Feature Extraction and Grouping for Robot Vision Tasks

Miguel Cazorla & Francisco Escolano

1. Introduction

This paper focuses on feature extraction and perceptual grouping in computer vision. Our main purpose has been to formulate and test new methods for feature extraction (mainly those related to corner identification and junction classification) and grouping (through junction connection). Our purpose is to build a geometric sketch which can be useful for a wide range of visual tasks. The context of application of these methods, robot vision, imposes special real-time constraints over their practical
use, and thus, the following requirements are observed: efficiency, robustness, and flexibility.

The paper is structured in three parts which cover junction classification, grouping, and estimation of the relative orientation of the robot with respect to the environment (an example of visual task). We follow a bottom-up exposition of the topics (Cazorla, 2000):

- Junction Classification: It relies on combining corner detectors and template matching. Corner detection is performed through two well known operators: SUSAN and Nitzberg. These operators provide an initial localization of the junction center. Given this localization, junction classification is posed in terms of finding the set of angular sections, or wedges, that explain as better as possible the underlying evidence in the image. We propose two greedy methods which rely on elements of Bayesian inference. These methods are tested with indoor and outdoor images in order to identify their robustness to noise and also to bad localizations of junction centers. Experimental results show that these methods are saver than other methods recently proposed, like Kona, and their computational efficiency is even better. However, we have detected some problems due to the local extent of junction detection. These problems and those derived from bad center localization can be aliviated through local-to-global interactions, and these interactions are inferred by grouping processes.
- Grouping: Using classified junctions as starting elements, this stage is performed by finding connecting paths between wedge limits belonging to pairs of junctions, and these paths exist when there is sufficient contrast or edge support below them. Given that corners are usually associated to points of high curvature in the image, it can be assumed that connecting paths must be smooth if they exist. Contrast support and smoothness are quantified by a cost function. As it can be assumed that there will be no more than one path between two junctions through a pair of wedge limits, such a path can be found by the Bayesian version of the well known A* algorithm. This method recently proposed searches the true path, according to the cost function, in a population of false path, instead of the best path among a population of possible paths.

Source: Cutting Edge Robotics, ISBN 3-86611-038-3, pp. 784, ARS/plV, Germany, July 2005 Edited by: Kordic, V.; Lazinica, A. & Merdan, M.

Such a simplification yields a computational average cost which is linear with the length of the path. We propose two versions of the cost functions and explore the possibilities of the method for the task at hand by selecting adequate thresholds which control the penalization introduced by the lack of support or smoothness. Penalized partial paths can be discarded by a pruning rule. We also modify this rule to provide stronger pruning although the admissibility of the algorithm is not guaranteed in this case. Grouping is completed by end-path conditions, and the robustness of the process is ensured by bidirectional search. Experimental results show that the grouping stage improves individual junction identification, and in this sense grouping gives feedback to a low level task. But grouping also provides a schematic representation which is useful to higher-level tasks like computing relative orientation from a single image.

- Relative orientation: Grouping results followed by straight line detection are very useful to compute the relative orientation between the observer and the environment. A recent proposal, which also relies on Bayesian inference, has proved that this task can be performed by using a single image, at least when the horizontal viewing direction is assumed. This method exploits perspective information and assumes that lines follow a particular configuration known as Manhattan world. We have replaced the original pixel-based strategy by the edges resulting from grouping in the preceding stage, yielding robust and fast results.

A graphical example of the complete process is shown in Fig. 1.

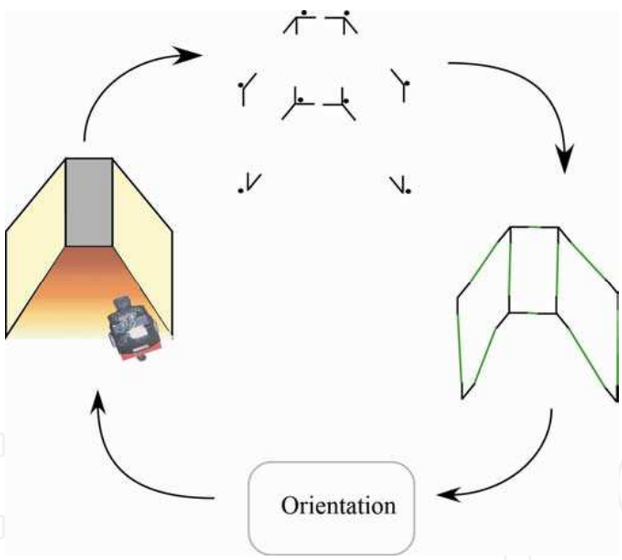

Figure 1. Sequence in the process: first, the corners and junctions are calculated; then the grouping step is applied; the last step is the orientation (compass) estimation, which is used to guide the robot

2. Juction Classification

A generic junction model can be encoded by a parametric template

$$\Theta = \left(x_c, y_c, r, M, \{\phi_i\}, \{W_i\}\right) \tag{1}$$

where (x_c, y_c) is the center, r is the radius, M is the number of wedges (sectors of near constant intensity), $\{\phi_i\}$ with $i = 1, \ldots, N$ are the wedge limits (supposed to be placed on the edge segments convergent in the center) and $\{W_i\}$, the intensity distributions associated to the wedges (see Fig. 2).

Some practical assumptions will reduce the number of parameters to be estimated by a junction detector. First, potential junction centers (x_c, y_c) may be localized by a local filter, like the Plessey detector (Harris & Stephens, 1988) or the more recently the SUSAN detector (Smith & Brady, 1997) which declares a corner when the intensity of the center is similar to that of a small fraction of points in the neighborhood. Second, although the optimal radius r may be found (Parida, Geiger & Hummel, 1998) the cost of doing it is prohibitive for real-time purposes. This is why this parameter is assumed to be set by the user. Furthermore, in order to avoid distortions near the junction center, a small domain with radius $R_{\min}$ around it should be discarded and then $r = R_{\max} - R_{\min}$ where $R_{\max}$ is the scope of the junction.

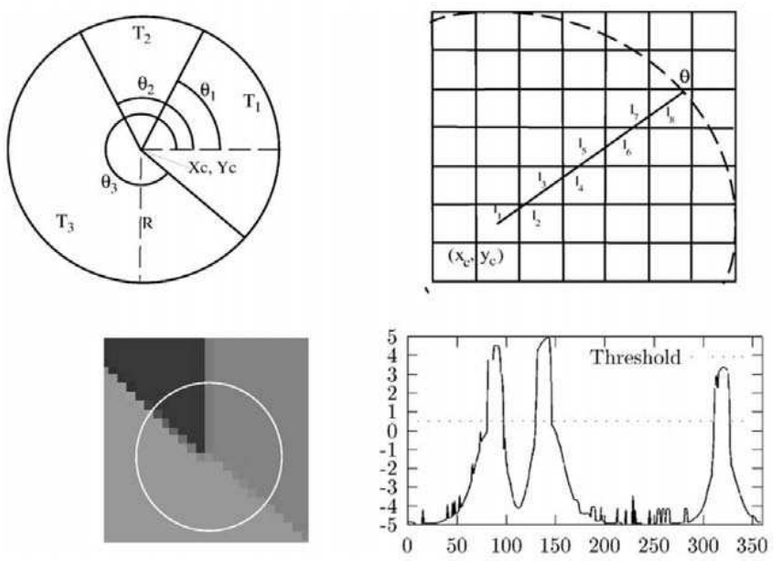

Figure 2. (Top-left) Parametric model for junctions; (top-right) discrete accumulation of intensity along a direction; (bottom-left) example of an ideal junction; (bottom-right) intensity profile of the junction where each peak represents the location of a wedge limit

Given the latter simplifications, a junction classification method will focus on finding the optimal number of wedges M, the wedge limits $\{\phi_i\}$ and the wedge intensity distributions $\{W_i\}$. To that purpose one may follow either a region-based or an edge-based approach, exploiting both of them the fact that wedge intensity distributions are assumed to be near constant (that is, junctions are built on piecewise smooth areas in the image). In the region-based approach the optimal position of a wedge limit is the equilibrium point between the neighbouring wedge regions which can be in turn fused into a greater one. However, in the edge-based method, edges emanating from the junction center are detected and thresholded. We have proposed these methods in (Cazorla & Escolano, 2003) and we have found that the second one yields a lower failure rate although both of them are prone to over-segmentation. This is why we describe here the edge-based method. Consequently, finding the wedge limits is addressed by analyzing the one-dimensional contrast profile (see Fig. 2, bottom right) associated to the junction. Such a profile is estimated by computing, for each angle $\phi \in [0, 2\pi]$ the averaged accumulated contrast $\tilde{I}_\phi$ along the radius in such direction

$$\tilde{I}_\phi = \frac{1}{r}\sum_{i=1}^{N} l_i \times \mathbf{E}_\mathbf{u} \tag{2}$$

where $\mathbf{E_u}$ is the intensity contrast of the pixel $\mathbf{u}=(u,v)$ associated to segment l_i as it is shown in Fig. 2 (top right), being N the number of segments needed to discretize the radius along a given direction and not necessarily the same for all directions. As the reliability of the junction detector depends on how "edgeness" is defined. As starting point we choose $\mathbf{E_u}=|G*\nabla I(\mathbf{u})|$, that is, the Gaussian smoothed gradient (with typically unitary standard deviation). The robustness of such a measure can be improved by embodying it into a decision test which performs a good classification of both edge and non-edge pixels. Recent studies in edge modeling applied to road tracking tasks (Geman & Jedynak, 1996), (Coughlan & Yuille, 1999), point towards building such decision test on the log-likelihood ratio, that is, the logarithm of the ratio between the probability of being "on" and "off" an edge. This criterion guarantees an optimal decision in the sense that it minimizes the Bayesian classification error, but the underlying distributions for "on" and "off" must be known beforehand. Such distributions can be estimated empirically by gathering and quantizing the frequencies of the filter responses in both cases. Then, the empirical probability that a given response is associated to an edge pixel is denoted by $P_{on}(\mathbf{E_u})$ and the empirical probability that a given response corresponds to a non-edge pixel is denoted by $P_{off}(\mathbf{E_u})$. In this paper we will use the empirical distributions presented in (Yuille & Coughlan, 2000). These distributions, were extracted from a range of images and quantized to take 20 values. Both distributions, and the corresponding plots of log-likelihood ratios, are shown in Fig. 4. Taking into account the log-likelihood ratio equation 2 is rewritten as

$$\tilde{I}_{\phi}=\frac{1}{r}\sum_{i=1}^{N} l_i \times \log\frac{P_{on}(\mathbf{E_u}\,|\,\phi^*)}{P_{off}(\mathbf{E_u})} \tag{3}$$

Given an input image, as shown in Fig. 3, the log-likelihood ratio between both the "on" and "off" probabilities gives a robust identification of edge pixels. However, the latter probabilistic model of the smoothed gradient filter can be easily improved by incorporating the orientation of such gradient in the definition of both the "on" and "off" probabilities. Consequently, the smoothed gradient at a given point is defined by $\mathbf{E_u}=(E_{\mathbf{u}},\sigma_{\mathbf{u}})$, where $\sigma_{\mathbf{u}}$ is the local estimation of σ^* the true orientation of the edge to which the pixel belongs. In (Coughlan & Yuille, 1999), where the local estimations of the gradient are used to accumulate evidence through a Hough-like process addressed to estimate the vanishing points in an image, the "on" probability is defined in terms of $P_{on}(\mathbf{E_u}\,|\,\sigma^*)$, the conditional probability of a gradient vector given the true orientation, that is, as a function of the true orientation. Such definition makes sense because the probability of an edge being on must decrease as the estimated orientation diverges from the true orientation, and conversely it must increase as both orientations converge. Furthermore, it can be assumed that the magnitude of the gradient is independent from its orientation and viceversa, which leads to the factorization of the "on" probability into two terms, one of them depending on the gradient vector and the other one on the divergence between the real and the estimated orientations:

$$P_{on}(\mathbf{E_u}\,|\,\sigma^*)=P_{on}(E_{\mathbf{u}})P_{ang}(\sigma_{\mathbf{u}}-\sigma^*) \tag{4}$$

where $P_{ang}(\sigma_{\mathbf{u}}-\sigma^*)$ is the probability of having the correct orientation. Although this probability can be estimated empirically its shape is consistent with having a maximum both in 0 (when both orientations coincide) and π (when both orientations are opposite). In this paper we have used this simple definition as shown in Fig. 3 (bottom right).

On the other hand, "off" probability can be redefined without considering the dependence between the estimated orientation and the true orientation. Therefore such probability $P_{off}(\mathbf{E_u})$ does only depends on the gradient vector. Also assuming independence between gradient magnitude and orientation the resulting probability depends also on two terms:

$$P_{off}(\mathbf{E_u}) = P_{off}(E_{\mathbf{u}})U(\sigma_{\mathbf{u}}) \tag{5}$$

where $U(\sigma_{\mathbf{u}}) = 1/2\pi$ is the uniform distribution. The effectiveness of this model for estimating "edgeness" is shown in Fig. 3 where we represent the log-likelihood ratio and the magnitude and orientation of the gradient.In Fig. 3 we represent the log-likelihood ratio and the magnitude and orientation of the gradient.

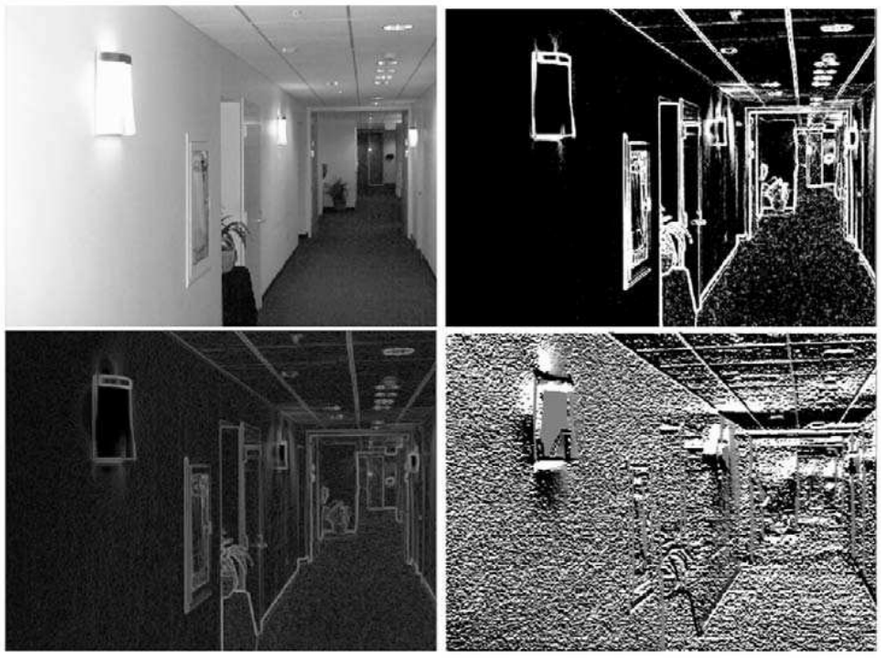

Figure 3. (Top) Sample image and the value of the log-likelihood ratio for pixels; (bottom) magnitude (left) and orientation (right) of the gradient. In the case of orientation, grey is 0, white π and black is $-\pi$

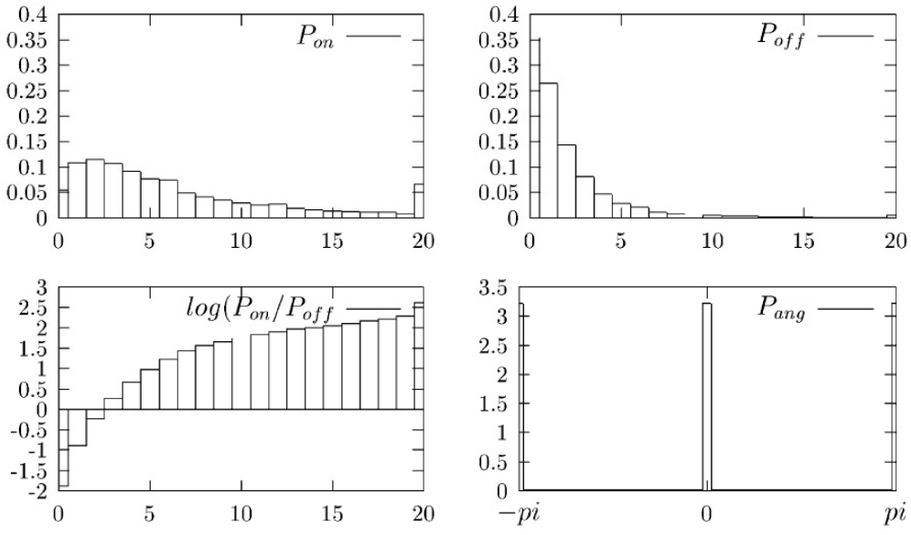

Figure 4. (Top) Plot of the P_{on}/P_{off} information. (Bottom) $\log P_{on}/P_{off}$ and angle distribution

Given the latter profile, junction classification simply consists of thresholding it properly. Furthermore, as the quality of the contrast profile depends on the correct localization of the junction center, we compensate for small localization errors (2-3 pixels) by replacing the average $\tilde{I}_\phi$ by the median $\hat{I}_\phi$. In Fig. 5 (top) we represent the resulting contrast profile. In practice, after thresholding we filter junctions with less than two wedges or with two wedges defining a quasi-straight line. We show some results in Fig. 5 (bottom). The parameters typically used were: $R_{\min}=4, R_{\max}=10$ that is $r=6$, being the threshold $H=0.5$. Due to incorrect junction scopes or to bad localizations, the latter method may yield either false positives or false negatives (see Fig. 5). However, some of these errors may be corrected by a proper grouping strategy along potential connecting edges between wedges of different junctions.

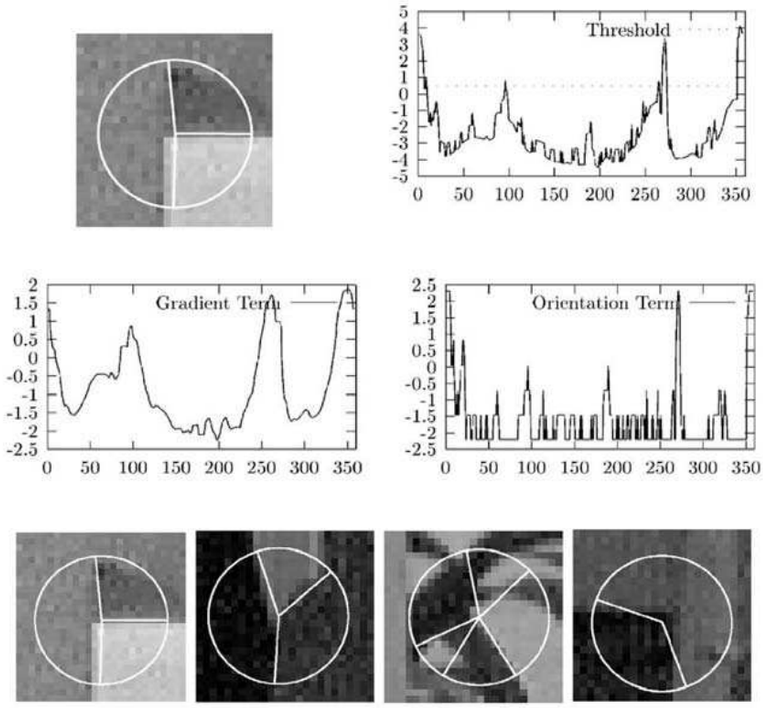

Figure 5. (Top and middle) Contrast profile using the Bayesian edge model. Peaks above the threshold represent suitable wedge limits; (bottom) some results

3. Connecting and Filtering Junctions

The grouping strategy relies on finding "connecting paths". A connecting path P of length L rooted on a junction center (x_c, y_c) and starting from the wedge limit defined by ϕ is defined by a sequence of connected segments $p_1, p_2, \ldots, p_L$ with fixed or variable length. As points of high curvature are usually associated to corners and these corners are placed at the origin or at the end of the paths, we assume that the curvature of these paths is smooth. For describing curvature we define second-order orientation variables $\alpha_1, \alpha_2, \ldots, \alpha_{L-1}$ where $\alpha_j = \phi_{j+1} - \phi_j$ is the angle between segments p_{j+1} and p_j. Then, following (Yuille & Coughlan, 2000) a connecting path P^* should maximize

$$E\left(\{p_j, \alpha_j\}\right) = \sum_{j=1}^{L} \log\left\{\frac{P_{on}(p_j)}{P_{off}(p_j)}\right\} + \sum_{j=1}^{L-1} \log\left\{\frac{P_{\Delta G}(\alpha_{j+1}-\alpha_j)}{U(\alpha_{j+1}-\alpha_j)}\right\} \quad (6)$$

It is straightforward to transform the latter cost function into the typical cost function for snakes if we assume a Gibbs function whose exponent is given by Equation (5). Actually the first term is related to the external energy whereas the second one is related to the internal energy (curvature). More precisely, the first term is the "intensity reward" and it depends on the edge strength along each segment p_j. Defining the intensity reward of each segment of fixed length F in terms of the same edge model that was used to compute the contrast profile yields

$$\log\left\{\frac{P_{on}(p_j)}{P_{off}(p_j)}\right\} = \frac{1}{F}\sum_{i=1}^{N} l_i \times \log\frac{P_{on}(\mathbf{E_u} \mid \phi^*)}{P_{off}(\mathbf{E_u})} \tag{7}$$

Alternatively, $P_{on}(p_j)$ and $P_{off}(p_j)$ may be built on a non-linear filter depending on the gradient magnitude and also on the relative orientation of the segment with respect to the underlying edge.

On the other hand, the second term is the "geometric" reward", which relies on a first-order Markov chain on orientation variables which implements geometric smoothness by exponentially penalizing angular variations

$$P(\alpha_{j+1} \mid \alpha_j) = P_{\Delta G}(\alpha_{j+1} - \alpha_j) \tag{8}$$

and

$$P_{\Delta G}(\Delta\alpha_j) \propto \exp\left\{-\frac{C}{2A}\left|\Delta\alpha_j\right|\right\} \tag{9}$$

where $\Delta\alpha_j = \alpha_{j+1} - \alpha_j$, C modulates the rigidity of the path, and $U(\alpha_{j+1} - \alpha_j)$ is the uniform distribution of the angular variation, defined to keep both the intensity and geometric terms in the same range. Such a choice of the geometric reward is dictated by our smoothness assumption but it is desirable to learn it from the data of the application domain. As we will see later, the effectiveness of this reward depends on its departure from the uniform distribution.

Once the cost function is defined in Equations (6-9) its maximization is addressed by the Bayesian A* algorithm (Coughlan & Yuille, 1999). Given an initial junction center (x_c^0, y_c^0) and an orientation ϕ^0 the algorithm explores a tree in which each segment p_j may expand Q successors. Although there are Q^N paths for path lengths of $N = L$, the Bayesian A* exploits the fact that we want to detect one target path against clutter, instead of taking the best choice from a population of paths. Then, the complexity of the search may be reduced by pruning partial paths with "too low" rewards. Then, the key element of Bayesian A* is the pruning rule. The algorithm finds the best path surviving to the pruning with an expected convergence rate of $O(N)$. In order to do so the pruning relies on evaluating the averaged intensity and geometric rewards of the last L_0 segments of a path. These segments are called the "segment block" and the algorithm discards them when their averaged intensity or geometric reward is below a given threshold:

$$\frac{1}{L_0}\sum_{j=zL_0}^{(z+1)L_0-1} \log\left\{\frac{P_{on}(p_j)}{P_{off}(p_j)}\right\} < T_I \tag{10}$$

$$\frac{1}{L_0}\sum_{j=zL_0}^{(z+1)L_0-1} \log\left\{\frac{P_{\Delta G}(\Delta\alpha_j)}{U(\Delta\alpha_j)}\right\} < T_G \tag{11}$$

being T_I and T_G are respectively the intensity and geometric thresholds. These thresholds determine the "minimum averaged reward" required for survive the pruning. What is key in this context is the thresholds are not arbitrary and they must satisfy the following conditions

$$-D(P_{off} \parallel P_{on}) < T_I < D(P_{on} \parallel P_{off}) \tag{12}$$

and

$$-D(U_{\Delta G} \parallel P_{\Delta G}) < T_G < D(P_{\Delta G} \parallel U_{\Delta G}) \tag{13}$$

being $D(.\|.)$ a distance between two distributions, so called Kullback-Leibler divergence, defined as

$$D(P_1 \parallel P_2) = \sum_{i=1}^{M} P_1(e_i) \log \frac{P_1(e_i)}{P_2(e_i)} \tag{14}$$

The latter thresholds are typically chosen as close to their upper bounds as possible. The rationale of the latter conditions is summarized as follows: If P_{on} diverges from P_{off} the then T_I may have a high value and then the pruning cuts many partial paths; otherwise, the pruning will be very conservative. The same reasoning follows for $P_{\Delta G}$ and $U_{\Delta G}$. More precisely, $D(P_{on} \parallel P_{off})$ quantifies the quality of the edge detector. Then, having large divergences means that the log-likelihood ratio will decide easily between being "on" or "off" the edge, and we will discover false paths soon. Otherwise, it is hard to know whether a given segment is a good choice or not and the algorithm will wait more before discarding a partial path. Similarly, $D(P_{\Delta G} \| U_{\Delta G})$ measures the departure from "geometric ignorance" (dictated by the uniform distribution). The more geometric knowledge the better it helps to constrain the search.

The latter rationale on divergences is independent on the algorithm because there are fundamental limits for the task of discriminating the true task among clutter. There is an order parameter K whose value determines whether the task may be accomplished (when it is positive) or not:

$$K = 2B(P_{on}, P_{off}) + 2B(U_{\Delta G}, P_{\Delta G}) - \log Q \tag{15}$$

being $B(.,.)$ the Battacharyya bound between two distributions:

$$B(P_1, P_2) = -\log\left\{\sum_{i=1}^{M} P_1^{1/2}(e_i) P_2^{1/2}(e_i)\right\} \tag{16}$$

The order parameter depends on the quality of the edge detector and the quality of the geometric knowledge and determines $\tilde{F}$, the expected number of completely false paths having greater reward than the true paths: $K < 0$ implies $\tilde{F} \to \infty$, $K > 0$ implies $\tilde{F} = 0$, and for $K = 0$ there is a phase transition.

Although the $O(N)$ convergence is ensured, the existence of real-time constraints motivates the extension of the basic pruning rule introducing an additional rule, though it is not-admissible in terms of optimality: In practice we prime the "stability of long paths". As long paths are more probable than shorter ones to be close to the target, because they have survived to more prunes, we will also prune paths with length L_j when

$$L_{best} - L_j > Z \times L_0 \text{, that is, } L_j > L_{best} - Z \times L_0 \tag{17}$$

where L_{best} is the length of the best path so far and $Z \geq 0$ sets the minimum allowed difference between the best path and the rest of path. For low Z we introduce more pruning and consequently increase the risk of loosing the true path. When Z is high, shorter paths may survive. It is desirable to find (experimentally) a value for Z representing a trade-off between admissibility and efficiency.
Then, the Bayesian A* algorithm with the latter pruning expands the search tree until it reaches the center (x_c^f, y_c^f) within a given neighbourhood at the end of the selected path. Such search is done by means of a "range tree". The cost of building it is $O(J \log J)$ being J the number of junctions whereas the cost of a query is logarithmic in the worst case.
After reaching a new junction it is checked that the last segment of the path coincides with a wedge limit ϕ^f and, if so, this limit is labelled as "visited". However, the search may finish without finding a junction (when the search queue is empty) or at a "termination point" whose coordinates must be stored. In the first case, if the length of the path is below the block size L_0 we assume that it corresponds to a false wedge limit. In the second case we assume that we have discovered a potential junction. Then, our "local-to-global" grouping algorithm performs path searching from each non-visited limit and an edge may be tracked in both directions.

Figure 6. Junction detection results (Top) and grouping results (Bottom)

4. Obtaining the Compass Angle

Here we explain a method for obtaining the compass angle of a camera, using the features obtained previously. Structured (man-made) environments (like a building, roads, corridors, doors, and so on) share some common features: they are usually built using straight forms. It seems convient to exploit this fact to make inferences about quantitative properties like the relative position of the observer.

4.1 3D Geometry

We define now some geometrical concepts used later on. Let Ψ be the camera orientation angle in the Manhattan world (Faugeras, 1993): the camera is oriented in the direction $\cos\Psi\mathbf{i} - \sin\Psi\mathbf{j}$ (see Fig. 7). Coordinates in the image plane $\mathbf{u} = (u, v)$ are related with world coordinates (x, y, z) by

$$u = \frac{f \cdot (-x\sin\Psi - y\cos\Psi)}{x\cos\Psi - y\sin\Psi} \qquad v = \frac{f \cdot z}{x\cos\Psi - y\sin\Psi} \tag{18}$$

where f is the camera focal length.

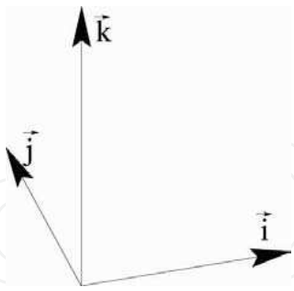

Figure 7. Axis labeling in the Manhattan world

The vanishing points in the directions **i** and **j** are found at the points (-ftanΨ,0) and $(-f \cdot \tan\Psi, 0)$, respectively, in the image plane. Lines in the **k** direction are verticals due to the assumption of being in a structured world.

Finally, we need to relate the intensity gradient in a pixel with the orientation angle. An image point **u**=(u,v) with intensity gradient $(\cos\theta,\sin\theta)$ is consistent with a line **i**, i.e. it points to the vanishing point, if -vtanθ=u+ftanψ . This equation does not change if we add $\pm\pi$ to θ. In a similar way we have vtanθ=-u+fcotΨ for **j** lines.

4.2 Initial Bayesian Model

Let we describe the Bayesian model proposed in (Coughlan, 99). The main difference from this model with respect to other methods is that this does not need to decide wich pixel is on or off an edge. Furthermore, it allows labelling a pixel as an edge pixel of one of the three Manhattan types: **i, j,k** .

Using the previous model defined in Section 2, let $\mathbf{E_u}$ be the gradient (magnitude and orientation) in an image point $\mathbf{u}$. This value can be explained with one of the following models $m_\mathbf{u}$: $m_\mathbf{u} = 1,2,3$ for each of the edges generated by **i**, **j**, **k** , respectively; $m_\mathbf{u} = 4$ means that the gradient has been generated by an edge in a random direction (not **i, j, k**); and, finally, $m_\mathbf{u} = 5$ means the pixel is not in an edge (off the edge). We have set the a priori probability $P(m_\mathbf{u})$ for each model experimentally.

We assume that the gradient probability of the image $\mathbf{E_u}$ has two factors corresponding to the magnitude $(E_\mathbf{u})$ and orientation $(\sigma_\mathbf{u})$:

$$P(\mathbf{E_u} \mid m_\mathbf{u}, \Psi, \mathbf{u}) = P(E_\mathbf{u} \mid m_\mathbf{u}) P(\sigma_\mathbf{u} \mid m_\mathbf{u}, \Psi, \mathbf{u}) \tag{19}$$

where $P(E_\mathbf{u} \mid m_\mathbf{u})$ is $P_{off}(E_\mathbf{u})$ if $m_\mathbf{u} = 5$ or $P_{on}(E_\mathbf{u})$ if $m_\mathbf{u} \neq 5$, and $P(\sigma_\mathbf{u} \mid m_\mathbf{u}, \Psi, \mathbf{u})$ is $P_{ang}(\sigma_\mathbf{u} - \theta(m_\mathbf{u}, \Psi, \mathbf{u}))$ if $m_\mathbf{u} = 1,2,3$ or $U(\sigma_\mathbf{u})$ if $m_\mathbf{u} = 4,5$. The orientation term $\theta(m_\mathbf{u}, \Psi, \mathbf{u})$ is the normal determined by the equation -vtanθ=u+ftanΨ for **i** lines, vtanθ=-u+fcotΨ for **j** lines and $\theta = 0$ for **k** lines. Ψ is the camera orientation angle.

Instead of determining which model better describe the pixel, the evidence is accumulated for the five models:

$$P(\mathbf{E_u} \mid \Psi, \mathbf{u}) = \sum_{m_\mathbf{u}=1}^{5} P(\mathbf{E_u} \mid m_\mathbf{u}, \Psi, \mathbf{u}) P(m_\mathbf{u}) \tag{20}$$

Thus, we can determine the evidence for the camera angle Ψ without knowing to which of those five models owns the pixel.
Now, we want to calculate the evidence for the complete image $\{E_u\}$. We assume that the data is condicionally independent in all the pixels, given an orientation Ψ:

$$P(\{\mathbf{E_u}\}|\Psi) = \prod_{\mathbf{u}} P(\mathbf{E_u}|\Psi,\mathbf{u}) \tag{21}$$

Thus, the a posteriori distribution over the orientation is

$$\prod_{\mathbf{u}} P(\mathbf{E_u}|\Psi,\mathbf{u})P(\Psi)/Z \tag{22}$$

where Z is a normalization factor and $P(\Psi)$ is the uniform a priori distribution over Ψ. In order to find the Maximum A Posteriori (MAP), we need to maximize (we ignore Z, as it is independent from Ψ):

$$\Psi^* = \arg\max_{\Psi} P(\{\mathbf{E_u}\}|\Psi)P(\Psi) \tag{23}$$

Applying log to the previous equation:

$$\log\left[\arg\max_{\Psi} P(\{\mathbf{E_u}\}|\Psi)P(\Psi)\right] = \log P(\Psi) + \sum_{m_u=1}^{5} \log\left[\arg\max_{\Psi} P(\mathbf{E_u}|m_u,\Psi,\mathbf{u})P(m_u)\right] \tag{24}$$

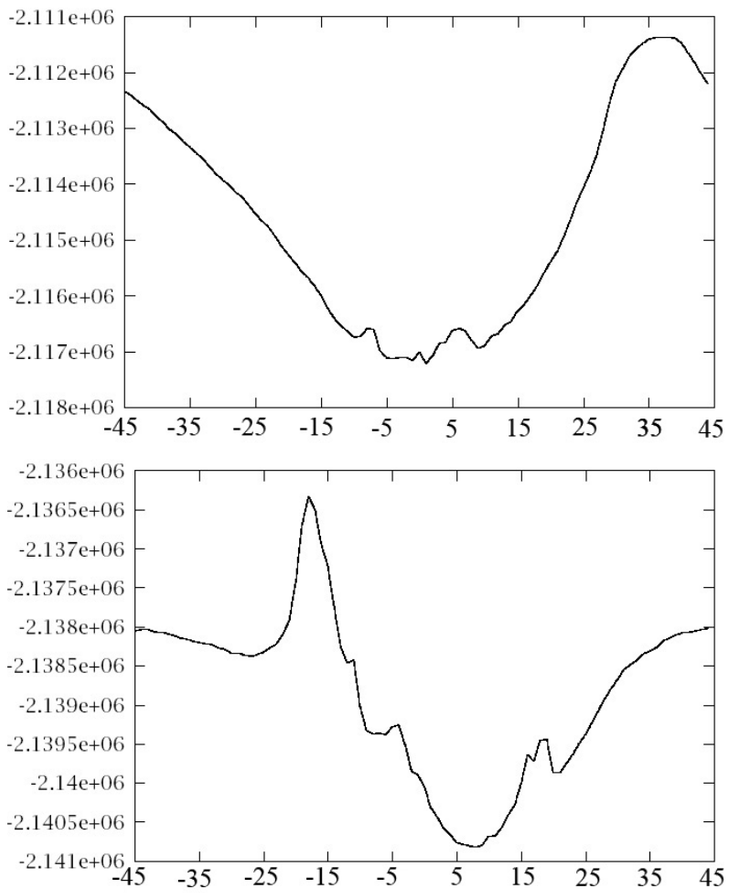

Figure 8. Plots of the equation 24 evaluated at each angle, from -45º to 45º

We need to find the angle Ψ maximizing the latter equation. It is easily calculated evaluating the function for each angle, from -45º to 45º. Fig. 8 shows a plot of this evaluation. The maximum value is the camera orientation angle.However, this is a slow process and we need to speed up. This part is explained in the next section.

4.3 Obtaining the Compass Angle From Edge Information

Now, we are interested in using the grouping information in order to speed up the overall process. The only thing we need to do is redefine the P_{on} / P_{off} functions. They are defined now as:

$$P_{on}(E_{\mathbf{u}}) = \begin{cases} 0.99 & \text{if } E_{\mathbf{u}} \in A \\ 0.01 & \text{otherwise} \end{cases}$$

$$P_{off}(E_{\mathbf{u}}) = \begin{cases} 0.99 & \text{if } E_{\mathbf{u}} \notin A \\ 0.01 & \text{otherwise} \end{cases} \tag{25}$$

where A is the set of edges obtained during the grouping process.

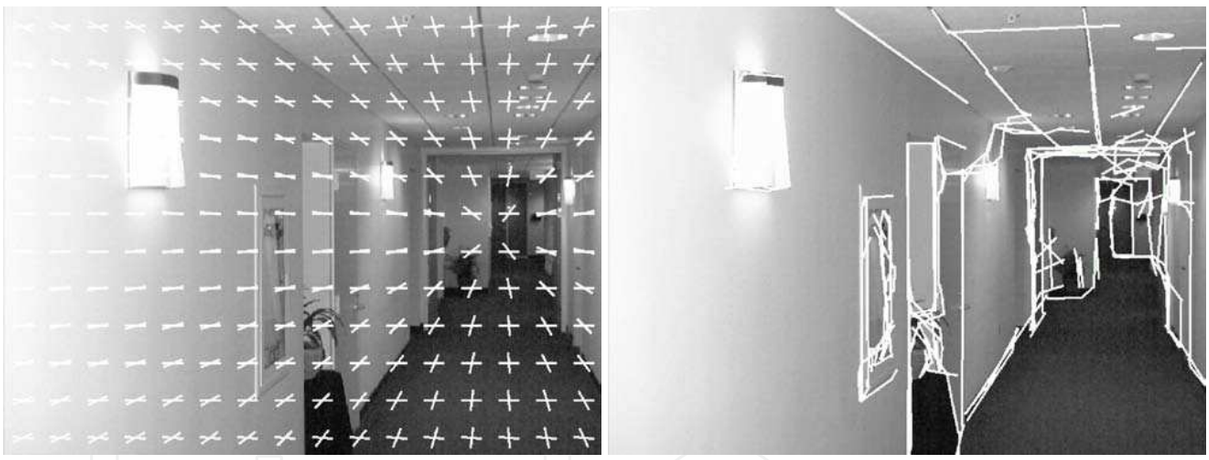

Figure 9. Grouping information and orientation angle obtained

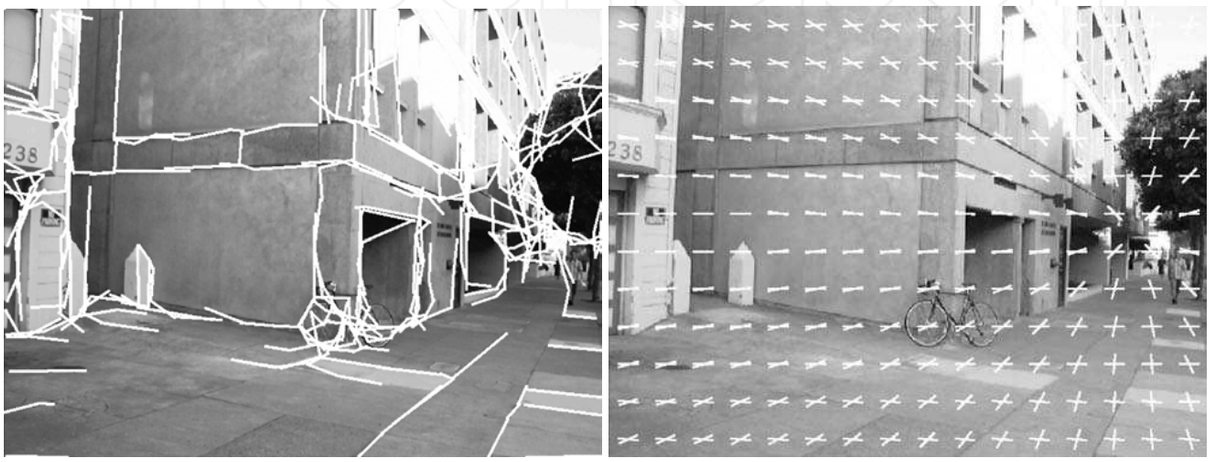

Figure 10. Grouping information and orientation angle obtained

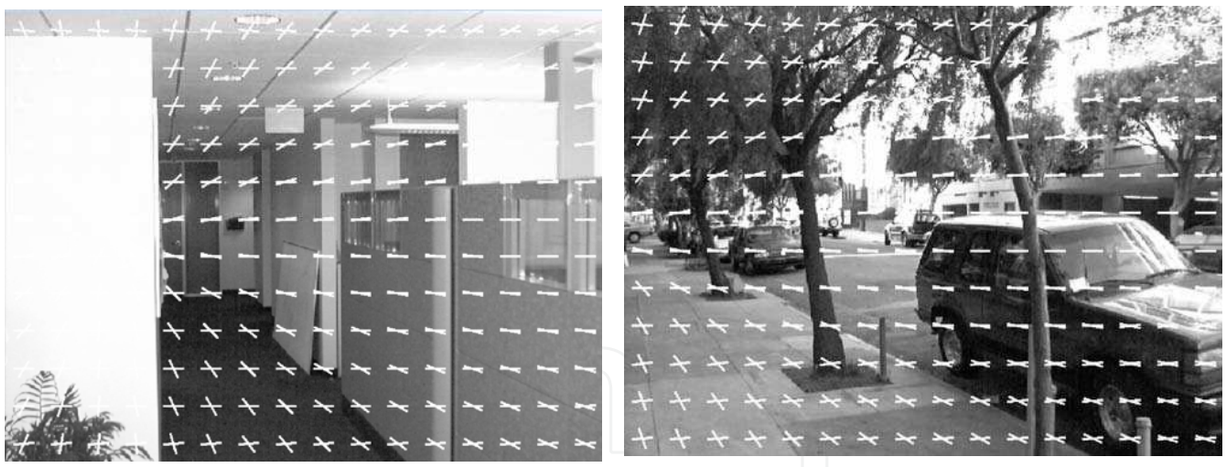

Figure 11. Two more examples of calculating the orientation angle

We now do not need to calculate the image gradient, as we have it included in the grouping information. So the pixel orientation is obtained as the normal in a point in the edge. With this information we now know which pixels are on and off the edge. The overall process is simple: just take the grouping output and, for each edge evaluate Equation 24 for each angle. The maximum value (see Fig. 8) is the camera orientation angle.

Figures 8, 9 and 10 are examples of applying this algorithm using grouping information. In both figures we have first the grouping information obtained in one of the image and its corresponding result. The crosses are a way to show the calculated angle.

5. Conclusions and Future Work

Exploiting computer vision for performing robotic tasks, like recognizing a given place in the environment or simply computing the relative orientation of the robot with respect to the environment, requires an in depth analysis of the vision modules involved in such computations. In this paper, we have considered three types of computations: local computations (for the estimation of junctions), local-to-global computations (for finding a geometric sketck) and voting-accumulation computations (for obtaining the relative orientation of the robot). We have addressed the analysis of the latter modules from the point of view of three practical requirements: reliability (robustness), efficiency and flexibility. These requirements are partially fullfiled by the methodology used: the three modules (junction detection, grouping, and orientation estimation) share elements of Bayesian inference. Sometimes, as in the case of junction detection, these elements yield statistical robustness. In other cases, as in the grouping module, the Bayesian formulation has a deep impact in the reduction of computational complexity. The Bayesian integration of feature extraction, grouping and orientation estimation is a good example of how to get flexibility by exploiting both the visual cues and the prior assumptions about the environment in order to solve a given task. On the other hand, as the basic visual cues are edges and there are fundamental limits regarding whether certain tasks relying on edge cues may be solved or not, independently of the algorithm, we also stress the convenience of having this bounds in mind in order to devise practical solutions for robotics tasks driven by computer vision.

6. References

Cazorla, M. (2000) PhD Thesis. University of Alicante, Spain

Cazorla, M. & Escolano, F., Gallardo, D. & Rizo, R. (2002). Junction detection and grouping with probabilistic edge models and Bayesian A*. Pattern Recognition. Vol. 35. pp. 1869-1881

Cazorla, M. & Escolano, F. (2003). Two Bayesian methods for junction classification. IEEE Transactions on Image Processing. Vol. 12, No. 3, pp. 317-327

Coughlan, J. & Yuille, A.L. (1999) Bayesian A* tree search with expected O(N) convergence for road tracking. LNCS 1654, pp. 189-2004

Coughlan, J. & Yuille, A. (2003) Manhattan World: Orientation and Outlier Detection by Bayesian Inference. Neural Computation. Vol. 15, No. 5, pp. 1063-88.

Faugeras, O. Three-Dimensional Computer Vision. The MIT Press, Cambridge, Massachusetts.

Geman, D. & Jedynak, B. An active testing model for tracking roads in satellite images. IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. 18 No. 1. pp. 1-14.

Harris, C.G. & Stephens, M. (1988). A combined corner and edge detection. Proceedings of the Fourth Alvey Vision Conference . pp. 147-151

Parida, L., Geiger, D. & Hummel, R. (1998). Junctions: detection, classification and reconstruction. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 20, No. 7. pp. 687-698.

Smith, S.M. & Brady, J.M. (1997). SUSAN = a new approach to low level image processing. International Journal on Computer Vision. Vol. 23, No. 1 pp. 45-78

Yuille, A.L. & Coughlan, J. (2000) Fundamental limits of Bayesian inference: order parameters and phase transitions for road tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 22, No. 2. pp. 160-173

Cutting Edge Robotics

Edited by Vedran Kordic, Aleksandar Lazinica and Munir Merdan

ISBN 3-86611-038-3

Hard cover, 784 pages

Publisher Pro Literatur Verlag, Germany

Published online 01, July, 2005

Published in print edition July, 2005

This book is the result of inspirations and contributions from many researchers worldwide. It presents a collection of wide range research results of robotics scientific community. Various aspects of current research in robotics area are explored and discussed. The book begins with researches in robot modelling & design, in which different approaches in kinematical, dynamical and other design issues of mobile robots are discussed. Second chapter deals with various sensor systems, but the major part of the chapter is devoted to robotic vision systems. Chapter III is devoted to robot navigation and presents different navigation architectures. The chapter IV is devoted to research on adaptive and learning systems in mobile robots area. The chapter V speaks about different application areas of multi-robot systems. Other emerging field is discussed in chapter VI - the human- robot interaction. Chapter VII gives a great tutorial on legged robot systems and one research overview on design of a humanoid robot.The different examples of service robots are showed in chapter VIII. Chapter IX is oriented to industrial robots, i.e. robot manipulators. Different mechatronic systems oriented on robotics are explored in the last chapter of the book.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen