Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

On-Road Computer Vision Based Obstacle Detection

Zheng-Tie Sun, Li-Chen, Fu, and Shin-Shinh, Huang

Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.
lichen@csie.ntu.edu.tw

Abstract

Applying computer technology to vehicle driving has been studied for many years. Many university labs, governmental organizations, and private companies in the world have spent a lot of efforts on the so-called ITS (Intelligent Transportation Systems). In this research field, obstacle detection plays an important role in assisting drivers with warning mechanism when some dangerous situations may happen. In this paper, we propose a fast method for detecting and tracking bikes, pedestrians, and vehicles in front of a moving vehicle.

In order to detect bikes and pedestrians efficiently, we apply a simplified fast stereovision method to estimate their approximate positions. On the other hand, we apply the so-called sign pattern technique to estimate the vehicle positions. After that, different methods are used to classify and confirm different kinds of obstacles for adapting their heterogeneity.

1. Introduction

Many countries have developed the safety for autonomous vehicle driving devotedly for decades. Those research groups combine hardware and software technologies to develop a certain mechanism for adding more safety to vehicle. Safety bags can be operated for instant lifesaving, and then a GPS (Global Positioning System) and mobile communication system can inform the traffic control center of those emergency conditions. Some sensors, like millimeter-wave radars, acoustic sensors, infrared images, and CCD (Charge-Coupled Device) cameras are applied for object and/or distance detection over the environment of the vehicles we are driving. This kind of sensing may be very useful when some potential collisions are going to take place, either due to the unawareness of the drivers or unreachable sight of some objects at a blind spot. In the proposed system of paper, we will focus on the latter.

Related Works

Tan *el al.* [1] developed a model-independent method for detecting the vehicles based on a GHT (Generalized Hough Transform)-like method. Willersinn *et al.* [2] used the optical flow method and clustered some similar optical flow vectors into a group for detecting and tracking of moving obstacles on the road. Matthews *et al.* [3] used the horizontal information especially pertaining to the vehicles to detect the approximate positions of the vehicles, and then applied some pattern classification technique for the sake of authenticity. In ARGO autonomous vehicle system [4], they constructed a symmetry map for detecting possible positions of vehicles, and then used the perspective constraints to generate the bounding box.

On the other hand, applying stereovision technologies have also been researched. Weber *et al.* [5] used the so-called Helmholtz Shear to extract the image regions above the road under flat ground plane assumption. Bertozzi and Broggi [6] used an inverse perspective transform to obtain the bird-view map from the image data, and the polar histograms of the difference of stereo bird-view maps are used to detect the vehicles.

In pedestrian recognition on the road, Broggi *et al.* [7] used a Shape-based method to detect pedestrian based on the symmetry map and applied the geometrical knowledge to erase background content.

Mori *et al.* [8] performed a sign pattern-based method for recognizing pedestrian by the rhythm of walking. Curio *et al.* [9] performed a similar method but also used a human walking model. Papageorgiou *et al.* [10] used the so-called wavelet templates to store the features of the pedestrians, and then applied the SVM (Support Vector Machine) technique to the training of those features so as to recognize pedestrians. Some similar methods based on SVM can be found in [11]. Gavrila and Philomin [12] used a distance transform based matching to detect pedestrians and to recognize road signs. In our

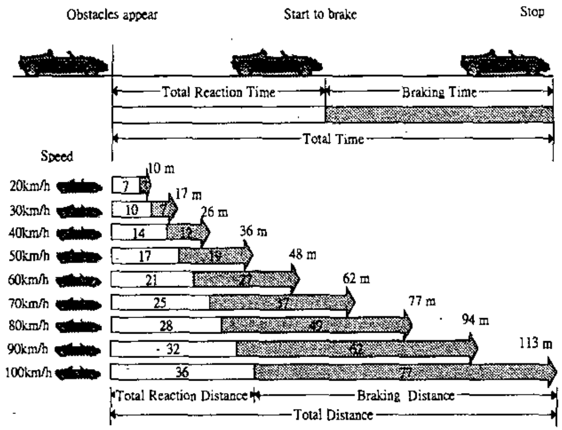


Figure 2-1: Total stopping distances

previous work, Liu [13] used a simplified stereovision method to detect positions of potential obstacles and applied M-estimated Hausdorff distance transform based hierarchical template matching procedure.

2. Preliminary

The obstacles on the ground we concern about are pedestrian, vehicle, and bike. The entire system in the paper can be defined as follows:At time t, the system obtains information of the traffic condition $Info(t)$, and then must answer the following four problems repeatedly: 1) Existence Problem: Is there any possible obstacle? 2) Verification and Classification Problem: Is there any pedestrian, vehicle, or bike in the group of detected possible obstacles? 3) Location Problem: Where are those obstacles? 4) Warning Decision Problem: In such a situation, should the system warn the driver to pay more attention to the current traffic scene at time $t+1$?

When a safety warning system is applied, a **Total Stopping Distance** for confirming the warning signal can be defined as below.

$$V \cdot (warning \quad system \ + 2 \times (perception \quad time) + reaction \quad time + V / (2 g \cdot f))$$

where g is the acceleration of gravity $9.8\ m/s^2$, f is friction coefficient, and V is the vehicle speed before braking.

For a dry ground, the friction coefficient is about 0.8, and for a wet ground, the friction coefficient is about 0.5. Here, we set warning system to 200*ms* (5 frame/s), friction coefficient to 0.5, perception time to 400*ms*, and reaction time to 300*ms* for a fast-reaction driver, and the total stopping distance with respect to various vehicle speeds can be summarized in Figure 2-1.

3. Obstacle Detection

Potential Obstacles for Pedestrian and Bike

For pedestrians and bikes, we use a simplified stereovision method to detect those obstacles efficiently, the so-called structure classification [14]. In the proposed system, two cameras are placed along the horizontal lines (see Figure 3-1) and we use the color information to do more precise structure classification. It can be expressed as follows:

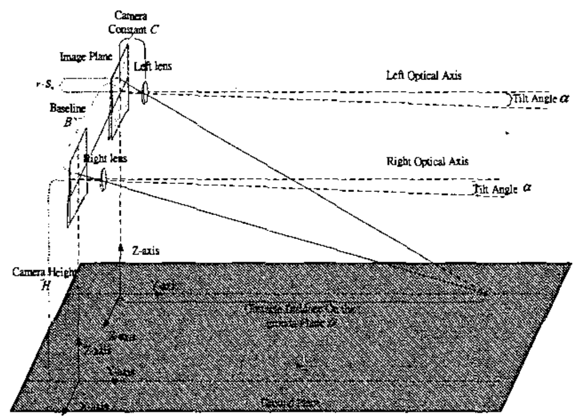


Figure 3-1 : Camera placement

$$g(i,x) = c_k(i) - c_l(x)$$

$$d(i) = \begin{cases} (|g(i,x)| + s/2)/s + d & \text{if } g(i,x) > 0 \\ (|g(i,x)| + s/2)/s & \text{if } g(i,x) < 0 \\ 0 & \text{else} \end{cases}$$

	0	
3	x	1
	2	

D(3)	D(2)	d(1)	d(0)

Class Index

***Figure 3-2:** Encoding scheme of the structure classification based on a 4-pixel neighborhood*

where k and l represent different color component R, G, or B, and c is the color value of that pixel, and s is a number for dividing the color range and the plus with $s/2$ is to round up or round down the result with respect to the number s. Lastly, the number d is added to distinguish the result whether the difference between the neighbor pixel i and the center pixel x is greater than zero or less than zero. Finally, we set the class index: $Class \quad Index = \sum_{i=0}^{3} 8^i \cdot d(i)$

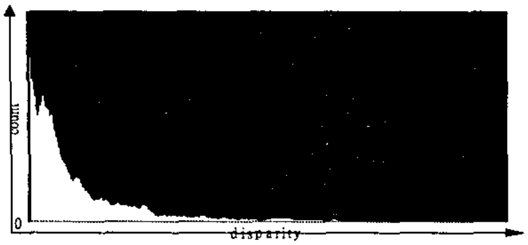


Figure 3-3: Disparity Histogram

For solving correspondence problem faster and more precisely, we only choose the edge points of an image as feature points and match them against edge feature points with the same class index of another image in the horizontal line. Therefore, the depth map will be constructed by calculating the distance between the correspondent points.

After constructing a sparse depth map, the disparity histogram is built from this map as shown in the following Figure 3-3. We can check the disparity value in the peak position to detect the obstacles. For pedestrian and bike detection, the potential obstacles for next verification step is generated here by checking all peaks in the disparity histogram.

In the paper, we select the *M-HD* to be the similarity measure for template matching. *M-HD* replaces the Euclidean distance by the cost function that can eliminate outliers. The cost function ρ is defined as:

$$\rho(x) = \begin{cases} |x|, & |x| \le \tau \\ \tau, & |x| > \tau \end{cases}$$

where τ is a threshold value.

And the directed Hausdorff distance based on M-estimation is defined as:

$$h_{M-HD}(A,B) = \frac{1}{N_A}\sum_{a \in A} \rho(D(a,B)),$$

N_A denotes the number of points of set A

Bike Detection Procedure

After getting the potential obstacles for bikes and pedestrians, we first classify them through the bike detection procedure. If they are not bikes, the pedestrian classification procedure will be continued to verify.

The detection for bike is based on *M-HD* template matching. Because the bikes are semi-rigid bodies, there are some methods that can reduce the number of comparisons:

1) The symmetry property: The bike is heading the same direction as or the opposite direction against that of our vehicle. Then, we always see the near-front or near-back side of bikes.
2) Templates property: Because of the property of semi-rigid body and fixed sitting appearance of human rider, we only need to choose some meaningful templates for template matching.

Pedestrian Detection & Template Voting Procedure

The performance of the template matching procedure depends on the number of matches in searching the best result and the accuracy for finding the best result. In the paper, we propose a different approach to resolve this problem fast, called TVP (Template Voting Procedure) The TVP has six steps as described in the following:

1) Establish the degree of similarity between templates by using the closeness of matching scores:

   Let the matching score between *A* and *B* templates be *M(A,B)*, and let the matching score between *A* and *C* templates be *M(A,C)*. Here, we make an assumption that the templates in the database are abundant so that *B* is similar to *C*, if *M(A,B)* is very close to *M(A,C)*.
2) Select a matching method among templates:

   We use Hausdorff distance to provide the matching scores. Because the distance measure becomes the matching score, the similarity is higher while the matching score is smaller.
3) Construct the matching score sorting queue for TVP:

   We give each template an index, from *1* to *n*, and select a template *i* to compare with other templates by the proposed matching method. Here, *S(i,1)* represents the matching score between template *i* and template *1*. The quick sort is applied to sort those matching scores *S(i,1)*, *S(i,2)*, ..., *S(i,n)* from low score to high score. We name this sorted queue as *Sort_Queue(i)*. Finally, we generate the score queue for all templates *Sort_Queue(i)*, *i=1~n*.
4) Voting procedure in TVP:

   After the construction of *Sort_Queue(i)*, *i=1~n*, we select some reference templates as the root of the overall comparison procedure. For explanation, we use an example with small number of templates, illustrated by Figure 3-5. There are totally eight templates in the database, and three reference templates *r1*, *r2*, and *r3* from those eight templates. When an image *t* for comparison comes, we generate the scores *S(r1,t)*, *S(r2,t)*, and *S(r3,t)* by the predefined matching function, and those scores are used as the similarity checking criteria. For reference template *r1*, we choose three candidates that are the closest scores to *S(r1,t)* in the score sorting queue *Sort_Queue(r1)*, say, *S(r1,1)*, *S(r1,4)*, and

$S(r1,6)$. Those templates form the set called $Possible\_Set(r1,t) = \{1,4,6\}$. The template sets $Possible\_Set(r2,t) = \{2,6,7\}$ and $Possible\_Set(r3,t) = \{6,8,7\}$ are also generated.

Finally, we use those template sets to construct a histogram $Candidate\_Histogram(t)$ illustrated by Figure 3-6. The template in peak position of this histogram is the most similar to the incoming image t.

In this procedure, we reduce the original number of matches dramatically from eight times, matching against all the templates, to four times, three with the reference templates, and one with the template in peak position of the histogram, to find out the most similar template in the template database.

5) Different votes for different candidates:

After the preceding example, the approach with fixed votes for each template was described, that is, each candidate gets a vote. Here, we want to give different votes for different candidates according to the reference templates. If there is a matching score $S(i,j)$ smaller than $S(k,j)$, which means i is more similar to j than k is. Thus, template j should obtain more vote relative to $S(i,j)$, since the higher the matching score is, the higher the variability is.

6) The selection of the reference templates:

A good TVP depends on the number of the reference templates and the closest candidates in the voting procedure. In the paper, we want to maximize the spans among each element in the score sorting queue:

The goal is to keep certain span among each template element in the score sorting queue:

$$Sort\_Queue(i) \text{ has elements from } 1 \text{ to } n$$
$$S'(i,j) = \text{the } j_{th} \text{ element in } Sort\_Queue(i), 1 \le j \le n$$
$$DSum(i) = \max_{i=1\sim n} \sum_{k=2}^{n} \rho(S'(i,k)\text{-}S'(i,k\text{-}1))$$
$$\rho(x) = \begin{cases} d & \text{if } x > d \\ x & \text{else} \end{cases}$$
$$d \text{ is a predefined difference threshold}$$

ρ is used to reduce the dominance of some very large distance scores. And we want to maximize $DSum$ to choose the suitable reference templates. There is also another method to reach this function. That is to keep the minimum variance of the distance scores in the sorting queue, but also to keep the maximum value of $DSum$ without constraint ρ.

$$DSum(i) = \max_{i=1\sim n} \sum_{k=2}^{n} (S'(i,k)\text{-}S'(i,k\text{-}1))$$
$$DMean(i) = \frac{1}{n-1}\sum_{2}^{n} (S'(i,k)\text{-}S'(i,k\text{-}1))$$
$$DVariance(i) = \frac{1}{n-1}\sum_{2}^{n} (S'(i,k)\text{-}S'(i,k\text{-}1) - DMean(i))^2$$

After the above steps, we can be sure that every element in a score-sorting queue keeps a certain distance from the neighbor elements. It will make the candidates more meaningful under some measure distances.

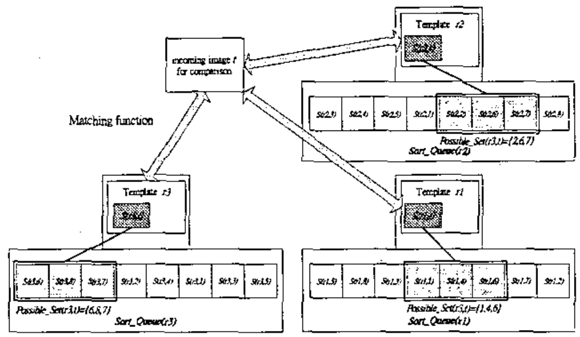


***Figure 3-4:** Possible candidates for image t by some reference templates*

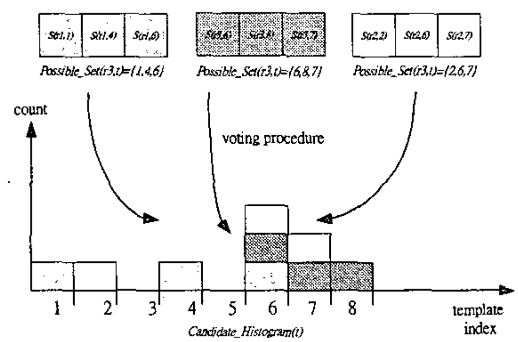


***Figure 3-5:** Candidate histogram of TVP*

Then, we apply TVP to match those image regions with potential obstacles against the pedestrian templates in the pedestrian template database. Unlike bikes and vehicles, pedestrians are non-rigid bodies so we need an abundant template database to meet the matching requirement. And, TVP can assist to speed up the matching procedure.

Vehicle Detection Procedure

For vehicles, we make use of the horizontal edges [3] and SP (Sign Pattern)[8] of the vehicles to estimate the possible vehicle existences. Different from bikes and pedestrians, vehicles are rigid bodies. The shapes of them are simpler and fixed. Although the Hausdorff distance can also be used for vehicle matching, we apply an easier and fast method based on SP for the vehicle, and it denotes darker underneath. Moreover, the spare depth map seems to be not enough to construct a complete position for vehicles. So, we find the potential vehicles by finding the horizontal edges from near to far (bottom to up in the image). There are five steps to detect the vehicle.

1) Generating the horizontal edges.
2) Finding horizontal edges with enough length.
3) Verifying those potential vehicles with SP.
4) Verifying those potential vehicles with the existence of enough edges.
5) Verifying those potential vehicles with the symmetry.d

4. Obstacle Tracking

The tracking step plays an important role in detecting any object, because it can reduce the cost to searching the objects. Because pedestrians are non-rigid bodies, they cannot be verified in the next location by cross-correlation based methods. Instead, we use the color information to accommodate the flexibility of the pedestrians; a so-called color indexing [15] methods for checking the similarity between two objects are in different time frames.

Unlike the previous method that RGB colors are used as features, we first transformed from RGB into YUV color formation, where Y denotes illumination vector and UV denote the chromatism vectors for RB, respectively, and only use UV as features to construct the histogram to remove the effect of illumination:

$$Color\_Score = \frac{\sum_u \sum_v \min(H_1(u,v), H_2(u,v))}{\min(|H_1|, |H_2|)}$$

We use the extrapolation method for to implement tracking. In the initialization of tracking procedure, the combination of two obstacles in different time frames will be performed by their types, their positions, and color indexing matching. After the combination, the previous obstacle will be discarded and the current obstacle will be reserved. The reserved one will tag what is tracked, and the predefined tracking model will be used to calculate the velocities.

5. Experiment

The binocular images are grabbed from the two SONY DCR PC-100 digital video cameras (DV), through Macally IEEE1394 PCMCIA card. The system runs on a notebook with an Intel Pentium III 650MHz CPU and 256MB RAM.

There are totally 1124 shape images stored for pedestrian matching. We only store 10 meaningful templates for bike matching. The height of the template is scaled to 120 pixels, and then the size of the template is normalized to 120x120 pixels in the database of our system. In our experiment, we use two different focal lengths 9.16*mm* and 4.2 *mm* to adapt to different vehicle speeds.

In Figure 5-1, (A) denotes the successful detection as two bikes that are with enough symmetry property, but the left detection window of (B) denotes the unsuccessful bike detection due to lack of symmetry. It is actually detected as a pedestrian.

In order to choose good TVP matching parameters for our pedestrian database which contains 1124 templates with the # of reference number equal to 3 and the # of candidate template being 400 as the parameter settings. With this setting, the average # of total matches for any an incoming image can be lowered down to 9. In Figure 5-2, there are some illustrations for pedestrian detection.

For vehicle detection, we test the proposed method for different lighting conditions as shown in Figure 5-3. In tracking, the $C_{tracking}$ is the interval to apply tracking procedure. When it increases infinitely, the system may be slower than the non-tracking system with frequent tracking failures. If tracking is quite correct, the detection rate will be better than the non-tracking system. We show the result in Table 5-1.

6. Conclusion

In this paper, a fast stereovision method is applied to extract the potential obstacles in front of the vehicle. Color information is mixed into the structure classification method to enhance the accuracy of the stereo reconstruction. Although this method construct sparse depth map, the presence of the obstacles in front of vehicles with significant image size in the scene can be obvious via this map. Therefore, we can extract the approximate positions of the obstacles in the image.

For detecting pedestrians and bikes, we choose M-estimation Hausdroff distance as our similarity measure. It uses the shape of an object as the matching feature, and provides a smoother matching score to adapt to the variability of pedestrians and bikes. A template voting procedure (TVP) is proposed in this paper. This procedure can also be applied with different template matching methods to minimize the # of total matches. In our experiment, the # of comparison is decreased from 1124 down to 9 in average. The more the abundant templates are in the database, the more precise the matching is. Also, the effect of TVP is more obvious to decrease the # of total comparisons.

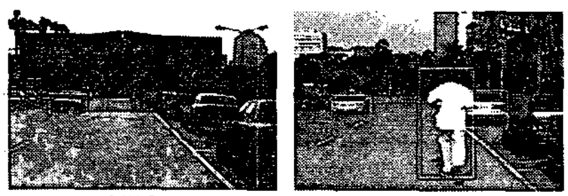
The M-HD based template matching is based on the shape of the image. Therefore, choosing a good

(A) Bike Detection (B) Detect Bike as Pedestrian

Figure 5-1: *Bike Detection*

(A) Single Pedestrian Detection (B) Multiple Pedestrian Detection

Figure 5-2: *Pedestrian detection*



(A) Vehicle Detection (B) Obstacle Detection

Figure 5-3: *Vehicle detection*

Table 5-1: *System Time and Detection Rate*.

$C_{tracking}$	Non-tracking System Time (fps)	Tracking System Time (fps)	Non-tracking - Detection Rate	Tracking - Detection Rate
5	4.5 ~ 7.7	5.4 ~ 8.6	85% / 84%	84% / 83%
4		5.9 ~ 9.2		82% / 82%
3		6.5 ~ 9.7		78% / 80%
2		6.7 ~ 10.3		72% / 75%

(Detection rate for *bike and pedestrian / vehicle*, fps denotes "frame per seconds")

threshold to distinguish edge from rough edge parts under different weather and illumination is very critical. Some image segmentation and analysis techniques can be added to increase the correctness of the shape edge. But those steps may become another computational burden to the whole system. Therefore, it is a trade-off for the shape matching methods.

Reference

[1] T. N. Tan, K. D. Baker, and G. D. Sullivan, "Model-independent recovery of object orientations," *Robotics and Automation, IEEE Transactions on,* Vol. 13 4 , Aug. 1997, pp. 602-606.

[2] D. Willersinn and W. Enkelmann, "Robust obstacle detection and tracking by motion analysis," *Intelligent Transportation System, 1997. ITSC '97, IEEE Conference on*, 1997, pp. 717-722.

[3] N. D. Matthews, P. E. An, and C. J. Harris, "Vehicle detection and recognition for autonomous intelligent cruise control," *Image, Speech and Intelligent Systems*, 1995/6 Research Journal.

[4] *http://millemiglia.ce.unipr.it/ARGO/english/techdet/vd.html*, Vehicle Detection in ARGO System

[5] Q.-T. Luong, J. Weber, D. Koller, and J. Malik, "An integrated stereo-based approach to automatic vehicle guidance," *Computer Vision, 1995. Proceedings, Fifth International Conference on,* 1995, pp. 52-57.

[6] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, Vol. 7 1, Jan. 1998, pp. 62-81.

[7] T. N. Tan, G. D. Sullivan, and K. D. Baker, "Efficient image gradient-based object localisation and recognition," *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, 1996, pp. 397-402.

[8] H. Mori, N. M. Charkari, and T. Matsushita, "On-line vehicle and pedestrian detections based on sign pattern," *Industrial Electronics, IEEE Transactions on*, Vol. 41 4, Aug. 1994, pp. 384-391.

[9] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. v. Seelen, "Walking pedestrian recognition," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 1 3 , Sept. 2000, pp. 155-163.

[10] C. Papageorgiou, T. Poggio, "Trainable Pedestrian Detection," *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on,* Vol. 4, 1999, pp. 35 -39 vol.4.

[11] A. Mohan, "Object detection in images by components," *A.I. Memo 1664, Center for Biological and Computational Learning, M.I.T., Cambridge, MA*, 1999.

[12] D. M. Gavrila, and V. Philomin, "Real-time object detection for "smart" vehicles," *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on,* Vol. 1, 1999, pp. 87 -93 vol.1.

[13] Liu, C.-Y. and L.-C. Fu, "Computer Vision Based Movable Obstacle Detection and Recognition for Vehicle Driving," *Department of Computer Science and Information Engineering, National Taiwan University, Master Paper.*

[14] U. Franke and I. Kutzbach, "Fast stereo based object detection for Stop&Go traffic," *Intelligent Vehicles Symposium, 1996, Proceedings of the 1996 IEEE, 1996*, pp. 339-344

[15] M. J. Swain and D. H. Ballard, "Color Indexing," *Int. Journal of Computer Vision*, 7(1):11-32,1991.