New Measurements and Corner-Guidance for Curve Matching With Probabilistic Relaxation

Ying Shan* and Zhengyou Zhang
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399, USA
Email: zhang@microsoft.com

Abstract

Reliable curve matching is a difficult yet important problem in many vision-based applications including image-based modeling. We describe in this paper two aspects of our research in this area: a new algorithm for curve matching (including lines) within a probabilistic relaxation framework, and an approach of incorporating previously matched points/corners to guide curve matching. We propose similarity-invariant unary and binary measurements suitable for curves, and introduce an additional measurement to model the uncertainty of the binary measurements. The uncertainty measure is proven to be very important in computing the matching support from neighboring matches. We also show how to use a set of previously matched points/corners to guide the curve matching. The role of the corner guidance is explicitly modeled by a set of unary measurements and a similarity function under the same relaxation framework. Preprocessing techniques contributing to the success of our curve matching techniques are also developed and discussed. Experiments with complex real scenes show that the rate of correct matching is higher than 98%.

Keywords: Stereoscopic vision, Curve matching, Probabilistic relaxation, Image features, Edge matching, Epipolar geometry.

1 Introduction

Information that can be used for curve matching falls into three categories, i.e., the geometrical constraint, the similarity between the curves, and the compatibility among the neighborhood matches. The most important geometric constraint between views is the epipolar constraint, which has been used in all curve-based stereo matching algorithms

*Current address: Sarnoff Corporation; Email: yshan@sarnoff.com

to reduce the search space. An early example exploiting this constraint was the PMF method [1, 2], where the epipolar lines are horizontal scan lines. The technique described in [3] also uses this constraint, but does not restrict the epipolar lines to be horizontal (the fundamental matrix is used to define the epipolar geometry).

The *similarity function* between two curves is usually defined in a high dimensional feature space. The features may include attributes such as the intensity in the neighborhood [3], orientation [4], and the local shape characteristics [5]. These are the *unary measurements*.

The *compatibility function* among neighboring matches is usually defined by the relationship between neighboring pairs of matches. The compatibility function is usually related to the local affinity or similarity assumption, and the *binary measurements*, defining the relationship, could be the angle and distance between a pair of neighboring curves [6, 7]. Relaxation techniques [8, 9, 7, 10] are useful methods that integrate the similarity function and the compatibility function to progressively reduce the matching ambiguity.

The goal of this paper is to attack the curve matching problem, and the cameras are not required to be calibrated. For point matching with calibrated cameras, please refer to the nice paper by Belhumeur [11]. In our context, curves include both straight and non-straight edges. Many techniques exist for detecting edges. We detect edge points in sub-pixel accuracy by finding the zero-crossing between the integer pixels on the DOG (difference of Gaussian) image. Adjacent edge points are then linked together into connected edge chains (also called curves for simplicity). The linking process does not include any heuristic processing such as one-pixel gap filling or straight line fitting. An edge chain (curve) $\mathcal{C}(s)$ is represented as a linked list, and is parameterized by the index variable s.

Techniques for line matching have been successfully applied to scenes containing mainly planar surfaces [3, 12, 13]. These techniques, however, have two fundamental difficulties when applied to more general scenes. First, they are not suitable for scenes containing curves because the line model is insufficient to describe curves. Second, they are not suitable for scenes that are taken by a camera closed to the scene, where the local affinity or similarity assumption for long line segments is no more valid. Furthermore, the projection of straight lines in 3D onto images may not be straight anymore due to radial lens distortion.

Two central problems related to curve matching are the design of good unary and binary measurements, and the definition of appropriate similarity and compatibility functions. Previous work on curve matching such as [2, 5, 14, 15] gave good examples on the unary measurements and similarity function between curves, but were weak when dealing with binary measurements and compatibility functions. Indeed, their compatibility functions were usually computed from measurements such as disparity [5] or disparity gradient [2], which are only suitable for the description of relationships between two pairs of points, and are not scale invariant. Another important issue to be considered is the uncertainty in the binary measurements. Obviously, the local affinity or similarity assumption is only valid within a limited area of the image. Binary measurements obtained from curves far away from each other have more uncertainty than those from nearby curves. This should be reflected in the computation of the compatibility function. Recently, Alibhai and Zucker [16] proposed to use differential structure of curves in contour-based stereo

matching within a relaxation framework. They defined a compatibility measure of edge points based on the relationship of the differential structure in the images to the geometry of curves in space. Note that their algorithm uses edges as primitives while ours uses curves (linked edges). Their compatibility measure could certainly be adapted to our framework.

When the image pair is taken by cameras with general configuration [17], the epipolar geometry is usually computed from a set of matched corners. It is then natural to ask the question whether the matched corners set can be useful for curve matching. The answer is definitely yes. In fact, the matched corner set gives a one-to-one mapping from a corner point p in one image to a corner point p' in the other image, which is stronger than the point-to-line mapping given by the epipolar constraint. The mapping given by the matched corner set will be referred to as the *corner matching constraint*. The progressive stereo matching algorithm proposed in [18] exploits a similar constraint in an iterative process by adding progressively new point matches. At each stage, the current reliable matches constrain the search range for new matches, and only unambiguous matches are selected in accordance with the least commitment strategy.

The major difference between the epipolar constraint and the corner matching constraint is that the latter is sparse and is only defined on the points in the matched corner set. In general, since the criterion and the process used to detect corners and edge points are different, these two types of features are unlikely to be overlapped. It is therefore improbable to expect that a corner point lies on a curve. Nevertheless, it is reasonable to assume that corner points can usually be found near a curve. Since object surfaces are usually continuous at least in a small neighborhood, local affinity or similarity constraint can be used to describe the relationship of a corner and a curve.

In this paper, we develop a curved-edge matching algorithm within a probabilistic relaxation framework similar to that in [7]. The framework described in [7] only deals with straight line segments. We adapt the framework to deal with curves and explicitly model the uncertainty in the binary measurements. Epipolar geometry is used to reduce the matching ambiguity. A piecewise linear model is used to approximate a curve with line segments. A set of binary measurements for line segments are proposed, which are similarity-invariant and measured in the same physical space. They are then integrated to provide a set of binary measurements and a compatibility function for the complete curve. A set of matched corners are also used to guide the curve matching. The role of the corner guidance is explicitly modeled by a set of unary measurements and a similarity function that can be integrated into the above frame work easily. Other contributions include a similarity-invariant unary measurement and two useful preprocessing techniques.

The paper is organized as follows. Section 2 outlines our curve matching algorithm. Section 3 introduces a probabilistic relaxation framework and the notion of combined unary and binary measurements. The combined unary measurements and the similarity function are discussed in Section 4, followed by discussions on the *combined binary measurements* and the compatibility function in section 5. Section 6 describes how corner matches, if available (e.g., during determination of the epipolar geometry of the image pair), can guide curve matching to achieve higher efficiency. Section 7 describes two preprocessing methods. Section 8 provides experimental results with several complex real scenes.

2 Outline of the curve matching algorithm

We outline the curve matching algorithm in Figure 1.

We consider an image pair. If its epipolar geometry, in terms of fundamental matrix [19, 20], is unknown, then use Harris operator to detect corners in each image, match them between the image pair and estimate the fundamental matrix using the robust technique described in [17]. Other techniques can be used to perform this task without affecting the remaining curve matching algorithm.

Curves are extracted through edge detection and linking. They first go through high curvature breaking and epipolar breaking (see Section 7), and a set of connected line segments is then generated to approximate each curve. Both unary and binary measurements are computed for line segments, which are then integrated to produce unary and binary measurements for curves (see Section 4 and 5). Similarity and compatibility of curves between images are computed and used in a probabilistic relaxation framework to generate curve matches (see Section 3).

The process shown in the dashed box in Figure 1 is the corner guidance. If a set of corner matches is available, it can provide useful information in guiding the search for curve matches. This information is formulated as a unary measurement in Section 6, and is seamlessly integrated into the probabilistic relaxation framework. Note that this process is optional. In our applications, corner matches are already available anyway after the estimation of the fundamental matrix, and using them as guidance indeed increases considerably the efficiency of curve matching.

3 A probabilistic relaxation framework

The probabilistic relaxation framework given in this section is based on the nice work of [7] on structural matching. We also adopt most of its notation for the convenience of comparison reading.

3.1 Notation

Curve matching is modeled as a labeling problem from the object space (left image) to the model space (right image). There are N curves in the left image, and M in the right. The curves in the left image form the object space, denoted by $A = \{a_1, \ldots, a_N\}$. The curves in the right image form the model space, each labeled as ω_j. We wish to match the object to the model. We therefore assign to each object a_i a label θ_i, which may take as its value any of the $M+1$ model labels that form the set $\Omega = \{\omega_0, \omega_1, \ldots, \omega_M\}$, where ω_0 is the null label used to label curves for which no match in the right image is appropriate.

Four sets of indices are defined for convenience as $N_0 \equiv \{1, \ldots, N\}$, $N_i \equiv \{j \mid j \in N_0, j \neq i\}$, $M_0 \equiv \{0, \ldots, M\}$, and $M_i \equiv \{j \mid j \in M_0, j \neq i\}$. The unary measurement set is defined as $\mathcal{X} = \{\mathbf{x}_i \mid i \in N_0\}$, where $\mathbf{x}_i$ is the unary measurement vector for the object a_i. Unary measurements include intensity similarity, and will be addressed in section 4. The

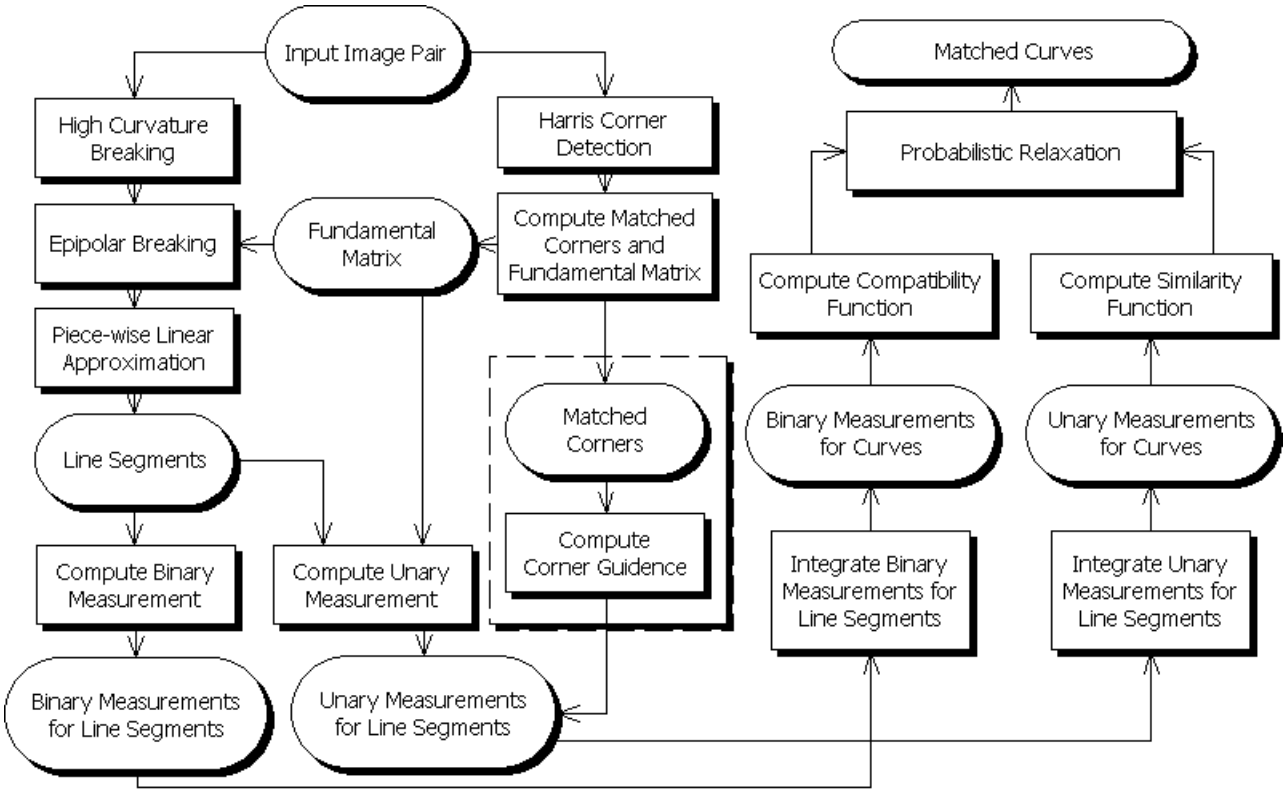

Figure 1: The outline of the curve matching algorithm. The process in the dashed box is for the optional corner guidance.

binary measurement set for the object a_i is defined as $\mathcal{A}_i = \{\mathbf{A}_{ij} \mid j \in N_i\}$, where $\mathbf{A}_{ij}$ is the vector of binary measurements between objects a_i and a_j. Binary measurements include distance between two curves, and will be addressed in section 5. A special measurement set $\mathcal{B}_i = \{\mathbf{B}_{ij} \mid j \in N_i\}$ is also defined, where $\mathbf{B}_{ij}$ is the uncertainty measurement to be defined later. To follow the traditional notation, we will use the upper-case P to denote the probability of an event, the lower-case p to denote the probability density function, and $\mathcal{N}_{\mathbf{v}}(\mu, \Sigma)$ to denote the Gaussian probability density function of a random vector $\mathbf{v}$ with the mean μ and the covariance matrix Σ.

3.2 A framework with uncertainty measurement

With the above notation, the matching problem becomes that of finding for object a_i a model label ω_{θ_i} with the highest probability given the measurements $\mathcal{X}$, $\mathcal{A}_i$, and $\mathcal{B}_i$:

$$\begin{aligned} & P(\theta_i = \omega_{\theta_i} \mid \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) \\ = & \max_{\omega_\lambda \in \Omega} P(\theta_i = \omega_\lambda \mid \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) \end{aligned} \tag{1}$$

For the convenience of the discussion that follows, we introduce the notation of the

event set

$$\mathcal{L}^{\lambda} = \{\theta_i = \omega_{\lambda}\} \bigcup \{\theta_j = \omega_{\theta_j} \mid j \in N_i\}$$

to indicate that object a_i is labeled with a given label ω_{λ} while other objects can change their labels. Obviously, for $\mathcal{L}^{\theta_i}$, the ith event $\mathcal{L}_i^{\theta_i}$ is equivalent to event $\theta_i = \omega_{\theta_i}$. Using Bayes' formula and the theorem of total probability, we have

$$\begin{aligned} & p(\mathcal{L}_i^{\theta_i} \mid \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) \\ = & \frac{p(\mathcal{L}_i^{\theta_i}, \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i)}{p(\mathcal{X}, \mathcal{A}_i, \mathcal{B}_i)} \\ = & \frac{\sum_{\{\omega_{\theta_j} \in \Omega, j \in N_i\}} p(\mathcal{L}^{\theta_i}, \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i)}{\sum_{\omega_{\lambda} \in \Omega} \sum_{\{\omega_{\theta_j} \in \Omega, j \in N_i\}} p(\mathcal{L}^{\lambda}, \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i)} \end{aligned} \tag{2}$$

By applying the product rule for $p(\mathcal{L}^{\lambda}, \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i)$, we have

$$\begin{aligned} & p(\mathcal{L}^{\lambda}, \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) \\ = & p(\mathcal{X} \mid \mathcal{L}^{\lambda}, \mathcal{A}_i, \mathcal{B}_i) p(\mathcal{L}^{\lambda}, \mathcal{A}_i, \mathcal{B}_i) \\ = & p(\mathcal{X} \mid \mathcal{L}^{\lambda}) p(\mathcal{A}_i \mid \mathcal{L}^{\lambda}, \mathcal{B}_i) p(\mathcal{L}^{\lambda}, \mathcal{B}_i) \\ = & p(\mathcal{X} \mid \mathcal{L}^{\lambda}) P(\mathcal{L}^{\lambda}) p(\mathcal{A}_i \mid \mathcal{L}^{\lambda}, \mathcal{B}_i) p(\mathcal{B}_i) \\ = & P(\mathcal{L}^{\lambda} \mid \mathcal{X}) p(\mathcal{A}_i \mid \mathcal{L}^{\lambda}, \mathcal{B}_i) p(\mathcal{X}) p(\mathcal{B}_i) \end{aligned} \tag{3}$$

where the second equation holds because $\mathcal{X}$ (unary measurement set) does not depend on $\mathcal{A}_i$ (binary measurement set) and $\mathcal{B}_i$ (uncertainty of the binary measurement set), and the third equation holds because $\mathcal{L}^{\lambda}$ and $\mathcal{B}_i$ are independent. Since there is no knowledge about $\mathcal{X}$ and $\mathcal{B}_i$, $p(\mathcal{X})$ and $p(\mathcal{B}_i)$ are constants. Substituting (3) into (2), we then have

$$P(\mathcal{L}_i^{\theta_i} \mid \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) = \frac{\sum_{\{\omega_{\theta_j} \in \Omega, j \in N_i\}} P(\mathcal{L}^{\theta_i} \mid \mathcal{X}) p(\mathcal{A}_i \mid \mathcal{L}^{\theta_i}, \mathcal{B}_i)}{\sum_{\omega_{\lambda} \in \Omega} \sum_{\{\omega_{\theta_j} \in \Omega, j \in N_i\}} P(\mathcal{L}^{\lambda} \mid \mathcal{X}) p(\mathcal{A}_i \mid \mathcal{L}^{\lambda}, \mathcal{B}_i)} \tag{4}$$

Assuming that the events in the $\mathcal{L}^{\lambda}$ are independent, we have

$$\begin{aligned} & P(\mathcal{L}^{\lambda} \mid \mathcal{X}) \\ = & P(\theta_i = \omega_{\lambda} \mid \mathbf{x}_i) \prod_{j \in N_i} P(\theta_j = \omega_{\theta_j} \mid \mathbf{x}_j) \end{aligned}$$

Since $\mathbf{A}_{ij}$ only depends on the ith and jth events in $\mathcal{L}^{\lambda}$ and $\mathbf{B}_{ij}$, and since $\mathbf{A}_{ij}$s are independent from each other, we have

$$\begin{aligned} & p(\mathcal{A}_i \mid \mathcal{L}^{\lambda}, \mathcal{B}_i) \\ = & \prod_{j \in N_i} p(\mathbf{A}_{ij} \mid \theta_i = \omega_{\lambda}, \theta_j = \omega_{\theta_j}, \mathbf{B}_{ij}) \end{aligned}$$

With the following simple notation

$$\mathbf{P}_j^{\lambda} = P(\theta_j = \omega_{\theta_j} \mid \mathbf{x}_j) p(\mathbf{A}_{ij} \mid \theta_i = \omega_\lambda, \theta_j = \omega_{\theta_j}, \mathbf{B}_{ij})$$

we then have

$$\begin{aligned} & \sum_{\{\omega_{\theta_j} \in \Omega, j \in N_i\}} P(\mathcal{L}^\lambda \mid \mathcal{X}) p(\mathcal{A}_i \mid \mathcal{L}^\lambda, \mathcal{B}_i) \\ = \; & P(\theta_i = \omega_\lambda \mid \mathbf{x}_i) \sum_{\omega_{\theta_1} \in \Omega} \mathbf{P}_1^\lambda \cdots \sum_{\omega_{\theta_N} \in \Omega} \mathbf{P}_N^\lambda \\ = \; & P(\theta_i = \omega_\lambda \mid \mathbf{x}_i) \prod_{j \in N_i} \sum_{\omega_{\theta_j} \in \Omega} \mathbf{P}_j^\lambda \end{aligned} \tag{5}$$

Substituting (5) into (4) leads immediately to

$$\begin{aligned} & P(\theta_i = \omega_{\theta_i} \mid \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) \\ = \; & \frac{P(\theta_i = \omega_{\theta_i} \mid \mathbf{x}_i) \mathbf{Q}(\theta_i = \omega_{\theta_i})}{\sum_{\omega_\lambda \in \Omega} P(\theta_i = \omega_\lambda \mid \mathbf{x}_i) \mathbf{Q}(\theta_i = \omega_\lambda)} \end{aligned} \tag{6}$$

where the *support function* is given by

$$\begin{aligned} \mathbf{Q}(\theta_i = \omega_\alpha) &= \prod_{j \in N_i} \sum_{\omega_\beta \in \Omega} \mathbf{P}_j^\beta \\ &= \prod_{j \in N_i} \sum_{\omega_\beta \in \Omega} P(\theta_j = \omega_\beta \mid \mathbf{x}_j) \\ &\quad p(\mathbf{A}_{ij} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \end{aligned} \tag{7}$$

where the first item in the summation is the similarity function, and the second is the compatibility function that takes into account explicitly the uncertainty measurement $\mathbf{B}_{ij}$.

Based on (6), we update the labeling probability in a relaxation scheme according to the following iterative equation

$$\begin{aligned} & P^{(n+1)}(\theta_i = \omega_{\theta_i}) \\ = \; & \frac{P^{(n)}(\theta_i = \omega_{\theta_i}) \mathbf{Q}^{(n)}(\theta_i = \omega_{\theta_i})}{\sum_{\omega_\lambda \in \Omega} P^{(n)}(\theta_i = \omega_\lambda) \mathbf{Q}^{(n)}(\theta_i = \omega_\lambda)} \end{aligned} \tag{8}$$

where n is the iteration number, and $P^{(0)}(\theta_i = \omega_{\theta_i}) = P(\theta_i = \omega_{\theta_i} \mid \mathbf{x}_i)$. The iteration process terminates if the change in the probabilities is less than a predefined small value or the maximum number of iterations is reached. The reader is referred to [7, 21, 8] for more details.

As has been mentioned in section 1, candidate curves in the right view for the current curve a_j in the left view are found by the epipolar geometry. This is equivalent to reducing the model label set for a_j from Ω to Ω_j in the above derivations. Since the size of Ω_j is usually much smaller than that of Ω, we increase considerably the computational efficiency of the relaxation process.

3.3 Adapted framework with combined measurements

In our case, the unary and binary measurements depend on information in both object and model space (i.e., left and right image). Therefore, we should consider measurements for combined object-model pairs. Let $\mathbf{x}_i^{(\alpha)}(\alpha \in M_0)$ be the combined unary measurement defined for the pair of the ith object and the αth model. The unary measurement of object a_i, $\mathbf{x}_i$, is then itself a set of combined unary measurements, i.e, $\{\mathbf{x}_i^{(\alpha)} \mid \alpha \in M_0\}$. Let $\mathbf{A}_{ij}^{(\alpha\beta)}(\alpha \in M_0, \beta \in M_0)$ be the combined binary measurement defined for two object-model pairs (i, α) and (j, β). Then, the binary measurement $\mathbf{A}_{ij}$ is itself a set of combined binary measurements, i.e., $\{\mathbf{A}_{ij}^{(\alpha\beta)} \mid \alpha \in M_0, \beta \in M_0\}$.

In order to adapt the framework for working with the combined measurements, consider the similarity function in (7) first. Since the event $\theta_j = \omega_\beta$ does not depend on the combined unary measurements other than the $\mathbf{x}_j^{(\beta)}$, we have

$$P(\theta_j = \omega_\beta \mid \{\mathbf{x}_j^{(\beta)} \mid \beta \in M_0\}) = P(\theta_j = \omega_\beta \mid \mathbf{x}_j^{(\beta)}) \tag{9}$$

Similarly, $A_{ij}^{(\alpha\beta)}$'s are independent of each other, and we have

$$\begin{aligned} & p(\mathbf{A}_{ij} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \\ = \;& p(\{\mathbf{A}_{ij}^{(kl)}\} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \\ = \;& \prod_{\substack{k \in M_0 \\ l \in M_0}} p(\mathbf{A}_{ij}^{(kl)} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \end{aligned}$$

For $\mathbf{A}_{ij}^{(kl)}$ which are not equal to $\mathbf{A}_{ij}^{(\alpha\beta)}$ (i.e., k and l are not matched with i and j), there is no information about how to compute the binary measurement $\mathbf{A}_{ij}^{(kl)}$. Hence, we assume the density function to be a uniform distribution, i.e., a constant, within its domain. In consequence, we have

$$\begin{aligned} & p(\mathbf{A}_{ij} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \\ = \;& \epsilon \; p(\mathbf{A}_{ij}^{(\alpha\beta)} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \end{aligned} \tag{10}$$

where $\{\mathbf{A}_{ij}^{(kl)}\}$ is the abbreviation of $\{\mathbf{A}_{ij}^{(kl)} \mid k \in M_0, l \in M_0\}$, and $\epsilon = \prod_{\{kl \neq \alpha\beta\}} p(\mathbf{A}_{ij}^{(kl)} \mid \mathbf{B}_{ij})$ is a constant. By substituting (10) and (9) into (7), (7) into (6), and eliminating the constant ϵ, (6) can be rewritten as

$$\begin{aligned} & P(\theta_i = \omega_{\theta_i} \mid \mathcal{X}, \mathcal{A}_i, \mathcal{B}_i) \\ = \;& \frac{P(\theta_i = \omega_{\theta_i} \mid \mathbf{x}_i^{(\theta_i)}) \mathbf{Q}(\theta_i = \omega_{\theta_i})}{\sum_{\omega_\lambda \in \Omega} P(\theta_i = \omega_\lambda \mid \mathbf{x}_i^{(\lambda)}) \mathbf{Q}(\theta_i = \omega_\lambda)} \end{aligned} \tag{11}$$

and (7) as

$$\mathbf{Q}(\theta_i = \omega_\alpha) = \prod_{j \in N_i} \sum_{\omega_\beta \in \Omega} \mathbf{P}_j^\beta$$

$$= \prod_{j \in N_i} \sum_{\omega_\beta \in \Omega} P(\theta_j = \omega_\beta \mid \mathbf{x}_j^{(\beta)}) \\ p(\mathbf{A}_{ij}^{(\alpha\beta)} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) \quad (12)$$

4 The combined unary measurement and the similarity function

In this section, a combined unary measurement invariant to similarity (scaled Euclidean) transformation is developed and the similarity function is computed based on it. This part of the algorithm is similar to the PMF [2] algorithm in the sense that

- Only selected seed points on both the left and the right curves are considered. The right seed points are obtained by intersecting the epipolar lines of the left seed points with the curve in the right image; and
- For each seed point on the left curve and its corresponding point on the right curve, we compute a correlation score between their neighborhoods.

However, a major change has been made to ensure that the score computation is less sensitive to the image transformation, as we detail below.

4.1 Seed score

Refer to Fig. 2. For each seed point k_L in the left image, a neighborhood $N(k_L)$ is assigned. The neighborhood $N(l_R)$ of the corresponding seed point l_R on the right image is defined by $N(k_L)$ and a similarity transformation (see Appendix A). The similarity transformation can be estimated from a pair of points inside the neighborhood and their corresponding points in the right image. We have already the pair of seed points, and we need to select another pair. We can choose the extreme point of $N(k_L)$ along the curve, say $\mathbf{p}_1$, and compute its corresponding point $\mathbf{p}_1'$ by intersecting the epipolar line of $\mathbf{p}_1$, l_{p_1}, with the curve in the right image. The estimation of the similarity transformation from two pairs of points is given in Appendix A.

The seed score is computed as the correlation between these two corresponding neighborhoods, and is given by

$$S(k_L, l_R) = \sum_{\substack{\xi \in N(k_L) \\ \eta \in N(l_R)}} \frac{[I_L(\xi) - \bar{I}(k_L)]\,[I_R(\eta) - \bar{I}(l_R)]}{N_n\, \sigma(I_{k_L})\, \sigma(I_{l_R})} \quad (13)$$

where $I_i(x)$ is the intensity value on a point x in the ith image, $N(x)$ is the neighborhood for seed point x, η is a point in the second image corresponding to point ξ in the first image according to the similarity transformation described above, and N_n is the number of points in the neighborhood. The details of computing the mean ($\bar{I}(x)$) and the standard

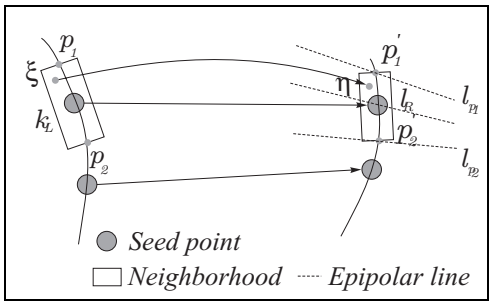

Figure 2: The neighborhood of the seed point in the left image and its correspondence in the right image

deviation ($\sigma(x)$) in $N(x)$ can be found in [19]. It is obvious that the correlation score thus computed is invariant under the similarity transformation.

4.2 The curve score as the combined unary measurement

The curve score $L(i_L, j_R)$ for the ith curve in the left image and the jth curve in the right image, which is the combined unary measurement in our context, is the average of all possible seed scores. That is,

$$L(i_L, j_R) = \frac{1}{N_s} \sum_{\substack{k_L \in Sd(i_L) \\ l_R \in Sd(j_R)}} S(k_L, l_R) \tag{14}$$

where N_s is total number of seed points on i_L, and $Sd(i_L)$ and $Sd(j_R)$ are the seed point sets of the left and right curves, respectively.

4.3 Similarity function

We can now compute the similarity function. The combined unary measurement $\mathbf{x}_i^{(\alpha)}$ in (11) is a scalar in our case, and will be denoted by $x_i^{(\alpha)}$. It is equal to $L(\theta_i, \omega_\alpha)$ computed above. According to Bayes' theorem, we have

$$\begin{aligned} & P(\theta_i = \omega_\alpha \mid x_i^{(\alpha)}) \\ = \; & \frac{p(x_i^{(\alpha)} \mid \theta_i = \omega_\alpha)\hat{P}(\theta_i = \omega_\alpha)}{\sum_{\omega_\lambda \in \Omega} p(x_i^{(\alpha)} \mid \theta_i = \omega_\lambda)\hat{P}(\theta_i = \omega_\lambda)} \end{aligned} \tag{15}$$

where $\hat{P}(\theta_i = \omega_\lambda)$ is the prior probability equal to a prefixed value ζ if $\lambda = 0$ (i.e., no match), and to $(1 - \zeta)/M_0$ otherwise. We assume that $p(x_i^{(\lambda)} \mid \theta_i = \omega_\lambda) = \mathcal{N}(\mu_l, \sigma_l)$. The value of μ_l and σ_l can be computed from the histogram of $x_i^{(\lambda)}$ using the initial matches. The initial matches could be obtained by selecting only the labeling with $x_i^{(\lambda)}$ that is the highest in both $\{x_i^{(\beta)} \mid \beta \in M_0\}$ and $\{x_{(j)}^{(\lambda)} \mid j \in N_0\}$. Obviously, the quality of the initial

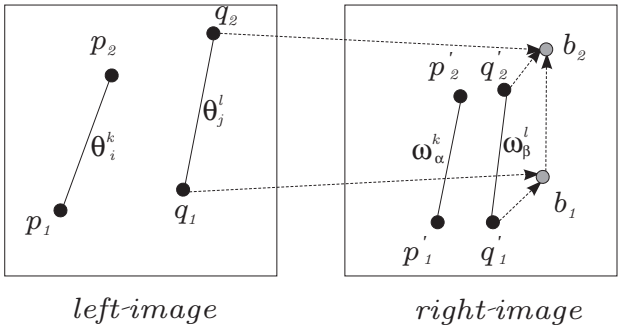

Figure 3: Configuration of two pairs of line segments in both images

matches will affect the accuracy of the similarity function, and therefore affect the final result. However, since the computational framework is quite robust, the impact is not significant.

The unary measurements are used in (15) to compute $P(\theta_i = \omega_\alpha \mid x_i^{(\alpha)})$. It is then used to initialize the relaxation scheme, that is: $P^{(0)}(\theta_i = \omega_\alpha) = P(\theta_i = \omega_\alpha \mid x_i^{(\alpha)})$.

It is interesting to note that since our unary measurement is similarity invariant, there is no need to introduce a motion related term as in [7] for the computation of the similarity function.

5 The combined binary measurement and the compatibility function

In order to efficiently compute combined binary measurements, we approximate a curve by a set of line segments connected by the seed points mentioned in section 4. This approximation is accurate enough provided that an edge chain is broken at the high curvature points (to be detailed in section 7), and that seed points are not very far from each other. A set of combined binary measurements for the line segments is proposed. These measurements are grouped together in an optimal way to form the combined binary measurements for curves. The compatibility function for curves can then be computed. The uncertainty measurements for the combined binary measurements are also given in this section.

5.1 Combined binary measurements for line segments

As we mentioned before, the similarity transformation is a reasonable mapping from a neighborhood in one image to the corresponding neighborhood in the other. Consider two pairs of line segments illustrated in Fig. 3. The similarity transformation $\mathbf{x}' = s\mathbf{R}_\theta\mathbf{x} + \mathbf{t}$ can be computed from $(\mathbf{p}_1, \mathbf{p}'_1)$ and $(\mathbf{p}_2, \mathbf{p}'_2)$, as described in Appendix A. We then compute $\mathbf{b}_1 = s\mathbf{R}_\theta\mathbf{q}_1 + \mathbf{t}$, $\mathbf{b}_2 = s\mathbf{R}_\theta\mathbf{q}_2 + \mathbf{t}$, and form a random vector $\mathbf{z}$ as the follows

$$\mathbf{z} = [\mathbf{v}_1^{(x)}, \mathbf{v}_1^{(y)}, \mathbf{v}_2^{(x)}, \mathbf{v}_2^{(y)}]^\top \tag{16}$$

where $\mathbf{v}_1 = \mathbf{b}_1 - \mathbf{q}'_1$, and $\mathbf{v}_2 = \mathbf{b}_2 - \mathbf{q}'_2$. Ideally, if the local similarity transformation is valid and the point coordinates are noise-free, we have $\mathbf{z} = \mathbf{0}$. These conditions are not satisfied in practice, and we assume that the components of $\mathbf{z}$ are i.i.d, and have the same standard deviations σ, i.e., $\mathbf{z} = \mathcal{N}_{\mathbf{z}}(\mathbf{0}, \Sigma)$, where $\Sigma = diag(\{\sigma\})$.

There are several advantages of the binary measurements $\mathbf{z}$ given here over the traditional ones.

- Measurements are invariant under the similarity transformation. Traditional measurement such as the distance between the midpoints of line segments may suffer from the scale change between views.
- The set of binary measurements is complete. Under similarity transformation, two pairs of line segments have 8 equations, and there are 4 degrees of freedom for the similarity transformation. Therefore, the number of invariants is equal to 4, and we have the same number of measurements.
- Measurements are evaluated in the same physical space, namely, they are quantified in pixels in the second image. This means that we only need one σ for all the measurements, whose computation is given in section 5.3.

5.2 Combined binary measurement for curves and the compatibility function

Now consider a pair of curves a_i and a_j. The numbers of line segments are k_i and k_j, respectively. For each line segment k on the shorter curve, the closest segment from the other curve is selected, and measurement $\mathbf{z}_k$ as defined in (16) is computed. The rationale of choosing the closest line segments is that the similarity transformation better applies to a small neighborhood than to a larger one. For the vector set $\{\mathbf{z}_j \mid j \in [1, \ldots, K]\}$, where $K = \min(k_i, k_j)$, consider the following measurement vector

$$\widetilde{\mathbf{z}} = \zeta_1 \mathbf{z}_1 + \cdots + \zeta_K \mathbf{z}_K \tag{17}$$

where ζ_k's are coefficients and $\sum_{k=1}^{K} \zeta_k = 1$. It is obvious that $\widetilde{\mathbf{z}}$ is also a joint Gaussian with $\widetilde{\mathbf{z}} = \mathcal{N}_{\widetilde{\mathbf{z}}}(\mathbf{0}, \widetilde{\Sigma})$, where $\widetilde{\Sigma} = diag[\{\widetilde{\sigma}\}]$, and

$$\widetilde{\sigma}^2 = \zeta_1^2 \sigma_1^2 + \cdots + \zeta_K^2 \sigma_K^2 \tag{18}$$

We then select ζ_j to minimize (18), which yields

$$\zeta_i = [\sigma_i^2 \sum_{j=1}^{K} \sigma_j^{-2}]^{-1} \tag{19}$$

Equation (17) is an unbiased estimate of the measurement error with minimal variance. The compatibility function is then given by

$$\begin{aligned} &p(\mathbf{A}_{ij}^{(\alpha\beta)} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \mathbf{B}_{ij}) = \\ &\quad p(\widetilde{\mathbf{z}} \mid \theta_i = \omega_\alpha, \theta_j = \omega_\beta, \widetilde{\Sigma}) = \mathcal{N}_{\widetilde{\mathbf{z}}}(\mathbf{0}, \widetilde{\Sigma}) \end{aligned} \tag{20}$$

where $\mathbf{A}_{ij}^{(\alpha\beta)} = \widetilde{\mathbf{z}}$ is the combined binary measurements, and $\mathbf{B}_{ij} = \widetilde{\Sigma}$ is the uncertainty measurements.

5.3 The distance between segments and the standard deviation

The standard deviation σ of vector $\mathbf{z}$ in (17) should be a function of the distance d between the line segments. This is because the local similarity assumption becomes weaker when d becomes larger. The function that we use is

$$\sigma(d) = \frac{\rho}{\sqrt{2\pi}}[(1-\gamma)(1-e^{-\frac{d^2}{\tau^2}}) + \gamma] \tag{21}$$

where $\rho = \min(W, H)$, W and H are respectively the image width and height, τ is a positive scalar that controls the range within which the contribution from neighboring segment is effective, and $\gamma = \sqrt{2\pi}\sigma^0/\rho$ where σ^0 is the desired standard deviation when the neighboring segment is very close to the segment under consideration. Several observations can be made for the function defined in (21):

- When d is small, $\sigma(d) \approx \sigma^0$ which is the desired standard deviation when neighboring segments are close.
- When d is much larger than τ, $\sigma(d) \approx \rho/\sqrt{2\pi}$. In this case, $\mathcal{N}_{\mathbf{z}}(\mathbf{0}, \Sigma) \approx 1/\rho^4$, indicating that the probabilistic density function approaches to a uniform distribution, and provides no information about the compatibility between the two pairs of line segments.
- The intermediate value of $\sigma(d)$ varies smoothly between the above two extreme cases.

As a consequence of the second point above, when all the distance measurements $\{d_j \mid j \in [1, \ldots, K]\}$ are much larger than τ, the compatibility function in (20) is approximately a constant $1/\rho^4$. As a result, if all the other curves are far from the current curve a_i, it is not difficult to see, from (8) and the definition of $\mathbf{Q}$, that $P^{(n+1)}(\theta_i = \omega_{\theta_i}) = P^{(n)}(\theta_i = \omega_{\theta_i})$. This is in accordance with our intuition that the labeling probability can not be improved when the neighborhood support is weak.

The adjustment of the parameter τ depends on the type of the scene in question. If the scene is close to the camera, there is a severe perspective distortion, and τ needs to be small enough to discard the support from curves far away. On the other hand, if the scene is far from the camera, τ could be large in order to include support from as many neighboring curves as possible.

6 Corner guidance

The method presented in this section exploits the corner matching constraint and the local similarity constraint to provide stronger guidance for curve matching. In our implementation, corners are detected by Harris corner detector, and corners across images are matched

using the technique described in [17]. In the following, we describe our corner-guidance algorithm within the same probabilistic relaxation framework as described in Section 3, and the role of the corner guidance is explicitly defined by a set of similarity-invariant unary measurements and a similarity function. Uncertainty measurements for the unary measurements can also be computed. These measurements are related to the distance from a corner to a curve, and are used to control the impact of corner guidance on curve matching. The similarity function for corner guidance can then be defined in this high dimensional measurement space.

The advantage of introducing the corner guidance can be seen from (7) and (8). Obviously, the computational complexity for each probability update in the relaxation scheme is proportional to $\overline{N}_s[\overline{\mathcal{S}}(\Psi_i)][\overline{\mathcal{S}}(\Omega_i)]^2$, where $\overline{N}_s$ is the average number of curve segments, Ω_i denotes the candidate set of a_i, Ψ_i denotes the neighbor set of a_i, $\mathcal{S}(\cdot)$ denotes the number of elements in a set, $\overline{\mathcal{S}}(\Psi_i) = \sum_{i=0}^{N-1} \mathcal{S}(\Psi_i)/(N-1)$, and $\overline{\mathcal{S}}(\Omega_i) = \sum_{i=0}^{N} \mathcal{S}(\Omega_i)/N$. If all other objects are regarded as neighbors and all models are regarded as candidates, relaxation will be extremely slow. To reduce the computational complexity, objects in Ψ_i are only selected for those within a certain distance to a_i. It is however, more difficult to reduce the size of Ω_i. Epipolar constraint is used by previous work for this purpose. Use the algorithm developed in this section, it is possible to further reduce the size of Ω_i, i.e. by a factor of 6, which means to accelerate the probability update in (8) by a factor of 36. As a result of reducing the size of Ω_i, matching ambiguity is also reduced, and results for curve matching are more accurate.

Our treatment of the corner guidance is quite similar to the binary measurements discussed in Section 5. A curves is approximated by a set of line segments as before. A corner is said to be the neighbor of a curve if its distance to the curve is less than a predefined threshold. The distance from a corner to a curve is defined by the minimum distance from the corner to the midpoints of all line segments on the curve. We again assume that the mapping from a neighborhood in one image to the corresponding neighborhood in the other can be approximated by a local similarity transformation. Consider a pair of line segments and a pair of corners illustrated in Fig. 4. The similarity transformation $\mathbf{x}' = s\mathbf{R}_\theta\mathbf{x} + \mathbf{t}$ can be computed from $(\mathbf{p}_1, \mathbf{p}_1')$ and $(\mathbf{p}_2, \mathbf{p}_2')$. We then compute $\mathbf{b} = s\mathbf{R}_\theta\mathbf{q} + \mathbf{t}$, and form a random vector $\mathbf{z}$ as the follows

$$\mathbf{z} = [\mathbf{v}^{(x)}, \mathbf{v}^{(y)}]^\top \tag{22}$$

where $\mathbf{v} = \mathbf{b} - \mathbf{q}'$.

Now consider a pair of curves θ_i and ω_α. For each neighboring corner of θ_i, the closest segment is selected, and measurement $\mathbf{z}_k$ as defined in (22) is computed. Because of the similarity between (22) and (16), we can apply the same methods as in (17) to integrate line measurements $\mathbf{z}_k$s into curve measurement $\widetilde{\mathbf{z}}$. We can also compute the uncertainty of $\mathbf{z}$ from the distance between corner and the line segment in a similar way as in (21). The similarity function of corner guidance can then be computed by (15), where the likelihood function is

$$\begin{aligned} &p(\mathbf{x}_i^{(\alpha)} \mid \theta_i = \omega_\alpha) = \\ &\quad p(\widetilde{\mathbf{z}} \mid \theta_i = \omega_\alpha) = \mathcal{N}_{\widetilde{\mathbf{z}}}(\mathbf{0}, \widetilde{\sigma}^2\mathbf{I}_2) \end{aligned} \tag{23}$$

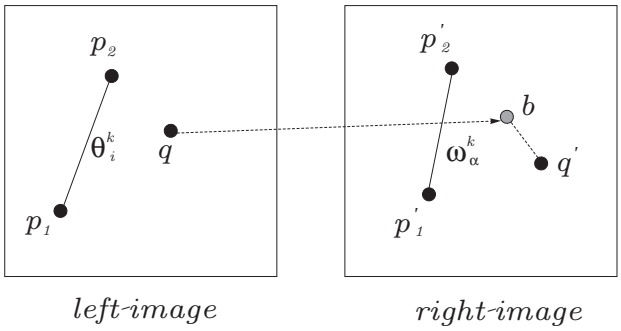

Figure 4: Configuration of a pair of line segments and a pair of corners in both images

and $\mathbf{x}_i^{(\alpha)} = \widetilde{\mathbf{z}}$ is the combined unary measurements for corner guidance.

7 Preprocessing

In our matching algorithm described above, we have made two major assumptions. The first is that a seed point on one curve can match at most one seed point on the other. The second is that in computing binary measurements a curve is approximated by a piece-wise linear model. In this section, we describe two preprocessing techniques to ensure that these assumptions are satisfied.

7.1 Epipolar breaking

When intersecting with a curve, an epipolar line may cut the same curve several times. In order to reduce the matching complexity, we have assumed that the epipolar line of a seed point in one image can only cut at most once the corresponding curve in the other image. The technique described below breaks curves into shorter pieces, and each piece only cuts an epipolar line once.

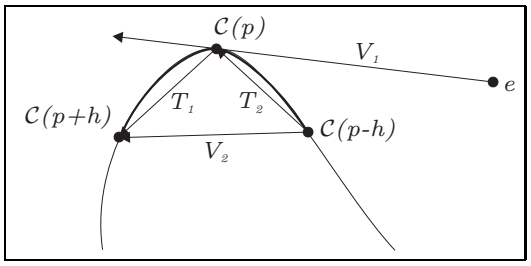

Figure 5: The computation of the high curvature and the epipolar break point

The technique is based on the following observations:

- The only epipolar line passing through a given point p on a curve in one image is the line connecting point p and epipole e in that image, as shown in Fig. 5.

- It is impossible for a curve to have multiple intersections with an epipolar line if there is no point on the curve whose tangent vector is parallel to the epipolar line passing through that point.

It follows that if we can break a curve at the point p where the tangent vector $\mathbf{T}(p)$ is parallel to the epipolar line $\mathbf{e}_p$, the problem will be solved.

We compute the cosine of the angle between the tangent vector and the epipolar line:

$$\varepsilon_p = \overline{\mathbf{T}(p)} \cdot \overline{\mathbf{e}_p}$$

where $\overline{\mathbf{v}}$ is the normalized vector of $\mathbf{v}$. As depicted in Fig. 5, an index worm $w_p(s) = p + s, s = [-h, \ldots, h]$ of size $2h + 1$, centered on the pth edge point, crawls along the edge chain, and ε_p is computed each time when an edge point is visited. The tangent vector $\mathbf{T}(p)$ is approximated by $\mathbf{V}_2$ in Fig. 5. The original edge chain is broken at point p if its ε_p is the local maxima or minima, and the absolute value $|\varepsilon_p|$ is larger than a predefined threshold (0.99 in our implementation). Therefore, We do not break curves at reflection points.

7.2 High curvature breaking

In order to have a good piecewise linear approximation to a curve, it is necessary to break the curve at points with high curvature. The technique works in a very similar way as the epipolar breaking technique described above. The curvature is approximated by

$$\kappa_p = \frac{\|\overline{T_1} - \overline{T_2}\|}{\|T_1\| + \|T_2\|} \tag{24}$$

where $T_1 = \mathcal{C}(w_p(h)) - \mathcal{C}(w_p(0))$, $T_2 = \mathcal{C}(w_p(0)) - \mathcal{C}(w_p(-h))$, and $\overline{T_j}$ denotes the normalized vector of T_j. The original edge chain is broken at point p if its curvature κ_p is the local maxima and is larger than a predefined threshold (0.85 in our implementation).

8 Experimental results

We present experimental results on three pairs of real images of complex scenes. Results of curve matching are shown by overlapping color coded curves onto the original images. An identical color for a pair of curves indicates a match, and a pair of pink arrows highlights a mismatch. For the success rate, we only counted the curves that survived after the matching process. There are several reasons that a curve may be killed during the matching process. For example, if the motion between an image pair is large, a number of curves in the left image are then not visible in the right image. Furthermore, edge detection and linking algorithms may sometimes produce inconsistent curves between an image pair. These curves will also be killed. Finally, curves can, of course, be mistakenly killed by the matching process itself. One known problem of our algorithm is that, if a physically

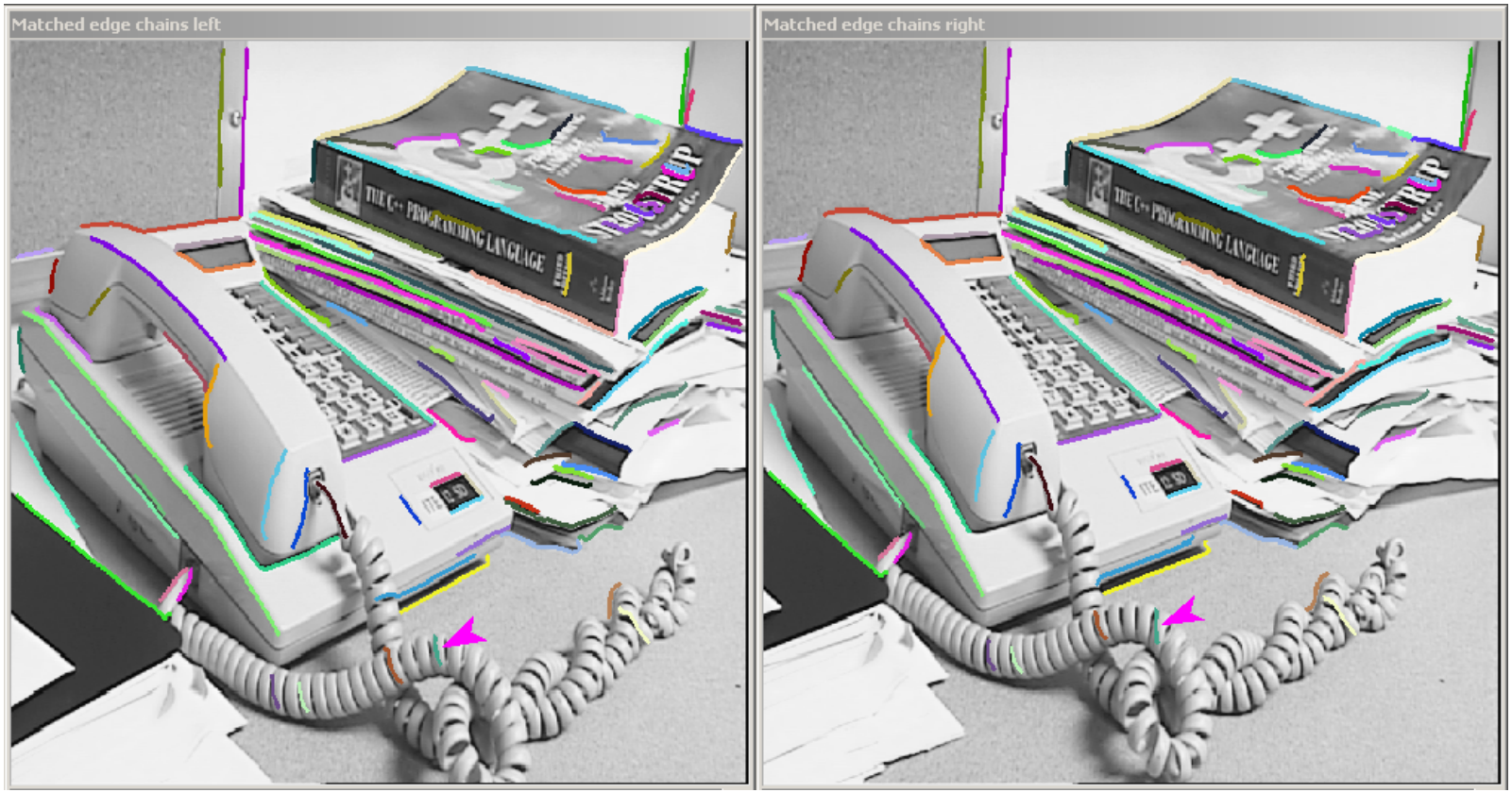

Figure 6: Matching results for an indoor scene taken by a camera closed to the scene. 107 out of 108 matches are correctly matched and the rate is above 99%. The bad match is marked with a pink arrow in the images for reader's convenience.

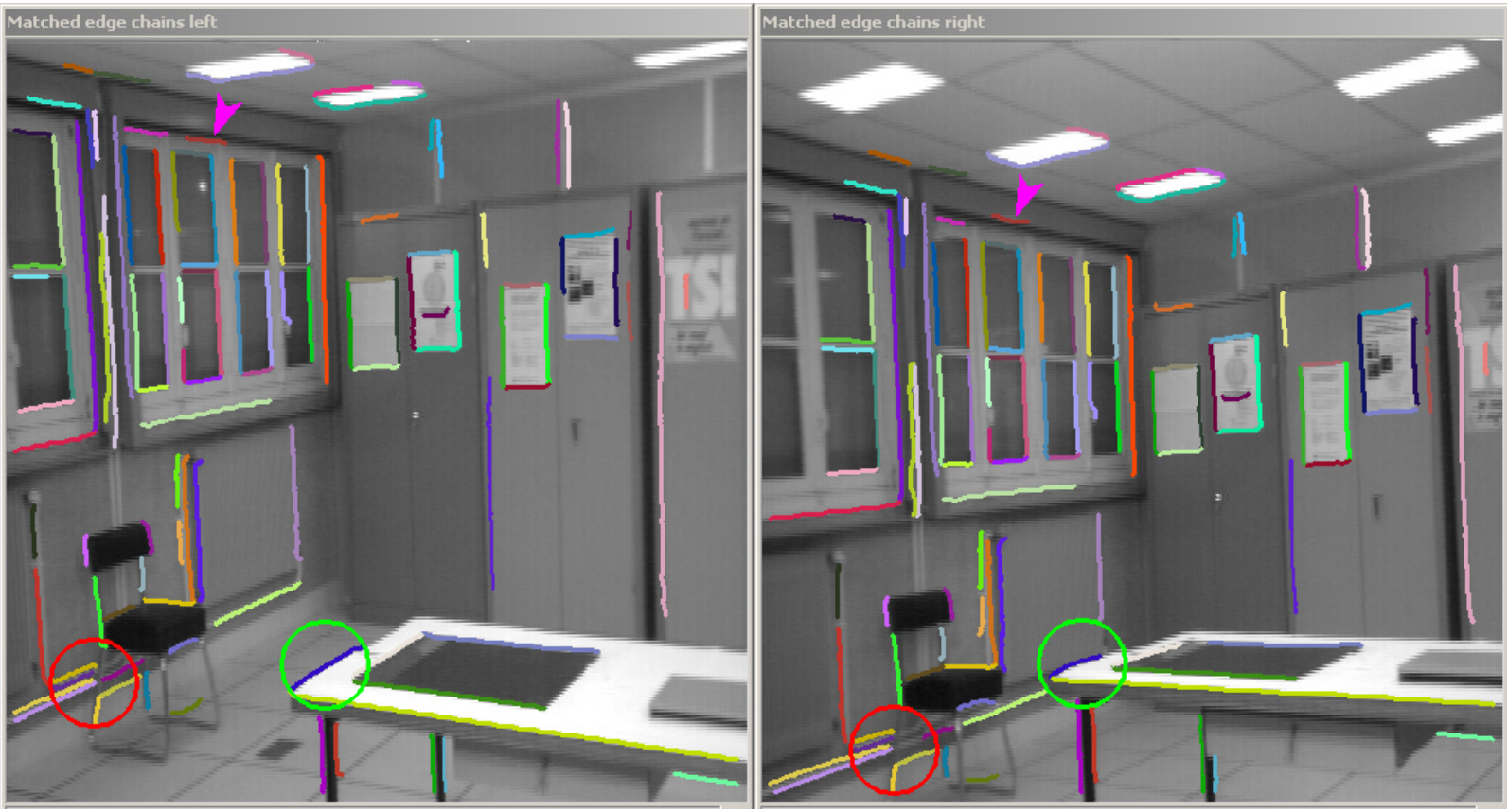

Figure 7: Matching results for an indoor scene. 109 out of 110 matches are correctly matched and the rate is above 99%. The bad match is marked with a pink arrow in the images for reader's convenience.

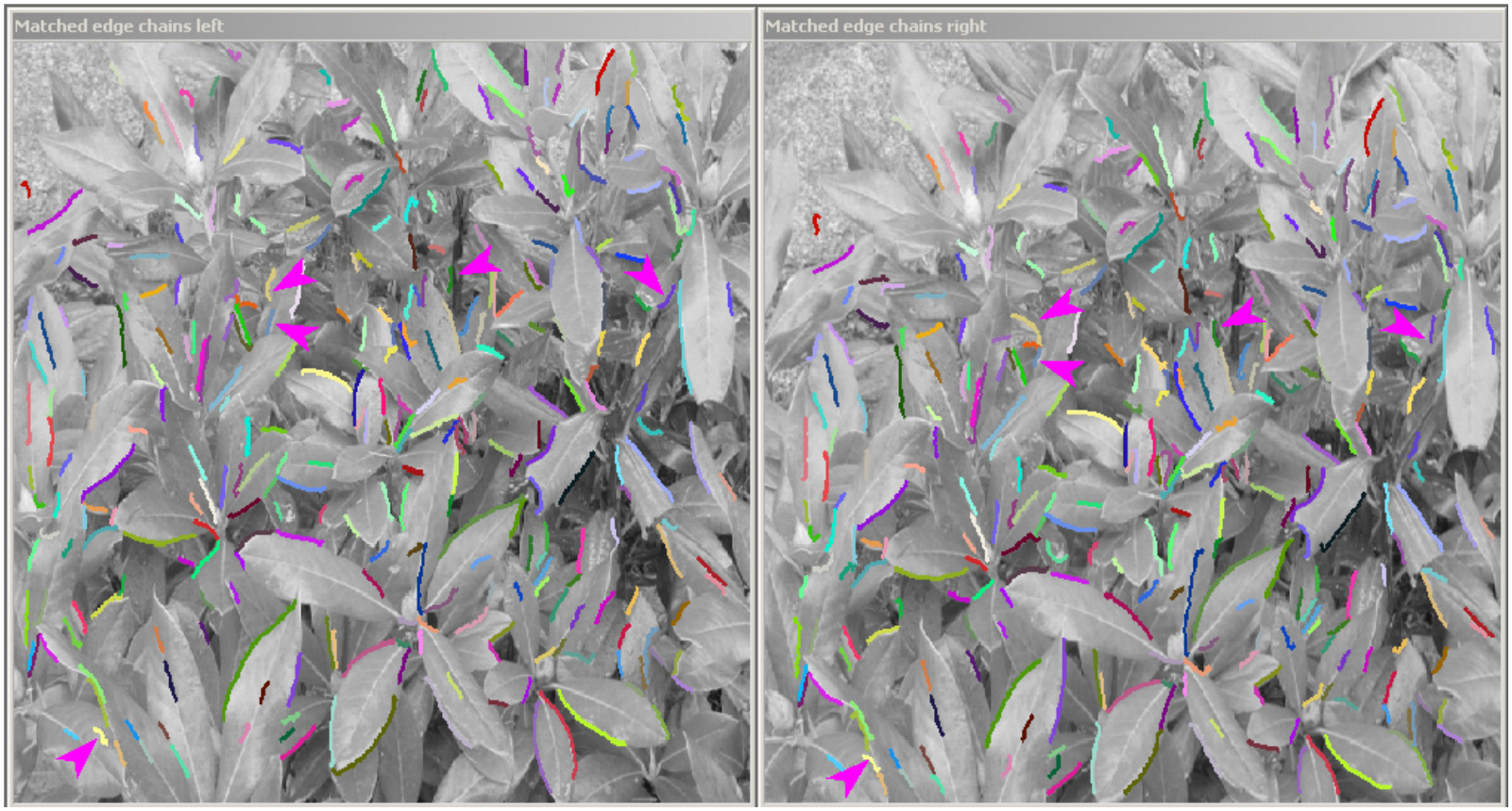

Figure 8: Matching results for a nature scene. 305 out of 310 matches are correctly matched and the rate is 98%. All bad matches are marked with pink arrows in the images for reader's convenience.

long curve is broken into several fragments, it can only match one fragment, leaving other fragments unmatched (killed). Since curves can be killed by a number of reasons that may not be directly related to the performance of our matching algorithm, we didn't count killed curves into the success rate. Another thing to notice is that the corners in the context of our experiments are those that have been matched. We used robust method to remove outliers such as the false corners on the occlusion boundaries. All experiments were conducted on a 850Hz PIII machine on the Windows 2000 platform, and the code was not optimized.

Fig. 6 is the result for an indoor scene which is taken by a camera close to the scene. There is a significant perspective distortion between the left and the right image. The program found 108 matches, and only one is bad. The running time for the whole process, including edge detection, edge linking and curve matching, is 38.2 seconds. In this example, the lines on the book sides are very close to the epipolar lines. This affects the first stage when we try to find the initial matches and to compute the unary measurements. As long as a right curve is on the epipolar line of a point on a left curve, the right curve will be regarded as a candidate of the left curve. In the later stages, the compatibility function and the relaxation scheme will take care of the rest part of the matching and the unary measurements do not play a significant role. This is why these lines are correctly matched.

Fig. 7 is another indoor scene extracted from an INRIA stereo sequence. Despite the significant camera motion between the two views, the program found 110 pairs, and only one is bad. The running time for the whole process is 23.3 seconds. Our algorithm may get confused on the occlusion boundaries only when the local similarity constraint is violated. This happens when the depth variation along the occlusion boundary is significantly large as compare to the distance from that part of the scene to the camera. In the case when the scene structure is relatively shallow, the compatibility function still works. This has been demonstrated with matched curves inside the red circles in both images. On the other hand, when there is little support from the neighborhood, the compatibility function will become a constant, and the similarity function will take over the control for that particular curve. This may help some curves on the occlusion boundaries to survive in the matching process (see the curves inside the green circles).

Fig. 8 is the result for an outdoor scene with many curves. This scene will be referred as leaf image hereafter. The program found 310 pairs, 305 of which are correctly matched. The running time for the whole process is 72.1 seconds, of which the time used for probability update in (8) is 23.8 seconds.

From all three tests above, we see that the average rate of correct matching is higher than 98%.

To demonstrate the power of corner guidance, we applied the method discussed in Section 6 directly to the leaf image without using relaxation and other unary measurements. A pair of curves whose similarity value (computed from (15) and (23)) larger than 0.5 is selected as a match. It can be seen in Fig. 9 that the corner matching constraint is able to give a reasonable result. Fig. 10 is result when the corner guidance is integrated into the relaxation method as a unary measurement, where 3 out of 301 matches are bad. As compared with the result in Fig. 8, the accuracy is slightly improved. On the other hand,

$\mathcal{S}_c$'s (the average number of candidate curves defined in Section 6) for the epipolar line constraint alone and the corner matching constraint combined are 32.1 and 5.8, respectively. The latter is almost 6 times less than the former. This reduces the time used for probability update from previous 23.8 seconds to 0.9 second. The running time for the whole process is reduced from 72.1 seconds to 44.2 seconds, of which 7.9 seconds are used to compute corner guidance.

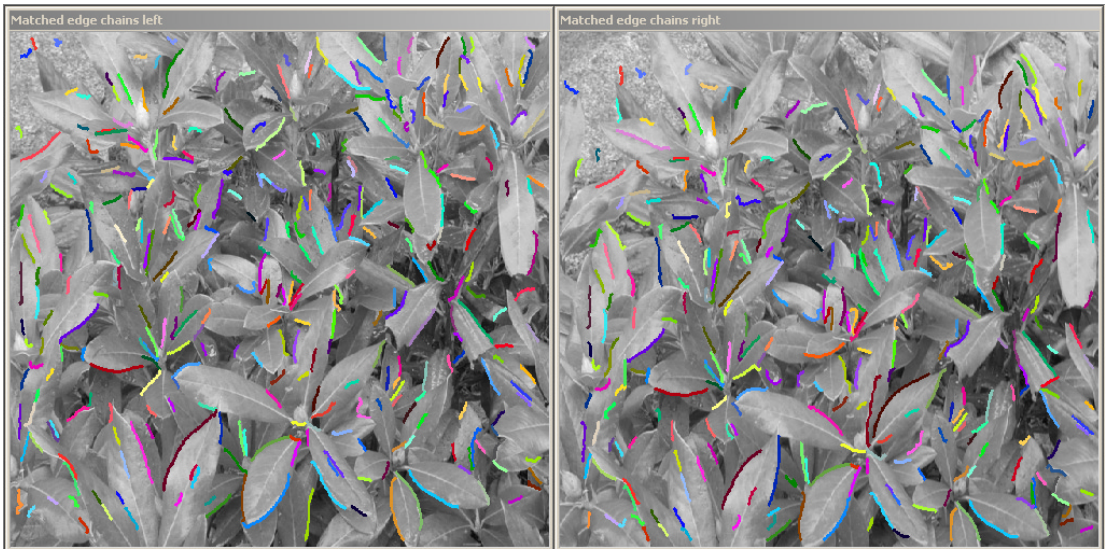

Figure 9: Initial set of matched curves obtained only with the corner matching constraint

9 Conclusions and extensions

A new algorithm for the curve matching has been developed within a probabilistic relaxation framework. We have proposed using the combined unary and binary measurements in this framework, which are similarity-invariant. We have modeled explicitly the uncertainty in binary measurements, which is very important in computing the matching support from neighboring matches. We have also introduced a novel unary measurement,the corner guidance, to improve the accuracy and the efficiency of the curve matching. Preprocessing techniques contributing to the success of the curve matching have been developed and discussed. Experiments with complex real scenes show that the rate of correct matching is higher than 98%.

Because of imperfect edge linking, a physically long curve may be broken into several small pieces in one image. With the current algorithm, only one piece will match the long curve. One interesting future work is to match all these small pieces to the long curve.

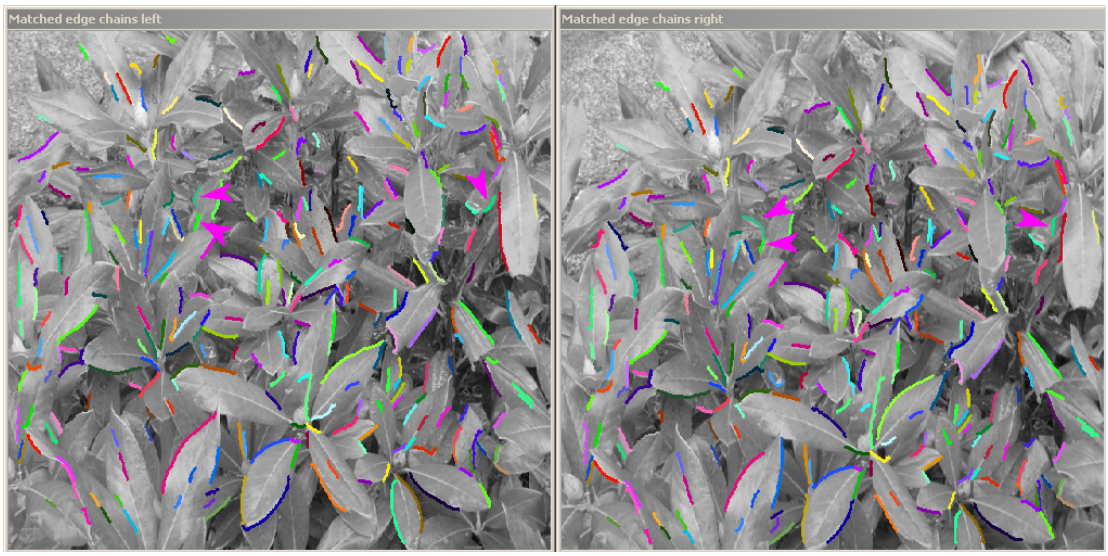

Figure 10: Matched curves with our curve matching method with the corner matching constraint integrated

A The similarity transformation

A similarity transformation is of the following form:

$$\mathbf{x}' = s\mathbf{R}_\theta \mathbf{x} + \mathbf{t} \tag{25}$$

where s is the scale, $\mathbf{R}_\theta$ is the rotation matrix with angle θ, and $\mathbf{t}$ is the translation vector. It maps a point $\mathbf{x}$ in one image to a point $\mathbf{x}'$ in the other. If we have two pairs of matched points $(\mathbf{p}_1, \mathbf{p}'_1)$ and $(\mathbf{p}_2, \mathbf{p}'_2)$, it is not difficult to obtain

$$\begin{cases} s &= \frac{\|\mathbf{p}'_1 - \mathbf{p}'_2\|}{\|\mathbf{p}_1 - \mathbf{p}_2\|} \\ \cos\theta &= (\overline{\mathbf{p}_1 - \mathbf{p}_2}) \bullet (\overline{\mathbf{p}'_1 - \mathbf{p}'_2}) \\ \mathbf{t} &= [(\mathbf{p}'_1 + \mathbf{p}'_2) - s\mathbf{R}_\theta(\mathbf{p}_1 + \mathbf{p}_2)]/2 \end{cases} \tag{26}$$

where $\overline{\mathbf{v}}$ denotes the normalized vector of $\mathbf{v}$.

Acknowledgment: We thank Rick Szeliski for his sub-pixel edge detection code and for checking the English, and reviewers for useful comments.

References

[1] Stephen B Pollard, John E. W. Mayhew, and John P Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. In John E. W. Mayhew and John P. Frisby, editors, *3D model recognition from stereoscopic cues*, pages 11–25. MIT Press, 1991.

[2] Stephen B Pollard, John E. W. Mayhew, and John P Frisby. Implementation details of the PMF stereo algorithm. In John E. W. Mayhew and John P. Frisby, editors, *3D model recognition from stereoscopic cues*, pages 33–41. MIT Press, 1991.

[3] Cordelia Schmid and Andrew Zisserman. Automatic line matching across views. In *CVPR'97*, pages 666–671, 1997.

[4] G. Médioni and R. Nevatia. Segment-based stereo matching. *CVGIP*, 31:2–18, 1985.

[5] Nasser M. Nasrabadi. A stereo vision technique using curve-segments and relaxation matching. *IEEE Trans. PAMI*, 14(5):566–572, 1992.

[6] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis*. Springer-Verlag, 1992.

[7] William J. Christmas, Josef Kittler, and Maria Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. PAMI*, 17(8):749–764, August 1995.

[8] J. Kittler and E.R. Hancock. Combining evidence in probabilistic relaxation. *Int'l J. Pattern Recognition and Artificial Intellegent*, 3:29–51, 1989.

[9] A. Rosenfeld, R.Hummel, and S.Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems, Man, and Cybernetics*, 6:420–433, 1976.

[10] M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *J. Math. Imageing Vision*, 7:309–323, 1997.

[11] Peter N. Belhumeur. A bayesian approch to binocular stereopsis. *IJCV*, 19(3):237–260, 1996.

[12] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *IJCV*, 1987.

[13] N. Ayache. *Stereovision and sensor fusion*. MIT Press, 1990.

[14] Luc Robert and Olivier Faugeras. Curve-based stereo: Figural continuity and curvature. In *Proc. CVPR'91*, pages 57–62, 1991.

[15] Frank Candocia and Malek Adjouadi. A similarity measure for stereo feature matching. *IEEE Trans. IP*, 6(10):1460–1464, October 1997.

[16] S. Alibhai and S.W. Zucker. Contour-based correspondence for stereo. In D. Vernon, editor, *Proceedings of the 6th European Conference on Computer Vision*, volume I, pages 314–330, Dublin, Ireland, June 2000.

[17] Z.Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence journal*, 78:87–119, 1995.

[18] Z. Zhang and Y. Shan. A progressive scheme for stereo matching. In M. Pollefeys et al., editor, *3D Structure from Images – SMILE 2000*, volume 2018 of *Springer LNCS*, pages 68–85. Springer-Verlag, 2001.

[19] Gang Xu and Zhengyou Zhang. *Epipolar geometry in stereo, motion and object recognition*. Kluwer Academic Publishers, 1996.

[20] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *The International Journal of Computer Vision*, 27(2):161–195, 1998.

[21] R.A. Hummel and S.W.Zucker. On the fundations of relaxation labeling process. *IEEE Trans. PAMI*, 5(3):267–286, May 1983.