis, e.g. image restoration, edge detection and clustering, and high-level image analysis, e.g. motion tracking and object recognition. As a probability distribution, it also has wide applications in other fields like reliability analysis, medical data analysis etc.

In both low-level image analysis and high-level image analysis, we can always generalize our problems as $Y = \Phi\ (X) + N$, where Y is the received data, X is the true data, N is the noise staying with Y and Φ is either a linear function or nonlinear function that projects data in domain of X to range of Y. Image restoration tries to find the original image X given degraded image Y where Φ is a linear one-to-one mapping; Edge detection tries to find edges X appearing in image Y; Image segmentation tries to find regions X standing in image Y; Motion tracking finds the real coordinate, X of an object at each time given image sequence Y; Object recognition detects the most possible object X in model database that generates given image Y. Each of the above problems, either low-level image analysis or high-level image analysis, falls into estimation problems. The beauties of Gibbs distribution, local property, convergence property and annealing etc. make it possible and much easier to solve the above problems.

As in Winkler (1995), they gave a definition of Gibbs distribution as follows.

Definition of Random Fields

Let S be a finite index set – the set of sites; for every site $s \in S$ let X_{s} be a finite space of states x_{s}. The product $X = \prod_{s \in S} X_s$ is the space of configurations $x = (x_s)_{s \in S}$. We consider probability measures or distributions Π on X, e.g. vectors $\Pi = (\Pi(x))_{x \in X}$ such that $\Pi\ (x) \geq 0$ and $\sum_{x \in X} \Pi\ (x) = 1$. Subsets $E \subset X$ are called events; the probability of an event E is given by $\Pi\ (E) = \sum_{x \in E} \Pi\ (x)$. A strictly positive probability measure Π on X, e.g. $\Pi\ (x) \geq 0$ for every $x \in X$, is called a stochastic or random field.

Definition of neighborhood system and cliques

A collection $\partial = \{\partial(s) : s \in S\}$ of subsets of S is called a neighborhood system, if (i) $s \notin \partial(s)$ and (ii) $s \in \partial(t)$ if and only if $t \in \partial(s)$. The sites $s \in \partial(t)$ are called neighbors of t. A subset C of S is called a clique if two different elements of C are always neighbors. The set of cliques will be denoted by C. We shall frequently write $\langle s, t \rangle$ if s and t are neighbors of each other. The neighborhood relation induces and undirected graph with vertices $s \in S$ and a bond between s and t if and only if s and t are neighbors. Conversely, an undirected graph induces a neighborhood system. The 'complete' sets in the graph correspond to the cliques.

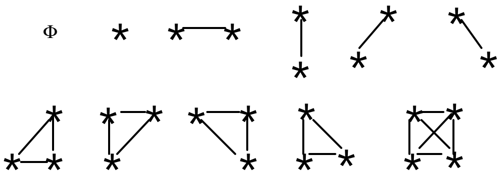


Fig. 5. Different cliques.

The random field Π is a Markov field w.r.t. the neighborhood system ∂ if for all $x \in X$, $\Pi(X_s = x_s \mid X_r = x_r, r \neq s) = \Pi(X_s = x_s \mid X_r = x_r, r \in \partial(s))$.

Definition of Potential
A potential is a family $\{U_A : A \subset S\}$ of functions on X such that
(1) $U_f = 0$
(2) $U_A(x) = U_A(y)\ if\ X_A(x) = X_A(y)$
The energy of the potential U is given by $H_U = \sum_{A \subset S} U_A$. Given a neighborhood system ∂ a neighbor potential w.r.t. ∂ if $U_A = 0$ whenever A is not a clique. Potential defines energy functions and thus random fields.

Gibbs distribution:

$$\Pi(x) = \frac{\exp(-H(x))}{\sum_{z \in X} \exp(-H(z))} \tag{13}$$

where Π is Gibbs field and H is the energy function. A random field Π is a Gibbs filed or Gibbs measure for the potential U and H is the energy H_U on a potential U. If U is a neighbor potential then Π is called a neighbor Gibbs field.

MRF relaxation labeling

To overcome the shortcomings of our mutilayer Hopfield network, we use a neighborhood Gibbs field to solve the same problem as we proposed in the last section. The system approach is the same as Fig. 2 while the neural network is instead replaced by MRF. Modestino and Zhang (1989) proposed a basic MRF approach for scene labeling. Li (1996) extended Modestino and Zhang 's idea and gave an approach how a basic labeling problem could to solved as MAP (Maxim A Posterior) of the MRF.

Suppose we have a set of extracted line segments $S_L^1 = \{L_i | i = 1\Lambda\ m_1\}$ where m_1 is the number of line segments and a set of extracted regions $S_R^1 = \{R_i | i = 1\Lambda\ m_2\}$ where m_2 is the number of regions. A given model are line segments $S_L^2 = \{l_j \mid j = 1\Lambda\ n_1\}$ where n_1 is the number of line segments and the set of extracted regions $S_R^2 = \{r_j \mid j = 1\Lambda\ n_2\}$ where n_2 is the number of regions. There are two kinds of cliques, first order, which corresponds to unary similarity in Hopfield network and the second order, which corresponds to binary similarity in Hopfield network.

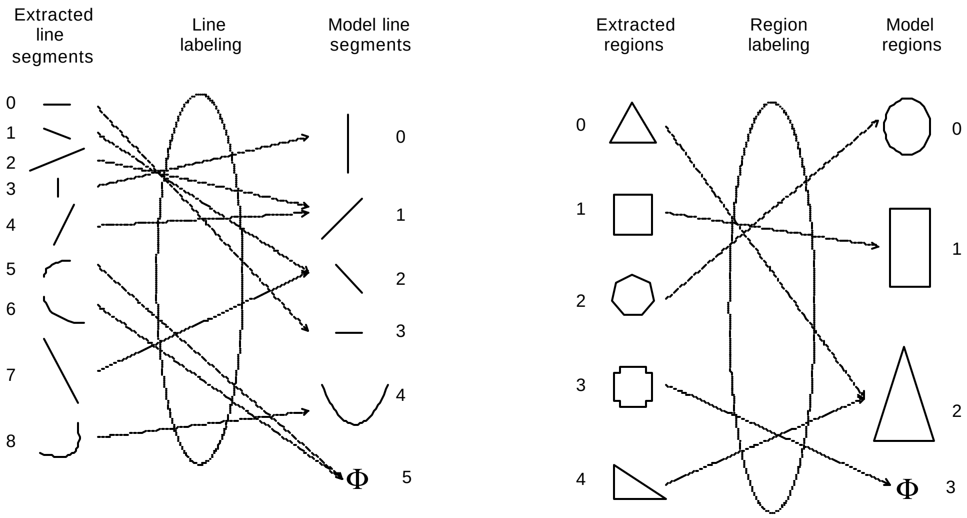


Fig.6. Label extracted regions and line segments with model line segments and regions

The first order cliques are as:

$C_1^1 = \{\{i\} \mid i \in S_L^1\}$ and $C_1^2 = \{\{j\} \mid j \in S_R^1\}$.

The second order cliques are as:
$C_2^1 = \{\{i_1, i_2\} \mid i_1, i_2 \in S_L^1 \; and \;\; i_1 \neq i_2\}$, $\quad C_2^2 = \{\{j_1, j_2\} \mid j_1, j_2 \in S_R^1 \; and \;\; j_1 \neq j_2\}$ and $C_2^3 = \{\{i, j\} \mid i \in S_L^1, j \in S_R^1 \; and \; S_L^1 \in S_R^1\}$.

Suppose random field F is a mapping $L \rightarrow l$ and $f_i = l_j$ when L_i is labeled with l_j. Suppose random field G is a mapping $R \rightarrow r$ and $g_i = r_j$ when R_i is labeled with r_j.
Let's define: $D^1(f_i)$ = similarity of L_i and l_j if $f_i = l_j$, $D^2(g_i)$ = similarity of R_i and r_j if $g_i = r_j$; $V^1(f_{i1}, f_{i2})$ = similarity of (L_{i1}, L_{i2}) and (l_{j1}, l_{j2}) if $f_{i1} = l_{j1}$ and $f_{i2} = l_{j2}$, $V^2(g_{i1}, g_{i2})$ = similarity of (R_{i1}, R_{i2}) and (r_{j1}, r_{j2}) if $g_{i1} = r_{j1}$ and $g_{i2} = r_{j2}$, and $T_{i,j}(f_i, g_j) = \begin{cases} 1 & f_i \in g_j \\ 0 & otherwise \end{cases}$.

The energy defined on cliques then becomes

$$H(f, g) = H_1(f) + H_2(g) + H_3(f, g) \tag{14}$$

where $H_1(f) = \sum\limits_{C_1^1 \subset S_1, i \in C_1^1} D_i^1(f_i) + \sum\limits_{C_2^1 \subset S_1, i1 \in C_2^1, i2 \in C_2^1} V_{i1,i2}^1(f_{i1,i2})$, (15)

$$H_2(f) = \sum_{C_1^2 \subset S_1, i \in C_1^2} D_i^2(g_i) + \sum_{C_2^2 \subset S_1, i1 \in C_2^2, i2 \in C_2^{21}} V_{i1,i2}^2(g_{i1,i2}), \tag{16}$$

and $H_3(f, g) = k \sum\limits_{C_2^3} T_{C_2^3}$ (17)

Given the above energy defined on cliques, we thus have the probability

$$\Pi(Y \mid F,G) = \frac{\exp(-H(f,g)/B)}{\sum_{f,g}\exp(-H(f,g)/B)} \tag{18}$$

where random field Y is the given extracted features.

The posterior probability is then

$$P(F,G \mid Y) = \frac{\Pi(Y \mid F,G)P(F,G)}{P(Y)} \tag{19}$$

where $P(F,G)$ is the prior probability labels.

For a given scene, $P(Y)$ is fixed and our final goal is to find the maximal value of $\Pi(Y \mid F,G)P(F,G)$ as the solution of MAP. The detailed description of how to travel on this probability space as Markov Random Chain will be discussed in the next section.

Compared with mutilayer Hopfield network, this MRF approach can be guaranteed to find the maximal probability, in other words, the minimal energy status. Hopfield network here is only a special case of MRF approach when $P(F,G)$ is identical every where in the domain of $F\ and\ G$ and $H(F,G) = E$ where E is the energy in equation (8).

5 Integrating Top-Down and Bottom-up Processes by Markov Chain Monte Carlo for Object Detection

In this section, a new object recognition system that uses an integrated top-down and bottom-up process by Markov Chain Monte Carlo is introduced.

5.1 Problem definition

The data we get from mobile mapping system are image sequences with georeferenced information, which are exterior orientation parameters and interior orientation parameters.

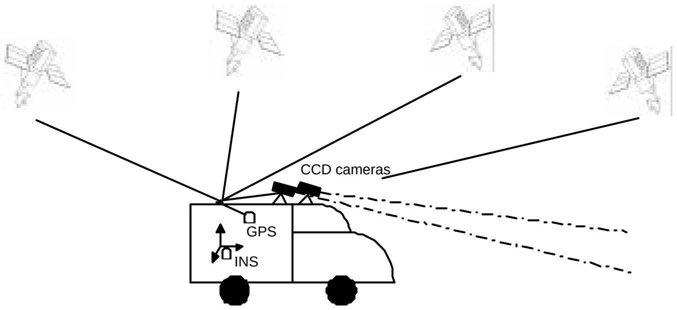
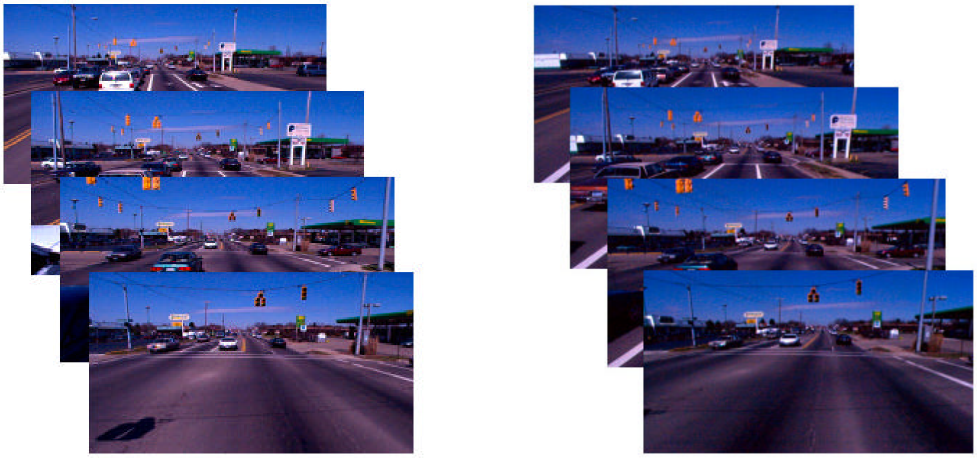


Fig.7. The basic mobile mapping system that has two CCD cameras, left and right, one INS (Inertial Navigation System) that measures orientation of the van and one GPS (Global Positioning System) that measures ground coordinates of the van.



(a) Left image sequences (b) Right image sequence

Fig. 8. Color image sequence taken by Mobile Mapping System. They are used as our input data in which civil infrastructures, e.g. traffic lights, stop signs , etc., are going to be recognized.

5.2 General approach

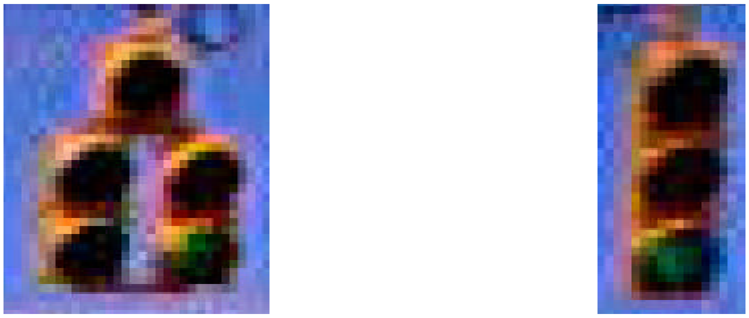


Fig. 9. Traffic light image pieces cut from color image sequences with size of 30X40 and 20X30 respectively.

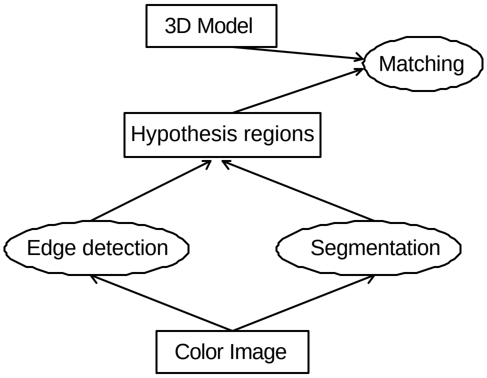


Fig. 10. A general framework for recognition of traffic signs in color image.

There are many civil infrastructure recognition systems existing in literature, most of which are designed for the purpose of autonomous driving. In Salgian and Ballard (198) color values and steerable filters are used in simulated scene to find traffic signs. The methods they proposed is quite simple since they are designed for on-line driving and won't work in real image sequences.

In Yuille et al. (1998), seeded regions that have similar color as models are picked and grown to hypothesis regions in which geometric information like edges are detected followed with matching between extracted boundaries and those of models. The system we design could be off-line instead of on-line to recognize traffic signs correctly and capture their spatial information accurately. The real image sequences we have are quite noisy where simple methods won't work.

5.2.1 Edge detection in color image

Edge detection methods have been studied in literature for quite many years and different edge model and different criterions lead to different edge detection algorithms. Many edge detectors, e.g. Canny, Log, Snake etc., are available to detect edges in gray value image. Finding edges in color image, however, is more complicated than in gray image. This is because we are looking for 2D edge points (x, y) in color image that has three bands and each of which has its corresponding edge map. In other words, each pixel in color image is a 3D vector, which will stay, under different considerations, in different 3D spaces, e.g., $[R \quad G \quad B]$, $[Y \quad U \quad V]$ etc. The procedure to find edges thus is a mapping from 3D space, color image, to 2D space, edge map. Lee and Cok proposed a new method to detect boundaries in vector field which gives a solution that extracts edge map in color image.

Suppose each pixel in color image is denoted as $[u(x,y)\ v(x,y)\ w(x,y)]'$ whose partial derivative matrix along x and y can then be denoted as

$$D = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} \end{bmatrix}. \tag{20}$$

The matrix $D^T D$ is denoted as $\begin{bmatrix} p & t \\ t & q \end{bmatrix}$ where

$p = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2$, $t = \left(\frac{\partial u}{\partial x}\right)\left(\frac{\partial u}{\partial y}\right) + \left(\frac{\partial v}{\partial x}\right)\left(\frac{\partial v}{\partial y}\right) + \left(\frac{\partial w}{\partial x}\right)\left(\frac{\partial w}{\partial y}\right)$ and

$q = \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2$.

The largest eigenvalue l and its corresponding eigenvector $\underline{g}$ of $D^T D$ are the gradient magnitude and the gradient direction at each edge point respectively. We can compute l as

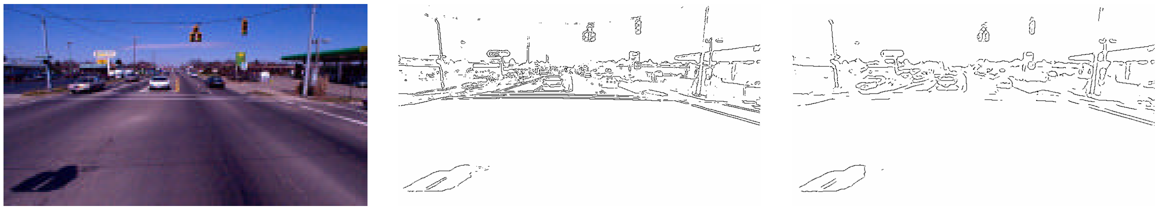
$$l = \frac{1}{2}\left(p + q + \sqrt{(p+q)^2 - 4(pq - t^2)}\right) \tag{21}$$

and thus have $\underline{g}$ as

$$\underline{g} = \begin{cases} \textit{any normalized vector} & p = t = q = 0 \\ [\,l,\ t]^T & t = 0 \textit{ and } l = q\,. \\ [t, l - p]^T & \textit{otherwise} \end{cases} \tag{22}$$

The color edge detection algorithm is as follows,

1. Canny operator is used separately in R, G and B layer to compute gradient maps.
2. $\underline{g}$ is computed as equation (22).
3. Same non-maximal suppression as Canny operator is used to get the final edge maps.



(a) Original color image (b) Edge map at $s = 1.0$ (b) Edge map at $s = 2.5$

Fig. 11. Color edge detector in mobile mapping images.

5.2.2 Mean shift clustering algorithm

Clustering algorithms have been explored by researchers for decades and are applied to both low-level image processing like image segmentation and high-level image processing like object recognition. Among hundreds of existing clustering algorithms, k-means clustering is a very famous one and widely used. In Cheng (1995) the author analyzed a more generalized algorithm, mean shift, in which k-means algorithm is a special case. Due to the overall performance, both convergence property and computational issue, we try this algorithm in color image segmentation and use this one in vanishing points detection and analysis of generalized Hough transformation.

Suppose we have a finite set S in the n-dimensional Euclidean space X in which S is normalized as $\sum_{x\in S} p(x) = 1$ where $p(x)$ is the value of each site x. For the sake of convenient, we treat x as random field defined on S. Let B_x be a $l-ball$ in X centered at x as $B_x = \{y \,|\, \|y-x\| \le l\}$. Given a site x and its corresponding $l-ball$ B_x we could compute the sample mean as

$$m(x) = E[y \mid B_x] = \int yp(y \mid B_x)dy = \frac{\int_{B_x} yp(y)dy}{p(B_x)}. \qquad (23)$$

We could approximate $p(y)$ using taylor expansion as $p(y) = p(x) + (y-x)^T \nabla p(x)$ so that the shift between $m(x)$ and x is

$$m(x) - x = \frac{\int_{B_x} (y-x)p(y)dy}{p(B_x)} = \frac{\int_{B_x} [(y-x)p(x) + (y-x)(y-x)^T \nabla p(x)]dy}{p(B_x)} \qquad (24)$$

where $\int_{B_x} (y-x)dy = 0$. The mean shift finally is

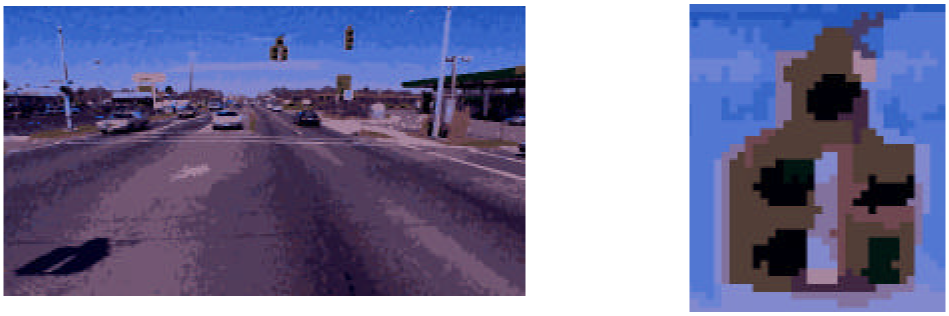
$$m(x) - x = \frac{\int_{B_x} (y-x)(y-x)^T dy}{p(B_x)} \nabla p(x) \qquad (25)$$

which is propoportational to gradient $\nabla p(x)$. We stop at a local maximal where $m(x) - x = 0$. This proves that this mean-shift algorithm is guaranteed to converge at the local maximal value. Detailed analysis could be found in Cheng (1995).

5.2.3 Color image segmentation

Many researchers have studied color image segmentation methods for many years. It is well known that the perfect segmentation of image, either in color or in gray value, could not be achieved without the full interoperation of scene. In other words, pure low-level image processing won't give us perfect results without the involvement of high-level image processing.

Huang et al. (1992) segmented color image by combining scale space filter and Markov random field together. Liu and Yang proposed a multiresolution color image segmentation method which actually is treated as a MAP problem to segments. Comaniciu and Meer (1997) used mean-shift algorithm as clustering method in classifying histogram. We test Comaniciu and Meer 's method on our color images and have the results as in Fig. 12.



(a) Segmentation results of real image (b) Zoomed traffic light cut from (a).

Fig. 12. Color image segmentation using mean-shift algorithm. The advantage of this algorithm is that it runs very fast. The disadvantage is that the results are not very good although we see well extracted regions when simple color images are tested. In (b) there are more than 10 regions existing even for a single traffic light which makes later recognition very difficult.

McCane (1996) proposed an adaptive segmentation method that combines and splits regions at different scales in gray image. We extend this method to different channels, R, G B so that regions at different scales and layers compete each other. The strategy is illustrated in Fig. 13.

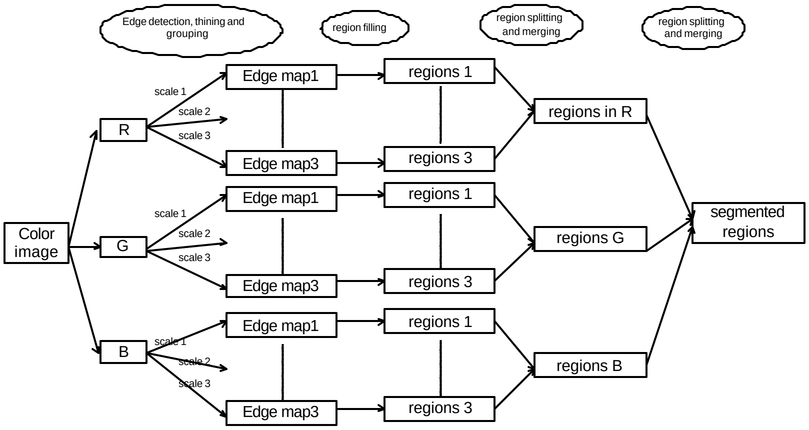
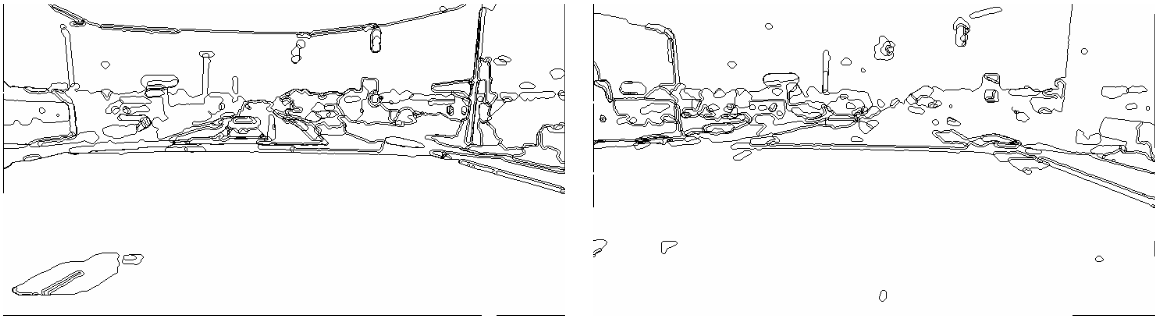


Fig. 13. Adaptive color image segmentation strategy.



(a) Segmentation results of left image with different scales, $s = 1.0$, $s = 2.0$ and $s = 3.0$.

(b) Segmentation results of right image with different scales, $s = 1.0$, $s = 2.0$ and $s = 3.0$.

Fig. 14. Segmentation results by adaptive algorithm. This method gives more reasonable regions than those given by mean-shift algorithm while spending much more time. However, the results, as we can see from (a) and (b), are not stable.

From above experiments we find that although the existing algorithms work fine in simulated scene and simple images, they all fail in real image sequences. In the next section we will propose a new method that runs faster than traditional segmentation method while giving much better results.

5.3 Integrating Top-down and Bottom-up processes by Markov Chain Monte Carlo for Object Recognition

In this section, we propose a complete new method that integrates Top-down and Bottom-up to recognize 3D objects, more specifically, traffic lights.

5.3.1 Interpretation of scene

Computer vision tries to understand 2D images, which are back-projections of 3D scenes. Recognition of 3D objects appearing in 2D image requires proper models to represent 2D images leading to proper models to represent 3D scene. Miller et al. (1995) and Miller et al. (1997) gave a basic random model to represent 3D scene in the recognition of objects by jump-diffusion.

Suppose we have detailed 3D models $\{O_i, i = 1\Lambda\ n\}$ that describe every possible existing objects in 3D scene and each of these models is paramized by 3D coordinates, pose and other parameters. Any possible scene x can be denoted as $x \in c \subset \cup_{i=1}^{n} \cup_{m=0}^{\infty} O_i^n$ where m is the occurance number of every object and n is the overall number of objects that may appear in the scenes. The imagery data could be denoted as $y \in \mathrm{Y}$ where Y is the observation space. We then have the likelihood function $L(\bullet \mid \bullet): Y \times X \to \Re$. The likelihood of y given observed scene x, $L(y \mid x)$, is the conditional probability. We can further define EOP (Exterior Orientation Parameter) and IOP (Interior Orientation Parameter) which are always given by GPS and INS as $e \in E$.

In Bayesian inference problems, posterior probability density is awlays wanted to estimate x given y. The posterior density is

$$p(x\,|\,y,e) = \frac{1}{Z(y,e)}\,p(x)L(y\,|\,x,e) \qquad (26)$$

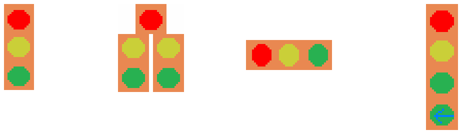
where $Z(y,e)$ is probability of $y\, and\, e$. Poor (1994) discussed the basic signal estimation problems, which are discussed a little bit below. Suppose we have a function $C : \Lambda \times \Lambda \rightarrow \Re$ such that $C[a,q]$ is the cost of estimating a true value of q as a, for a and q in Λ. Given such a function C we can then associate with an estimator $\hat{q}$ a conditional risk or cost averaged over Y for each $q \in \Lambda$; e.g., we have $R_q(\hat{q}) = E_q\{C[\hat{q}(Y),q]\}$. If we now adopt the interpretation that the actual parameter value q is the realization of a random variable q, we can define an average or Bayes risk as $r(\hat{q}) \underline{\underline{\Delta}} E\{R_q(\hat{q})\}$ and the appropriate design goal is to find an estimator minimizing $r(\hat{q})$. Different choices of risk function yield different Bayes estimators. The function $C[a,q] = (a-q)^2$ yields the Minimum-Mean-Square-Error estimator $\hat{q}_{MMSE} = E\{q \mid Y = y\}$; The function $C[a,q] = |\, a-q \,|$ yields the Minimum-Mean-Absolute-Error estimaor $\hat{q}_{MMSE} = median\, of\;\; p(q \mid Y = y)$; The function $C[a,q] = \begin{cases} 0 \; if \,|\, a-q \,| \leq \Delta \\ 1 \; if \,|\, a-q \,| > \Delta \end{cases}$ yields the Maximal-A-Posterior estimator $\hat{q}_{MAP} = \arg\{\max_{q}\, p(q \mid Y = y)\}$.

To recoginze 3D object in 2D images, we always choose the MAP estimator which finds the x that makes $p(x \,|\, y,e)$ to be the largest value. Since each observed image is just the 2D projection of we can have the expression as $\mathrm{Y} = c \times \Re^3_{e_1} \times \Re^2_{e_2} + N$ where $\Re^3_{e_1}$ is the 3D transformation in which e_1 is EOP, where $\Re^2_{e_2}$ is the 2D transformation in which e_2 is IOP and N is imposed noise. Many existing bottom-up methods try to find x, either implicitly or explicitly, with given data y. Among them indexing of 3D invariants is a straightforwd way to do. Generalized Hough transformation is another smart way that tries to find the most significant evidance by voting to c space. Direct indexing is straightword, easy to compute and runs fast but 3D invariants may not always, in most cases, exist. The Hough transformation sapce which is actaully a probability distribution, is a rough approximation of $p(x \,|\, y,e)$ and it works only in well defined situation. That's why line detection using Hough transformation is widely used where objects—lines are very simple and their background is clean. The method we are proposing tries to absorb the advantages of indexing and hough transformation, fast approach, and estimates x more accurately in Markov Chain Monte Carlo random process.

5.3.2 Description of models—traffic lights

If we focus on our task, recognition of traffic lights in outdoor images, the detailed description of parameters becomes important because we don't know otherwise what are the parameters to estimate.

(1) type t



(a) Type 1 (b) Type 2 (c) Type 3 (d) Type 4

Fig. 15. Different types of traffic lights

Fig. 15 shows 4 different types of traffic lights that appear the most time. Generally speaking, different types of objects should have different parameter space to describe. In the case of traffic lights it just happens to be same number and items of parameters for each type.

(3) illuminanc of shell, red light, yellow light and green light which are $c_s(R,G,B)$, $c_r(R,G,B)$, $c_y(R,G,B)$ and $c_g(R,G,B)$ respectively.

(4) size of the primitive (w,h)

For each model, we have the assumption that each type of traffic light is made by several primitives that have identical shape and size.

(5) spatial position (x,y,z)

(6) rotation angles (n,k,j) along X, Y and Z respectively

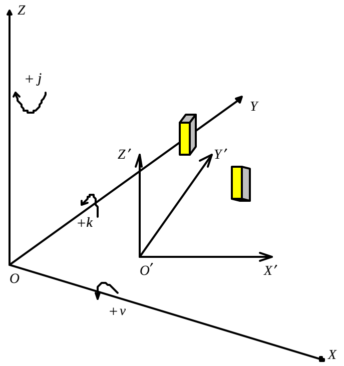


Fig. 16. Absolute ground coordinates system (X,Y,Z), local coordinates (X',Y',Z') and occruances of traffic lights.

Fig. 16 shows the basic coordinates systems in which 3D objects live where (X,Y,Z) is the absolute ground coordinates system and (X',Y',Z') is the local coordinates system. The reason why we define this local coordinates is because the occurance of trafic lights shows nice propertis that meet our aspect framework in Fig.1. Let (n',k',j') be the rotation angles of traffic light in terms of local coordinates system (X',Y',Z'). We may have the assumptions that $n'=0$ and $k'=0$ because traffic lights are always hung to be perpendicular to ground. Rotation angle j' is close to one of four major aspects, $0^0, 90^0, 180^0$ *and* 270^0. We thus have the prior probability of j' as

$p(j') = \frac{f(j')}{Z}$ where

$$f(j') = \frac{1}{\sqrt{2p s}} \exp\{-\frac{1}{2s^2}(j'-0)^2\} + \frac{1}{\sqrt{2p s}} \exp\{-\frac{1}{2s^2}(j'-90)^2\} +$$

$$\frac{1}{\sqrt{2p}s}\exp\{-\frac{1}{2s^2}(j'-180)^2\}+\frac{1}{\sqrt{2p}s}\exp\{-\frac{1}{2s^2}(j'-270)^2\}+\frac{1}{\sqrt{2p}s}\exp\{-\frac{1}{2s^2}(j'-270)^2\} \qquad (27)$$

and $Z=\sum_{j'=0}^{360} f(j')$.

Fig. 17 shows the prior pdf of j'.

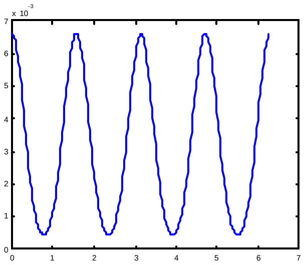


Fig. 17 Prior probability of j'.

Let (n_1,k_1,j_1) be the rotation angles of local coordinates system interms of global coordinates system, we can compute (n,k,j) as follows,

$n=n_1+n'$,

$k=k_1+k'$,

and $j=j_1+j'$ where n', and k' could be approximated as 0. (n_1,k_1,j_1), however, could be solved by vanishing points detection which we will discuss larter.

5.3.3 Vanishing points detection

As we state above, it's important to known (n_1,k_1,j_1) to compute (n,k,j). It is well known that a set of paralle lines in 3D scene generates a set of lines in 2D image that converge to a single point which is called vanishing point. Although it is true that there are infinite numbers of paralle line sets existing in real scene, the most dominate directions are along (n_1,k_1,j_1) in mobile mapping imageries. Due to this fact we may get (n_1,k_1,j_1) by detecting vanishing points in a single image.

Brillault-O'Mahony (1991) proposed a new method to detect vanishing points in a new accumulator space other than Gaussian sphere. Lutton et al. (1994) tried to detect (n_1,k_1,j_1) in Gaussian Sphere. Shufelt (1996) used vanishing point in the detection of buildings in aerial images. Coughlan and Yuille (1999) tried to use gradient of edge points instead of direction of extracted straight lines to determine (n_1,k_1,j_1) in Bayesian Inference where $n_1\, and\, k_1$ are actually approximated with 0.

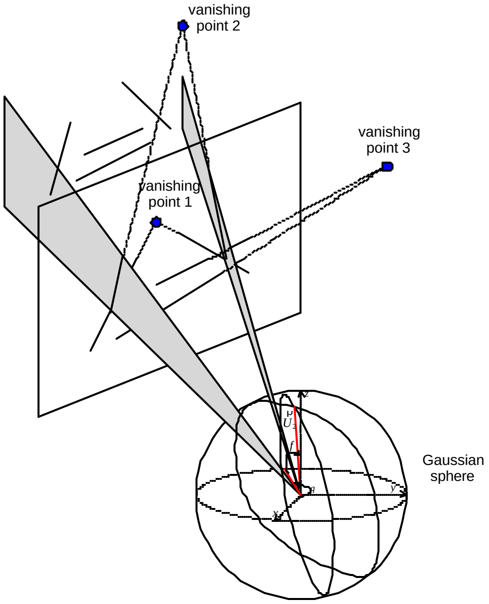


Fig. 18. Vanishing points geometry and their corresponding Gaussian sphere. (This graph is modified from Shufelt (1996).

As stated in Lutton et al. (1994), let $\overset{\mu}{U}(q_u, f_u)$ be the direction of a vanishing point direction in Gaussian sphere and $\overset{\mu}{N}_i(q_i, f_i)$ be the norm of a surface that passes the origin of the Gaussian sphere and two extremes of straight line segments. We have the basic knowledge $\overset{\mu}{N}_i \bullet \overset{\mu}{U} = 0$ from which we have the equation as $\cos(q_i - q_u)\sin f_i \sin f_u + \cos f_i \cos f_u = 0$.

The method to detecting vanishing points we are using is:

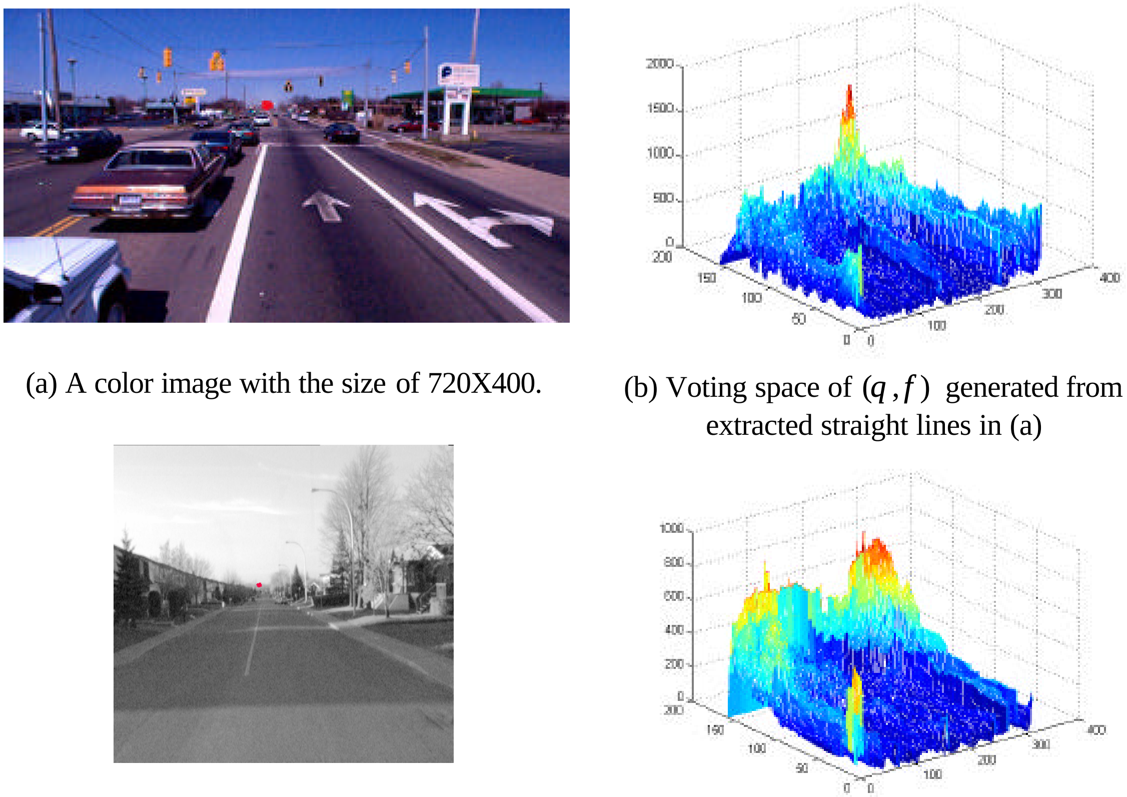
(1) Get edge maps using color edge detection algorithm or Canny in gray value image at large scale s .

(2) Apply edge thining and following methods to get line segments.

(3) Use Lowe edge split method to split line segments into straight line segments.

(4) For each extracted straight line segment, $\overset{\mu}{N}_i(q_i, f_i)$ is computed and thus every $\overset{\mu}{U}(q_u, f_u)$ that meets the above equation is voted in to (q, f) space where q is not equally spaced because same piece on different place of the Gaussian sphere cover different area. The interval of q is selected as $\Delta q(k) = \dfrac{\Delta S}{\Delta f} \cdot \dfrac{1}{\cos(k-1) - \cos(k)}$.

(5) Mean-shfit clustering algorithm is applied to find vanishing points.



(a) A color image with the size of 720X400.

(b) Voting space of (q,f) generated from extracted straight lines in (a)

(c) A gray image with the size of 520X400.

(d) Voting space of (q,f) generated from extracted straight lines in (c)

Fig. 19. Vanishing points detection algorithm applied in both color image and gray image.

This algorithm tests many images in either color or gray value and we found it's robust under different circumstances. The directions of (n_1,k_1,j_1) thus can be easily computed in a signle image.

5.3.4 Top-down and Bottom-up method

If we go back to Fig. 1, which shows the basic framework of a 3D object recognition system, we could find that our conditions here nicely fit this framework. Each traffic light is made of several primitives and each of them has four aspects, which can be approached by vanishing points detection method.

As we state before, color image segmentation methods are time-consuming and the results are not promising. To make our algorithm practical, we develop a new method that integrates bottom-up and top-down methods to recognize traffic lights fast and correctly. The basic bottom-up and top-down strategy is stated below.

Bottom-up approach

1. Edges are detected from color image.
2. (n_1,k_1,j_1) are computed using vanishing points detection algorithm.
3. Different aspect images of primitives are used as template in histogram filtering to compute hot spot map at different scale.

4. Minimal risk signal detection method is used to get hot spot map as a typical signal detection problem and one image is used as training set.
5. Image pieces that contain hot spots are extracted.

Top-down approach

1. Markov Chain Monte Carlo is used to recognize traffic lights thus to get the best estimations of parameters that describe each traffic light.

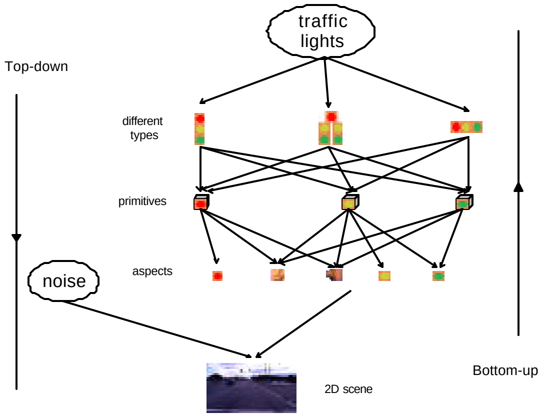


Fig. 20. Realization of Fig. 1 as bottom-up and top-down method in traffic light recognition.

5.3.5 Histogram filtering

Many researchers have been trying to recognize objects in color images using color invariants and geometric invariants. Swain and Ballard (1991) initiated a new method called “color indexing”, which actually compare histograms of given image with object stored in database in black-white, red-green and blue-yellow spaces. To capture more invariant information, Funt and Finlayson (1995) used Laplacian and four directional first derivatives to convolve with color image and compute the histogram again. Slater and Healey (1996) used bcal color pixel distributions instead of whole image to recognize objects. Nagao and Grimson (1997) combined photometric invariants and geometric invariants together to recognize 3D object under different views.

Color values of each pixel in an image could be denoted as a vector $r = (r_1, r_2, \Lambda\ r_m)^T$ where $r_k, k = 1\mathrm{K}\ m$ represents scalar response of the kth sensor channel. r_k can be denoted as

$$r_k(x) = \int S(x, l)E(x, l)R_k(l)dl \qquad (28)$$

where $S(x, l)$ is the spectral reflectance funtion of the object surface at x and $E(x, l)$ is the spectal power distribution of the ambient light and $R_k(l)$ is the spectral sensitivity of the kth sensor.

As proved in Nagao and Grimson (1997), final invariants could be awarded with

$$e_{ij} = \frac{r_i}{r_j} \Big/ \frac{r'_i}{r'_j} = \frac{E(l_i)}{E(l_j)} \Big/ \frac{E'(l_i)}{E'(l_j)} . \qquad (29)$$

The local color invariants are important to us because we only want to extract the hot spots that are most likely to be traffic lights insead of the whole image. Using this notion we make three aspects of the primitives of traffic lights as 2D templates, which are specifed in Fig. 20. In the advoidence of segmentation, we develop a new algorithm that captures both photometric and geometric invariants to get hot spot maps using histogram filtering. The algorithm is as follows:

(1) Original color image in (R, G, B) space is converted to L^*, u^*, v^* space (Wyszecki and stiles , 1982).

(2) Several 2D image templates, $I_i, i = 1\Lambda\ n$, are generated as 2D projections of traffic light primitives at major views.

(3) Color template images $I_i, i = 1\Lambda\ n$ are transformed from (R, G, B) to L^*, u^*, v^* space and histograms are computed as

$H_i^{L^*}(j) = \frac{1}{Z} \sum_{s \in I_i, L^*(s) = j} 1$ where j is each bin value in the domain of L^* where Z is the normalization factor so that $\sum_j H_i^{L^*}(j) = 1$,

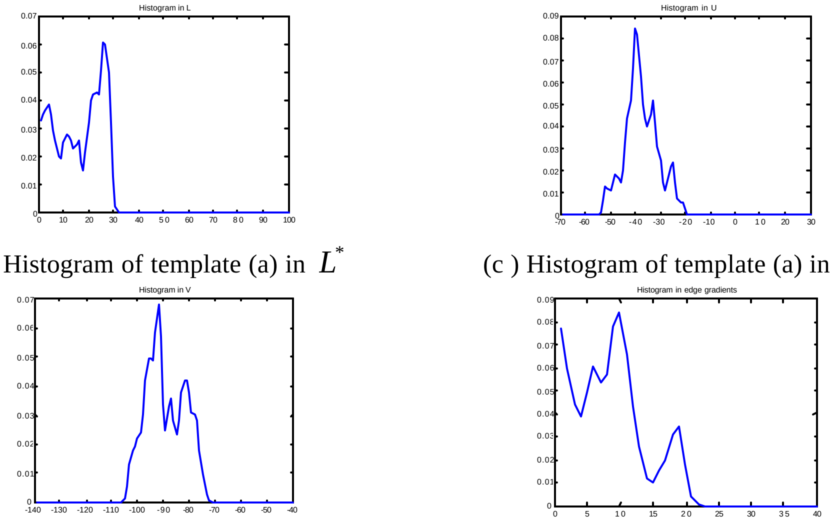
$H_i^{u^*}(j) = \frac{1}{Z} \sum_{s \in I_i, u^*(s) = j} 1$ where j is each bin value in the domain of u^* where Z is the normalization factor so that $\sum_j H_i^{u^*}(j) = 1$,

and $H_i^{v^*}(j) = \frac{1}{Z} \sum_{s \in I_i, v^*(s) = j} 1$ where j is each bin value in the domain of v^* where Z is the normalization factor so that $\sum_j H_i^{v^*}(j) = 1$.

As for geometric invariants, edge map is obtained using color edge detection method we state before at $s = 1.0$. A large s value is not satisfied because aspect image is small. The set of edge pixels are denoted as $I_i^E = \{s \mid s \in I \ and \ s \, is \, edge \, pixle\}$. Histogram of gradients of edge point is computed as

$H_i^E(j) = \frac{1}{Z} \sum_{s \in I_i^E} 1$ where j is each bin value in the domain of discrised gradients values where Z is a normalization factor so that $\sum_j H_i^E(j) = 1$. Also, we get edge pixel $p_i^{on} = \frac{\sum_{s \in I_i^E} 1}{\sum_{s \in I_i} 1}$ and $p_i^{off} = 1 - p_i^{on}$.

(a) Template image at aspect 1 of primitive which is facing to us



(b) Histogram of template (a) in L^*

(c) Histogram of template (a) in u^*

(d) Histogram of template (a) in v^*

(e) Histogram of template (a) in edge gradients

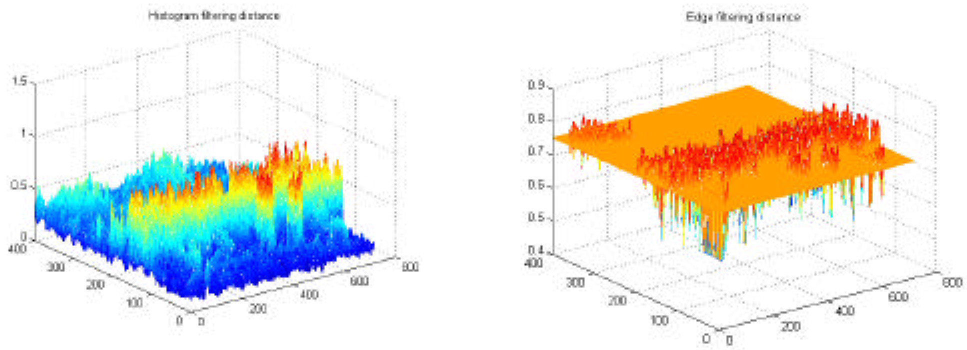
Fig. 21. One aspect template and its corresponding histograms in L^*, u^*, v^* and edge gradients.

(4) Three square windows that have different sizes, in other words, under different scales, are moved around the image. Let $W_1(s)$, $W_2(s)$ and $W_3(s)$ be three windows centered at pixel s. The histograms of each window centered at every pixel is computed as $H^{L^*}_{W_i(s')}(j) = \frac{1}{Z}\sum_{s\in W_i(s'),L^*(s)=j} 1$, $H^{u^*}_{W_i(s')}(j) = \frac{1}{Z}\sum_{s\in W_i(s'),u^*(s)=j} 1$ and $H^{v^*}_{W_i(s')}(j) = \frac{1}{Z}\sum_{s\in W_i(s'),v^*(s)=j} 1$. $H^{E}_{W_i(s)}(j) = \frac{1}{Z}\sum_{s\in I^E_{W_i(s)}} 1$ and $p_i^{on} = \frac{\sum_{s\in I^E_{W_i(s)}} 1}{\sum_{s\in W_i(s)} 1}$ are computed as (3).

(5) The histograms are actually probability distribution functions that describe the distributions of L^*, u^*, v^* and edge gradients. The overall measurements of the similarity between $H_i^{L^*}$ and $H^{L^*}_{W_i(s')}$, the similarity between $H_i^{u^*}$ and $H^{u^*}_{W_t(s')}$, and the the similarity between $H_i^{v^*}$ and $H^{v^*}_{W_t(s')}$ tell us how likely the template I_i appears at s' with size of W_t. Let the overall photometric similarity be $\eta(i,t) = \sqrt{D^{L^*}(i,t)^2 + D^{u^*}(i,t)^2 + D^{v^*}(i,t)^2}$ where the distance of two pdf functions, $D(i,t)$, could be computed as

$$D(i,t) = 1 - \| p_i \mid p_t \| = 1 - \sum_j \min(\, p_i(j), p_t(j)) \,. \tag{30}$$

The geometric similarity $\Delta(i,t) = D^E(i,t)$ is computed where distance of two pdf functions are obtained the same way as that stated in above.



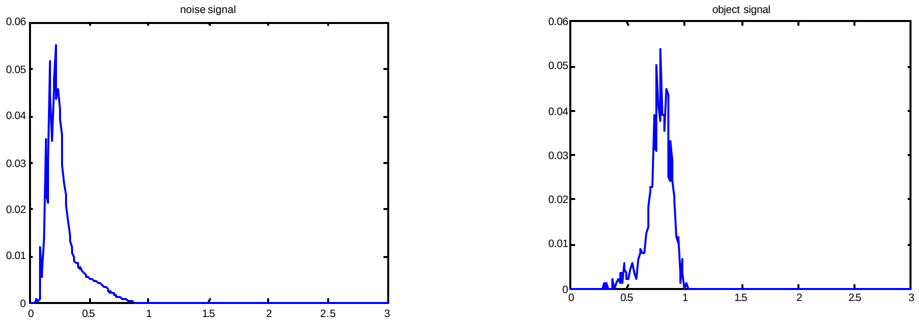
(a) Histogram similarity map in photometric invariants.

(b) Histogram similarity map in geometric invariants.

Fig. 22. Aspect template of Fig. 21 applied one image at window size 20x20.

(6) Given the above similarity map we still don't know what's the possible hot spots. The general approach is to set up a thresholding method. Every value that is larger than a fixed threshold would be set as 1 and every one that is smaller than it would be set as 0. How to set a proper threshold is difficult. Here we treat this kind of problem as a typical signal detection problem in which noise or signal is determined in terms of some crieterias using their probability distribution other than just thresholding. With this mehtod, the system could be trained with training data.

Several traffic lights that appear in one image known to have the same aspect as Fig. 21 are choosen as training samples. Same histogram similarity maps are obtained with same method applied. We pick up those pixels that appear in samples as object signal and all others are just noises so that we can have their pdf functions as shown in Fig. 23.



(a) pdf function of noise

(b) pdf function of object signal

Fig. 23. Pdf function of signal and noise.

Let p_0 be the prior probability of occurances of noise and p_1 be the prior probability of occurances of object signal. Given a histogram map and the pdf function of noise and the pdf function of object signal, we are going to tell which pixels in the map are the noises and which are the signals. Let C_{10} be the risk for each pixel that being the signal while assigned as noise; Let C_{00} be the risk for each pixel that being noise while assigned as noise; Let C_{01} be the risk for each pixel that being the noise while assigned as signal; Let C_{11} be the risk for each pixel that being the signal while assigned as signal;

The overall risk we have is thus

$$r(d) = R_0(d) + R_1(d)$$

where $R_j(d) = C_{1j}P_j(\Gamma_1) + C_{0j}P_j(\Gamma_0)$.
To minize the risk $R(d)$ we have

$$r(d) = \Pi_0 R_0(d) + \Pi_1 R_1(d) = \Pi_0 \int_{\Gamma_0} C_{00} p_0(y)dy + \Pi_0 \int_{\Gamma_1} C_{10} p_0(y)dy + \Pi_1 \int_{\Gamma_0} C_{01} p_1(y)dy + \Pi_0 \int_{\Gamma_1} C_{11} p_1(y)dy$$

$$= \int_{\Gamma_0} (\Pi_0 C_{00} p_0(y) + \Pi_1 C_{01} p_1(y))dy + \int_{\Gamma_1} \Pi_1 C_{10} p_0(y)dy + \Pi_0 C_{11} p_1(y). \qquad (31)$$

Apparently, we get the critrien as

$$\Gamma_1 = \{ y \in \Gamma \mid p_1(y) \geq tp_0(y) \} \qquad (32)$$

where $t \overset{\Delta}{=} \dfrac{p_0(C_{10} - C_{00})}{p_1(C_{01} - C_{11})}$.

With this method, it's straightforward to have the object signal map where the dark pixels mean object signal and bright pixels means noise. By cominbing these maps at different window sizes and different aspects together, we could have the final hot spot map.

This hot spot detection algorithm is robust at occlusion and illuminate. For a color image with 720X400 it just spends 2 minutes in a pentium III 500 PC machine which is much faster than segmentation methods while giving better results.

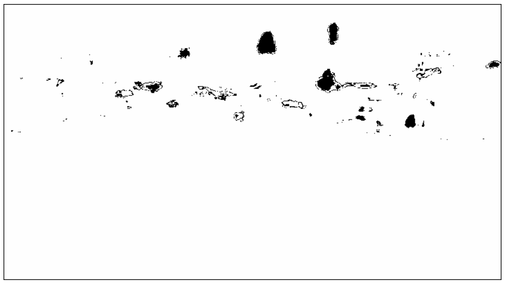


Fig. 24. Hot spot map

5.3.6 Traffic light recognition by MCMC in Top-down approach

Given the hot spot map as shown in Fig. 24, we may extract regions out of it. We have the assumption that occlusion won't happen then several rectangle pieces of image each of which encloses a connected hot spot region could be extracted. The size of every piece of image may be larger than its enclosed region because at this point we don't know what's the exactly position and size the traffic light would be.

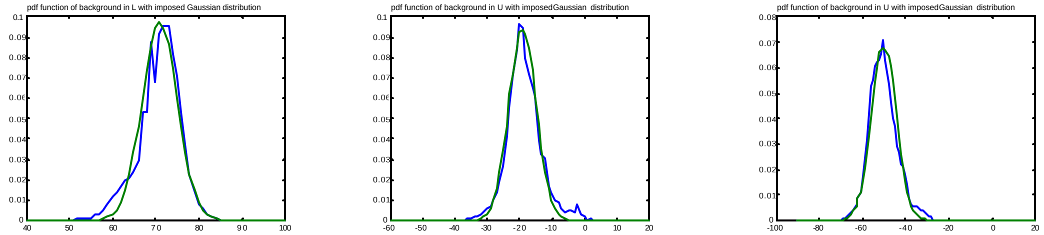
The remaining thing we should do is just to work on every piece of image trying to recognize traffic lights, may or may not appear, and their corresponding parameters.

Given every piece of image y and EOP and IOP parameters e, we want to find the x that maximize the posterior probability

$$p(x \mid y,e) = \frac{1}{Z(y,e)} p(x) L(y \mid x,e).$$

As our description in 5.3.1, we have $y \in \Upsilon = c \times \Re^3_{e_1} \times \Re^2_{e_2} + N$ where $\Re^3_{e_1}$ *and* $\Re^2_{e_2}$ are 3D transformation and 2D transformation respectively, c is the 3D scene in which only one traffic light is enclosed and N is superimposed noises.

Fig. 25 shows pdf of background color image in L^*, u^*, v^* with imposed Gaussian function approximation. We can see the nice fit of them which means background could be treated as Gaussian distributed noise. In more general cases Zhu and Mumford (1997) proposed a more general statistical description of background. We may use the same method to get the model by training through many image samples. In this case, traffic lights are generally hung against the sky and we just suppose a simple Gaussian distribution of background.



(a) pdf of background in L^* with imposed Gaussian distribution at s =4.1.

(b) pdf of background in u^* with imposed Gaussian distribution at s =4.2.

(c) pdf of background in v^* with imposed Gaussian distribution at s =5.8.

Fig. 25 pdf of background image

Suppose the parameter x is decomposed to

$$[t, c_s(R,G,B), c_r(R,G,B), c_y(R,G,B), c_g(R,G,B), (x_o, y_o, z_o), (w,h), (n,k,j)].$$

In Ullman and Basri (1991), the authors proved that the perspective projection of a 3D object, when viewed from some distance could be approximated with orthogonal projection. We also have the assumptions that $n = 0$ and $k = 0$ which are true in real scene. These requirements could be met in our cases reasonably and the parameters are simplified as

$$[t, c_s(R,G,B), c_r(R,G,B), c_y(R,G,B), \ c_g(R,G,B), (x_I, y_I), (w,h), j \ ]$$

where (x_I, y_I) are the 2D corrdinates of the center of traffic light in image piece. Let $F(x,e)$ be the orthogonal projection of a traffic light paramterized by x. Let $F_s^{L^*}(x,e)$, $F_s^{u^*}(x,e)$ and $F_s^{v^*}(x,e)$ be the L^*, u^*, v^* value at each pixel site s in $F(x)$. Let m^{L^*}, m^{u^*} and m^{v^*} be the average value in L^*, u^*, v^* of background.

The likelihood is

$$L(y\,|\,x,e) = \prod_{s\in F(x,e)} \prod_{c=L^*,u^* \text{ and } v^*} \left[\frac{1}{\sqrt{2p}\,s^{c}} \exp\left(-\frac{1}{2(s^{c})^2}(y_s^c - F_s^{\ c}(x,e))^2\right)\right]\times$$

$$\prod_{s\notin F(x,e)} \prod_{c=L^*,u^* \text{ and } v^*} \left[\frac{1}{\sqrt{2p}\,s^{c}} \exp\left(-\frac{1}{2(s^{c})^2}(y_s^c - m^{c})^2\right)\right]. \qquad (33)$$

The log likelihood becomes

$$\log(L(y\,|\,x,e)) = \sum_{s\in F(x,e)} \sum_{c=L^*,u^* \text{ and } v^*} -\frac{1}{2(s^{c})^2}(y_s^c - F_s^{\ c}(x,e))^2 + \sum_{s\notin F(x,e)} \sum_{c=L^*,u^* \text{ and } v^*} -\frac{1}{2(s^{c})^2}(y_s^c - F_s^{\ c}(x,e))^2 + g$$

where g is a constanct value which equals to $m \sum_{c=L^*,u^* \text{ and} v^*} \log(\frac{1}{\sqrt{2p}\,s^{c}})$ where m is the number of pixels in y.

The distribution that is propotationl posterior probability, $p(x\,|\,y,e)$, we are caring about becomes

$$p(x) = e^{(p(x)+\log(L(y|x,e)))/B} \qquad (34)$$

where B is the temperature. The introduction of B won't change the x^* that maxims the posterior probability because expotional function is montonl. We could see here that $p(x)$ is exactly the Gibbs distribution as in section 4. We may rewrite the above equation as

$$p(x) = e^{-H(x)/B} \qquad (35)$$

where

$$H(x) = -(p(x) + \log(L(y\,|\,x,e))) \qquad (36)$$

is the energy function.

Gilks et al. (1996), Winkler (1995) and Li (1996) discussed MCMC in image analysis. Apparently, it's impossible to search every value of x, which is in a huge space making search algorithm run forever. There are three ways of sampling x to find the solution to this MAP:

(1) Gibbs sampler, whose detailed description can be found in chaper 5, Winkler (1995)
(2) Steepest descent approach
(3) Metropolis sampler which can be seen in Gilks et al. (1996) and Winkler (1995).

Gibbs sampling algorithm runs too slow and is not efficient. Because different aspects of traffic light will give complete 2D images steepest descent algorithm doesn't have a nice surface whose second derivative, Hessian matrix, has all negative eigen values.

Metropolis sampler, specifically Metropolis-Hasting, is used here to find the solution to the MAP. The basic metropolis sampling method is in the following as Winkler (1995):

(1) A new configuration x_2 is proposed by sampling from a probability distribution $G(x_1,\cdot)$ on X where $G(x_1,\cdot)$ is called proposal matrix

(2) The energy at x_2 is computed and is compared with x_1

   (a) If $H(x_2) \leq H(x_1)$ then x_2 is accepted as the new setp

   (b) If $H(x_2) > H(x_1)$ then x_2 is accepted with the probability $\exp((H(x_1) - H(x_2))/B)$

   (c) If x_2 is not accepted then x_1 will be kept

The transformation matrix $p(x_1, x_2)$ becomes

$$p(x_1,x_2) = \begin{cases} G(x_1,x_2)\exp(-(H(x_2)-H(x_1))^+/B) & if\ x_1 \neq x_2 \\ 1-\sum_{z\in X\backslash\{x\}} p(x,z) & if\ x_1 = x_2 \end{cases}$$ where

$$(H(x_2)-H(x_1))^+ = \begin{cases} 0 & H(x_2)-H(x_1) \geq 0 \\ -(H(x_2)-H(x_1)) & H(x_2)-H(x_1) < 0 \end{cases}.$$

It could easily be proven that $p(x_1)p(x_1,x_2) = p(x_2)p(x_2,x_1)$, which meets the requirement of the convergence of Markov Chain.

A more efficient method in Metropolis algorithm is Metropolis-Hastings algorithm whose Markov transformation matrix can be denoted as

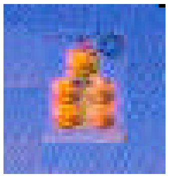
$$p(x_1,x_2) = \begin{cases} G(x_1,x_2)A(x_1,x_2) & if\ x_1 \neq x_2 \\ 1-\sum_{z\in X\backslash\{x_1\}} p(x_1,z) & if\ x_1 = x_2 \end{cases} \tag{37}$$

where

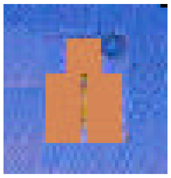
$$A(x_1,x_2) = \min\left\{1, \frac{p(x_2)G(x_2,x_1)}{p(x_1)G(x_1,x_2)}\right\}. \tag{38}$$

It is trival to prove the convergence of Markov random process, $p(x_1)p(x_1,x_2) = p(x_2)p(x_2,x_1)$.

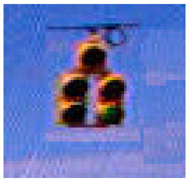
The important thing remaining is how to generate proposal matrix $G(x_1,x_2)$. As we stated before, traditional method like Generalized Hough Transformation votes for a solution x, may or may not be the solution to MAP, given image y. To combine the advantage of Generalized Hough Transformation, the speed, and the advantage of MCMC, perfect result, we choose the result of Generalized Hough Transformation as proposal matrix $G(x_1,x_2)$. The voting space of Generalized Hought Transformation actually gives a distribution of every possible parameters.



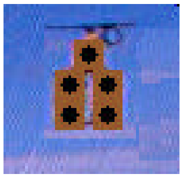
(a) image piece



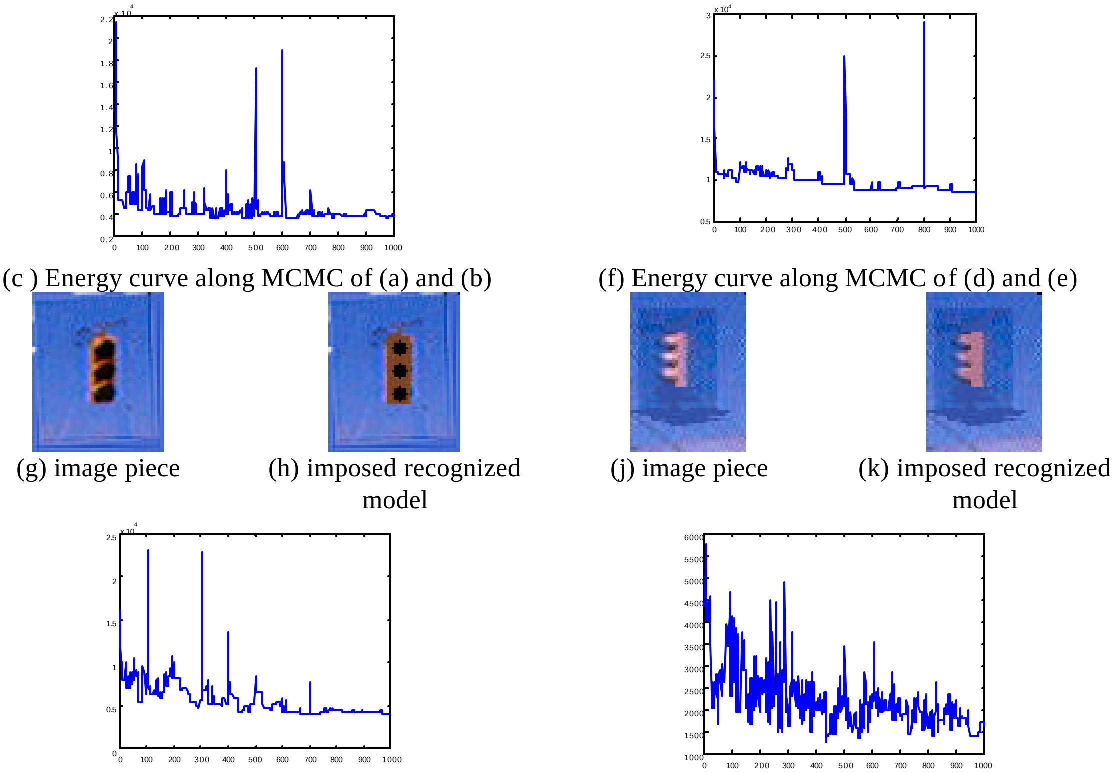
(b) imposed recognized model



(d) image piece



(e) imposed recognized model



(c) Energy curve along MCMC of (a) and (b)

(f) Energy curve along MCMC of (d) and (e)

(g) image piece

(h) imposed recognized model

(j) image piece

(k) imposed recognized model

(i) Energy curve along MCMC of (g) and (h)

(l) Energy curve along MCMC of (j) and (k)

Fig. 26. Original image pieces with the recognized traffic and the energy curve along MCMC simulations. We can see the nice matching between original image and imposed 3D object. In (c) it takes around 2 minutes to reach the final status. In (f) it takes one and a half minutes. It just takes less than one minute for (i) and (l) to reach the final steps.

6 Conclusions

In this article, the framework of a 3D object recognition is discussed. A new multilayer Hopfield Neural Network followed by a more general method Gibbs relaxation labeling in 3D invariants matching is proposed. Capturing the main heart of this framework, a novel method that integrates bottom-up and top-down is introduced. As to this idea, a real system that recognizes traffic lights in real image sequences is proposed. It takes fifteen minutes for the system to recognizes a color image with 720X400 starting from the low-level processing. The results we get are promising and show the great potential of using Markov Chain Monte Carlo method in recognizing 3D object in estimation problems. In this bottom-up and top-down by MCMC, we combine traditional method like indexing and Generalized Hough Transformation and show that they could be nicely integrated in random processes.

7 Reference

1. Amit, Y. and D. Geman, 1998, A Computational Model for Visual Selection, Department of Statistics University of Chicago, *Technical Report.*
2. Ballard, D.H., 1980, Generalizing the Hough Transform to Detect Arbitrary Shapes, *Matching, Model Fitting, Deduction and Information Integration*, pp. 714-725.

3. Bebis, G., G. Papadourakis and S. Orphanoudakis, 1999, Curvature Scale Space Driven Object Recognition with and Indexing Scheme based on Artifical Neural Networks, *Pattern Recognition*, Vol. 32, No. 7, pp. 1175-1201.
4. Bergevin, R. and M.D. Levine, 1993, Generic Object Recognition: Building and Matching Coarse Descriptions from Line Draws, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 15, No.1, pp. 19-36.
5. Brillault-O'Mahony, B., 1991, New Method for Vanishing Point Detection, *CVGIP: Image Uncerstanding*, Vol. 54, No. 2, pp. 289-300.
6. Bulthoff, H.H., Shimon Y.E. and Michael J.T., 1994, How are three-dimensional objects represented in the brain?, A.I. Memo No. 1479, *MIT*.
7. Canny, J., A computational Approach to Edge Detection, 1996, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 8, No. 6.
8. Carlotto, M.J., 1987, Histogram Analysis Using a Scale-Space Approach, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 9, No.1, pp. 121-129.
9. Cheng Y. 1995, Mean Shift, Mode Seeking, and Clustering, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 17, No.8, pp. 790-799.
10. Clemens, D., 1991, Region-Based Feature Interpretation for Recognizing 3D Models in 2D Images, PhD dissertation, AI-TR-1307, *MIT*.
11. Comaniciu, D. and P. Meer, 1997, Robust Analysis of Feature Spaces: Color Image Segmentation, *IEEE Conference on Computer Vision and Pattern Recognition*.
12. Coughlan J.M. and A.L. Yuille, 1999, Manhattan World: Compass Direction from a Single Image by Bayesian Inference.
13. Dickinson, S.J., A.P. Pentland and A. Rosenfeld, 1992, From Volumes to Views: An Approach to 3-D Object Recognition, *CVGIP: Image Understanding*, Vol. 55, No. 2, pp. 130-154.
14. Drew, M.S., J. Wei and Z.-N. Li, 1997, Illumination-Invariant Color Object Recognition via Compressed Chromaticity Histograms of Normalized Images, Technical Report, CMPT-TR 97-09, *Simon Fraster University School of Computing Science*.
15. Funt, B.V. and G.D. Finlayson, 1995, Color Constant Color Indexing, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 17, No.5, pp. 522-529.
16. Geman, S. and D. Geman, 1984, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. Signal Processing*, PAMI-6, No.6, pp. 721-741.
17. Gilks, W.R., S. Richardson and D.J. Spiegelhalter, 1996, *Markov Chain Monte Carlo in Practice*, Chapman & Hall.
18. Hopfield, J.J. and D.W. Tank, 1985, "Nerual" Computation of Decisions in Optimization Problems. *Biol. Cybern.* Vol. 52, pp.141-152.
19. Huang, C.-L., T.-Y. Cheng and C.-C. Chen, 1992, Color Image Segmentation Using Scale Space Filter and Markov Random Field, *Pattern Recognition*, Vol. 25, pp. 1217-1229.
20. Jacobs, D.W., 1992, Recognizing 3-D Objects Using 2-D Images, PhD dissertation, *MIT*.
21. Korn, M.R. and C.R. Dyer, 1987, 3-D Multiview Object Representations for Model-Based Object Recognition, *Pattern Recognition*, Vol. 20, No. 1, pp. 91-103.
22. Miller, M.I., U. Grenander, J.A. O'Sullivan and D.L. Snyder, 1997, Automatic Target Recognition Organized via Jump-Diffusion Algorithms, *IEEE Trans. Image Processing*, Vol. 6, No. 1, pp. 157-174.
23. Lee H.-C. and D.R. Cok, 1991, Detecting Boundaries in a Vector Field, *IEEE Trans. Signal Processing*, Vol. 39, No.5, pp. 1181-1194.
24. Lutton, E., H. Maitre, and J. Lopez-Krahe, 1994, Contribution to the Determination of Vanishing Points Using Hough Transform, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 16, No.4, pp. 430-438.
25. Lamdan, Y., J.T. Schwartz and H.J. Wolfson, 1990, Affine Invariant Model-Based Object Recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, No. 5, pp. 578-589.

26. Li, R. 1997. Mobile Mapping-An Emerging Technology for Spatial Data Acquisition. *Journal of Photogrammetric Engineering and Remote Sensing*, Vol. 63, No. 9, pp.1085-1092.
27. Li, S.Z., 1996, *Markov Random Field and Monte Carlo in Computer Visio*n, Springer Verlag.
28. Lin, W.-C., F.-Y. Liao, C.-K. Tsao and T. Lingutla, 1991, A Hierarchical Multiple-View Approach to Three-Dimensional Object Recognition. *IEEE Trans. on Neural Networks*, Vol. 2, No. 1, pp. 84-92.
29. McCane, J.B., 1996, Learning to Recognize 3D Objects from 2D Intensity Images, PhD dissertation, Department of Computer Science, *James Cook University of Norht Queenslan*d.
30. Miller, M.I., A. Srivastava and U. Grenander, 1995, Conditional-Mean Estimation Via Jump-Diffusion Processes in Multiple Target Tracking/Recognition, *IEEE Trans. Signal Processing*, Vol. 43, No.11, pp. 2678-2689.
31. Modestino, J.W. and J. Zhang, 1989. A Markov random filed model-based approach to image interpretation, *In Proceedings of the IEEE CVPR*, pp. 458-465.
32. Mokhtarian F. and A. Mackworth, 1986, Scale-Based Description and Recognition of Planar curves and Two-Dimensional Shapes, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 8, No.1, pp. 34-43.
33. Nagao, K. and W.E.L. Grimson, 1997, Using Photometric Invariants for 3D Object Recognition, *Computer Vision and Image Understanding*, Vol. 71, No. 1, pp. 74-93.
34. Panjwani, D.K. and G. Healey, 1995, Markov Random Field Models for Unsupervised Segmentation of Textured Color Images, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 17, No. 10, pp. 939-953.
35. Poggio, T. and S. Edelman, 1990, A network that learns to recognize 3D objects, *Nature*, 18th, 343, pp. 263-266.
36. Pontil, M. and A. Verri, 1998, Support Vector Machines for 3-D Object Recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 20, No. 6, pp. 637-646.
37. Poor, H.V., 1994, *An Introduction to Signal Detection and Estim*ation, Springer Verlag.
38. Rubner, Y., L. Guibas and C. Tomasi, 1997, Navigating through a Space of Color Image, *Stanford University*, Technical Report.
39. Salgian, G. and D.H. Ballard, 1998, Visual Routines for Autonomous Driving, *Proceedings of the 6-th ICCV*, pp. 876-882.
40. Seibert, M. and A.M. Waxman, 1992, Adaptive 3-D Object Recognition from Multiple Views, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 14, No. 2, pp. 107-124.
41. Shufelt, J.A., 1996, Projective Geometry and Photometry for Object Detection and Delineation, CMU-CS-96-164.
42. Slater D. and G. Healey, 1996, The Illumination-Invariant Recognition of 3D Objects Using Local Color Invariants, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, No.2, pp. 206-210.
43. Slater, D. and G. Healey, 1997, The Illumination-Invariant Matching of Deterministic Local Structure in Color Images, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No.10, pp. 1146-1151.
44. Stricker, M.A., 1992, Color and Geometry as Cues for Indexing, Department of Computer Science, *The University of Chicago*, Technical Report CS 92-22.
45. Suganthan, P.N., E.K. Teoh and D.P. Mital, 1995, Pattern Recognition by Homomorphic Graph Matching Using Hopfield Neural Network, *Image and Vision Comp.*, Vol. 13, No. 1, pp. 45-60.
46. Swain, M. and D. Ballard, 1991, Color indexing, International Journal of Computer Vision, Vol. 7, No. 1, pp. 11-32.
47. Tao, C., 1997, Automated Approach to Object Measurement and Feature Extraction from Georeferenced Mobile Mapping Image Sequences, PhD dissertation, Department of Geomatics Engineering, *The University of Calgary*.

48. Ullman S. and Basri R., 1991, Recognition by Linear Combinations of Models, 1991, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 10, pp. 992-1006.
49. Winkler, G., 1995, *Image Analysis, Random Fields and Dynamic Monte Carlo Met*hods, Springer Verlag.
50. Wong, A.K.C., S.W. Lu and M. Rioux, 1989, Recognition and Shape Synthesis of 3-D Objects Based on Attributed Hypergraphs, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11, No. 3, pp. 279-290.
51. Wyszecki, G. and W.S. Stiles, 1982, *Color Science: Concepts and Methods, Quantitative Data and Formulas*, Second Ed., New York: Wiley.
52. Young, S.S., P.D. Scott and M.N. Nasser, 1997, Object Recognition Using Multilayer Hopfield Neural Network, *IEEE Trans. on Image Processi*ng, Vol. 6, No. 3, pp. 357-371.
53. Yuille A. and T. Poggio, 1986, Scaling theorems for zero crossings*, IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 8, No.1, pp. 15-25.
54. Yuille, A.L., D. Snow and M. Nitzberg, 1998, Signfinder: Using Color to Detect, Localize and Identify Informational Signs, Technical Report, Smith-Kettlewell Eye Research Institute.
55. Zhu, S.C. and A. Yuille, 1996, Region Competition: Unifying Snakes, Region Growing and Bayes/MDL for Multi-band Image Segmentation, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, No.9, pp. 884-900.
56. Zhu, S.C. and D. Mumford, 1997, Prior Learning and Gibbs Reaction-Diffusion, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No.11, pp. 1236-1250.