Corner Detection Using Support Vector Machines

Minakshi Banerjee, Malay K. Kundu and Pabitra Mitra
Machine Indian Statistical Institute
Kolkata 700108
e-mail: {minakshi_r,pabitra_r,malay}@isical.ac.in

Abstract

A support vector machine based algorithm for corner detection is presented. It is based on computing the direction of maximum gray-level change for each edge pixel in an image, and then representing the edge pixel by a four dimensional feature vector constituted by the count of other edge pixels lying in a window centred about and having each of the possible four directions as their direction of maximum local gray-level change. A support vector machine is designed using this feature vectors and the support vectors, representing critical points in a classification problem, correspond to the corner points. The algorithm is straightforward and does not involve computation of complex differential geometric operators. It has implicit learning capability resulting in good performance for a wide range of images.

1. Introduction

Corners are high curvature points lying on the lines of steep gray-level slope in an image. These points play a dominant role in shape perception by human. Using corner points a shape can be represented in an efficient and compact for many shape analysis problems. There are mainly two categories of corner detectors, namely, template based and geometry based methods [4]. Template based corner detection involves determining similarity, or correlation, between a given template of size $m \times m$ and all windows of size $m \times m$ in a given image. Geometry based corner detection algorithms are more popular and are either edge related, topology based, or using auto-correlation measures. Edge related methods considers corners as the junction of two or more edge lines. They usually operate on the chain code of the edges to detect the corners. Cornerness measures computed using differential geometric operators are also used. Topological corner detectors are based on the measurement of topological features of differential geometry of the corners in the image surface. They can work directly on grayscale images without the need for edge detection. Harris corner detector [2] may be considered as belonging to this category. Corners can also be detected by considering a local window in the image and determining the average changes of intensity which results from shifting the window by a small amount in various directions. A corner is detected when the minimum change produced by any of the shift is larger than a threhold value. These methods constitute the auto-correlation approach.

The methods described above has been successfully used in several applications in shape analysis and retrieval. However, some drawbacks of these methods which needs to be addressed include: (a) lack of guideline for construction of templates in template based methods, (b) need for computation of chain code in edge based methods, (c) need for computation of complex defferential geometric operators in topological methods, and (d) image dependency of performance of the corner detectors and lack of adaptiveness and learning capabilities. In this article we present a simple corner detection strategy which addresses some of these issues. The proposed algorithm, though requiring edge detection, does not need computation of chain codes or complex differential geometric operators. It also does not require construction of corner templates and performs corner detection by learning on a given image. Since learning is involved 'optimal' performance of the method is not restricted to a particular class of images.

The algorithm involves computing the direction (out of possible four, 0, +45, 90, and -45) of maximum gray-level gradient of all the edge point of an image. Then a local window about a edge point is considered, and the count of (other) edge pixels in that window having different maximum gradient direction is used to constitute a four dimensional feature vector for the edge pixel in the centre of the window. The pixel is assigned a class denoting its own direction of maximum gradient, thus obtaining a labeled data set having four classes. A support vector machine (SVM) is then designed on this data. It is evident on analysis that the feature vectors for non-corner points will be interior points and those for corner points will be points near

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

the class boundaries and correspond to the support vectors. Thus the support vector points generates the corner pixels. The corners obtained by the proposed algorithm are visually compared with the popular Harris corner detector for two graylevel images.

Next we present the preliminaries of support vector machine, before describing the corner detection algorithm in detail.

2. Support Vector Machine Preliminaries

Support vector machines are a general class of learning architecture inspired from statistical learning theory that performs *structural risk minimisation* on a nested set structure of separating hyperplanes [3]. Given a training data, the SVM training algorithm obtains the optimal separating hyperplane in terms of generalization error. We describe below the SVM design algorithm for a two class problem. Multiclass extensions can be done by designing a number of one-against-all on one-against-one two class SVMs.

Algorithm:
Suppose we are given a set of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, $\mathbf{x} \in R^N, y_i \in \{-1, +1\}$. We consider functions of the form $sgn((\mathbf{w} \cdot \mathbf{x}) + b)$, in addition we impose the condition

$$\inf_{i=1,\ldots,l} |(\mathbf{w} \cdot \mathbf{x}_i) + b| = 1. \quad (1)$$

We would like to find a decision function $f_{\mathbf{w},b}$ with the properties $f_{\mathbf{w},b}(x_i) = y_i;\ i = 1, \ldots, l$. If this function exists, condition (1) implies

$$y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1, \quad i = 1, \ldots, l. \quad (2)$$

In many practical situations, a separating hyperplane does not exist. To allow for possibilities of violating Equation 2, slack variables are introduced like

$$\xi_i \geq 0, \quad i = 1, \ldots, l \quad (3)$$

to get

$$y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l. \quad (4)$$

The support vector approach for minimizing the generalisation error consists of the following:

$$\textbf{Minimize}: \qquad \Phi(\mathbf{w}, \xi) = (\mathbf{w} \cdot \mathbf{w}) + C \sum_{i=1}^{l} \xi_i \quad (5)$$

subject to the constraints (3) and (4).

It can be shown that minimizing the first term in Equation 5, amounts to minimizing the VC-dimension, and minimizing the second term corresponds to minimizing the missclassfication error [1]. The above minimization problem can be posed as a constrained quadratic programming (QP) problem.

The solution gives rise to a decision function of the form:

$$f(\mathbf{x}) = sgn\left[\sum_{i=1}^{l} y_i \alpha_i (\mathbf{x} \cdot \mathbf{x_i}) + b\right].$$

Only a small fraction of the α_i coefficients are non-zero. The corresponding pairs of $\mathbf{x_i}$ entries are known as *support vectors* and they fully define the decision function. The support vectors are geometrically the points lying near the class boundaries. We use linear kernels for SVM. However, nonlinear kernels like polynomial, sigmoidal and radial basis functions may also be used [1].

The aforesaid two class SVM can easily be extended for multiclass classification by designing a number of one-against-all two class SVMs, e.g., a k-class problem is handled with k two class SVMs.

3. Corner Detection Algorithm

The corner detection algorithm consists of two phases. In the first phase, the edges of the given image are extracted along with the direction of maximum gray-level change of the edge pixels. Then a support vector machine is designed on a labelled set of four dimensional feature vectors. The support vectors generated in the design process correspond to corner points.

3.1. Construction of feature vectors for edge points

The first step of corner detection is extraction of edges of a given image. Any edge detection algorithm may be used for this purpose. Canny's edge detector is used in our experiments. For each of the edge points the direction of maximum gray-level change is detected next. This is done in the following manner. For every edge pixel, consider a 3×3 window centred about it. The change in gray-level value along the four directions 0, 45, 90, and -45 degrees are computed by taking the absolute differences of the pixel values $|a_1 - a_2|$, $|b_1 - b_2|$, $|c_1 - c_2|$, and $|d_1 - d_2|$ as shown in Figure 1. The direction of maximum gray-level change is considered as the direction of that edge pixel.

The next step is to construct the four dimensional feature vector describing an edge pixel. This is done as follows. Consider an $m \times m$ window about an edge pixel p_i. Count the number of other edge pixels within this $m \times m$ window which have directions 0, +45, 90, and -45. Let this counts be n_0, n_{+45}, n_{90}, and n_{-45} respectively. Let the direction for pixel p_i itself be $d_i \in \{0, +45, 90, -45\}$ which is the majority of the counts. Thus the feature vector for pixel p_i is $[n_0, n_{+45}, n_{90}, n_{-45}]$ and class label is d_i.

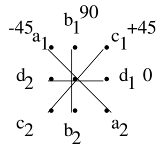


Figure 1. Computation of gray-level changes.

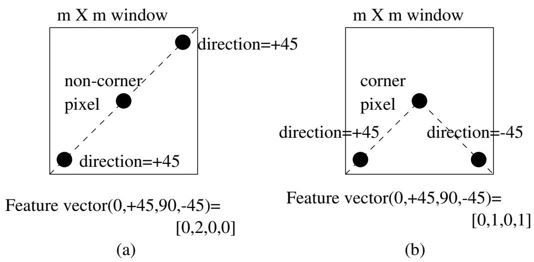


Figure 2. Direction of neighboring pixels in a window for, (a) non-corner pixel, and (b) corner pixel.

3.2. Corners as support vectors

It may be noted that for a non-corner edge pixel, all other edge points in the $m \times m$ window about it has the same direction as itself. As a consequence only one of the $n_0, n_{+45}, n_{90}, n_{-45}$ counts is non-zero and the other three are zero. While for corner points multiple of these counts are non-zero. This is illustrated in Figure 2. Thus if one considers the scatter plot of these four dimensional feature vectors, the non-corner points lie on the axes and are interior points of the corresponding class, while the corner points are off-axis points and constitute the boundary region of the class. This is illustrated in Figure 3.

As discussed in the previous section support vectors are points lying near class boundaries. If a support vector machine is designed on the above mentioned four dimensional labelled feature vectors the corner points correspond to the extracted support vectors.

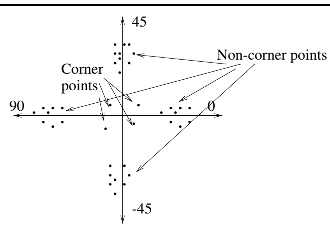


Figure 3. Visualization of the distribution of corner and non-corner points.

In our experiments we have used linear support vector machines with soft margin. The constant C of Equation 5 can be varied to increase or decrease the number of support vectors, thus leading to multiscale detection of corners. Another issue is to choose the window size $m \times m$. The value of m determines the amount of smoothing performed. Though no guideline for choosing it is provided here, one should choose the window as large as possible but with its diagonal smaller than an edge segment.

4. Experimental Results

The corners detected using the support vector machine based algorithm and the popular Harris corner detector is presented in Figures 4 and 5 for two grayscale images. The Harris corner detector [2] computes the locally averaged moment matrix computed from the image gradients, and then combines the eigenvalues of the moment matrix to compute a corner strength, of which maximum values indicate the corner positions. We have chosen threshold to be 1000 and the standard deviation of the Gaussian used for local averaging to be 1. In the SVM based algorithm the value of C is taken to be 1000, and the window size for constructing the feature vector is taken to be 7×7. The corners detected using both Harris and the SVM based algorithm are found to be satisfactory for both the images.

5. Conclusions

A support vector machine based algorithm for corner detection is presented. It is based on computing the direction of maximum gray-level change for each edge pixel, and then representing the edge pixel by a four dimensional feature vector constituted by the count of pixels in a window centred about it having each of the possible four directions of maximum gray-level change. A support vector machine is designed using this feature vectors and the support vectors correspond to the corner points.

The algorithm does not involve computation of complex differential geometric operators and has implicit learning capability. It may be further refined using other non-linear support vector machines and more complex feature vectors like gradient values instead of direction pixel counts.

References

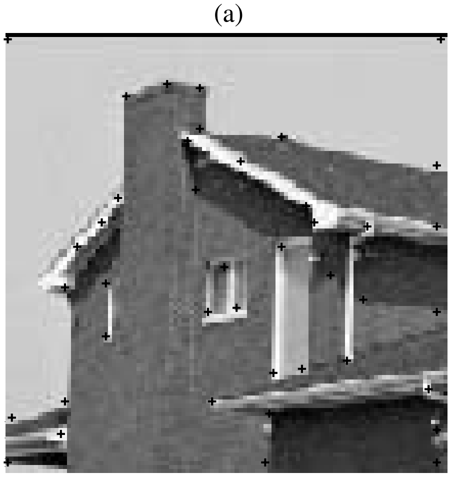
[1] C. J. C. Burges. A tutorial on support vector machines. *Data Mining and Knowledge Discovery*, 2, 1998.

[2] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conf.*, pages 147–151, 1988.

[3] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

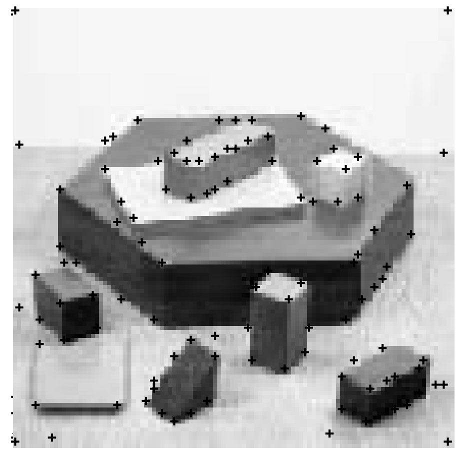
[4] Z. Zheng, H. wang, and E. K. Teoh. Analysis of gray level corner detection. *Pattern Recognition Letters*, 20:149–162, 1999.

(a)

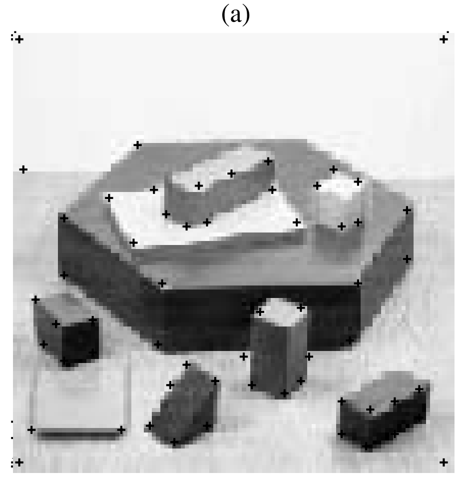


(b)

Figure 4. Corners obtained for the house image using (a) Harris, and (b) SVM based corner detectors



(a)



(b)

Figure 5. Corners obtained for the blocks image using (a) Harris, and (b) SVM based corner detectors