Junctions: Detection, Classification and Reconstruction

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

Abstract

Junctions are important features for image analysis and form a critical aspect of image understanding tasks such as object recognition. We present a unified approach to *detecting* (location of the center of the junction), *classifying* (by the number of wedges – lines, corners, 3-junctions such as T or Y junctions, or 4-junctions such as X-junctions) and *reconstructing* junctions (in terms of radius size, the angles of each wedge and the intensity in each of the wedges) in images. Our main contribution is a modeling of the junction which is complex enough to handle all these issues and yet simple enough to admit an effective dynamic programming solution. Broadly, we use a template deformation framework along with a gradient criterium to detect radial partitions of the template. We use the minimum description length principle to obtain the optimal number of partitions that best describes the junction. Kona [27] is an implementation of this model. We (quantitatively) demonstrate the stability and robustness of the detector by analyzing its behavior in the presence of noise, using synthetic/controlled apparatus. We also present a qualitative study of its behavior on real images.

I. Introduction

A critical component of most recognition systems is a 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 [6], [7], [33]. They are also critical features for motion as suggested by Walach experiments [34] and by Movshon and Adelson [1]. A study of the role of junctions in stereo is presented in Malik [21].

There have been basically two different views/approaches for detecting junctions: edge detection followed by grouping of edges to form junctions [25], [24], [4], [2], [13], and, treating a junction as a template matching phenomenon [8], [12], [15]. In the former, it is assumed that the presence (or absence) of a junction is determined by "grouping" the intensity gradients 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 [25] are interesting but not exhaustive. In the latter approach 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.

Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, U.S.A. This research has been supported by AFOSR grant # F49620-96-1-0159 and DARPA grant # MDA972-93-1-0011.

We use the template deformation framework to develop a "junction detector," to find 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 a neighborhood of the point. Our approach is to use a combination of the two paradigms: grouping of edges (via Dynamic Programming) and fitting templates. We use a template deformation framework, using the minimum description length (MDL) principle, and we include the gradient criterium in order to detect the radial partitions of the template as a grouping mechanism. 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. The minimum length encoding principle was first suggested by J. Rissanen [28], [30], [31], and is very close in spirit to the Kolmogorov sufficient statistic, as discussed in [3].

Kona [1] [27] is an implementation of our model. To test the stability/robustness of the detector we have experimentally analyzed its behavior for the location and junction parameters against noise. We used a synthetic/controlled apparatus for a quantitative study. We have also qualitatively studied the stability/robustness of detector for real stereo images.

Our model is in contrast to other approaches which we briefly review here. In [13] , in order to find corners, a local operator is first applied to detect possible junction locations. Then, inside an area around this, the edge lines are detected and depending on this number (of edge lines), a parametric model is used to fit the data is fit in a small region (25×25 pixels).

The idea of performing local feature detection by projecting image data onto a subspace is fundamental in [8], [10]. 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 [8], [10], and lines [9]. A generalization to junctions is suggested in [32], using steerable filters, and in [18], through the use of principal component analysis (PCA). A possible drawback for the PCA is that it does not have an explicit model of the junctions (to be adapted).

The authors in [15], [12] employ an iconic model of the junctions, giving forms similar to our $\mathcal{R}$ (equation (5) in Section II). See Section II for a comparison.

In [16], 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 [17].

The paper is organized as follows: Section II describes the junction model. Section III studies detailed issues of the optimization process to estimate the junction parameters. Section IV presents the results with a study of the stability of the detector. Section V concludes the paper.

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

II. 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 disc centered at this point (see Figure 1). 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" [11] that determines the position where the feature is located in a region, possibly pre-defined. Another corner-detector by [35] performs a similar role of detecting locations that have a corner. This uses a predetermined window/mask on each pixel and compares the intensity of every pixel with that of the center of the mask to determine whether there is a corner at the location.

We can formulate the junction detection problem as one of finding the parameter values that yield a junction that best approximates the local data using minimum description, 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. Further, let I denote the input signal.

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

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

where $\lambda \geq 0$.

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.

We note that the input to the system is an image I^{raw} that can be filtered by a Gaussian mask to yield a smooth image I. Smoothing I^{raw} is needed due to the discretization of the image lattice and how it affects the junction detection. Our analysis of this effect is shown in Section IV-B.

The second term, $\mathcal{G}$, is a measure of the distance of the *gradient* using the L^2 norm.

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

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

Note that,

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

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

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,

$$\mathcal{G} = \mathcal{A}\mathbf{e}_\theta + \mathcal{R}\mathbf{e_r},$$

where,

$$\mathcal{A} = \int_0^\infty \int_0^{2\pi} \frac{1}{r^2}\left(\frac{\partial I}{\partial \theta} - \frac{\partial T}{\partial \theta}\right)^2 g^*(r) r dr d\theta, \tag{4}$$

$$\mathcal{R} = \int_0^\infty \int_0^{2\pi} \left(\frac{\partial I}{\partial r}\right)^2 g^*(r) r dr d\theta. \tag{5}$$

Taking $g^*(r) = r^2 G_\sigma(r)$ where $G_\sigma(r)$ is the Gaussian function with standard deviation σ, our function $\mathcal{R}()$ is the same as the regularity measure $S()$ for junctions as in equation (4) of [15]. Also, it is the same as ω_d defined in equation (31) in [12] which is the measure of spirality omitting the term normal to the radial direction. However, in [12], a general method is presented to detect circular symmetry; thus our function $\mathcal{R}$ could be considered as a special case.

A. On the Exact form of Template T & Energy E

This section gives a more rigorous definition of T and E as a *continuous* function of two variables. The reader may skip this section without loss of continuity.

Let the image be given by $I(x, y)$. T is a template that partitions the image in N regions, an "N-junction" template. Between the partition lines (wedges), say p and $p+1$, we assign a constant value T_p for the template. Let us further consider a template for the edges that is zero everywhere except along the partition lines, $\theta = \theta_p$. We can write both templates as

$$\begin{aligned} T(\theta) &\equiv \{(T_p, \theta_p); p \in (1, 2, \ldots, N)\} = \sum_{p=1}^{N} T_p u(\theta - \theta_p) u(\theta_{p+1} - \theta) \\ T'(\theta) = e(\theta) &\equiv \{(e_p, \theta_p); p \in (1, 2, \ldots, N)\} = \sum_{p=1}^{N} e_p \delta(\theta - \theta_p)\ , \end{aligned} \tag{6}$$

where $\theta_{N+1} \equiv \theta_1$, $\delta(x)$ is the Dirac delta function and $u(x)$ is the step function (or Heaviside function), i.e.

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1 \quad \delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases} \quad \text{and} \quad u(x) = \int_{-\infty}^{x} \delta(x)\, dx = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}.$$

To fit a junction template $T(\theta)$ to the image $I(x, y)$ and to fit an edge template $e(\theta)$ to the image gradient along the angle θ (perpendicular to the radial line), an error function based on the L^2 norm weighted by a function $g(r)$ is used. Using the polar coordinate system,

$$r=\sqrt{x^2+y^2}\,,\ \theta=\arctan\frac{y}{x}\,,\ \text{and}\quad x=r\,\cos\theta\,,\ y=r\,\sin\theta\,,$$

the image gradient is

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

Thus the energy E at a point (i,j) is:

$$E=\int_0^\infty\int_0^{2\pi}\Big\{[I(r,\theta)-T(\theta)]^2g(r)+\lambda|\frac{1}{r}\frac{\partial I(r,\theta)}{\partial\theta}-e(\theta)|^2g^*(r)\Big\}rdrd\theta, \tag{7}$$

where $g(r)$ and $g^*(r)$ are functions that weights the importance of the piecewise constant fit according to the distance from the center $r=0$.

III. Energy Minimization

Recall the energy equation:

$$\begin{aligned}E &= \lambda\mathcal{R}+(\mathcal{D}+\lambda\mathcal{A})\\ &= \lambda\mathcal{R}+\overline{E}.\end{aligned} \tag{8}$$

$\mathcal{R}$ is independent of the junction template and is used to filter out the image locations. $\overline{E}$ is minimized to obtain the most appropriate junction parameters. We now discuss the details of computing $\mathcal{R}$ (scale and location) and $\overline{E}$ (junction parameters).

A. Scale and Location (On $\mathcal{R}$)

We have considered $g(r)=1/r$. Notice that for λ to be used in equation (1), $g^*(r)=r^2g(r)=r$ is a choice that makes $\mathcal{D}$ and $\mathcal{G}$ have the same unit (intensity^2/length). (Thus λ is unitless as expected.)

Estimating R_0 and R_1 (scale): We further refine $g(r)$ as follows:

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

Notice that we introduce a "hole" of size $R_0>0$ and R_1 is the size of the window. To compute R_0 and R_1, we study $\mathcal{R}^r$, which is $\mathcal{R}$ for r, the radius being considered. Figure 2 shows the plot of the relative values of $\mathcal{R}^r$, e^r, which is

$$e^r=\frac{\mathcal{R}^r}{\mathcal{R}^{r+1}}. \tag{9}$$

Notice that $\mathcal{R}$ has comparatively larger values close to the origin, which we remove by introducing a "hole". We look for a range of r values where e^r is less than a threshold, τ_w. This leads to obtaining two values that correspond to R_0 and R_1. See Figure 2 and 3 for examples.

The weight function $a(x)$ which is defined in equation (37) in [12] is similar to our $g(r)$ function.

The use of a hole is also supported by psychophysical experiments [34] that suggest humans may also utilize such a method. The experiments suggested by Wallach [34] have shown that different motion effects can be produced depending on the junction movement. In a Wallach type of experiment, junctions occur due to the intersection of lines, and, junctions move due to the movement of the lines. The results of the experiments in [34] seem to suggest that junctions are detected although there is a small gap (R_0) between the lines.

Estimating the location: We do the following to select the best location in an image region[2]. We compute $\mathcal{R}^r = \mathcal{R}/r$ for all the points in the region (with not necessarily the same window size). The one with the minimum value defines the location. We illustrate this in Figure 4.

B. Reconstructing the junction (On $\overline{E}$)

In this section we explain how the junction parameters (number of wedges, wedge angles and wedge intensities) are estimated. Recall that the last section discussed the estimation of the radius (size) of the junction and the size of the hole.

We can carry out an appropriate numerical integration to obtain the value of the first term $\mathcal{R}$. 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 [3], $\{\theta_p\}, \{T_p\}, p = 1 \ldots N$, 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, whereas $\mathcal{V}$ does. Recall $g^*(r) = r^2 g(r)$. With some straightforward manipulations we obtain

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

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

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

$$\tilde{\mathcal{V}} = \sum_{p=1}^{N} \left[(\theta_{p+1} - \theta_p)(-2T_p \bar{I}_{\theta_p,\theta_{p+1}} + T_p^2 C) + \lambda(-2T'_p(\partial_\theta \tilde{I})_{\theta_p} + C^* T'^2_p) \right],$$

where,

$$C = \int_0^\infty g(r) r dr = R_1 - R_0, \quad (10)$$

$$\tilde{I}(\theta) = \int_0^\infty I(r,\theta) g(r) r dr, \quad (11)$$

[2] Notice that $\mathcal{R}$ has unit (intensity)^{2}length; thus $\mathcal{R}^r$ has unit (intensity)^2.

[3] 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.

$$
\begin{aligned}
C^* &= \int_0^\infty g^*(r)\, r\, d\, r = \int_0^\infty r^2\, d\, r = \frac{R_1^3 - R_0^3}{3}, \\
(\partial_\theta \tilde{I})_{\theta_p} &\equiv \frac{\partial \tilde{I}(\theta)}{\partial \theta}|_{\theta_p} = \frac{\partial \int_0^\infty I(r,\theta) g(r) r\, d\, r}{\partial \theta}|_{\theta_p}, \\
\bar{I}_{\theta_p,\theta_{p+1}} &= \Delta\theta \sum_{j=j_p}^{j_{p+1}} \tilde{I}(j\, \Delta\theta).
\end{aligned} \tag{12}
$$

Further, $\tilde{I}(\theta)$, $(\partial_\theta \tilde{I})_{\theta_p}$ can be approximated numerically. For the sake of brevity, we have omitted the details of the derivations here, and present it in Appendix A.

Estimating wedge angles and intensities: Although the energy equation E 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. $\mathcal{R}$ 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 $\tilde{I}(\theta)$ (equation (11)) can be viewed as integrating the intensity along a radial line. Thus, the two-dimensional image is projected on to a one-dimensional coordinate θ, appropriately discretized.

Let $\tilde{I}(\theta_i)$ be defined for $\theta_1, \theta_2, \ldots, \theta_d$, and, $(\theta_{i+1} - \theta_i) = 2\pi/d$, $\forall i$ [4]. For a p-junction, $T_1, T_2, \ldots, T_p$, are the template intensities and the wedge boundaries are at $\theta_{k_1}, \theta_{k_2}, \ldots, \theta_{k_p}$, $\theta_{k_1} \leq \theta_{k_2} \ldots \leq \theta_{k_p}$.

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 $\lambda = 0$, $\frac{\partial \mathcal{V}}{\partial T_p} = f(\theta_p)$, for some function $f()$. Thus fixing θ_p's, the T_l's can be shown to be the following:

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

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

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

When $\lambda \neq 0$, $\overline{E}^p$ has some extra terms. It can be verified that T_l is the same as before, and, $\overline{E}^p$ is:

$$
\overline{E}^p = \sum_{l=1}^{p} \sum_{j=k_l}^{k_{l+1}} \left(\tilde{I}(\theta_j) - T_l \right)^2 + \lambda \sum_{i=k_1}^{k_p} (\tilde{I}(\theta_{k_i}) - \tilde{I}(\theta_{k_i - 1}))^2.
$$

We compute the θ_p's by exploring all possible set of θ_p's. We summarize the dynamic program

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

[4] If $i = d$, $i + 1$ is defined to be 1.

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

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

This dynamic programming solution has been implemented in Kona.

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

Estimating the number of wedges: We estimate the number of wedges by measuring the rate of increase of energy by thresholding the ratio of the energy measurement, $\overline{E}^n$. More precisely, the (optimal) number of wedges, N, is computed by thresholding the relative error, r^n,

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

Notice that by definition r^n is unitless. Although, in principle, we are looking for the minimum r^n, in practice we terminate the computation when r^n drops below a pre-defined threshold, say τ_c, as n is increased. (See Figure 5 for an example.) Note that as the number of parameters increases, $\overline{E}^n$ decreases, i.e., $r^n < 1$.

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

1. Compute $\mathcal{R}$, the measure of radial variation, and R_0, the size of the hole and R_1, the size of the template at every point.
2. Filter the locations using a threshold on $\mathcal{R}^r$.
3. In a neighborhood of a filtered 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 filtered locations.
4. Compute the junction parameter for all the filtered locations.

Kona is an implementation of this junction model. It is programmed in C. A version in C^{++} and a parallel implementation of Kona is also available.

IV. Results of Experiments

We carried out a series of experiments on synthetically generated images (see Figures 6 and 7) to test the stability of the algorithm in the presence of noise in the image, and to further understand the role of smoothing. In these experiments, we suppressed the use of threshold values using a fixed $R_1 = 4$ and fixed $N = 3$, i.e., forcing the algorithm to pick up the optimal 3-corners. We then carried the experiments on real images to test the general performance of the algorithm.

A. Stability of the Algorithm

Figure 6 shows the result of one such experiment where we look for 3-corners in the center region of the image. It shows that when the image edges are sharp ($\sigma = 0 \ldots 3$ in the figure), the error in the angles and the intensities are slightly higher. The errors are least in the range of $\sigma = 6 \ldots 20$, and increase further on.

Note that the (Manhattan) distance of the location of the 3-corner is less than 6 pixels; the error in the angles is bounded by 30° at the very worst and the intensity differs by 40 units in the worst situation. It also shows that, in the best situation the angle differs, on an average, by 15° from the true answer.

B. Effect of Image Smoothing

We present in Figure 7 the results of one such experiment. The images used here are the smoothed version of the ones shown in Figure 6. Note the reduction in error around the sharp images or low values of σ.

It is worth noting that even for sharp junctions (without noise added) the algorithm performs better when smoothing is introduced. It is possible that smoothing the image just along radial directions would be more appropriate, to avoid smoothing along wedges. In our experiments this possible effect was not noticed, while the substantial speed/efficiency of filtering on x and y (and not along r) encouraged us to use an overall Gaussian blur.

C. Results on Real Images

The threshold for deciding N of the junction, τ_c, was set to 0.4, and the threshold for deciding R_0 and R_1, τ_w, was set to 2.1, for all the images we used[5]. We observed that using the number of discretization units of the angle as 16 has worked well: 8 was found to be too coarse and 32 did not result in any significant higher accuracy[6].

Figure 8 shows the function/template that is fit to different points on the images. Figure 9 shows an image where the X-junctions have been detected and Figure 10 shows the results of Kona on a region of the image. After the filtering it computes the parameters for about sixteen junctions in a minute, on a Sun Sparc Station.

We demonstrate the stability of the junction detector by running it on a pair of stereo images as shown in Figure 11. (The results of this is intended for use by stereo algorithms.) The analysis of the result of this is presented in Figure 12. We measure the stability by first finding a correspondence between location (x_l, y_l) in the left image and (x_r, y_r) in the right stereo image and, then, comparing the number of junctions in a small neighborhood of the two locations.

V. Conclusions

To conclude, we summarize the contributions of this paper:

- A modeling of the junction that includes the gradient term in the model, and, supports the removal of a small disc at the center of the junction.

[5] Recall that τ_c and τ_w are thresholds for relative errors, hence are unitless.
[6] We choose a multiple of 4 because of the four quadrants – using any arbitrary number could do as well, in principle.

- Using the model to obtain the scale and location of a junction.
- 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.

Kona is an implementation on this model. This detector is being currently tested in a multitude of tasks such as object recognition (based on geometric hashing), detection of illusory contours, and 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.

VI. Acknowledgments

We are grateful to J. Morel and T. Binford for their comments and suggestions. We wish to thank the anonymous referees for their detailed comments that substantially improved the paper. Thanks to R. Chandrasekar for his careful reading of the paper.

References

[1] E.H. Adelson, J.A. Movshon, Phenomenal Coherence of Moving Visual Patterns, Nature, vol 300, pp 523-525, 1982.

[2] V. Caselles, B. Coll, J. M. Morel, *A Kanizsa Programme*, Technical Report, Universitat de les Illes Balears, Spain, 1996.

[3] Cover T., Thomas J., *Elements of information theory* , John Wiley & Sons, Inc., 1991.

[4] W. Freeman, E. Adelson, *Junction Detection and Classification*, Proc. ARVO 1991.

[5] D. Geiger, K. Kumaran, L. Parida, *Visual Organization for Figure/Ground Separation* , CVPR-96, San Francisco, 1996.

[6] A. Guzman, *Decomposition of a Visual Scene into Three-Dimensional Bodies*, Proc. AFIPS 1968 Fall Joint Computer Conference, 1968.

[7] D.A. Huffman, *A Duality Concept for the Analysis of Polyhedral Scenes*, Machine Intelligence, Vol 6, Edinburgh Univ. Press, Edinburgh, U.K., 1971.

[8] M. F. Hueckel, *An operator which locates edges in digitized pictures*, J, Assoc. Compt. Mach. Vol 18, pp 113-125, 1971.

[9] M. F. Hueckel, *A local operator which recognizes edges and lines*, J, Assoc. Compt. Mach. Vol 20, pp 634-647, 1973.

[10] R. Hummel, *Feature Detection Using Basis Functions*, Computer Graphics and Image Processing, Vol 9, pp 40-55, 1979.

[11] 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, pp 281-305, 1987.

[12] J. Bigun, *A structure feature for some image processing applications based on spiral functions*, Compter Vision, Graphics and Image Processing, vol 51, pp. 166-194, 1990.

[13] K. Rohr, *Recognizing Corners by Fitting Parametric Models*, International Journal of Computer Vision, 9:3, pp. 213-230, 1992.

[14] U. Weidner, *Parametric Information-Preserving Surface Restoration*, Lecture Notes in Computer Science, Vol 801, Jan-Olof Eklundh (Ed), Computer Vision - ECCV '94, pp 218-224, 1994.

[15] W. Forstner, *A Framework for Low Level Feature Extraction*, Lecture Notes in Computer Science, Vol 801, Jan-Olof Eklundh (Ed), Computer Vision - ECCV '94, pp 383-394, 1994.

[16] 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, pp. 530-535, June, 1993.

[17] G. Giraudon and R. Deriche. *On corner and vertex detection.* In Proceedings of Computer Vision and Pattern Recognition, Hawaii, pp 650-655, 1991.

[18] Parametric Feature Detection S. Nayar, S. Baker, H. Murase *Parametric Feature Detection*, CVPR-96, San Francisco, 1996.

[19] K. Brunnstorm, T. Lindeberg, J. O. Eklundh, *Active Detection and Classification of Junctions by Foveation with a head-eye system Guided by the scale-space primal sketch*, TR ISRN KTH/NA/P-91/31-SE, CVAP, Royal Institute of Technology, 1992.

[20] B. M. Ter Haar Romey, L. M. J. Florack, J. J. Koenderink, M. A. Viergever, *Invariant Third-order Properties of Isophotes: T-junction Detection*, In Procc. Scandinavian Conference on Image Analysis, August, 13-16, 1991.

[21] J. Malik. *On Binocularly viewed occlusion junctions.* In Lecture Notes in Computer Science-Computer Vision-ECCV'96, Eds. B. Buxton and R. Cipolla, Springer Verlag, 1996.

[22] J. Morel, S. Solimini, Variational Methods in Image Segmentation, Birkhauser Boston, pp 47-58, 1995.

[23] D. Mumford, T. Shah, *Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems*, Comm. on Pure and Applied Mathematics, Vol XLII, No 5, pp 557-685, July 1989.

[24] M. Nitzberg, D. Mumford, T. Shiota, *Filtering, Segmentation, and Depth*, Springer Verlag Berlin 1993.

[25] D. Beymer, Massachusetts Institute of Technology Master's thesis, *Junctions: Their detection and use for grouping images*, 1989.

[26] D. J. Beymer, *Finding Junctions using the Image Gradient*, In Computer Vision and Pattern Recognition, pp. 720-721, Lahania, Maui, Hawai, June 3-6, 1991.

[27] L. Parida, D. Geiger, B. Hummel, *Kona: A Multi-Junction Detector Using Minimum Description Length Principle*, Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'97), Marcello Pelillo, Edwin Hancock (Eds), LNCS vol 1223, pp 51-65, May 1997.

[28] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Annals Statistics, vol. 11, pp, 416-431, 1983.

[29] Lamdan Y., Wolfson H. J., Geometric Hashing: A general and efficient model-based recognition scheme in *Second IEEE International Conference on Computer Vision*, 238-249, 1988.

[30] J. Rissanen. Stochastic complexity and modeling. *Ann. Stat.*, 14:1080-1100, 1986.

[31] J. Rissanen. Stochastic complexity (with discussions). *Journal of the Royal Statistical Society*, 49: 223-239, 252-265, 1987.

[32] E. Simoncelli and H. Farid, *Steerable wedge filters for local orientation analysis.* IEEE Transactions on Image Processing, Sept. 1996.

[33] D. L. Waltz, *Understanding Line Drawings of Scenes with Shadows*, The Psychology of Computer Vision, McGraw-Hill, New York, 1972.

[34] S. Wuerger, R. Shapley, N. Rubin, "On the visually perceived direction of motion" by Hans Wallach: 60 years later, vol 25, pp 1317-1367, 1996.

[35] S. Smith and J. Brady, *SUSAN - a New Approach to Low Level Image Processing*, DRA Technical Report TR95SMS1c, Defense reasearch Agency, Farnborough, Hampshire, GU14 6TD, UK, 1995.

Appendix

I. Computing E

Exploiting the radial invariance of the templates $T(\theta)$ and $e(\theta)$ we can write equation (7) as follows:

$$
\begin{aligned}
E &= \int_0^\infty \int_0^{2\pi} [I^2(r,\theta) - 2\,T(\theta)\,I(r,\theta) + T^2(\theta)]\,g(r)\;r\;d\,r\;d\,\theta \\
&\quad + \lambda \int_0^\infty \int_0^{2\pi} [\,(\frac{1}{r}\frac{\partial I(r,\theta)}{\partial \theta})^2 - 2\,e(\theta)\,\frac{1}{r}\frac{\partial I(r,\theta)}{\partial \theta} + e^2(\theta)\,]\,g^*(r)\;r\;d\,r\;d\,\theta \\
&= \int_0^\infty \int_0^{2\pi} [\,I^2(r,\theta) + \lambda(\frac{\partial I(r,\theta)}{\partial \theta})^2\,]\,g(r)\;r\;d\,r\;d\,\theta
\end{aligned}
$$

$$+\sum_{p=1}^{N} lim_{\epsilon\to 0}\int_{\theta_p-\epsilon}^{\theta_{p+1}-\epsilon}\int_0^\infty [T_p^2 - 2T_p I(r,\theta) + \lambda(re^2(\theta) - 2re(\theta)\frac{\partial I(r,\theta)}{\partial\theta})]g(r)rdrd\theta,$$

where we have assumed $g^*(r) = r^2 g(r)$ We can now exploit the radial invariance of the templates to perform the integrals over dr. Define the following:

$$\begin{aligned}
\mathcal{F} &= \int_0^\infty\int_0^{2\pi}[\,I^2(r,\theta) + \lambda\left(\frac{\partial I(r,\theta)}{\partial\theta}\right)^2\,]\,g(r)\;r\;d\,r\;d\,\theta,\\
C &= \int_0^\infty g(r)r\;d\,r = R_1 - R_0,\\
C^* &= \int_0^\infty g^*(r)\;r\;d\,r = \int_0^\infty r^2\;d\,r = \frac{R_1^3 - R_0^3}{3},\\
\tilde{I}(\theta) &= \int_0^\infty I(r,\theta)g(r)r\;d\,r. \qquad (13)
\end{aligned}$$

Note that

$$\int_0^\infty \frac{\partial I(r,\theta)}{\partial\theta}g(r)r\;d\,r = \frac{\partial}{\partial\theta}\int_0^\infty I(r,\theta)g(r)r\;d\,r = \frac{\partial\tilde{I}(\theta)}{\partial\theta}\,.$$

The radial integrals can be performed numerically. Thus,

$$E = \mathcal{F} + \sum_{p=1}^{N} lim_{\epsilon\to 0}\int_{\theta_p-\epsilon}^{\theta_{p+1}-\epsilon}[-2T_p\tilde{I}(\theta) + T_p^2 C + \lambda(-2\,re(\theta)\frac{\partial\tilde{I}(\theta)}{\partial\theta} + C^* e^2(\theta))]d\theta.$$

We can now perform the integrals over $d\theta$ to obtain the following:

$$E \;=\; \mathcal{F} + \sum_{p=1}^{N}[(\theta_{p+1}-\theta_p)(-2T_p\bar{I}_{\theta_p,\theta_{p+1}} + T_p^2 C) + \lambda(-2re_p(\partial_\theta I)_{\theta_p} + C^* e_p^2)],$$

where we have used that $e(\theta) = \sum_{p=1}^N e_p\delta(\theta-\theta_p)$, and,

$$\begin{aligned}
(\partial_\theta I)_{\theta_p} &= \frac{\partial\tilde{I}(\theta)}{\partial\theta}|_{\theta_p} = \frac{\partial\int_0^\infty I(r,\theta)g(r)r\;d\,r}{\partial\theta}|_{\theta_p},\\
e_p F(\theta_p) &= \lim_{\epsilon\to 0}\int_{\theta_p-\epsilon}^{\theta_{p+1}-\epsilon} e(\theta)F(\theta)\,d\theta,\\
\bar{I}_{\theta_p,\theta_{p+1}} &= \frac{1}{(\theta_{p+1}-\theta_p)}\int_{\theta_p}^{\theta_{p+1}}\tilde{I}(\theta)\;d\,\theta,\\
&= \Delta\theta\sum_{j=j_p}^{j_{p+1}}\tilde{I}(j\,\Delta\theta).
\end{aligned}$$

where the integral $\int_{\theta_p}^{\theta_{p+1}}\tilde{I}(\theta)\;d\,\theta$ is adapted to the image lattice, by computing $\tilde{I}(\theta)$ at every interval $\Delta\theta$ and with $(\theta_{p+1}-\theta_p) = (j_{p+1}-j_p)\Delta\theta$.

II. Numerical Approximation of $\tilde{I}(\theta)$

Using the definition of $g(r) = 1/r$ and equation (11), we have

$$\begin{aligned} \tilde{I}(\theta) &= \int_0^{\infty} I(r,\theta) g(r) r \, dr \\ &= \int_{R_0}^{R_1} I(r,\theta) \left(\frac{1}{r}\right) r \, dr \end{aligned}$$

To carry out the numerical approximation we define a grid, as shown in Figure 13, such that each cell in the grid corresponds to a pixel on the image. A ray centered at (x_0, y_0) of length R_1 is broken by the grid into k segments $l_1, l_2, \ldots, l_k$ where each segment lies in a rectangular cell (pixel) with intensity i_j, $j = 1, 2, \ldots, k$. Note that $R_1 = l_1 + \ldots + l_k$. Thus we have the following approximation: $\tilde{I}(\theta) \approx \sum_{j=1}^{k} l_j i_j$.

Further, if $R_0 > 0$, the segments are $l_m, l_{m+1}, \ldots . l_k$, for some $1 \leq m \leq k$, and, $R_1 = l_m + l_{m+1} + \ldots + l_k$, and $\tilde{I}(\theta) \approx \sum_{j=m+1}^{k} l_j i_j$.

□

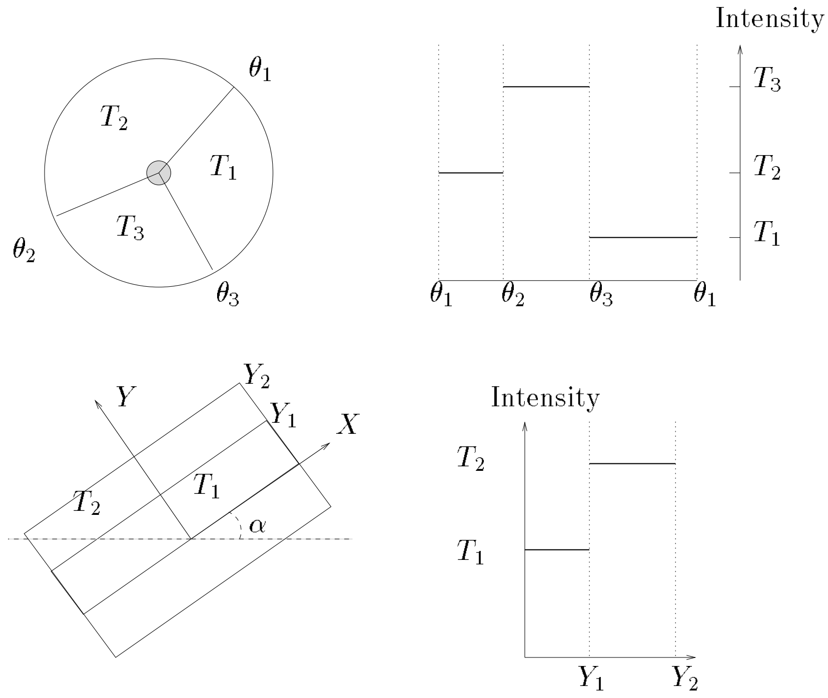
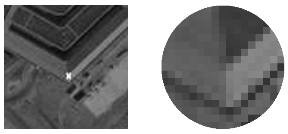
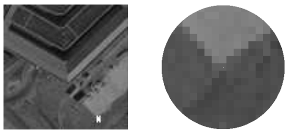
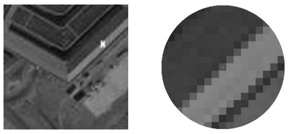
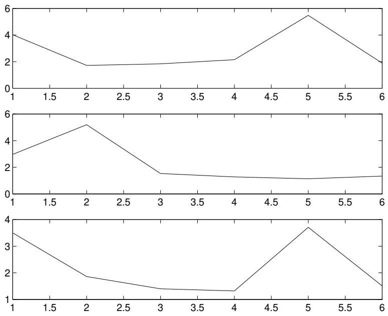


Fig. 1. Piecewise constant features. A junction detector is shown on the top row: 3-junction template with edges at angles θ_1, θ_2 and θ_3 with intensities T_1, T_2 and T_3. A bar detector is shown in the bottom row: it is characterized by an angle α which gives the inclination of the bar and the intensities T_1 and T_2 at "heights" Y_1 and Y_2 from the center of the location.

Marked Images.

		
Location 1 (46,50)	Location 2 (68,81)	Location 3 (71,27)

e^r versus r plots for the three locations.



The computed windows.

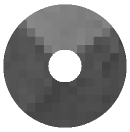
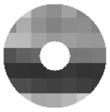
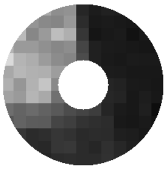
		
Location 1 (46,50)	Location 2 (68,81)	Location 3 (71,27)
$R_0 = 1$, $R_1 = 4$ pixels.	$R_0 = 2$, $R_1 = 8$ pixels.	$R_0 = 1$, $R_1 = 4$ pixels.

Fig. 2. Estimating R_0 and R_1, the size of the hole and the size of the window respectively. The relative values of $\mathcal{R}$, e^r is plotted against the radius r. The threshold value of $\tau_w = 2.1$ was used to obtain the values of R_0 and R_1 as shown. Notice that when we pick the lower bound (R_0), we take 1 unit less than the picked interval (for instance, the range for the first graph is $r = 2 \ldots 4$ and $R_0 = 1$, $R_1 = 4$). We do this since we sample the values of $\mathcal{R}$ only at integral values and $\mathcal{R}^r = \mathcal{R}^s$, $r - 1 < s \leq r$ where s is a rational.

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

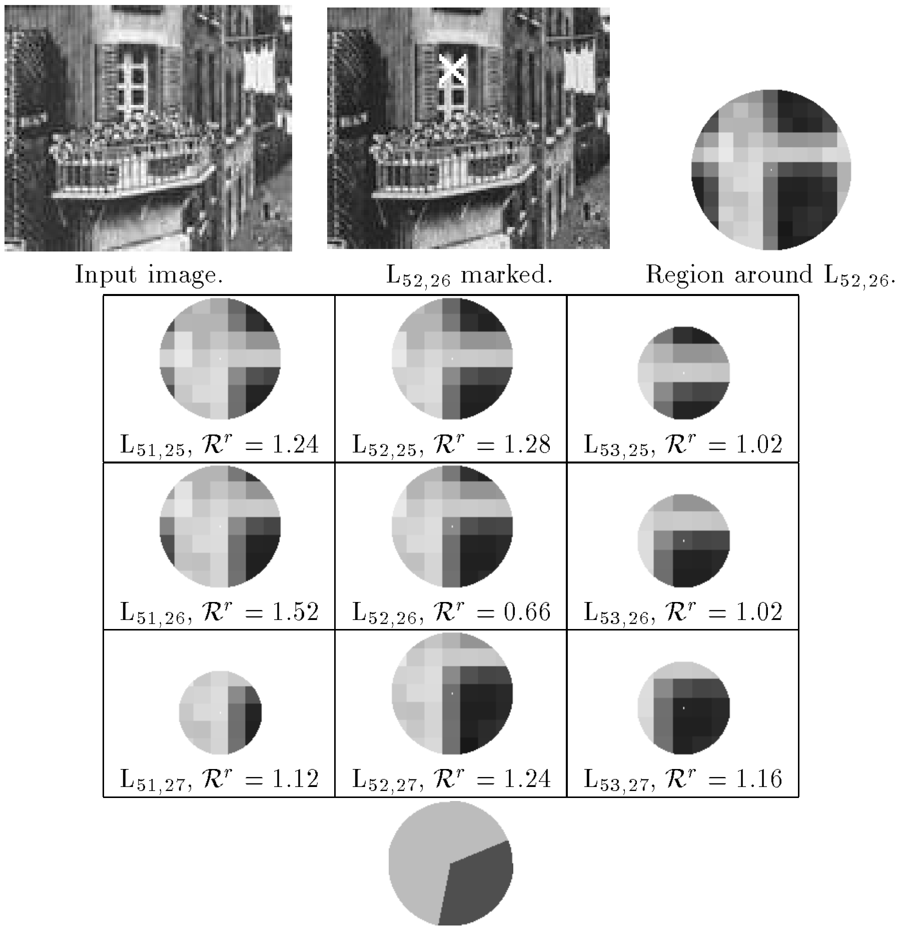
 

(c) Adaptive window at (a). ($R_0 = 1, R_1 = 3$ pixels) (d) Adaptive window at (b). ($R_0 = 2, R_1 = 6$ pixels)

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

Fig. 3. An example to show the dynamic computation of R_0 and R_1 at different locations on the image. We have used $\tau_w = 2.1$.



The junction at location $L_{52,26}$.

Fig. 4. Estimating the location of a junction: The use of $\mathcal{R}^r$, to locate the center of the junction. $L_{x,y}$ indicates the x and y coordinates on the image. For convenience, the input data is normalized, that is, the intensities are scaled to lie in $[0, 1]$. The unit of each value is $(\text{intensity})^2$. Note that the location $L_{52,26}$ has the minimum value in the neighborhood. ($\tau_w = 2.1$)

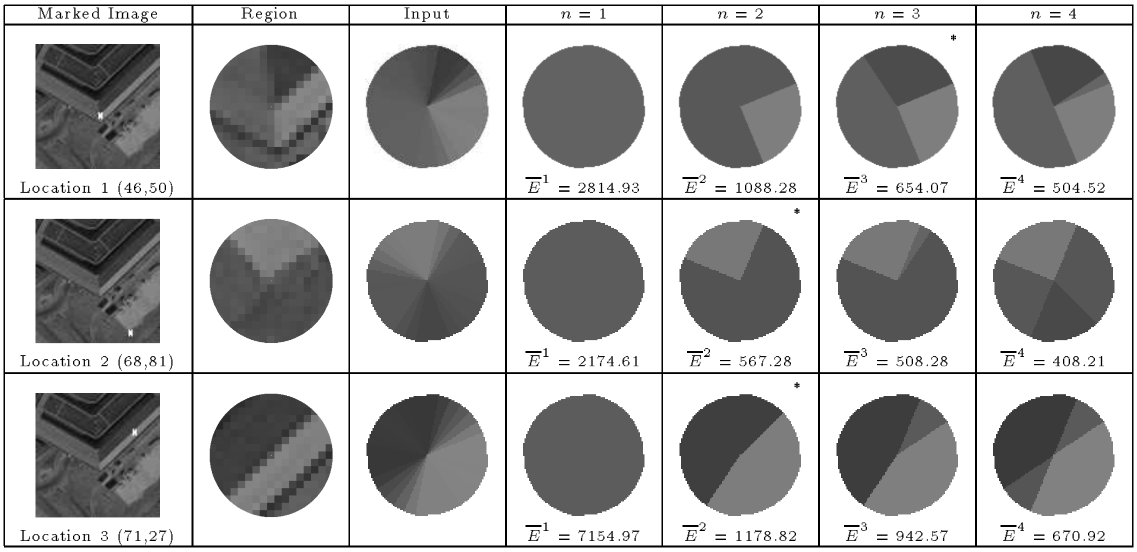


Fig. 5. Estimating the number of wedges: τ_c is used to threshold the relative errors to declare a N-junction. The asterisk denotes the computed optimal N using $\tau_c = 0.4$. The unit of $\overline{E}^n$ is (intensity)^2$*$length. The thresholding process works in the following manner: we examine the successive relative errors, in increasing n, until we reach a n where the relative error exceeds the threshold, τ_c. The previous value of n is declared as the optimal value.

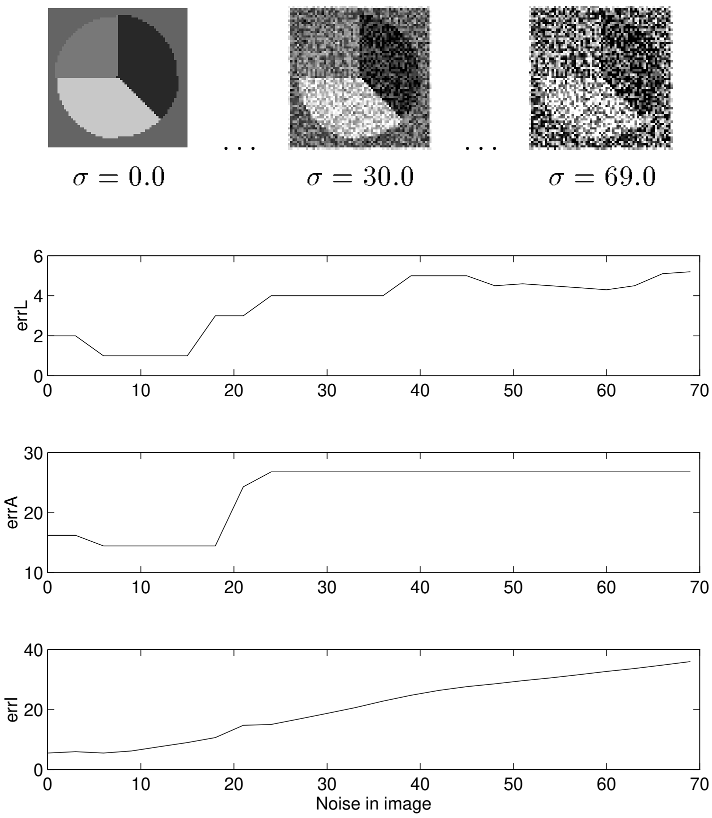


Fig. 6. Test of stability: The range of intensities in the image is $0 - 255$. A noise with a Gaussian distribution, with standard deviation σ, is introduced randomly at each pixel. The first image in the top left has $\sigma = 0$ (that is, no noise) and the angles of the 3-corner at the center, (x, y), are $a_{1f} = 90°$, $a_{2f} = 180°$ and $a_{3f} = 315°$ with intensities $i_{1f} = 120$, $i_{2f} = 200$ and $i_{3f} = 40$. σ is varied from 0 to 69 to obtain 24 images, three of which are shown in the top row for illustration. The 3-corners around the center, (x_σ, y_σ), of each image is extracted and the following errors are computed: error in location, $errL_\sigma = |x_\sigma - x| + |y_\sigma - y|$, error in intensities, $(errI_\sigma)^2 = \sum_{j=1}^{3}(i_{j\sigma} - i_{jf})^2/3$, error in angles, $(errA_\sigma)^2 = \sum_{j=1}^{3}(a_{j\sigma} - a_{jf})^2/3$. Finally, $errL_\sigma$ vs σ, $errA_\sigma$ vs σ, and, $errI_\sigma$ vs σ are plotted. See the text for further explanations.

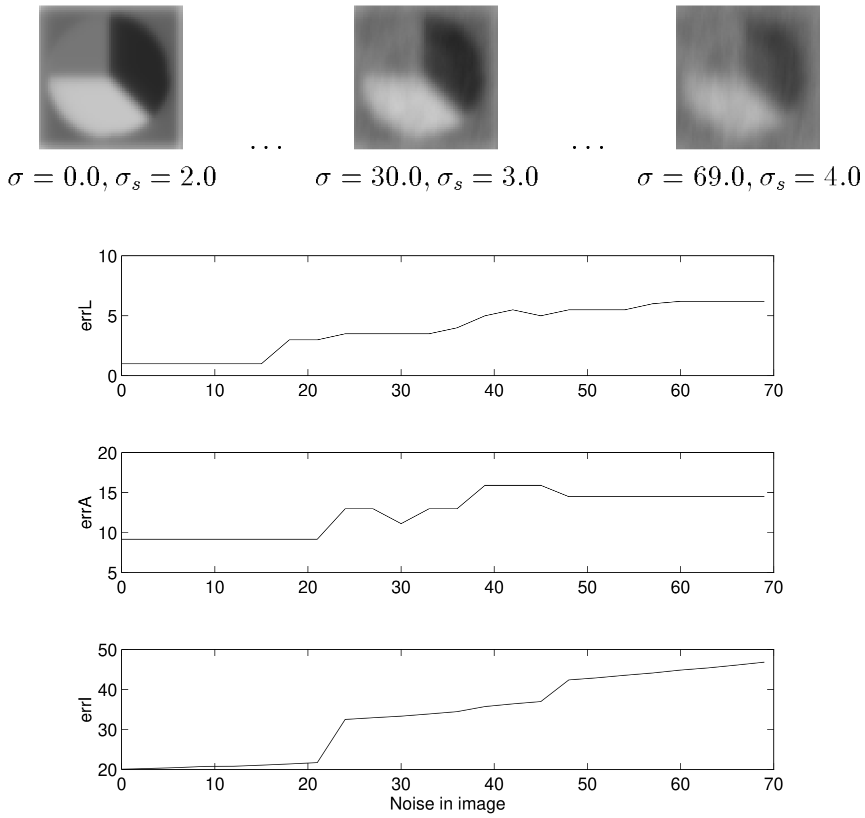


Fig. 7. Test of stability over smoothed noisy images: We use the same set of (noisy) images as used in Figure 6, except that these images are now smoothed. We increase the smoothing factor as the image gets noisier. The first eight images ($\sigma = 0, 3, \ldots, 21$) are smoothed using a $\sigma_s = 2.0$, the next eight ($\sigma = 24, 27, \ldots, 45$) uses $\sigma_s = 3.0$ and the last set ($\sigma = 48, 51, \ldots, 69$) uses $\sigma_s = 4.0$. We plot the errors as in Figure 6. As expected we are able to get rid of the high errors for the very sharp images (low values of σ).

(a) Position on the image.

(b) Position on the image.

(c) Adaptive window for (a).
($R_0 = 1, R_1 = 4$ pixels)

(d) Adaptive window for (b).
($R_0 = 2, R_1 = 6$ pixels)

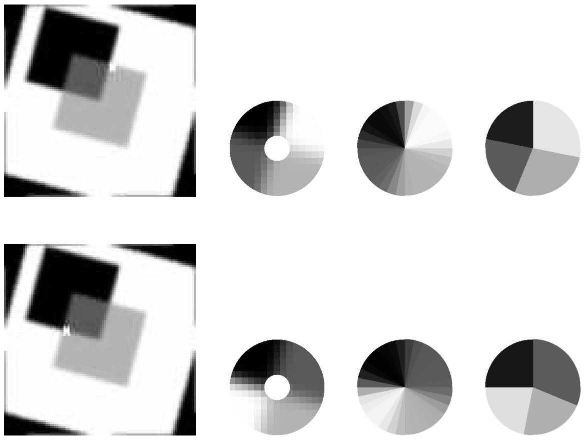
(e) Projection of (c) to 1D signal.

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

(g) Best-fit to (e).

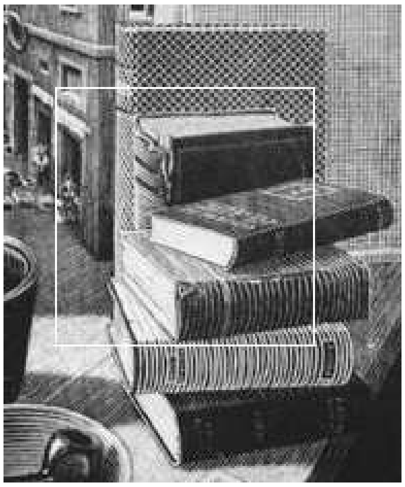
(h) Best-fit to (f).

Fig. 8. Examples of a sharp corner (column 1) and a T-junction on an image (column 2). We have used $\tau_w = 2.1$ to get R_0 and R_1 values, and $\tau_c = 0.4$ to get the N-junctions.

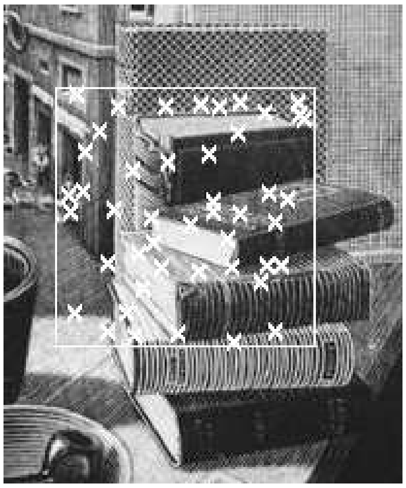


(a) (b) (c) (d)

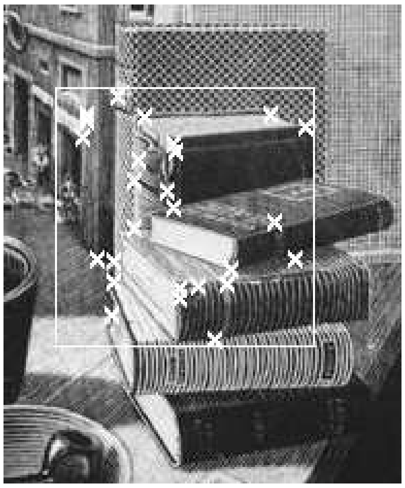
Fig. 9. The image of a transparent object occluding a dark object in the background. We show two X-junctions detected and reconstructed by the algorithm. (a) The marked images. (b) The window around the marked points. (c) The corresponding 1D signals. (d) The 4-junctions at the points.



(a) Marked input image.

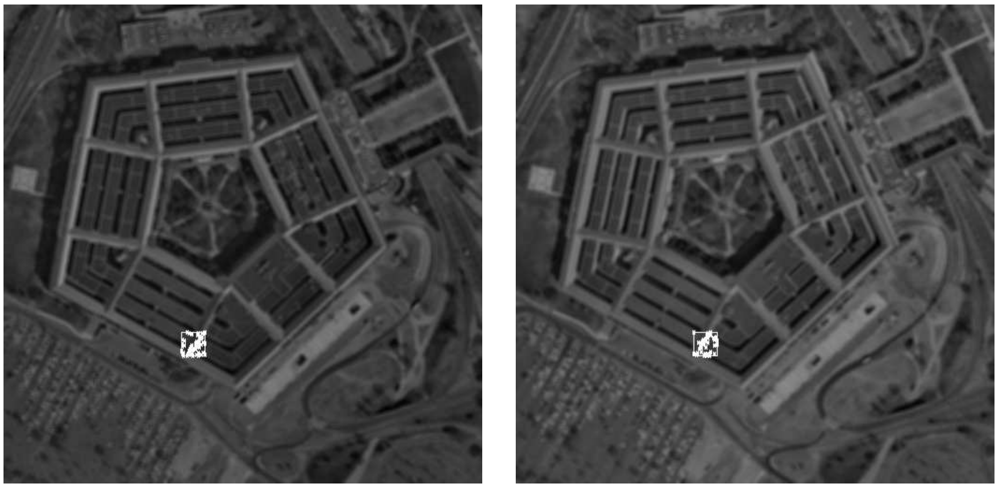


(b) 2-junctions.



(c) 3-junctions.

Fig. 10. Example of an image showing the results of the junction detector Kona. The 2-junctions have been filtered which has removed the large angle corners (>160 degrees). Also, the low-contrast 3-junctions have been filtered out. The values of the various control parameters are: $\tau_c = 0.4$ (to obtain N the number of wedges), $\tau_w = 2.1$ (to obtain the size of hole, R_0, and the radius, R_1, of the template), number of radial lines $= 32$.



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

Fig. 11. Kona is run on a 25×25 size window (at location $(190, 350)$) on a pair of stereo images. It detects junctions, with significant contrast, on similar locations on both the images. The analysis of this is presented in a simplified form in Figure 12.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	2	2
0	0	0	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**2**	**2**	**2**	**3**	**3**
0	0	0	0	0	0	**0**	0	0	0	1	1	2	2	3	3	3	3	4
0	0	0	0	0	0	**0**	0	1	1	2	2	3	3	4	4	4	4	5
0	0	0	0	0	0	**0**	0	1	1	2	2	3	4	5	5	5	5	6
0	0	0	0	0	0	**0**	0	1	1	2	3	4	5	6	6	6	6	6
0	0	0	0	0	0	**0**	0	1	1	2	3	4	5	6	6	6	5	5
0	0	0	0	0	0	**1**	1	2	2	3	4	6	6	6	6	6	5	5
0	0	0	0	1	1	**2**	2	3	4	5	5	6	6	6	6	5	4	5
0	0	0	0	1	1	**2**	2	3	4	4	4	5	5	5	5	4	4	4
0	0	0	0	1	1	**3**	3	3	4	4	4	5	4	4	4	4	4	4
1	1	1	2	3	3	**5**	4	4	5	5	5	6	4	4	4	4	3	3
2	2	2	3	4	3	**5**	4	4	5	5	4	5	3	3	3	3	2	3
2	3	3	4	5	4	**6**	6	5	6	6	5	6	4	3	3	3	2	4
3	4	4	5	6	5	**5**	5	4	5	5	4	4	3	2	2	2	1	3
3	4	4	5	5	5	**5**	5	4	4	4	4	3	2	1	1	3	2	3
3	4	4	5	5	5	**5**	5	4	4	4	4	3	2	1	1	3	2	3

Matrix corresponding to the left stereo image.

0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	2
0	0	0	0	0	1	1	1	1	2	2	2	2	2	2	2	2	2	2
0	0	0	0	0	1	2	2	2	3	3	4	4	3	3	4	4	4	3
0	0	0	0	1	2	3	3	3	4	4	4	4	3	3	4	4	4	3
0	0	1	1	2	3	4	5	5	5	5	5	5	4	3	4	4	4	3
0	0	1	1	2	4	5	6	6	6	6	6	5	4	3	4	3	3	2
0	0	1	2	3	4	5	6	6	6	5	5	5	4	3	4	3	4	3
0	1	2	3	4	5	6	7	6	6	6	6	5	4	3	4	3	3	2
0	1	2	3	4	5	6	7	6	5	5	5	4	3	2	3	3	3	2
1	2	3	4	5	7	7	7	6	5	5	4	2	2	1	1	1	2	2
1	2	3	4	4	6	6	6	5	4	4	4	2	2	1	1	1	2	2
1	2	2	3	3	5	5	4	3	3	3	3	1	1	1	1	1	2	2
1	2	2	3	3	4	4	3	2	2	2	2	1	1	1	2	2	3	3
1	2	2	2	2	3	3	2	1	1	2	2	1	1	1	2	2	2	2
1	1	1	1	1	2	2	1	1	1	1	1	0	0	0	1	1	2	2
2	2	2	2	2	3	3	1	1	1	1	1	0	0	0	1	1	2	2
1	1	2	2	2	2	2	1	2	1	1	1	1	1	1	1	1	2	2
1	1	2	2	2	2	2	1	2	1	1	1	1	1	1	1	1	2	2
2	2	3	3	3	3	2	1	2	1	1	1	1	1	1	1	1	2	2

Matrix corresponding to the right stereo image.

Fig. 12. Kona is run on a 25×25 size window as shown in Figure 11. The 19×19 matrix gives a count of the number of junctions Kona detected in a 7×7 window centered at that location. (Thus the $(1,1)$ entry of the matrix corresponds to location $(193, 353)$ of the image and gives the number of junctions detected.) By inspection, we have indicated the alignment points of the pair of images by presenting the numbers in bold. It can be seen that the number of junctions (after aligning the two images), match up reasonably well (i.e., gives good correlation values).

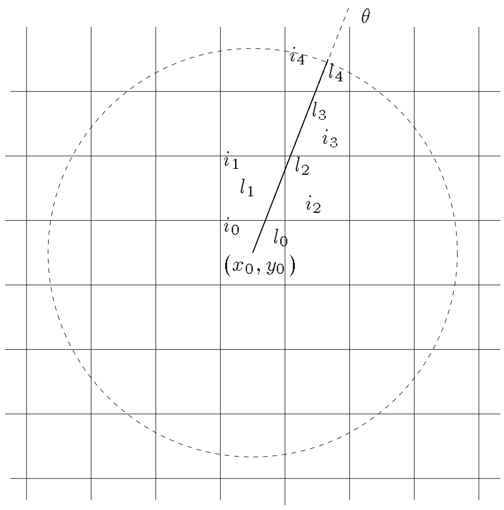


Fig. 13. The pixels shown as rectangular cells. The ray along θ shown by the solid line starting at (x_0, y_0) is split into segments $l_0, l_1, \ldots, l_4$. Each of these segment lies in a distinct pixel having intensity $i_0, i_1, \ldots, i_4$ respectively.