Kona: A Multi-Junction Detector using Minimum Description Length Principle

Laxmi Parida, Davi Geiger, Robert Hummel
parida@cs.nyu.edu, geiger@cs.nyu.edu, hummel@cs.nyu.edu.

Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, U.S.A.

Abstract. Corners, T-, Y-, X-junctions give vital depth cues which is a critical aspect of image understanding tasks like object recognition: junctions form an important class of features invaluable in most vision systems. The three main issues in a junction (or any feature) detector are: scale, location, and, the junction (feature) parameters. The junction parameters are (1) the radius, or size, of the junction, (2) the kind of junction: lines, corners, 3-junctions such as T or Y, or, 4-junction such as X-junction, etcetera, (3) angles of the wedges, and, (4) intensity in each of the wedges. Our main contribution in this paper is a modeling of the junction (using the minimum description length principle), which is complex enough to handle all the three issues and simple enough to admit an effective dynamic programming solution. Kona is an implementation of this model. A similar approach can be used to model other features like thick edges, blobs and end-points.

1 Introduction

A critical component of most recognition systems is stable, representative, feature extraction from images. One of the key features used in recognition is junctions: T-junctions, Y-junctions, X-junctions and so on. These junctions are also critical for stereo vision modules or motion modules, since these are places where occlusions can be identified. Such points, for example, coincide with the images of trihedral vertices of an object. These are critical features for recognition as suggested by [2], [3], [5]. We use the template deformation framework to develop a "junction detector," to finds corners (two-junctions), tri-corners (tri-junctions), quad-corners (quad-junctions), etc., defined as points where two or more homogeneous surface patches are located within an arbitrarily small neighborhood of the point.

There have been basically two different paradigms for detecting junctions: edge detection followed by grouping of edges to form junctions [13, 12, 1, 16], and, treating junction as a template matching phenomenon [4], In the former, it is assumed that the presence (or absence) of a junction is determined by "grouping" the intensity gradient near a hypothesized junction. Usually one is interested in examining large gradients in the direction perpendicular to the hypothesized radial line. Experiments in this framework are limited and even the richest ones shown in [13] are interesting but not exhaustive. In the latter it is assumed that

a (suitably small) local neighborhood is sufficient to detect a junction. The basic idea is to fit a junction-model to the input signal in a neighborhood. This involves minimizing an energy function which gives a measure of the "distance" of the junction-model from the input signal.

The idea of performing local feature detection by projecting image data onto a subspace is fundamental in [4, 7]. Basically, the input is orthogonally projected onto a finite dimensional subspace of the Hilbert space of functions. An energy function (which is the L^2 norm of the the difference of the input and the fitted function) is minimized in this finite dimensional space. The two main issues are finding an orthonormal basis that spans a good finite dimensional subspace and minimizing the energy function. This approach can give closed form solutions for edges [4, 7], and lines [6]. But, the generalization to junctions is complex and the solution is not apparent.

In [9], corners and junctions (which are modeled as two adjacent corners) are represented by functions (models) that are blurred with a Gaussian (or an exponential filter) where the authors use a closed form solution. In general, numerical methods are used to obtain parameters that minimize the distance to the input data using an L^2 norm. This is also the case in [10].

Our approach is to use a combination of the two paradigms: grouping of edges and fitting templates. We use a template deformation framework, using the minimum description length (MDL) principle, that includes the gradient information in order to detect the radial partitions of the template as a grouping mechanism. In other words, the task is to find the minimum number of wedges that best describes the junction. Note that as we increase the number of wedges, the junction description gets more accurate, hence the task is to use the MDL principle to obtain an optimal number of such wedges.

2 The Junction Model

We model a junction as a region of an image where the values are piecewise constant in wedge-shaped regions emanating radially from a central point, covering a small disk centered at the point and omitting a (much) smaller disk centered at this point (see figure1). The parameters of a junction consist of (i) the radius of the junction-disk, (ii) the center location, (iii) the number of radial line boundaries, (iv) the angular direction of each such boundary, and (v) the intensity within each wedge. The radius of the disk addresses the "scale" issue, and the location of the center is a kind of "interest operator" [8] that determines the position where the feature is located in a region, possibly pre-defined.

We can formulate the junction detection problem as one of finding the parameter values that yield a junction that best approximates the local data, and declaring local minima of the error as junctions. The best-fit parameter values provide attributes of the detected junction.

Let T denote the piecewise constant function/template. It has N angles and N intensities if N is the number of constant pieces. The interested reader may

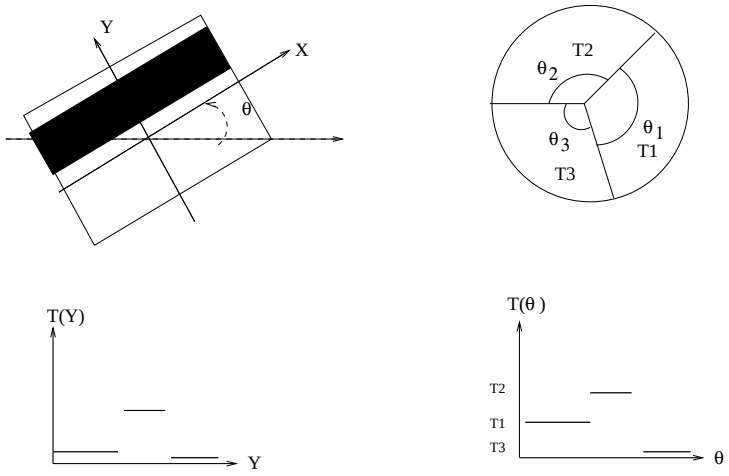


Fig. 1. Piecewise constant features. A bar detector and a junction detector.

look at [14] for a more rigorous definition of the template as a smooth function of two variables. Further, let I denote the input signal.

Define the energy function, at a point (i, j) on the image as follows

$$E = \mathcal{D} + \lambda^2 \mathcal{G} \, . \tag{1}$$

The first term, $\mathcal{D}$, is a measure of the distance of the fitted function from the *data* using the L^2 norm:

$$\mathcal{D} = \int_0^{\infty} \int_0^{2\pi} [I(r,\theta) - T(\theta)]^2 g(r) r dr d\theta; \tag{2}$$

where $g(r)$ is an appropriate modulating function that goes to zero for large r, thus defining the template size.

The second term, $\mathcal{G}$, is a measure of the distance of the *gradient* using the L^2 norm. Why is $\mathcal{G}$ required? We show an example, in Figure 2, where considerable improvement is achieved by the use of this term.

$$\mathcal{G} = \int_0^{\infty} \int_0^{2\pi} |\nabla I(r,\theta) - \nabla T(\theta)|^2 g^*(r) r dr d\theta. \tag{3}$$

where $g^*(r)$ is an appropriate modulating function, not necessarily the same as $g(r)$.

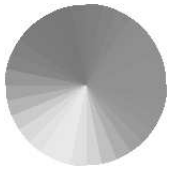
It may be pointed out that this form is different from the Mumford-Shah functional [11] particularly because of the presence of the term $\nabla I(r,\theta)$ in our model. As we explain in the next section, this term is critical in obtaining the portion of the image, or window, to which we fit a multi-junction template.

Note that

$$\nabla I = \frac{\partial I(r,\theta)}{\partial r} \mathbf{e_r} + \frac{1}{r} \frac{\partial I(r,\theta)}{\partial \theta} \mathbf{e}_\theta$$

(a) Point on the original image.

(b) Original data.



(c) Projection of (b) to 1D signal.

(d) Gradient is not used.

(e) Gradient is used.

Fig. 2. The use of gradient information in (e) gives a more accurate corner than in (d) where it is not used.

$$\nabla T(\theta) = \frac{1}{r}\frac{\partial T(\theta)}{\partial \theta}\mathbf{e}_\theta$$

where $\mathbf{e_r}$ and $\mathbf{e}_\theta$ are the orthonormal vectors in the r and θ direction respectively, evaluated at (r, θ).

Separating the angular and the radial terms, $\mathcal{G}$ can be written as,

$$\mathcal{G} = \mathcal{A} + \mathcal{R}, \tag{4}$$

where,

$$\mathcal{A} = \int_0^\infty \int_0^{2\pi} \frac{1}{r^2}\left(\frac{\partial I}{\partial \theta} - \frac{\partial T}{\partial \theta}\right)^2 g^*(r) r dr d\theta, \tag{5}$$

$$\mathcal{R} = \int_0^\infty \int_0^{2\pi} \left(\frac{\partial I}{\partial r}\right)^2 g^*(r) r dr d\theta. \tag{6}$$

3 Energy Minimization

This section deals with minimization of the engery function 1. Using equations 1 and 4, the energy equation can be written as: $E = \lambda^2\mathcal{R} + \overline{E} = \lambda^2\mathcal{R} + (\mathcal{D} + \lambda^2\mathcal{A})$. $\mathcal{R}$ is independent of the junction feature and handles the scale and the location issue. $\overline{E}$ is minimized to obtain the most appropriate junction template.

3.1 Scale & Location

Consider a convenient and meaningful definition of $g(r)$ as follows:

$$g(r) = \begin{cases} 0 & r < R_0 \\ \frac{1}{r} & R_0 \leq r \leq R_1 \\ 0 & r > R_1 \end{cases} .$$

A user-defined threshold bounds $\lambda^2\mathcal{R}$: this defines R_1. R_0 is often a user-defined fraction of R_1: this allows a small hole in the center. A significant observation, from the series of experiments, has been the use of non-zero R_0.

We do the following to obtain the location in a window: $\lambda^2\mathcal{R}$, (with not necessarily the same R_1), is evaluated for the points and the one with the minimum value defines the location. We illustrate this in Figure 3.

3.2 Junction Parameters

We can carry out an appropriate numerical integration to obtain the value of the first term $\lambda^2\mathcal{R}$. The second factor, $\overline{E}$, is used to estimate the junction template. The unknown factors are: N, the number of intersecting lines (or wedges) at the junction [1], $\{\theta_p\}, \{T_p\}, p = 1 \ldots N$, where N is the number of wedges, θ_p's are the angles where the partitions occurs, T_p's are the intensity values.

We can write down $\overline{E}$ as

$$\overline{E} = \mathcal{F} + \mathcal{V},$$

where, $\mathcal{F}$ is fixed, that is, it does not depend on the unknown parameters, where as, $\mathcal{V}$ does. Setting $g^*(r) = r^2 g(r)$, and, with some simple manipulations we obtain

$$\mathcal{F} = \int_0^{\infty} \int_0^{2\pi} \left[I^2(r,\theta) + \lambda^2 \left(\frac{\partial I(r,\theta)}{\partial \theta} \right)^2 \right] g(r) r dr d\theta.$$

$\mathcal{F}$ can be approximated numerically. Also, $\mathcal{V}$ can be approximated as,

$$\mathcal{V} = \sum_{p=1}^{N} \left[(\theta_{p+1} - \theta_p)(-2T_p \bar{I}_{\theta_p, \theta_{p+1}} + T_p^2 C) + \lambda^2 (-2T_p'(\partial_\theta \tilde{I})_{\theta_p} + C T_p'^2) \right],$$

[1] N for homogeneous region is 1, for line and corner is 2, for junctions like T-, Y-junction is 3, 4 for X-junction and so on.

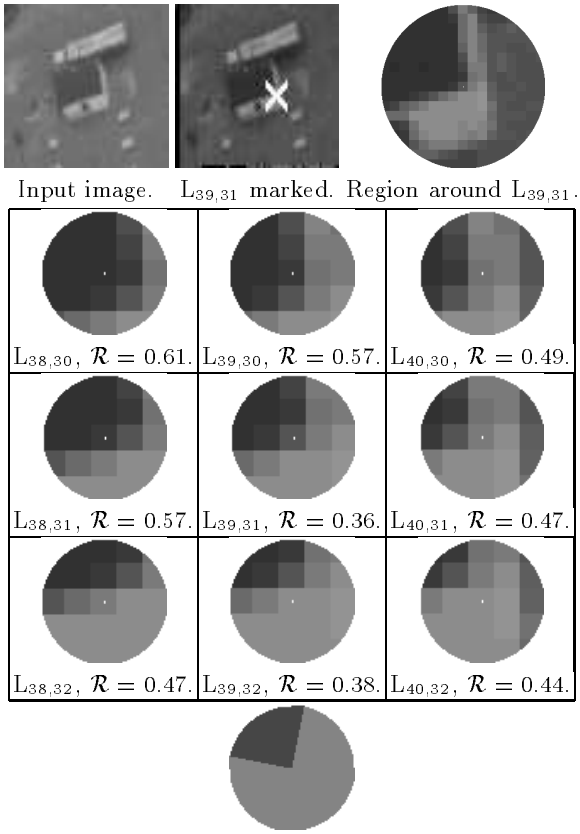


Fig. 3. The use of $\mathcal{R}$, to locate the center of the junction. $L_{x,y}$ indicates the x and y coordinates on the image. Note that the location $L_{39,31}$ has the minimum $\mathcal{R}$ in the neighborhood. Incidentally, R_1, the size of the template is the same for all the nine locations.

where,

$$
\begin{aligned}
C &= \int_0^\infty g(r) r dr = R_1 - R_0 , \\
\tilde{I}(\theta) &= \int_0^\infty I(r,\theta) g(r) r dr \\
(\partial_\theta \tilde{I})_{\theta_p} &\equiv \frac{\partial \tilde{I}(\theta)}{\partial \theta}|_{\theta_p} = \frac{\partial \int_0^\infty I(r,\theta) g(r) r \; d\, r}{\partial \theta}|_{\theta_p} \\
\bar{I}_{\theta_p,\theta_{p+1}} &= \Delta\theta \sum_{j=j_p}^{j_{p+1}} \tilde{I}(j\,\Delta\theta).
\end{aligned}
$$

Further, $\tilde{I}(\theta)$, $(\partial_\theta \tilde{I})_{\theta_p}$ can be approximated numerically. For the sake of brevity,

(a) Position on the image. (b) Position on the image.

(c) Adaptive window at (a). (d) Adaptive window at (b).

(e) The best-fit to (c). (e) The best-fit to (d).

Fig. 4. An example to show the dynamic computation of the windows at different locations on the image.

we have omitted the details of the derivations here; they appear in [14].

Estimating wedge angles and intensities: Let us assume the number of wedges is fixed. In the next section we discuss how to estimate the number of wedges. We propose a dynamic programming formulation, although in practice we use the simpler version which is reasonably effective (See section 4). Let A_1, A_2, ..., A_K denote the range of admissible intensities, θ_1, θ_2, ..., θ_k, the discretized angles. Then,

$$\mathcal{E}^{ij} = \begin{cases} -2A_j\tilde{I}(\theta_i) + A_j^2 C & i = 1, \\ \min_{1\le s\le K}\left(\mathcal{E}^{(i-1)s} - 2A_j\tilde{I}(\theta_i) + \lambda^2(C\Delta A_{js}^2 - 2\Delta A_{js}\partial_\theta\tilde{I}(\theta_i))\right) & i > 1. \end{cases}$$

where $\Delta A_{jk} = A_j - A_k$. The energy is,

$$\overline{E} = \min_{s=1,K} \mathcal{E}^{ks}.$$

The constraint that the number of wedges is fixed can be imposed by increasing the energy to infinity whenever violated. Also, for the dynamic program, we rearrange θ's so that in the rearrangement, θ_1 is such that $\tilde{I}(\theta_1)$ is closest to $\tilde{I}(\theta_k)$. This ensures that θ_1 and θ_k are part of the same wedge.

Estimating the number of wedges: There are at least two ways of deciding on the optimal number of wedges: one is by penalizing for the increase in the number of junctions by introducing a term such as αn, and, the other is by measuring the amount of decrease in the energy.

The former has a straightforward underlying variational form by adding the term $+\alpha \Delta A_{js}$ to the energy obtained in the previous paragraph. We have observed that, in practice, this criteria has not worked well. Instead, measuring the rate of increase of energy by thresholding the ratio of the energy measurement, $\overline{E}^n$, works better. More precisely, the number of wedges, N, is computed by thresholding the relative error, r^n,

$$r^n = \frac{\overline{E}^{n+1}}{\overline{E}^n}.$$

Note that as the number of parameters increase, $\overline{E}^n$ decreases, i.e., $r^n < 1$. A variational form that may justify this approach is to use, as suggested in the book of Morel and Solimini [19], a split and merge algorithm for the energy $\tilde{E} = log(r^n)$.

4 Implementation

Although the energy equation E of the last section looks fairly complex, it has a remarkably simple and natural interpretation. Once the role of each factor of the energy equation is ascertained, some computations (like evaluating $\mathcal{F}$, for instance), can be omitted without changing the solution. To recapitulate, the factors are used thus:

1. $\lambda^2 \mathcal{R}$ is used for scale, i.e., to determine the size of the window, R_1. This is also used to obtain the exact location of the junction in a neighborhood.
2. $\overline{E}$ is used to obtain the junction parameters.

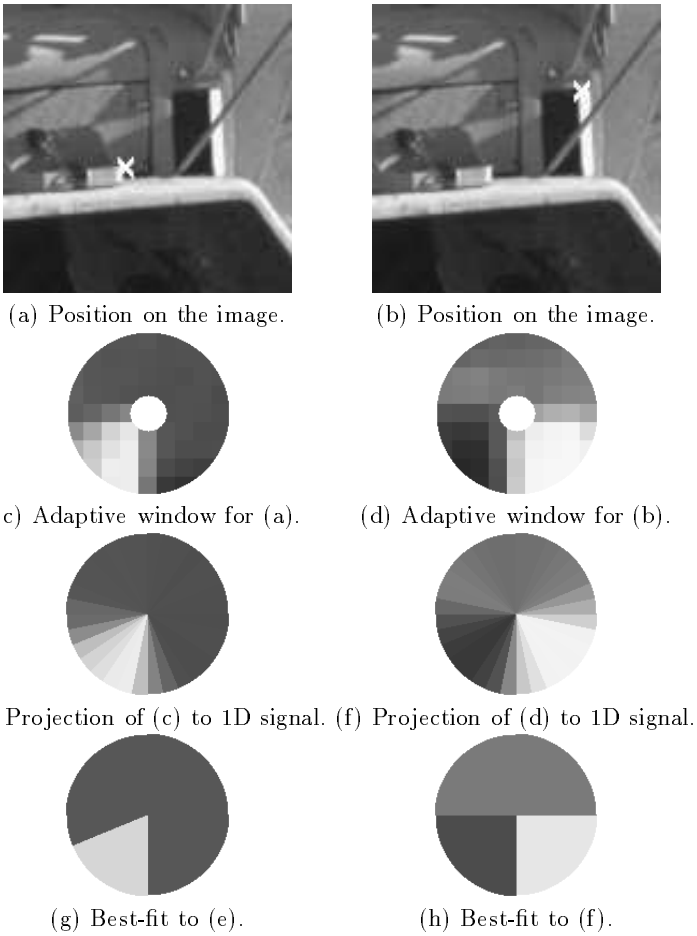
The $\tilde{I}(\theta)$ can be viewed as integrating the intensity along a radial line. Thus the two-dimensional image is projected on to a one-dimensional coordinate $D(\theta)$, where θ is appropriately discretized.

Let $D(\theta_i)$ be defined for $\theta_1, \theta_2, \ldots, \theta_d$, and, $(\theta_{i+1} - \theta_i) = 2\pi/d$, $\forall i$ [2]. For a p-junction, $T_1, T_2, \ldots, T_p$, are the template intensities and the wedge boundaries are at $\theta_{k_1}, \theta_{k_2}, \ldots, \theta_{k_p}$, $k_1 \leq k_2 \ldots \leq \theta_{k_p}$.

Assuming, we know the θ_p's, we can obtain T_p's by setting $\frac{\partial \mathcal{V}}{\partial T_p} = 0, \forall p$. When $\lambda^2 = 0$, fixing, θ_{k_i}'s, the T_l's can be shown to be the following:

$$T_l = \frac{\sum_{j=k_l}^{k_{l+1}} D(\theta_j)}{k_{l+1} - k_l}.$$

[2] If $i = d$, $1 = i + 1$.



(a) Position on the image. (b) Position on the image.

(c) Adaptive window for (a). (d) Adaptive window for (b).

(e) Projection of (c) to 1D signal. (f) Projection of (d) to 1D signal.

(g) Best-fit to (e). (h) Best-fit to (f).

Fig. 5. Examples of a sharp corner and a T-junction.

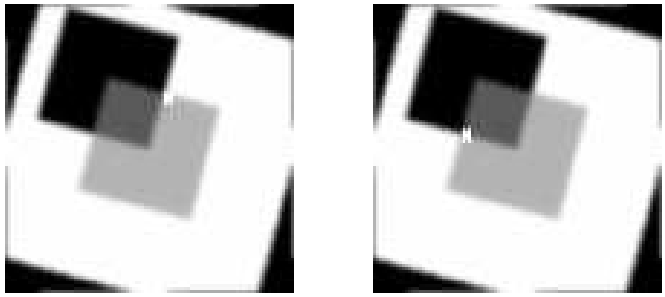
In other words, the T_l is a piecewise constant fit which is the average value of the data in that region. The energy for the fit is:

$$\overline{E}^p = \sum_{l=1}^{p} \sum_{j=k_l}^{k_{l+1}} \left(D(\theta_j) - T_l \right)^2 .$$

When $\lambda^2 \neq 0$, $\overline{E}^p$ has some extra terms. Since, the template intensities are close to the image intensities, we make the following approximation:

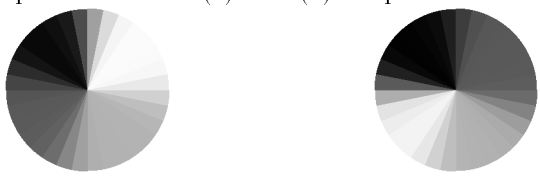
$$T_p' \mathbf{e}_\theta \approx (\partial_\theta \tilde{I})_{\theta_p} .$$

Now, it can be easily verified that the T_l and $\overline{E}^p$ is the same as before. We compute the θ_p's by exploring all possible set of θ_p's. We summarize the dynamic program



(a) Position on the image. (b) Position on the image.

(c) Adaptive window for (a). (d) Adaptive window for (b).



(e) Projection of (c) to 1D signal. (f) Projection of (d) to 1D signal.

(g) Best-fit to (e). (h) Best-fit to (f).

Fig. 6. A transparent object occludes a dark object in the background. The X-junctions detected by the program.

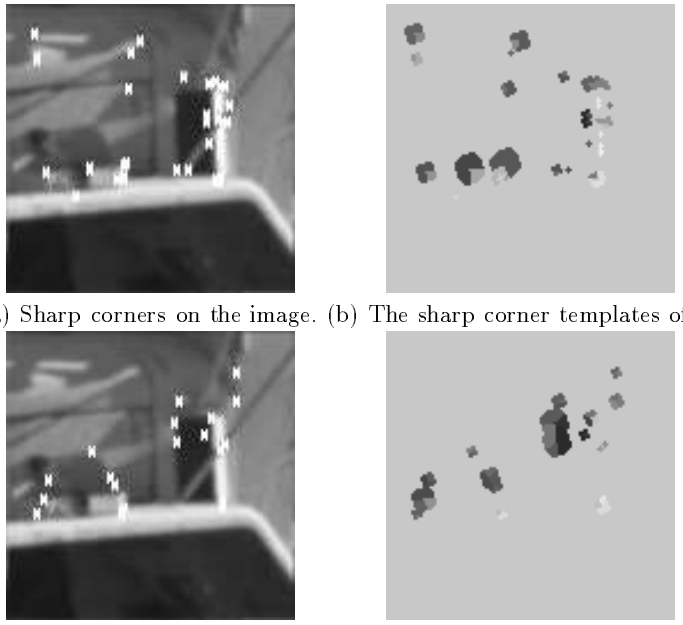
as follows:

$$C_{jl}^{p} = \begin{cases} \text{cost of fitting } T_p \text{ to } \theta_j, \theta_{j+1} \ldots \theta_l, & l \geq j, \\ \text{cost of fitting } T_p \text{ to } \theta_l, \theta_{l+1} \ldots \theta_1 \ldots \theta_j, & l < j. \end{cases}$$

$$\mathcal{E}_{sj}^{p} = \begin{cases} C_{sj}^{1} & p = 1, \\ \min_{i<j} \left(\mathcal{E}_{si}^{p-1} + C_{(i+1)j}^{p} \right) & \text{otherwise}. \end{cases}$$

$$\overline{E} = \min_{s=1,m} \mathcal{E}_{sm}^{N}$$

$\mathcal{F}$ is a constant (fixed), depending only on the image, and, does not influence the junction parameters. The scenario where $\mathcal{F}$ could play a role is at the time of



(a) Sharp corners on the image. (b) The sharp corner templates of (a).

(c) Junctions on the image. (d) The junction templates of (c).

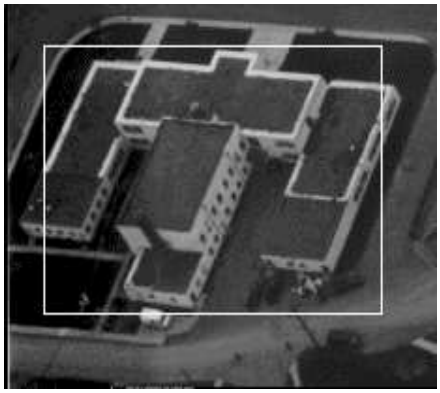
Fig. 7. Kona is run on an image. The images on the left show the points at which the features were detected and the images on the right show the feature templates at the points, with the size of the template proportional to the size of the window computed by the program. The bi-corners are filtered to show sharp corners (angle < 100) and with not-too-small contrast (> 20). The tri-corners are filtered to show the ones with not-too-small contrast (> 40).

comparing the energy in a neighborhood. But we use $\mathcal{R}$ to filter the neighborhood so $\mathcal{F}$ may be ignored without any damage.

The threshold used to dynamically determine the window size R_1, is the same at all places on the image, irrespective of the local contrast. We achieve this by normalizing the input data (i.e., rescaling the intensities in the input signal to lie in a standard range, say $[0, 1]$). Thus, regions with low contrast get comparable values to regions with high contrast. Hence, a single global threshold works.

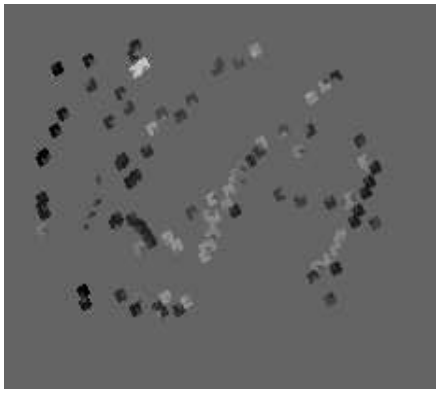
To summarize, the following steps are involved in detecting junctions on a large region of an image:

- Compute $\mathcal{R}$, and, R_1, the size of the template on every point.
- Filter the locations using a threshold on $\mathcal{R}$.
- In a neighborhood of a surviving location, pick the one with minimum $\mathcal{R}$, and remove all other locations within a radius of R_1 of this. Repeat this for all the surviving locations.
- Compute the junction parameter for all the surviving locations.

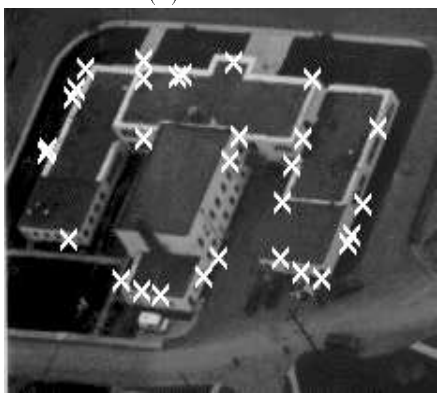


(a) Marked input image.

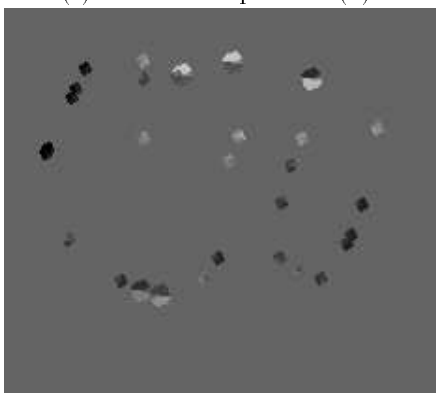
(b) 2-corners.



(c) Junction templates of (b).



(d) 3-corners.



(e) Junction templates of (d).

Fig. 8. Example of another image showing the results of Kona.

Kona[3] is an implementation of this junction model. It is programmed in C. A parallel implementation of this is also available.

5 Results of Experiments

In our experiments we have used $\lambda^2 = 1$.

The threshold for obtaining the scale (and, subsequently R_1), is lower for very sharp images, like binary images, where we use a value of 0.4 and for other real-life images we have used a value of 0.8 with very satisfactory results.

The threshold for deciding N of the junction was set to 0.4, for all the images we used, with very satisfactory results.

We observed that using the number of discretization units of the angle as 16 has worked very reasonably: 8 was found to be too coarse and 32 did not result in any significant higher accuracy [4].

An interesting observation of our experiments is that, using a small hole at the center ($R_0 > 0$) has worked very well. We have also noticed that the sharper images need larger holes. The other alternative is to blur the image first.

Figure 5 shows the function/template that is fit at two points on the image. Figure 6 shows an image where the X-junctions have been detected and Figures 7 and 8 show the results of Kona on regions of images.

We demonstrate the stability of the junction detector by running it on a pair of stereo images as shown in Figure 9. The results of this is used by an algorithm that runs on stereo images and estimates depth of objects in the scene. Figure 10 shows some results of obtaining illusory contours from junctions on an image. See [17] for details of the algorithm.

6 Conclusions

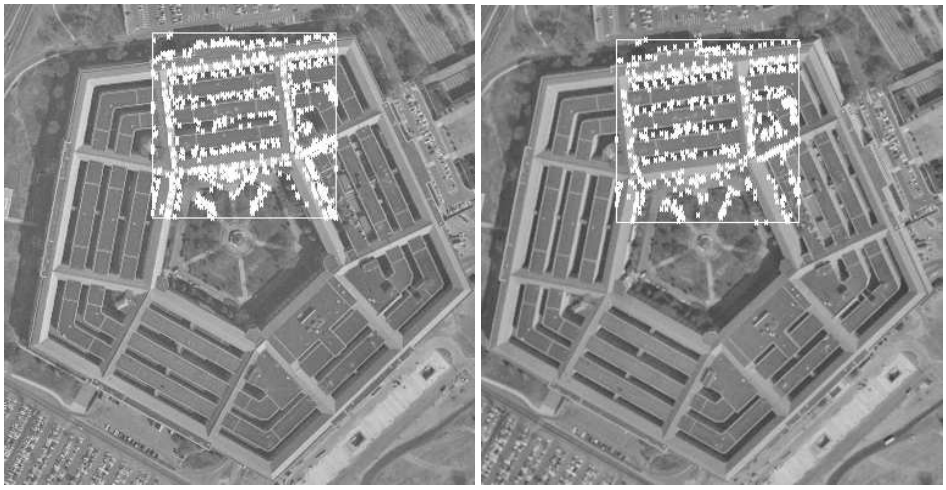
To conclude, we summarize the contributions of this paper as follows. Firstly, an effective modeling of the junction that includes a gradient term in the model, and, the removal of a small disk at the center of the junction. Secondly, using the model to obtain the scale and location of a junction, and, obtaining a simple dynamic programming solution to the optimization problem.

This paper demonstrates the successful use of piecewise constant functions, using a minimum description length principle, to detect features like junctions and corners. Similar piecewise constant functions may be used for features like bar detectors, blobs, end-points etc.

In this paper we have focussed solely on the task of junction detection, however it is being currently tested in a multitude of tasks such as object recognition (based on geometric hashing), detection of illusory contours, depth estimation from stereo pairs of images. It is also being tested in a novel approach to image compression using image diffusion and multi-junctions.

[3] The word for "corner" in Hindi is *Kona*.

[4] We choose a multiple of 4 because of the four quadrants - using any arbitrary number could do as well, in principle.



(a) Left Image. (b) Right Image.

Fig. 9. Kona is run on a pair of stereo images: it detects junctions, with significant contrast, on similar locations on both the images.

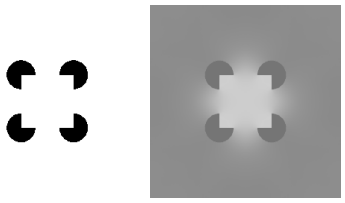


Fig. 10. The well known four-packman picture, and, the illusory contours of a square in front, as a result of the algorithm, is shown on the right.

Acknowledgments: We are grateful to J. Morel and T. Binford for their insightful comments and feedback. We also thank E. Hancock for his comments on the paper.

References

1. W. Freeman, E. Adelson, *Junction Detection and Classification*, Proc. ARVO 1991.
2. A. Guzman, *Decomposition of a Visual Scene into Three-Dimensional Bodies*, Proc. AFIPS 1968 Fall Joint Computer Conference, 1968.
3. D.A. Huffman, *A Duality Concept for the Analysis of Polyhedral Scenes*, Machine Intelligence, Vol 6, Edinburgh Univ. Press, Edinburgh, U.K., 1971.
4. M. F. Hueckel, *An operator which locates edges in digitized pictures*, J, Assoc. Compt. Mach. Vol 18, 1971.
5. D. L. Waltz, *Understanding Line Drawings of Scenes with Shadows*, The Psychology of Computer Vision, McGraw-Hill, New York, 1972.

6. M. F. Hueckel, *A local operator which recognizes edges and lines*, J, Assoc. Compt. Mach. Vol 20, 1973.
7. R. Hummel, *Feature Detection Using Basis Functions*, Computer Graphics and Image Processing, Vol 9, 1979.
8. W. Forstner, E. Gulch, *A Fast Operator for Detection and Precise Location of Distinct Points, Corners, and Centres of Circular Features*, Procc of Intercommssion Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 1987, pp 281-305.
9. R. Deriche, T. Blaszka, *Recovering and Characterizing Image Features Using An Efficient Model Based Approach.* In Proceedings of Computer Vision and Pattern Recognition, New York, 1993.
10. G. Giraudon and R. Deriche. *On corner and vertex detection.* In Proceedings of Computer Vision and Pattern Recognition, Hawaii, 1991.
11. D. Mumford, T. Shah, *Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems*, Comm. on Pure and Applied Mathematics, Vol XLII, No 5, July 1989.
12. M. Nitzberg, D. Mumford, T. Shiota, *Filtering, Segmentation, and Depth*, Springer Verlag Berlin 1993.
13. D. Beymer, Massachusetts Institute of Technology Master's thesis, *Junctions: Their detection and use for grouping images*, 1989.
14. L. Parida, D. Geiger, B. Hummel, *Junction Detection Using Piecewise Constant Functions*, 1996.
15. J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Annals Statistics, vol. 11, pp, 416-431, 1983.
16. V. Caselles, B. Coll, J. M. Morel, *A Kanizsa Programme*, Technical Report, Universitat de les Illes Balears, Spain, 1996.
17. D. Geiger, K. Kumaran, L. Parida, *Visual Organization for Figure/Ground Separation* , CVPR-96, San Francisco, 1996.
18. Y. Lamdan, H. J. Wolfson, Geometric Hashing: A general and efficient model-based recognition scheme in *Second IEEE International Conference on Computer Vision*, 238-249, 1988.
19. J. Morel, S. Solimini, Variational Methods in Image Segmentation, Birkhauser Boston, 1995.

This article was processed using the LaTeX macro package with LLNCS style