2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Minimum-Error Active Matching for Real-time Vision

Zhibin Liu, Zongying Shi, Wenli Xu

Abstract— **As an integral part of real-time vision system, there are two most important requirements for feature matching mechanisms: high computational efficiency for meeting the real-time demands, and high correct matching rate for ensuring the convergence and consistency of state estimation. Both of these are addressed and solved as an integrated whole by the efficient minimum-error active matching scheme proposed in this paper. Image processing is performed in a dynamically guided fashion by checking only parts of the image where positive matches are most probable. For achieving the global consensus matchings, rigorous analysis on how to minimize the matching errors in active matching by choosing an optimal search order is made. After that, practical feature matching algorithms are given, which have naturally absorbed the ideas of nearest neighbor (NN) and joint compatibility branch and bound (JCBB) approaches. Both statistical simulations and real-world experimental results have verified the proposed methods can perform better than the state-of-the-art algorithms, i.e. being able to obtain the best global consensus matchings with much lower computational cost.**

I. INTRODUCTION

Feature matching, which is also often referred to as data association problem in target-tracking and robotics communities, is most critical for ensuring the convergence and consistency of all kinds of filtering or other estimation algorithms used in a variety of tracking [1], structure from motion [2], visual odometry [3] and visual SLAM [4], [5] applications. Usually, there are two most important requirements for any feature matching mechanism to be implemented in real-time vision systems: high computational efficiency and high correct matching rate.

The existing works on feature matching can be roughly divided into two classes: passive methods [2]–[4] and active methods [5]–[8]. In passive methods, image processing is treated as a separable step — a bottom-up operation applied uniformly to incoming images, detecting or re-finding various features. On the contrary, active methods usually do not scan the whole image exhaustively, but rather in a guided and dynamically planned way. So, generally, higher computational efficiency can be expected from active methods.

In [5], Davison presented a feature-by-feature search algorithm which can be viewed as the embryo of active matching, and it has been widely used in a lot of state-of-the-art works on visual SLAM [9]–[12]. Then, in [7], more rigorous algorithms for active search have been proposed, which makes explicit decisions on the problem about which feature should be searched first, by examining a kind of mutual information scores. Recently, Chli *et al.* [8] has proposed an extension of this method which uses multiple hypothesis tracking to deal with the cases where matching ambiguity arises. However, there are still some limitations: 1) as the number of spurious candidate matchings for the features gets large, the number of hypotheses will increase and the multi-hypothesis tracking will be less efficient; 2) there are risks that correct hypothesis may be assigned with low weight and falsely pruned, especially when the number of features and spurious candidate matchings gets large; 3) its performance is sensitive to some parameters which have to be finely tuned.

In this paper, an efficient minimum-error active matching scheme is proposed. It not only performs image processing in a dynamically guided fashion, but also directly aims for the best global consensus matchings. So, the two problems, i.e. enhancing computational efficiency and seeking for the best all-consistent matchings, are solved simultaneously. The rest of this paper is organized as follows: Section II presents rigorous analysis on how to minimize the matching errors in active matching by choosing an optimal search order. Theoretical lower bound for the objective function as well as the necessary and sufficient condition for achieving the lower bound have been derived. Then, in Section III, two practical minimum-error active matching algorithms are presented. The statistical simulations and real-world experiments results are presented and discussed in Section IV. Finally, conclusions are drawn in Section V.

II. OPTIMAL SEARCH ORDER FOR MINIMUM-ERROR ACTIVE MATCHING

In either model-based tracking or visual-SLAM, the current state of knowledge of an object or scene is usually modeled by a multi-dimensional probability distribution over a finite vector of parameters $\mathbf{x}$, representing the position, dynamics and other factors of interest. Usually there are one or more features corresponding to an object or scene structure. A measurement of feature i yields a vector of parameters $\mathbf{z}_i$, which describes for example the 2D image coordinates of a keypoint. At any given point in time, we usually have a set of features available to measure, and we define the stacked vector $\mathbf{z}_T = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)^\top$ for all candidate measurements, whose distribution is:

$$p(\mathbf{z}_T) = \int p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n|\mathbf{x})p(\mathbf{x})d\mathbf{x}. \quad (1)$$

According to the idea of active matching [7], [8], candidate matches are searched for feature-by-feature. We can factorize

This work is supported by the National Natural Science Foundation of China under Grant NO. 60732064.

The authors are with Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing, 100084, China liu-zb04@mails.tsinghua.edu.cn; {szy,xuwl}@mail.tsinghua.edu.cn.

the above equation as follows:

$$p(\mathbf{z}_T) = p(\mathbf{z}_{j_1})p(\mathbf{z}_{j_2}|\mathbf{z}_{j_1}) \cdot \cdots \cdot p(\mathbf{z}_{j_n}|\mathbf{z}_{j_1} \ldots \mathbf{z}_{j_{n-1}}), \quad (2)$$

where j_i, $i = 1, \ldots, n$, are indicators whose values are taken from $\{1,\ldots,n\}$, and $\forall l \neq k,\ \ j_l \neq j_k$. We term the sequence $\langle j_1, j_2, \ldots, j_n \rangle$ a *reasonable search order* (RSO). Equation (2) suggests that the first feature in the search order could be searched for simply according to its marginal distribution $p(\mathbf{z}_{j_1})$, while the other feature, for example the ith feature, $\forall i,\ 1 < i \leq n$, in the search order, should be searched for according to its conditional distribution $p(\mathbf{z}_{j_i}|\mathbf{z}_{j_1} \ldots \mathbf{z}_{j_{i-1}})$.

Obviously, the problem of how to choose a search order from the extremely large set of $n!$ candidate search orders needs to be dealt with explicitly. For it, two requisites are needed: 1) the criterion for defining a *good* search order; 2) the method for picking out such a search order. In literature, the existing criterion, i.e. the mutual information (MI) based criterion used by [7], [8], considers *enhancing computational efficiency*. However, since computational efficiency has already been greatly enhanced under active matching framework, it may be more reasonable to put emphasis on *enhancing the probability of correct data association*, which is of the upmost importance for ensuring the convergence and consistency of estimate. For the sake of this, we suggest a new criterion, i.e. *minimum potential matching errors*.

A. Minimum-Error Criterion

When the probability density of $\mathbf{x}$ can be characterized as a multi-variate Gaussian, the observer's knowledge about the system is captured by a state vector $\hat{\mathbf{x}}$ and covariance matrix $\Sigma_{\mathbf{xx}}$. Additionally, we assume the measurement noise $\mathbf{n}_i$ can be described by a multi-variate zero-mean Gaussian with covariance $\mathbf{R}_i$, which is independent for each measurement. Suppose $p(\mathbf{z}_T, \mathbf{x})$ is Gaussian, then we know that $p(\mathbf{z}_T)$ and $p(\mathbf{z}_T|\mathbf{x})$ are Gaussian, and for any reasonable search order $\langle j_1, j_2, \ldots, j_n \rangle$, the distributions $p(\mathbf{z}_{j_1})$, $p(\mathbf{z}_{j_2}|\mathbf{z}_{j_1})$, $\ldots$, $p(\mathbf{z}_{j_n}|\mathbf{z}_{j_1}\mathbf{z}_{j_2} \ldots \mathbf{z}_{j_{n-1}})$ are all Gaussian. We denote their mean vectors as $\hat{\mathbf{z}}_{j_1}$, $\hat{\mathbf{z}}_{j_2|j_1}$, $\ldots$, $\hat{\mathbf{z}}_{j_n|<j_1 j_2 \cdots j_{n-1}>}$, and covariance matrixes as $\Sigma_{\mathbf{z}_{j_1}}$, $\Sigma_{\mathbf{z}_{j_2}|\mathbf{z}_{j_1}}$, $\ldots$, $\Sigma_{\mathbf{z}_{j_n}|\mathbf{z}_{<j_1 \cdots j_{n-1}>}}$. These covariance matrixes represent the shape of 2D Gaussian PDFs over image coordinates. Suppose the number of standard deviations being chosen is N_σ ($N_\sigma > 0$, and usually we can choose $N_\sigma = 2$ or 3 to cover the 95% or 99% mass of probability respectively), then the area of the elliptical search region of any feature F_{j_i} can be calculated as:

$$S_{j_i} = \pi N_\sigma^2 \sqrt{\left|\Sigma_{\mathbf{z}_{j_i}|\mathbf{z}_{<j_1 \cdots j_{i-1}>}}\right|}. \quad (3)$$

Let's denote the actual projection of feature F_{j_i} in image as $\boldsymbol{f}_{j_i}^*$, and suppose $\boldsymbol{f}_{j_i}^*$ lies within the search region S_{j_i}. Assume that there are n_{j_i} *ghost points* in S_{j_i}, which appear quite similar to the feature F_{j_i} and are difficult to distinguish. Define $\lambda_{j_i} = n_{j_i}/S_{j_i}$, and assume it is constant for feature F_{j_i} during the current matching process, then the probability that all of the n features can be correctly matched is:

$$p(\text{correct}) = \frac{1}{1+\lambda_{j_1}S_{j_1}} \times \frac{1}{1+\lambda_{j_2}S_{j_2}} \times \cdots \times \frac{1}{1+\lambda_{j_n}S_{j_n}}. \quad (4)$$

Our objective is to choose an optimal search order $\langle k_1, k_2, \cdots, k_n \rangle_{opt}$ that can maximize (4), or equivalently minimize its denominator, i.e.

$$\begin{aligned} \langle k_1, k_2, \cdots, k_n \rangle_{opt} &= \underset{\langle j_1, j_2, \ldots, j_n \rangle}{\arg\min}\ \psi \\ &\triangleq \underset{\langle j_1, j_2, \ldots, j_n \rangle}{\arg\min}\ (1+\lambda_{j_1}S_{j_1})(1+\lambda_{j_2}S_{j_2}) \cdots (1+\lambda_{j_n}S_{j_n}). \end{aligned} \quad (5)$$

However, it is quite difficult to find out such an optimal search order. Generally, the derivative of ψ with respect to the order $\langle j_1, j_2, \ldots, j_n \rangle$ is hard to compute explicitly according to (5), so the minimum value of ψ can't be calculated analytically, and the classical hill-climbing algorithms are also inapplicable. In the following subsections, we will first analyze the intrinsic characteristics of the features' search regions, and then present an efficient and quite effective method for determining a sub-optimal search order.

B. Intrinsic Characteristics of the Search Regions

First, let's look into the problem of how to calculate the mean vectors $\hat{\mathbf{z}}_{j_1}$, $\hat{\mathbf{z}}_{j_2|j_1}$, $\ldots$, $\hat{\mathbf{z}}_{j_n|<j_1 j_2 \cdots j_{n-1}>}$ and covariance matrixes $\Sigma_{\mathbf{z}_{j_1}}$, $\Sigma_{\mathbf{z}_{j_2}|\mathbf{z}_{j_1}}$, $\ldots$, $\Sigma_{\mathbf{z}_{j_n}|\mathbf{z}_{<j_1 \cdots j_{n-1}>}}$ for any given search order $\langle j_1, j_2, \ldots, j_n \rangle$.

We define $\mathbf{x}_m$ to be the vector which stacks the object state $\mathbf{x}$ and candidate measurements $\mathbf{z}_J = (\mathbf{z}_{j_1}, \mathbf{z}_{j_2}, \ldots, \mathbf{z}_{j_n})^\top$. Note that $\mathbf{z}_J$ is a rearranged version of $\mathbf{z}_T$. Given the knowledge of the state $\mathbf{x}$ (i.e. mean $\hat{\mathbf{x}}$ and covariance $\Sigma_{\mathbf{xx}}$), our knowledge of the stacked vector $\mathbf{x}_m$ can be described by its mean vector and full covariance:

$$\hat{\mathbf{x}}_m = \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}}_{j_1} \\ \hat{\mathbf{z}}_{j_2} \\ \vdots \\ \hat{\mathbf{z}}_{j_n} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}} \\ \mathbf{h}_{j_1}(\hat{\mathbf{x}}) \\ \mathbf{h}_{j_2}(\hat{\mathbf{x}}) \\ \vdots \\ \mathbf{h}_{j_n}(\hat{\mathbf{x}}) \end{pmatrix} \quad (6)$$

$$\Sigma_{\mathbf{x}_m\mathbf{x}_m} = \left[\begin{array}{c|c} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{x}\mathbf{z}_J} \\ \hline \Sigma_{\mathbf{z}_J\mathbf{x}} & \Sigma_{\mathbf{z}_J} \end{array}\right] = \left[\begin{array}{c|ccc} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{x}\mathbf{z}_{j_1}} & \cdots & \Sigma_{\mathbf{x}\mathbf{z}_{j_n}} \\ \hline \Sigma_{\mathbf{z}_{j_1}\mathbf{x}} & \Sigma_{\mathbf{z}_{j_1}} & \cdots & \Sigma_{\mathbf{z}_{j_1}\mathbf{z}_{j_n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\mathbf{z}_{j_n}\mathbf{x}} & \Sigma_{\mathbf{z}_{j_n}\mathbf{z}_{j_1}} & \cdots & \Sigma_{\mathbf{z}_{j_n}} \end{array}\right] =$$

$$\left[\begin{array}{c|ccc} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xx}}\frac{\partial \mathbf{h}_{j_1}}{\partial \mathbf{x}}^\top & \cdots & \Sigma_{\mathbf{xx}}\frac{\partial \mathbf{h}_{j_n}}{\partial \mathbf{x}}^\top \\ \hline \frac{\partial \mathbf{h}_{j_1}}{\partial \mathbf{x}}\Sigma_{\mathbf{xx}} & \frac{\partial \mathbf{h}_{j_1}}{\partial \mathbf{x}}\Sigma_{\mathbf{xx}}\frac{\partial \mathbf{h}_{j_1}}{\partial \mathbf{x}}^\top + \mathbf{R}_{j_1} & \cdots & \frac{\partial \mathbf{h}_{j_1}}{\partial \mathbf{x}}\Sigma_{\mathbf{xx}}\frac{\partial \mathbf{h}_{j_n}}{\partial \mathbf{x}}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{h}_{j_n}}{\partial \mathbf{x}}\Sigma_{\mathbf{xx}} & \frac{\partial \mathbf{h}_{j_n}}{\partial \mathbf{x}}\Sigma_{\mathbf{xx}}\frac{\partial \mathbf{h}_{j_1}}{\partial \mathbf{x}}^\top & \cdots & \frac{\partial \mathbf{h}_{j_n}}{\partial \mathbf{x}}\Sigma_{\mathbf{xx}}\frac{\partial \mathbf{h}_{j_n}}{\partial \mathbf{x}}^\top + \mathbf{R}_{j_n} \end{array}\right] \quad (7)$$

where $\hat{\mathbf{z}}_{j_i}$ denotes the *best guess* or *prediction* of the observations, $\mathbf{h}_{j_i}(\cdot)$ is the observation function, and the lower-right block of $\Sigma_{\mathbf{x}_m\mathbf{x}_m}$ (i.e. the sub-matrix $\Sigma_{\mathbf{z}_J}$) describes the uncertainty in this prediction, which is also known as the *innovation covariance matrix* in Kalman filter tracking.

From (6) and (7), we can directly get $\hat{\mathbf{z}}_{j_1}$ and $\Sigma_{\mathbf{z}_{j_1}}$. In order to obtain $\hat{\mathbf{z}}_{j_2|j_1}$ and $\Sigma_{\mathbf{z}_{j_2}|\mathbf{z}_{j_1}}$ we take advantage of the general formula for conditioning one partition of a state vector and covariance with respect to another, as presented very clearly by Eustice *et al.* [13]. First, we partition the

stacked measurements vector $\mathbf{z}_J$ as $\mathbf{z}_J = \left(\mathbf{z}_{j_1}, \mathbf{z}_{<j_2\cdots j_n>}\right)^\top$, then if we learn the exact values of all elements of $\mathbf{z}_{j_1}$ (i.e. a match to this feature is found), the state vector and covariance of $\mathbf{z}_{<j_2\cdots j_n>}$ can be updated as:

$$\hat{\mathbf{z}}_{<j_2\cdots j_n>|j_1} = \hat{\mathbf{z}}_{<j_2\cdots j_n>} + \Sigma_{\mathbf{z}_{<j_2\cdots j_n>}\mathbf{z}_{j_1}} \Sigma^{-1}_{\mathbf{z}_{j_1}} \left(\mathbf{z}_{j_1} - \hat{\mathbf{z}}_{j_1}\right) \tag{8}$$

$$\Sigma_{\mathbf{z}_{<j_2\cdots j_n>}|\mathbf{z}_{j_1}} = \Sigma_{\mathbf{z}_{<j_2\cdots j_n>}} - \Sigma_{\mathbf{z}_{<j_2\cdots j_n>}\mathbf{z}_{j_1}} \Sigma^{-1}_{\mathbf{z}_{j_1}} \Sigma_{\mathbf{z}_{j_1}\mathbf{z}_{<j_2\cdots j_n>}} \tag{9}$$

So, $\hat{\mathbf{z}}_{j_2|j_1}$ can be directly acquired by taking the top two elements from $\hat{\mathbf{z}}_{<j_2\cdots j_n>|j_1}$, and $\Sigma_{\mathbf{z}_{j_2}|\mathbf{z}_{j_1}}$ can be obtained by taking the top-left 2×2 block from $\Sigma_{\mathbf{z}_{<j_2\cdots j_n>}|\mathbf{z}_{j_1}}$, and the lower-right $2(n-1)\times 2(n-1)$ block of $\Sigma_{\mathbf{z}_{<j_2\cdots j_n>}|\mathbf{z}_{j_1}}$ is denoted as $\Sigma_{\mathbf{z}_{<j_3\cdots j_n>}|\mathbf{z}_{j_1}}$. The update for the predicted state of the subsequent features can be obtained similarly by iteratively partitioning the stacked measurements vector as $\mathbf{z}_{<j_2\cdots j_n>} = \left(\mathbf{z}_{j_2}, \mathbf{z}_{<j_3\cdots j_n>}\right)^\top$, $\mathbf{z}_{<j_3\cdots j_n>} = \left(\mathbf{z}_{j_3}, \mathbf{z}_{<j_4\cdots j_n>}\right)^\top$, and so on.

Then, based on mathematical induction, we can prove the following theorem (the details are omitted due to space limit).

Theorem 1: No matter what kind of reasonable search order $\langle j_1, j_2, \ldots, j_n\rangle$ is chosen, the product of the areas of the elliptical search regions for all of the sequentially searched features will be constant, i.e.

$$\begin{aligned}&\pi N_\sigma^2 \sqrt{\left|\Sigma_{\mathbf{z}_{j_1}}\right|} \cdot \pi N_\sigma^2 \sqrt{\left|\Sigma_{\mathbf{z}_{j_2}|\mathbf{z}_{j_1}}\right|} \cdots \pi N_\sigma^2 \sqrt{\left|\Sigma_{\mathbf{z}_{j_n}|\mathbf{z}_{<j_1\cdots j_{n-1}>}}\right|} \\ &= S_{j_1} \cdot S_{j_2} \cdots S_{j_n} = \left(\pi N_\sigma^2\right)^n \sqrt{|\Sigma_{\mathbf{z}_T}|} = \text{const}.\end{aligned} \tag{10}$$

C. Minimize the objective function

Now we turn back to the problem of minimizing the objective function ψ presented in (5). The function ψ can be factorized into the following form:

$$\begin{aligned}\psi =& 1 + \left(\lambda_{j_1}S_{j_1} + \cdots + \lambda_{j_n}S_{j_n}\right) + \left(\lambda_{j_1}S_{j_1}\lambda_{j_2}S_{j_2} + \cdots\right. \\ &\left. + \lambda_{j_1}S_{j_1}\lambda_{j_n}S_{j_n} + \cdots + \lambda_{j_{n-1}}S_{j_{n-1}}\lambda_{j_n}S_{j_n}\right) + \cdots \\ &+ \prod_{i=1}^{n} \lambda_{j_i}S_{j_i}\end{aligned} \tag{11}$$

The first term in (11) is constant. And, since λ_{j_i}, $i = 1, \cdots, n$, are constants, according to Theorem 1 we have no difficulty to find that the last term $\prod_{i=1}^{n} \lambda_{j_i}S_{j_i} = \left(\pi N_\sigma^2\right)^n \sqrt{|\Sigma_{\mathbf{z}_T}|} \cdot \prod_{i=1}^{n} \lambda_{j_i}$, which is also constant.

Besides the first term and the last term, there are $n-1$ terms left in (11), which are denoted $\{\Omega_k\}$, $k = 1, \cdots, n-1$, for short. The kth term Ω_k is the summation of $C_n^k \triangleq \frac{n!}{k!(n-k)!}$ sub-terms, each of which is the product of k arbitrarily chosen non-reduplicate $\lambda_{j_i}S_{j_i}$, $i \in \{1, \cdots, n\}$, and $\forall i$, $i \in \{1, \cdots, n\}$, the total times that $\lambda_{j_i}S_{j_i}$ appears in these C_n^k sub-terms are the same.

Theorem 2: No matter what kind of search order $\langle j_1, j_2, \ldots, j_n\rangle$ is chosen, the theoretical minimum of the objective function ψ in (11) is:

$$\psi_{LB} = 1 + \sum_{k=1}^{n} \left\{ C_n^k \left(\pi N_\sigma^2\right)^k \left(\sqrt{|\Sigma_{\mathbf{z}_T}|}\right)^{\frac{k}{n}} \left(\prod_{i=1}^{n} \lambda_{j_i}\right)^{\frac{k}{n}} \right\}, \tag{12}$$

which can be reached if and only if the following condition is satisfied:

$$\lambda_{j_1}S_{j_1} = \cdots = \lambda_{j_n}S_{j_n} = \left(\prod_{i=1}^{n} \lambda_{j_i}\right)^{\frac{1}{n}} \left(\sqrt{|\Sigma_{\mathbf{z}_T}|}\right)^{\frac{1}{n}} \pi N_\sigma^2. \tag{13}$$

Proof: According to the analysis presented above, we know that minimizing ψ is equivalent to minimizing the summation of the $n-1$ terms $\{\Omega_k\}$.

$\forall k \in \{1, \cdots, n-1\}$, there are $C_n^k = \frac{n!}{k!(n-k)!}$ terms in Ω_k. Each term is of the form $\lambda_{j_{i_1}}S_{j_{i_1}}\lambda_{j_{i_2}}S_{j_{i_2}}\cdots\lambda_{j_{i_k}}S_{j_{i_k}}$, $i_1, i_2, \cdots, i_k \in \{1, \cdots, n\}$ and $i_1 \neq i_2 \neq \cdots \neq i_k$. And, $\forall i$, $i \in \{1, \cdots, n\}$, the total times that $\lambda_{j_i}S_{j_i}$ appears in these C_n^k terms are the same, being equal to C_{n-1}^{k-1}. So, the product of these C_n^k terms is:

$$\begin{aligned}M_k &= \prod_{i=1}^{n} \left(\lambda_{j_i}S_{j_i}\right)^{C_{n-1}^{k-1}} = \left(\prod_{i=1}^{n} \lambda_{j_i}S_{j_i}\right)^{C_{n-1}^{k-1}} \\ &= \left[\left(\pi N_\sigma^2\right)^n \sqrt{|\Sigma_{\mathbf{z}_J}|} \cdot \prod_{i=1}^{n} \lambda_{j_i}\right]^{C_{n-1}^{k-1}} = \text{const}\end{aligned} \tag{14}$$

Then, according to the arithmetic-geometric mean inequality, we know that:

$$\Omega_k \geq C_n^k \cdot \left(M_k\right)^{\frac{1}{C_n^k}} = \left(\pi N_\sigma^2\right)^k \times C_n^k \times \left(\sqrt{|\Sigma_{\mathbf{z}_T}|}\right)^{\frac{k}{n}} \left(\prod_{i=1}^{n} \lambda_{j_i}\right)^{\frac{k}{n}} \tag{15}$$

The equality is satisfied if and only if all of the C_n^k terms in Ω_k are equal, which is equivalent to the condition that all of the $\lambda_{j_i}S_{j_i}$, $i = 1, \cdots, n$, are equal, i.e. (13) holds.

Now, it is clear that the necessary and sufficient condition for all of the $n-1$ terms $\{\Omega_k\}$ to achieve their lower bounds are common, i.e. (13) holds. So, (13) is also the necessary and sufficient condition for ψ to achieve its lower bound, which is given by (12).

D. Algorithm for Choosing Suboptimal Search Order

Theorem 2 indicates that, theoretically, the optimal search order should be the one that the resulted $\lambda_{j_i}S_{j_i}$ (which will be termed *weighted search area* later on) for all of the sequentially examined features are equal. However, in practice, this condition is hard to be satisfied in general cases. Under the guidance of Theorem 2, we have the following consideration: intuitively, we should seek to minimize the deviation of the $\lambda_{j_i}S_{j_i}$, $i = 1, \cdots, n$, from the optimal value in (13). There are two different cases that have to be noted:

1) If, at the ith step, the weighted search areas of all of the unmeasured features are larger than the optimal value in (13), then the feature with the minimum weighted search area would be chosen at this step, because: on one hand, the weighted search area of this feature is the closest to the optimal value; on the other hand, the areas of the other features' predicted search regions at the next step will reduce after the selected feature is measured, which in turn makes the weighted search areas of those features get closer to the optimal value.

2) If, at the *i*th step, the weighted search areas of some of the unmeasured features are smaller than the optimal value, then the feature with the minimum weighted search area would be chosen at this step, because if the feature with minimum weighted search area is not selected at this step, its search region will shrink further at the next step, making its weighted search area get more far-away from the optimal value.

To sum up, we propose the following algorithm for choosing a sub-optimal search order.

Algorithm 1: The *i*th feature in the sub-optimal search order is chosen as

$$j_i = \arg \min_{\substack{k_i \in \{1,\ldots,n\} \\ k_i \notin \{j_1,\ldots,j_{i-1}\}}} \lambda_{k_i} \pi N_\sigma^2 \sqrt{\left|\Sigma_{\mathbf{z}_{k_i}|\mathbf{z}_{j_1},\ldots,\mathbf{z}_{j_{i-1}}}\right|}, \quad (16)$$

where $\Sigma_{\mathbf{z}_{k_i}|\mathbf{z}_{j_1},\ldots,\mathbf{z}_{j_{i-1}}}$ is the 2×2 diagonal block of $\Sigma_{\mathbf{z}_{<j_i \cdots j_n>}|z_{j_1} z_{j_2} \ldots z_{j_{i-1}}}$ which corresponds to feature $\mathbf{z}_{k_i}$.

III. PRACTICAL ALGORITHMS FOR MINIMUM-ERROR ACTIVE MATCHING

Two practical minimum-error active matching algorithms are presented. The first one is a one-trial sequential matching algorithm termed MED-SCNN (see Fig. 1), which can simply apply the NN rule to achieve satisfactory consensus matchings; while the second one is a recursive algorithm termed MED-JCBB (see Fig. 2), which actually combines the idea of JCBB [14] and minimum-error active matching, and can be expected to be more robust than MED-SCNN when handling some extremely complex situations.

Algorithm MED-SCNN($\hat{\mathbf{z}}_T$, $\Sigma_{\hat{\mathbf{z}}_T}$, im_t, N_σ, η)

Input: the predicted feature positions $\hat{\mathbf{z}}_T$ and the innovation covariance $\Sigma_{\hat{\mathbf{z}}_T}$, the current image im_t, the gating number N_σ and the threshold η for template matching response.
Output: matchings $\mathbf{z}_T$ for the features.
Steps:

1. initialize $\mathbf{z}_T$=[ ]; // Null set
2. **while** is_not_empty($\hat{\mathbf{z}}_T$)
3. **for** i = 1: length($\hat{\mathbf{z}}_T$)
4. $a_i = \lambda_i \sqrt{|\Sigma_{\hat{\mathbf{z}}_i}|}$; //$\Sigma_{\hat{\mathbf{z}}_i}$ ith 2×2 diagonal block of $\Sigma_{\hat{\mathbf{z}}_T}$
5. **end for**
6. $k = \arg\min_i a_i$;
7. Measure the kth feature in $\hat{\mathbf{z}}_T$. Search for all positions in the neighborhood of $\hat{\mathbf{z}}_k$ (an elliptical region centered around $\hat{\mathbf{z}}_k$ with a shape gated at N_σ of $\Sigma_{\hat{\mathbf{z}}_k}$) whose template matching response is higher than η. Suppose m points have been found, and denote the points set as $\left\{\mathbf{y}_k^{(j)} | j = 1, \cdots, m\right\}$.
8. **if** $m > 0$
9. **for** j = 1 : m
10. $d_j = (\mathbf{y}_k^{(j)} - \hat{\mathbf{z}}_k)^\top \Sigma_{\hat{\mathbf{z}}_k}^{-1} (\mathbf{y}_k^{(j)} - \hat{\mathbf{z}}_k)$;
11. **end for**
12. $q = \arg\min_j d_j$;
13. $\mathbf{z}_T = [\mathbf{z}_T, \mathbf{y}_k^{(q)}]$; // choose $\mathbf{y}_k^{(q)}$ as the matching;
14. delete $\hat{\mathbf{z}}_k$in $\hat{\mathbf{z}}_T$;
15. delete the corresponding rows and columns in $\Sigma_{\hat{\mathbf{z}}_T}$;
16. $\hat{\mathbf{z}}_T = \hat{\mathbf{z}}_T + \Sigma_{\hat{\mathbf{z}}_T \hat{\mathbf{z}}_k} \Sigma_{\hat{\mathbf{z}}_k}^{-1} \left(\mathbf{y}_k^{(q)} - \hat{\mathbf{z}}_k\right)$; // positions
17. $\Sigma_{\hat{\mathbf{z}}_T} = \Sigma_{\hat{\mathbf{z}}_T} - \Sigma_{\hat{\mathbf{z}}_T \hat{\mathbf{z}}_k} \Sigma_{\hat{\mathbf{z}}_k}^{-1} \Sigma_{\hat{\mathbf{z}}_T \hat{\mathbf{z}}_k}^\top$; // innovation cov.
18. **else**
19. $\mathbf{z}_T = [\mathbf{z}_T, \Phi]$; // feature $\hat{\mathbf{z}}_k$ not matched;
20. delete $\hat{\mathbf{z}}_k$in $\hat{\mathbf{z}}_T$;
21. delete the corresponding rows and columns in $\Sigma_{\hat{\mathbf{z}}_T}$;
22. **end if**
23. **end while**
24. **return** $\mathbf{z}_T$

Fig. 1. MED-SCNN matching algorithm.

IV. EXPERIMENT RESULTS AND DISCUSSIONS

In this section, the effectiveness of our minimum-error active matching algorithms are experimentally validated through both simulation and real-world implementation. The performances of our MED-SCNN and MED-JCBB algorithms are compared with those of ICNN algorithm [5], [14], original active matching algorithm [7], [8] (will be referred as MIactive for short throughout the rest of this paper), and the original SCNN and JCBB algorithms [14].

A. Simulation Study

For simulation study, we make up a setting similar to that of [7]. We consider a specific visual tracking problem, where an object is assumed to move and rotate in a plane which is fronto-parallel to a single observing camera. We parameterize the location of the center of the object in image coordinates $\mathbf{r} = (u, v)^\top$ relative to the bottom-left image corner, and its orientation with the angle φ in radians, to give state vector $\mathbf{x} = (u, v, \varphi)^\top$. The locations of known measureable point features are defined in the object coordinate frame O as $\mathbf{f}_i^O = \left(f_{iu}^O, f_{iv}^O\right)^\top$. Then, the location of the feature in the image can be calculated by considering the transitional and rotational transformation. Assume the covariance of measurement noise as $\mathbf{R}_i = \text{diag}\left(\sigma_m^2, \sigma_m^2\right)$, representing independent uncertainty in horizontal and vertical feature location measurements with $\sigma_m = 1$ pixel. We set up a snapshot in tracking with object state and covariance:

$$\hat{\mathbf{x}} = \begin{pmatrix} 320.0 \\ 260.0 \\ 0.3 \end{pmatrix}, \Sigma_{\mathbf{xx}} = \begin{bmatrix} 7.0 & 0.0 & 0.0 \\ 0.0 & 7.0 & 0.0 \\ 0.0 & 0.0 & 0.007 \end{bmatrix}. \quad (17)$$

The size of the object's projection on the image is set to be 320×240 pixels. At the beginning of each simulation, we randomly generate the real state of the object according to the predicted state and covariance given by (17). Then, the real positions of the features are randomly generated within the physical range of the object. We also randomly generate some spurious points around each of these features to simulate the spurious candidate matches.

We have made a large amount of simulations with the number of features varying from 6 to 20. For any setting of the number of features, we have made 3×10^4 runs of simulation. For each algorithm, we record the times that at least one of the features has been erroneously matched. The total times that the algorithms have obtained erroneous matching results are shown in Fig. 3. It is clear that the

Algorithm MED-JCBB($\hat{\mathbf{z}}_T$, $\Sigma_{\hat{\mathbf{z}}_T}$, im_t, N_σ, η)

Input: the predicted feature positions $\hat{\mathbf{z}}_T$ and the innovation covariance $\Sigma_{\hat{\mathbf{z}}_T}$, the current image im_t, the gating number N_σ and the threshold η for template matching response.
Output: the best jointly compatible matchings $\mathbf{z}_{best}$.
Steps:

1. initialize $\mathbf{z}_{best}$=[ ], $\mathbf{z}_T$=[ ]; // Null set
2. Sub_DJCBB($\hat{\mathbf{z}}_T$, $\Sigma_{\hat{\mathbf{z}}_T}$, im_t, N_σ, η, $\mathbf{z}_T$);
3. return $\mathbf{z}_{best}$

procedure Sub_DJCBB($\hat{\mathbf{z}}_T$, $\Sigma_{\hat{\mathbf{z}}_T}$, im_t, N_σ, η, $\mathbf{z}_T$)

1. **if** is_empty($\hat{\mathbf{z}}_T$)
2. **if** number_matched($\mathbf{z}_T$) > number_matched($\mathbf{z}_{best}$)
3. $\mathbf{z}_{best}$=$\mathbf{z}_T$;
4. **end if**
5. **else**
6. **for** i = 1: length($\hat{\mathbf{z}}_T$)
7. $a_i = \lambda_i \sqrt{|\Sigma_{\hat{\mathbf{z}}_i}|}$;
8. **end for**
9. $k = \arg\min_i a_i$;
10. Measure the kth feature in $\hat{\mathbf{z}}_T$. Search for all positions in the neighborhood of $\hat{\mathbf{z}}_k$ (an elliptical region centered around $\hat{\mathbf{z}}_k$with a shape gated at N_σ of $\Sigma_{\hat{\mathbf{z}}_k}$) whose template matching response is higher than η. Suppose m points have been found, and denote the points set as $\{\mathbf{y}_k^{(j)} | j = 1, \cdots, m\}$.
11. **if** m = 0 // no candidate matches have been found
12. **if** number_matched($\mathbf{z}_T$)+length($\hat{\mathbf{z}}_T$) − 1 > number_matched($\mathbf{z}_{best}$) // can do better
13. $\mathbf{z}_T = [\mathbf{z}_T, \Phi]$; // feature $\hat{\mathbf{z}}_k$ not matched;
14. delete $\hat{\mathbf{z}}_k$ and corresponding rows & columns in $\Sigma_{\hat{\mathbf{z}}_T}$;
15. Sub_DJCBB($\hat{\mathbf{z}}_T$, $\Sigma_{\hat{\mathbf{z}}_T}$, im_t, N_σ, η, $\mathbf{z}_T$);
16. **end if**
17. **else** // some candidate matches have been found
18. **for** j = 1 : m
19. $d_j = (\mathbf{y}_k^{(j)} - \hat{\mathbf{z}}_k)^\top \Sigma_{\hat{\mathbf{z}}_k}^{-1} (\mathbf{y}_k^{(j)} - \hat{\mathbf{z}}_k)$;
20. **end for**
21. Sort $\{\mathbf{y}_k^{(j)} | j = 1, \cdots, m\}$ according to d_j;
22. **for** j = 1 : m
23. $\mathbf{z}_T = [\mathbf{z}_T, \mathbf{y}_k^{(j)}]$; // choose $\mathbf{y}_k^{(j)}$ as the matching
24. delete $\hat{\mathbf{z}}_k$ and corresponding rows & columns in $\Sigma_{\hat{\mathbf{z}}_T}$;
25. $\hat{\mathbf{z}}_T = \hat{\mathbf{z}}_T + \Sigma_{\hat{\mathbf{z}}_T \hat{\mathbf{z}}_k} \Sigma_{\hat{\mathbf{z}}_k}^{-1} \left(\mathbf{y}_k^{(j)} - \hat{\mathbf{z}}_k \right)$; // positions
26. $\Sigma_{\hat{\mathbf{z}}_T} = \Sigma_{\hat{\mathbf{z}}_T} - \Sigma_{\hat{\mathbf{z}}_T \hat{\mathbf{z}}_k} \Sigma_{\hat{\mathbf{z}}_k}^{-1} \Sigma_{\hat{\mathbf{z}}_T \hat{\mathbf{z}}_k}^\top$; // covariance
27. Sub_DJCBB($\hat{\mathbf{z}}_T$, $\Sigma_{\hat{\mathbf{z}}_T}$, im_t, N_σ, η, $\mathbf{z}_T$);
28. **end for**
29. **end if**
30. **end if**

Fig. 2. MED-JCBB matching algorithm.

performances of both MED-SCNN and MED-JCBB are dominantly better than all of the other algorithms, and we can observe the following facts:

1) The rate for ICNN to obtain erroneous matching results is significantly larger than that of any other algorithm. And, as the total number of features increases, the error rate of ICNN also increases.
2) The error rates of SCNN and JCBB appear relatively stable and irrelevant to the total number of features.
3) The error rate of MIactive has a trend to decrease gradually, but is generally higher than that of JCBB.

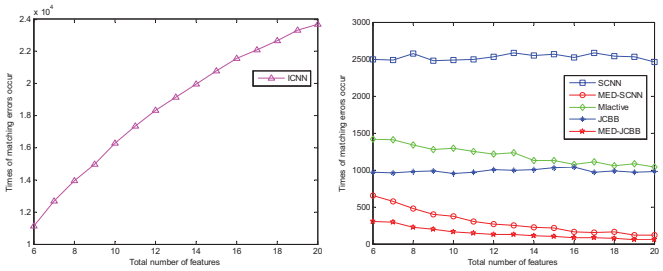


Fig. 3. Statistical results of different algorithms. Since the results of ICNN are far worse than any of the other algorithms, they are shown separately.

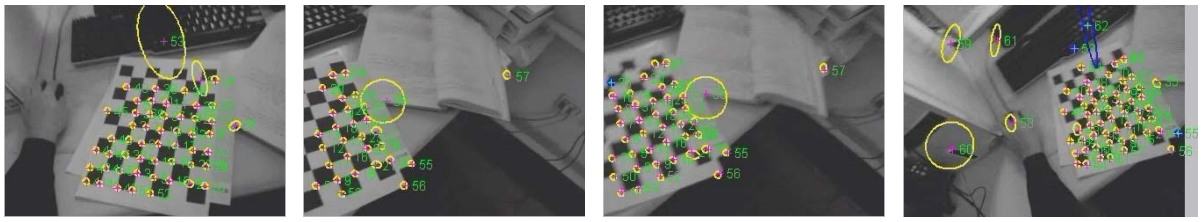
4) The error rates of both MED-SCNN and MED-JCBB are lower than that of JCBB, and have a trend to decrease as the number of features increases. In general, MED-JCBB can perform better than MED-SCNN, but as the feature number increases the difference between them becomes more and more insignificant. When the number of features is larger than 10, the error-rate of MED-SCNN is constantly lower than 1%.

B. Experiment Study

We set up a real-time monocular visual-SLAM system, using a hand-held low-cost USB web camera to move around in our lab to capture real-image sequences with a 320×240 resolution at 25 fps. The features are originally detected by Harris corner detector, and the main body of the SLAM algorithm uses an Extended Kalman Filter. The location of the 3D scene features are parametrized by following the *inverse depth parametrization* proposed by Civera *et al*. [9]. Whenever a new feature is initialized, its matching ambiguity parameter λ_i is roughly calculated by checking the number of similar points in its neighborhood, and in the mean time a relatively large (21×21 pixels in our experiments) image patch is extracted around this point feature to serve as a long-term landmark feature *template* (or *descriptor*). In order to adapt to significant scale change and view point change, we apply the method in [10] to manage the descriptors.

The testing sequence is a desk-top sequence where similar features are relatively dense. Due the limited space, only the detailed results of our MED-SCNN algorithm are shown in Fig. 4, while the statistical results of the other algorithms are summarized in Table I.

In Fig. 4, the yellow (light) and blue (dark) ellipses denote the image regions that have been examined by the algorithms. A yellow (light) ellipse means a match is successfully found for the corresponding feature and a red cross within the ellipse denote the matched position; while, a blue (dark) ellipse says that the corresponding feature is unsuccessfully matched, and a cyan cross within the ellipse represents the final predicted position of the feature. All features are numbered differently. It is clear that the performance of MED-SCNN is quite satisfactory, and during the whole sequence no erroneous matchings are observed, even when the camera is undergoing a jerk motion such as the case shown in Fig. 4(c). It is not difficult to discover that the features which



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

Fig. 4. Some sample results of MED-SCNN algorithm in desk-top sequence. (a) 37th frame, (b) 76th frame, (c) 78th frame, (d) 133rd frame.

TABLE I
STATISTICAL RESULTS FOR DESK-TOP SEQUENCE

	Frames error	Frames failed	Time (ms)
MED-SCNN	0	0	15.8
MED-JCBB	0	0	15.8
SCNN	62	62	38.5
JCBB	0	0	230.2
MIactive	68	68	51.8
ICNN	139	78	165.6

have been chosen to be measured first by MED-SCNN are always those features with sufficient distinctiveness, having no similar spurious matching candidates in its neighborhood that can lead to matching ambiguity and potential errors (see the features enclosed by the relatively big ellipses in Fig. 4).

The statistic results of the algorithms in desk-top sequence are presented in Table I. There are 140 frames in total in the desk-top sequence, and the correctness of the matching results are manually checked. The number of frames where at least one feature has been erroneously matched by the algorithms are listed in the column 'Frames error' in Table I. And, at any frame, if more than half of the features are erroneously matched or unmatched, we would count it as 'failed'. It is obvious that the performance of ICNN is the worst, in which matching errors occur throughout the whole sequence. For SCNN and MIactive algorithms, whenever matching errors occur, catastrophic failure would follow. For JCBB, though the final matchings obtained are satisfactory, its computational cost is too high. Both MED-SCNN and MED-JCBB are quite satisfactory, with the highest efficiency and lowest error-rate.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, an efficient minimum-error active matching scheme is proposed for solving the problems about enhancing the overall computational efficiency and seeking for the best global consensus matchings as an integrated whole. Two practical feature matching algorithms, i.e. MED-SCNN and MED-JCBB, have been presented. As verified by the statistical simulations and the real-world experiments, both MED-SCNN and MED-JCBB algorithms are quite efficient and effective, performing significantly better than the ICNN, SCNN, JCBB and MIactive algorithms, which are all popular methods used in the state-of-the-art works on visual SLAM.

An interesting phenomenon that deserves mention is that MED-SCNN has obtained identical results as MED-JCBB throughout the real-world experiments, which indicates that this relatively simpler algorithm may be good enough for real-world applications. However, as shown by the statistical simulation results, MED-JCBB can be more robust than MED-SCNN in handling extremely complex situations. Thus, we plan to examine the performances of these algorithms in more extensive real-world applications in future.

REFERENCES

[1] Yan Fei, W. Christmas, and J. Kittler, "Layered Data Association Using Graph-Theoretic Formulation with Application to Tennis Ball Tracking in Monocular Sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1814-1830, Oct., 2008.

[2] B. Micusik, T. Pajdla, "Structure from motion with wide circular field of view cameras," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1135-1149, July, 2006.

[3] D. Nistr, O. Naroditsky, and J. Bergen, "Visual Odometry for Ground Vehicle Applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3-20, Jan. 2006.

[4] S. Se, D. Lowe, J. Little, "Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks," *Int'l J. Robotics Research*, vol. 21, no. 8, pp. 735-758, 2002.

[5] A. Davison, I. Reid, N. Molton et al., "MonoSLAM: real-time single camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, June, 2007.

[6] M. Isard, A. Blake, "Contour tracking by stochastic propagation of conditional density," *Proc. European Conf. Computer Vision*, pp. 343-356, 1996.

[7] A. Davison, "Active search for real-time vision," *Proc. IEEE Int'l Conf. Computer Vision (ICCV'05)*, pp. 66-73, 2005.

[8] M. Chli, and A. Davison, "Active matching," *Proc. European Conf. Computer Vision (ECCV'08)*, pp. 72-85, 2008.

[9] J. Civera, A. Davison, and M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Tran. Robotics*, vol. 24, no. 5, pp. 932-945, 2008.

[10] D. Chekhlov, M. Pupilli, W. Mayol et al., "Robust real-time visual SLAM using scale prediction and exemplar based feature description," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'07)*, pp. 1-7, 2007.

[11] E. Eade, T. Drummond, "Monocular SLAM as a Graph of Coalesced Observations," *Proc. IEEE Int'l Conf. Computer Vision (ICCV'07)*, pp. 1-8, 2007.

[12] S. A. Holmes, G. Klein, and D. W. Murray, "An $O(N^2)$ Square Root Unscented Kalman Filter for Visual Simultaneous Localization and Mapping," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1251-1263, July, 2009.

[13] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly Sparse Delayed-State Filters for View-Based SLAM," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1100-1114, Dec. 2006.

[14] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robotics and Automation*, vol. 17, no. 6, pp. 890-897, 2001.