

Feature Point Detection for Real Time Applications

Neeta Nain, Vijay Laxmi and Bhavitavya Bhadviya *

Abstract— This paper presents a new feature point detector that is accurate, efficient and fast. A detailed qualitative evaluation of the proposed feature point detector for grayscale images is also done. Experiments have proved that this feature point detector is robust to affine transformations, noise and perspective deformations. More over the proposed detector requires only 22 additions per pixel to evaluate the interest point and its strength, making it one of the fastest detectors. The accuracy and speed of this algorithm makes it a strong contender for ASIC and hardware implementations and applications requiring real time feature point abstraction.

Keywords: Gray level, edge detection, corner detection, Real Time Application

1 Introduction

Point Correspondence between two or more images is crucial component for many computer vision and image analysis tasks. Most methods for 3D reconstruction, object detection and recognition, image alignment and matching and camera calibration techniques assume that feature points were extracted and put to reliable correspondence. In general, all possible pairs of points should be examined to solve this correspondence problem, and this is computationally very expensive. If two images have n pixels each, the complexity is $O(n^2)$. This process might be simplified if the correspondence is examined among a much smaller number of points, called **interest points**. Interest points are locations in the image where the signal changes two-dimensionally. Examples include corners and T -junctions as well as locations where the texture varies significantly. Of the most intuitive type of features, corners are very critical because they are invariant to rotation and little changes can be observed under different lighting. Corners also minimize the amount of data to be processed without losing the most important information of the gray level image. Corner detection is used as the first step of many vision tasks such as tracking, simultaneous localization and mapping and recognition. Corners in images can be located using local detectors; input to the corner detector is the gray-level image, and output

is the image $f(i, j)$ in which values are proportional to the likelihood that the pixel is a corner. A large number of corner detectors exist in the literature. They can be divided mainly into two categories: template based and geometry based corner detectors. Template based corner detection is to build a set of corner templates and determine the similarity between the templates and all the sub windows of the gray level image. The earliest geometry based corner detectors, first extract the boundary as a chain code, and then search for significant turnings at boundary. These kind of corner detectors also suffer from high algorithm complexity, as multiple steps are needed. Additionally any errors in the segmentation step will lead to different corner results. Afterwards, many geometry based corner detectors, which directly operate on the gray level image were introduced. The corner detector proposed in this work belongs to this last category. Also the terms interest points and feature points refer to the same concept and hence would be used interchangeably.

2 Literature Survey of Corner Detectors

The majority of grayscale feature detection algorithms work by computing a corner response function (C) across the image. Pixels which exceed a threshold (locally maximal) are then chosen. The simplest corner detector is the **Moravec detector** [9] which computes the sum-of-squared-difference (SSD) between a patch around a candidate corner and patches shifted a small distance in a number of directions. The Moravec operator's SSD is given by

$$SSD(i, j) = \frac{1}{8} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} |f(k, l) - f(i, j)| \quad (1)$$

C is then the smallest SSD , thus ensuring that extracted corners are those locations which change maximally under translations (on corners and sharp edges). Harris [5] builds on this by computing an approximation to the second derivative of SAD with respect to the shift. They modified the Moravec method to the famous Plessey corner detector by estimating the auto correlation from the first order derivatives.

Brown and Lowe [6, 2] obtained the scale invariance by convolving the image with a Difference of Gaussian (DoG)

*Department of Computer Engineering, Malaviya National Institute of Technology Jaipur, Jaipur, India - 302017. Tel: +91 141 2529140 Email: neetanain@yahoo.com, vlaxmi@yahoo.com, bhavitavya_bhadviya@yahoo.co.in

kernel at multiple scales, retaining locations which are optimal in scale as well as space. *DoG* is used because it is good approximation for the Laplacian of Gaussian(*LoG*) and much faster to compute. An approximation to *DoG* has been proposed which, provided that scales are $\sqrt{2}$ apart, speeds up computation by a factor of about two, compared to the straightforward implementation of Gaussian convolution.

Smith and Brady proposed a method called SUSAN [11] for feature point detection based on simple masking operations instead of gradient convolution. It uses a circular window consisting of 37 pixels to obtain the USAN area whose centroid is estimated to enhance the localization of the corner. The results showed a very good localization and noise robustness but they involved high computational cost due to a large window size for obtaining the USAN area and centroid. Trajkovic and Hedley used a similar idea in [13]: a patch is not self-similar if pixels generally look different from the center of the patch. This is measured by considering a circle and then comparing the local intensity changes in all directions with respect to the center. Although the method is fast and efficient, it is highly sensitive to noise and straight lines sometimes. Rosten and Drummond [10] proposed a criterion which operates by considering that an edge is a boundary between two regions and a corner occurs where the edge changes directly suddenly. A point is a corner if enough of the pixels around the point are in a different region from the point. It implements this by considering a circle around the candidate point, p . It looks for the largest arc where the intensities of all the points on the arc are above the intensity of $p(I_p)$ by some threshold, t , or the intensity of all points on the arc are below I_p by t . The point is a corner if $\theta \geq \theta_t$, where θ is the angle of the arc and θ_t is some threshold. Although this feature point detector is extremely fast it is not robust to high level noise and is also susceptible to 1-pixel thick straight lines. Mokhtarian and Suomela [8] described a method for image corner detection based on the curvature scale space(*CSS*) representation. This technique ECSS is suitable for recovering invariant geometric features also called curvature zero crossing points or extrema of a planar curve at multiple scales. This algorithm is highly accurate and robust to noise but has a very high time complexity. In [12] two oriented cross operators called COP(Crosses as Oriented Pairs) were used to detect corners and feature points. This technique is extremely fast and accurate but is unable to handle noise and image deformity.

3 Feature Point Detection Model

The proposed feature point detector is based on the fact that feature points are nothing but the sudden changes in 2 dimensions. The aim of the proposed technique is to enable detection of corners and all other points of interest in an image whose information content is greater

than the desired threshold. With the aim of satisfying the universal criterion of localization, consistency, accuracy and noise immunity in a real time environment we propose a short but effective five step algorithm. The algorithm has been designed to be a ready contender for hardware implementation in fields of Robotic vision and other image processing ASICs involved in feature detection and matching. For each pixel of the image under consideration:

1. Apply the Difference Mask with threshold parameter P_1 .
2. Apply the Partial Averaged Gaussian Mask to the points satisfying 1.
3. Once again calculate the Difference Mask with threshold parameter P_2 .
4. Eliminate False Positives.
5. Determine Localization of the Remaining feature points.

3.1 Apply the Difference Mask with threshold parameter P_1

Unlike the usual convolution masks of sizes varying from 3x3 to 7x7, we propose the use of very simple 2x2 mask. Moreover we replace the convolution with simple differences between the pixel intensity values and then comparing these differences with the threshold P_1 . The use

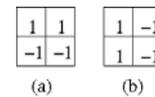


Figure 1: The 2x2 convolution kernel

of the masks as shown in figure 1 gives rise to four orthogonal difference values. The 4 orthogonal difference values of H_1 , H_2 , V_1 and V_2 as shown in Figure 2 are the four critical gradient values that responds positively to corners and diagonal edges. Each of this is defined as follows:

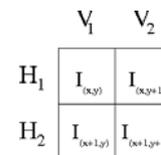


Figure 2: The 2x2 Operator

$$H_1 = | I_{(i,j)} - I_{(i,j+1)} | \tag{2}$$

$$H_2 = | I_{(i+1,j)} - I_{(i+1,j+1)} | \tag{3}$$

$$V_1 = | I_{(i,j)} - I_{(i+1,j)} | \tag{4}$$

$$V_2 = | I_{(i,j+1)} - I_{(i+1,j+1)} | \tag{5}$$

Here (i, j) is the current pixel under consideration. Further the standard responses of this operator on a set of standard edges and corner is shown in Figure 3, 4, 5 and 6. Further the response of this 2x2 operator can be grouped into three categories:

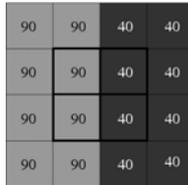


Figure 3: Figure showing the response of the Difference mask on a typical vertical edge. $H_1=H_2=90 - 40=50$ and $V_1=V_2=0$

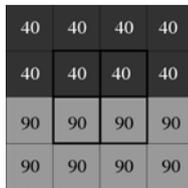


Figure 4: Figure showing the response of the Difference mask on a typical horizontal edge. $H_1=H_2=0$ and $V_1=V_2=90 - 40=50$

1. A vertical edge. (H_1 and H_2 both greater than zero)
2. A horizontal edge. (V_1 and V_2 both greater than zero)
3. Diagonal edges and interest point candidates. (Atleast one of the horizontal differences and vertical differences are non zero).

Since, for feature point detection we are only interested in the 3rd category of responses we will henceforth concentrate on these responses only. We also considered the responses of our mask on some blurred or gradual diagonal edges as shown in figure 5. A single response was generated when the threshold parameter is chosen suitably. Each pixel satisfying the third criteria is considered for further processing. The absolute sum of H_1 , H_2 , V_1 and V_2 gives the strength of cornerity at that pixel.

3.2 Apply the Partial Averaged Gaussian Mask to the Points Satisfying Step 1

In real case scenario we will never come across images with well defined edges or corners. Not only the feature points will be blurred but the image itself will consist of several types of noises. All this makes the Difference mask in the previous subsection respond to every noisy pixel. In order to overcome this and increase the noise immunity of our algorithm we propose the application of

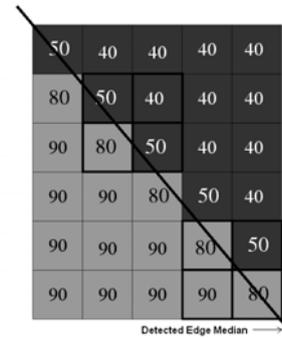


Figure 5: Figure showing a typical diagonal edge. The bold line across the image shows the position of diagonal edge when $P_1=20$

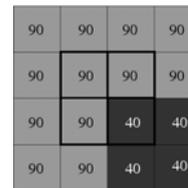


Figure 6: Figure showing a typical corner candidate. Here $H_1=0$, $H_2=90 - 40=50$, $V_1 = 0$ and $V_2 = 90 - 40 = 50$

a pseudo-gaussian kernel.

Usually A gaussian kernel requires the convolution of the image with a complex matrix generated by (Standard Gaussian Function) and a varying σ . In view of constrained time and performance intensive application scenario of real time feature point detection we propose to use a partial averaging kernel. This kernel was derived from a 5x5 gaussian kernel having $\sigma=1.3$. This was further modified and normalized so as to have all factors some exponent of 2. This results in just a shift of the bits during calculations and no expensive operations like multiplication or division are avoided.

Further we take advantage of our difference kernel and apply only a partial of the entire gaussian kernel. Then similar to a rotating averaging mask, this is rotated by 90° and applied to each pixel of the difference mask. This partial implementation as shown in figure 7 reduces the gaussian averaging overhead by 75% but produces the desired results on the set of 2x2 pixels under consideration.

3.3 Calculate the Difference Mask with threshold parameter P_2

Once the new gaussian averaged values of the four pixels under consideration are calculated, we once again apply the difference mask. But this time we use a different threshold P_2 . The use of this second thresholding parameter is encouraged so as to avoid missing weak feature points in presence of noise. One can look at this in a

		8	8		
		$I_{(i-2,j)}$	$I_{(i-2,j+1)}$		
	16	16	16	16	
	$I_{(i-1,j-1)}$	$I_{(i-1,j)}$	$I_{(i-1,j+1)}$	$I_{(i-1,j+2)}$	
8	16	64	64	16	8
$I_{(i,j-2)}$	$I_{(i,j-1)}$	$I_{(i,j)}$	$I_{(i,j+1)}$	$I_{(i,j+2)}$	$I_{(i,j+3)}$
8	16	64	64	16	8
$I_{(i+1,j-2)}$	$I_{(i+1,j-1)}$	$I_{(i+1,j)}$	$I_{(i+1,j+1)}$	$I_{(i+1,j+2)}$	$I_{(i+1,j+3)}$
	16	16	16	16	
	$I_{(i+2,j-1)}$	$I_{(i+2,j)}$	$I_{(i+2,j+1)}$	$I_{(i+2,j+2)}$	
		8	8		
		$I_{(i+3,j)}$	$I_{(i+3,j+1)}$		

Figure 7: Figure showing the pseudo-gaussian mask. Each of the four central pixel(weight=64) is averaged with the neighboring similar colored pixels with their respective weights(in top right corner). Note the weights are all a power of 2 and that the total weight of the partial mask is also $64+2*16+16+(2*8)=128$ (a power of 2).

way that a low P_1 will detect most of the interest points and a higher P_2 will give the user a control over the noise interference with the desired feature points.

3.4 Eliminate False Positives

As discussed in 3.1 it is very clear that the orthogonal difference masks will respond strongly to all diagonal edges. In order to avoid this we eliminate all the candidate pixels that are part of the diagonals and are only 2-pixel connected in the direction of the diagonal. If a candidate pixel is eliminated, we reduce its cornerity strength by half. This strength reduction plays a very important role in determining the locality of the still existing candidate points. This step is depicted in the figure 8.

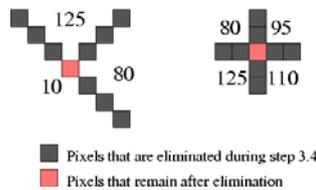


Figure 8: Figure showing the false positive candidates and marking the ones that are true feature points. The figures are just junctions where such elimination are prominent. The values denote the grayscale value of the image at the junction.

3.5 Determine Localization of the Desired Feature Points

This is the final step of the proposed algorithm and is similar to non-maxima suppression. Here we refer the cornerity strength of each of the four pixels of the 2x2

pixel patch and compare it with each other. Only the pixel having the highest cornerity value will be identified as the true position of the feature point. This gives our algorithm a localization accuracy of 1-pixel. Figure 9

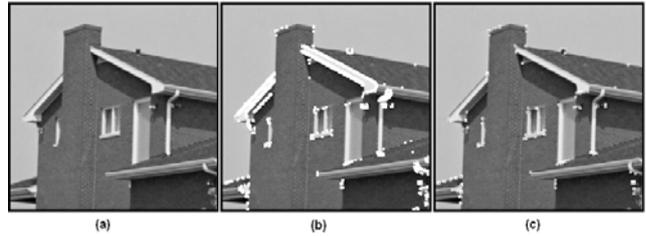


Figure 9: (a) Original Grayscale Image, (b) Output of the Algorithm after step 3.3 and (c) Final output of the algorithm after step 3.4 and 3.5

shows the output of the algorithm at intermediate stages. Invariably the entire algorithm has 3 major steps viz. the difference mask, elimination of false positives and then determining the localized points in the last step. Step 1 has an inbuilt gaussian kernel to handle noise and other deformities unlike other algorithms which use noise filter as an external process.

4 Qualitative Analysis of the Proposed Algorithm

Although a wide number of corner detection techniques are reported in literature, there is very less on the subject of comparing detectors. Mohanna and Mokhtarian [7] evaluated performance by wrapping test images in an affine manner by a known amount. They define two matrices: Consistency and Accuracy. Where consistency is defined as:

$$CCN = 100 * 1.1^{-|n_w - n_o|} \quad (6)$$

where n_w is the number of features in the wrapped image and n_o is the number of features in the original image. The accuracy is defined as:

$$ACU = 100 * \frac{\frac{n_m}{n_o} + \frac{n_m}{n_g}}{2} \quad (7)$$

where n_g are the number of 'ground truth' corners marked by humans and n_m is the number of matched corners compared to ground truth. This unfortunately relies on subjectively made decisions. [13] defines stability to be the number of 'strong' matches (detected over various frames in matching problems) divided by the total number of corners. This measurement is clearly dependent on both the tracking and matching methods used. When measuring reliability, what is important is if the same real-world features are detected from multiple views. This is the definition which will be used here. For an image pair, a feature is 'detected' if it is extracted in one image and appears in the second. It is 'repeated' if

it is extracted nearby in the second run. Schmid [3] proposes repeatability and information content as the two criteria for qualitative evaluation. The repeatability is the ratio of repeated features and detected features. Repeatability signifies that we obtain the same points in an image even after changes in the imaging condition. That is, a 3D point should be detected in both images if it is detected in one of them. The second criteria, information content, measures the distinctiveness of the local gray-level pattern at an interest point. repeatability and information content are the key criteria for image matching. In this context, at least a subset of the points have to be detected at the same relative position to allow feature correspondence. Additionally interest points should have distinctive patterns to be distinguishable. A point p_1 detected in image I_1 is repeated in image I_i if the corresponding point p_i is detected in image I_i , where I_1, I_i are two projections of the same image I . According to [7] it is clear that repeatability and information content are nothing but consistency and accuracy combined. So we tested algorithm for consistency and accuracy on the following experimental cases under various transformations:

1. Rotation variations: We rotated the test image house.gif to 45° in increments of 5° each. Algorithm was run on each rotated image and CCN and ACU numbers were calculated.
2. Scaling variations: Both uniform and nonuniform scaling is done. We skewed the image, scaled it horizontally and also uniformly scaled it to 125% its original size in 5% increments.
3. Blurring: We applied gaussian blurring and motion blurring on the test image to test our algorithm against diffused image boundaries. This was done to analyze the localization capability of the proposed algorithm.
4. 3-D projection: We projected our test image on a sphere and then calculated the CCN and ACU numbers. This test is a sure indication that our algorithm performs well from any viewpoint and in the presence of any perspective deformities.
5. Artificial Noise: We added artificial noise like poisson, gaussian noise($variance = 0.004$), salt & pepper noise($density = 0.004$) and speckle noise($variance = 0.004$). This test will evaluate our corner detector for robustness to real world noise and variations.

5 Performance Evaluation and Discussion

We used Matlab to perform all our tests. Figure 10 and Figure 11 depict the test results. Table 1 and Table 2 summarizes the mean CCN and ACU numbers for the

various transformations discussed in section 4. In most cases the CCN and ACU numbers are high, infact it gives very good results in case of noise and perspective deformation. Also from [12] and [14] we compare the algorithm complexity of our algorithm with some of the most popular existing feature point detectors. Our algorithm will incur at most 28 subtractions(or additions) at any pixel, but assuming that even in the scenario of highest noise content in the images only 80% of all the pixels will be candidate points that pass the first step of our algorithm, this gives an upper limit of average subtractions per pixel to 22. We involve no multiplication or division operations in our algorithm making it one of the most suitable contender for real time implementations as shown in Table 3.

	Plessy	Susan	ECSS	Our Algorithm
Rotation	32	24	74	72
Uniform Scaling	35	28	42	89
Non Uniform Deformation	28	31	68	82
Noise	14	9	33	41

Table 1: Shows the Consistency of the proposed Algorithm, CCN numbers(out of 100) of 3 other feature detectors are included for comparison.

Criteria	Plessy	Susan	ECSS	Our Algorithm
Accuracy	49.6	53.4	77.2	71.8

Table 2: Shows the Accuracy Factor of our Algorithm with respect to other popular feature detection algorithms. ACU number is listed as a percentage out of 100.

Algorithm	Additions	Multi-plications	Operations /Pixel
SCD	26	0	26
Wang-Brady	24.5	7.25	32
Plessy	95	22	117
Susan	32.25	0.75	33
Our Algorithm	22	0	22

Table 3: Algorithm complexity

6 Conclusions

We proposed a true real time feature point detector and successfully evaluated its performance over varying parameters such as noise, affine transformations, deformations speed and accuracy. It is based on the fact that any feature point is a variation in 2-dimension. Further robustness to noise and transformations is provided by the

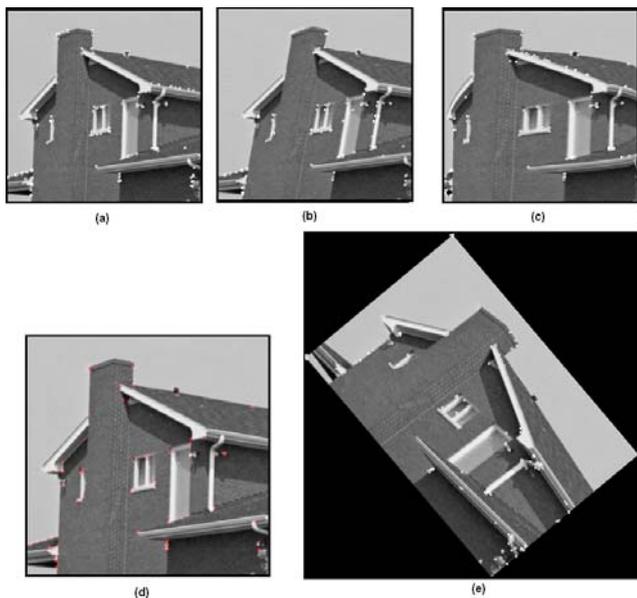


Figure 10: Feature Point detection on test image with various transformations (a) Original Image, (b) Image Skewed by 15° (Non-uniform scaling), (c) Image projected on a Sphere (Perspective Deformity), (d) Uniformly Scaled to 125%, and (e) Affine Transformation (Rotated by 30° and scaled horizontally by 50%)

use of a pseudo-gaussian kernel. The entire algorithm is simple but effective. This Noise and Affine transformation invariant feature Point detector is a true contender for Hardware implementations in the areas of Robotic Vision and real time image matching and object recognition.

References

- [1] D H Ballard and C M Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, NJ, 1982.
- [2] M Brown and D G Lowe. In *Invariant Features from Interest Point Groups*, pages 656–665. British Machine Vision Conference, 2002.
- [3] Roger Mohr C Schmid and C Bauckhage. In *Comparing and Evaluating Interest Points*, volume 655, pages 63–86. INRIA Rhone, Montbonnot, France, Europe, 2004.
- [4] R M Haralick and L G Shapiro. Addison-Wesley, 1993.
- [5] C. Harris and M. Stephens. In *A Combined Corner and Edge Detector*, volume 23, pages 147–151. Proceedings of 4th Alvey Vision Conference, Manchester, 1988.
- [6] D G Lowe. In *Distinctive Image Features from Scale-Invariant Keypoints*, volume 60, pages 91–110. International Journal of Computer Vision, 2004.

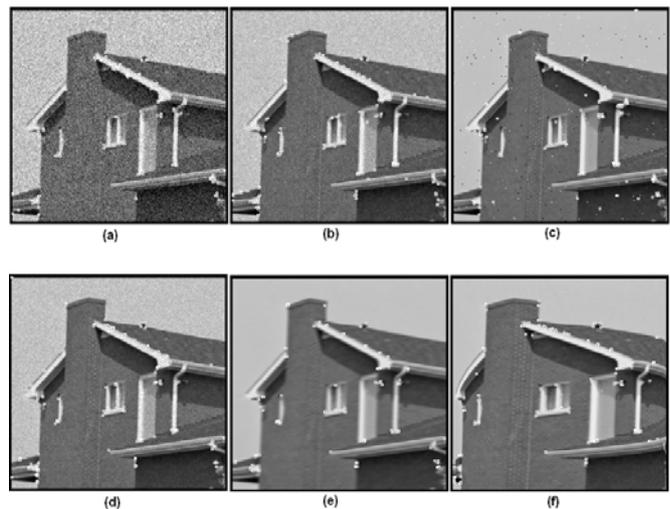


Figure 11: Results of our algorithm on the test image induced with artificial noise, (a) Gaussian Noise, (b) Poisson Noise, (c) Salt n Pepper, (d) Motion Blurring with $(x,y)=(5,10)$ and (e) Gaussian blurring of the spherically projected image.

- [7] M Mohannah and M Mokhtarian. Performance evaluation of corner detectors using consistency and accuracy measures. *Computer Vision and Image Understanding*, 102(1):81–94, 2006.
- [8] F Mokhtarian and R Soumela. In *Robust Image Corner Detection Through Curvature Scale Space*, volume 20, pages 1376–1381. IEEE Transactions on pattern Analysis and Machine Intelligence, 1998.
- [9] H P Moravec. In *Towards Automatic Visual Obstacle Avoidance*, page 584. Proceedings of the 5th International Joint Conference on Artificial Intelligence, 1977.
- [10] Edward Rosten and Tom Drummond. In *Machine Learning for high-speed Corner Detection*, volume 17, pages 211–224. ECCV, 2006.
- [11] Stephen M. Smith and J. Michael Brady. Susan - a new approach to low level image processing. *International Journal for Computer Vision*, 23(1):45–78, 1997.
- [12] Choong Don Yoo Sun Cheol, In So Kweon. In *Pattern Recognition Letter*, volume 23, pages 1349–1360. Elsevier, 2002.
- [13] M Trajkovic and M Hedley. Fast corner detection. *International Journal for Image and Vision*, 16(2):75–87, 1998.
- [14] H Wang and M Brady. In *Real-Time Corner Detection Algorithm for Motion Estimation*, volume 13, pages 695–703. Image and Vision Computing, 1995.