

# Automatic real-time SLAM relocalization based on a hierarchical bipartite graph model

DONG QiuLei\*, GU ZhaoPeng & HU ZhanYi

*National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

Received October 1, 2011; accepted January 6, 2012; published online September 25, 2012

**Abstract** The need to increase the robustness of a real-time monocular SLAM system raises the important problem of relocalization; namely, how to automatically recover a SLAM system after tracking failures. We address this problem by proposing a real-time relocalization algorithm based on a hierarchical bipartite graph model. When the SLAM system is lost, we use the latter model to find sufficient correspondences between the detected image and stored map features, thus achieving efficient, real-time relocalization. The model accounts for both temporal and spatial constraints. Experimental results on both synthetic and real data support the effectiveness of the proposed algorithm.

**Keywords** monocular SLAM, real-time relocalization, hierarchical bipartite graph model

**Citation** Dong Q L, Gu Z P, Hu Z Y. Automatic real-time SLAM relocalization based on a hierarchical bipartite graph model. *Sci China Inf Sci*, 2012, 55: 2841–2848, doi: 10.1007/s11432-012-4669-5

## 1 Introduction

Recently, there has been a growing interest in the problem of real-time localization and mapping using only a single camera, known as monocular SLAM in the field of computer vision [1–6]. Davison et al. [1] first demonstrated real-time monocular SLAM implementation, employing the extended Kalman filter (EKF) to estimate the camera pose and build a sparse map of point features. Eade et al. [2] used the FastSLAM algorithm [7] to perform recursive estimation with a Rao-Blackwellized particle filter. Clemente et al. [3] proposed a monocular SLAM system to build outdoor, closed-loop maps, based on the Hierarchical Map approach [8]. Klein et al. [4] presented two methods to improve the agility of a keyframe-based SLAM system, consisting of two separate threads, one for tracking, and the other for creating and expanding the map from a set of keyframes.

One of the main problems in most existing monocular SLAM implementations is a lack of robustness for use outside laboratory conditions. In particular, motion blur and occlusion make it difficult to track features reliably, and may cause tracking to fail and corrupt the estimated map.

There are several proposed solutions for automatic recovery of a SLAM system from tracking failures. For instance, Williams et al. [9] present a relocalization algorithm for an EKF-based monocular SLAM system, using correlation to find the candidate feature matches, and using RANSAC to recover the

\*Corresponding author (email: qldong@nlpr.ia.ac.cn)

camera pose from tracking failures. Unfortunately, these features are view-dependent, and it is difficult to use exhaustive correlation for obtaining correct feature correspondences. Furthermore, the update rate becomes slow when the number of the detected features increases. Based on the relocalization algorithm [9], an improved algorithm using a randomized tree classifier is proposed [10], enabling on-line feature learning without prior training. Although the latter system matches features more rapidly and accurately during relocalization, some computational costs have to be sacrificed during normal tracking. Chekhlov et al. [11] propose an appearance indexing method for SLAM relocalization using low order Haar wavelet coefficients. Eade et al. [12] extend the graph-based monocular system [13], and present a unified method for automatic relocalization from tracking failures and closing loops.

In this paper, we propose a hierarchical bipartite graph model (HBGM) for real-time SLAM relocalization. When the SLAM system fails to track features, we describe the similarities between the detected image and map features with a weighted bipartite graph in the first layer of the HBGM: this decomposes into a set of the detected features in the current frame and a set of the stored features in the map. We identify the weight of each edge, which describes similarities between the spatial-temporal neighborhoods of the two vertices of an edge, using an associated bipartite subgraph in the constructed second layer of the HBGM. We recover the pose of the lost camera in real time with a set of compatible matches by calculating the maximum weight maximum cardinality matching of the HBGM.

We organize the remainder of this paper as follows. Section 2 provides an overview of the EKF-based visual SLAM system. Section 3 presents the proposed relocalization algorithm. Section 4 reports experimental results in a real environment, and concluding remarks follow in Section 5.

## 2 EKF-based monocular SLAM implementation

Our proposed automatic relocalization algorithm is implemented as an extensive module to Davison's "SceneLib" EKF-based monocular SLAM system [1], and is applicable to other vision-based SLAM systems. Here, we give a brief review of the EKF-based monocular SLAM system and motivate the need for a relocalization module. Thereafter, we present details of the algorithm.

The goal of a monocular SLAM system is to simultaneously estimate unknown environmental structure and compute the camera pose based on observations in the captured video. Here, like [1,9], we define the camera's state vector  $\mathbf{x}_v$  by a 3D position vector  $\mathbf{r}$ , orientation quaternion  $\mathbf{q}$ , velocity vector  $\mathbf{v}$ , and angular velocity vector  $\boldsymbol{\omega}$ :

$$\mathbf{x}_v = (\mathbf{r}, \mathbf{q}, \mathbf{v}, \boldsymbol{\omega})^T. \quad (1)$$

Scene structure is defined by a map of  $N$  features  $(f_1, \dots, f_i, \dots, f_N)$ , assumed to be 3D points in the scene, i.e.

$$\mathbf{f}_i = (x_i, y_i, z_i)^T, \quad i = 1, 2, \dots, N, \quad (2)$$

where  $x_i, y_i, z_i$  denote the world frame coordinates of the feature  $\mathbf{f}_i$ . Hence, we represent the system state vector  $\mathbf{x}$  as a  $13 + 3N$  dimensional vector:

$$\mathbf{x} = (\mathbf{x}_v^T, \mathbf{f}_1^T, \dots, \mathbf{f}_i^T, \dots, \mathbf{f}_N^T)^T. \quad (3)$$

We assume that the system satisfies Gaussian statistics and Markov state evolution, and use the EKF to estimate the mean and covariance of the system state vector with a process model that defines the state time evolution as well as an observation model that defines the relationship between the system state vector and the observation vector in the current input frame. Here, 2D image points extracted by the "FAST" corner detector [14] define the observations, where each point corresponds to the perspective projection of a 3D point in the map. Fixed-sized image patches describe both observations and map features. Because of uncertainties in map features and camera pose, a corner point corresponding to a map feature in the current frame is only searched and recognized using template matching in the search region determined by the predicted position and innovation covariance of the map feature, but not searched in the whole frame, in order to speed up the SLAM system.

Note that it is mainly data association [1,15] (establishing correspondences between observations and map features) that affects the robustness of the EKF-based monocular SLAM system. When the experimental settings are strictly constrained, the uncertainties of the map features and the camera pose are low so that the search regions could be small, which can effectively constrain the spatial search for matches and reduce the computational cost as well as the likelihood of mismatch. However, for motion blur or occlusion in real environments, the uncertainties increase, leading to incorrect data associations and increased search regions, resulting in the ultimate failure of the SLAM system.

The above discussion forms the motivational basis for focusing on the development of a relocalization module for tracking failures of a monocular SLAM system. The module attempts to obtain successful data associations in subsequent frames after tracking failures, and relocalizes the moving camera based on the created map. Thereafter, the filter returns to the normal operation. The following section presents details of the relocalization algorithm.

### 3 Real-time relocalization based on the hierarchical bipartite graph model

We base our proposed relocalization algorithm on the HBGM, which associates detected features in the current frame with stored map features, when the pose information of the moving camera is unavailable, thus achieving rapid relocalization. There are two main steps in the relocalization algorithm: 1) The HBGM establishes potential compatible correspondences between the detected image and map features; 2) Based on these matches, the camera pose recovers and the SLAM system returns to the normal process.

#### 3.1 Hierarchical bipartite graph model for potential correspondences

When tracking failures arise, the map building is stopped and the “Fast” corner detector [14] is used to extract corner points from the subsequent frame. The key issue for relocalizing the camera’s pose is to find possible correspondences between extracted corners and stored map features. Because of the lack of constraints on the detected features, current SLAM relocalization matching techniques usually obtain many unreliable matches. In contrast, bipartite matching results in more accurate matches by constructing weights based on spatial-temporal constraints on these features. For this reason, we employ bipartite matching for constructing the HBGM for real-time relocalization.

We construct the HBGM to represent relationships between the observed corners and features in the map, and obtain possible correspondences by resolving the HBGM. Let us start by defining the concepts of a bipartite graph and a matching within the context of Graph Theory.

**Definition 1.** A bipartite graph is a set of graph vertices decomposed into two disjoint sets, such that no two graph vertices within the same set are adjacent.

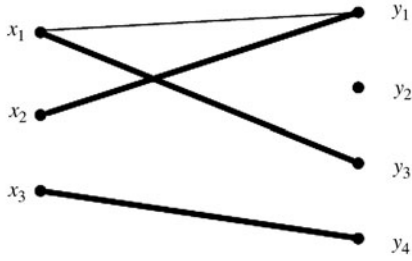
**Definition 2.** A matching on a graph  $G$  is a set of edges of  $G$ , such that no two of them share a common vertex.

Let us denote a set of descriptors of  $m$  extracted corners in the current frame by  $X = \{x_1, \dots, x_m\}$ , and a set of descriptors of  $n$  stored features in the map by  $Y = \{y_1, \dots, y_n\}$ . Then, we divide the whole process of constructing the HBGM and calculating the potential correspondences between  $X$  and  $Y$  into the following three steps:

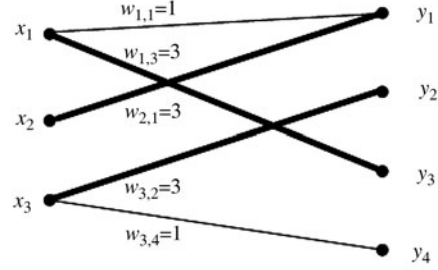
Step 1 Constructing a bipartite graph  $G$ : a bipartite graph  $G$  (the first layer in the HBGM) is constructed with the bipartition  $\{X, Y\}$ , and let  $e_{i,j} (i = 1, \dots, m, j = 1, \dots, n)$  denote a link between  $x_i$  and  $y_j$  in  $G$ .

Step 2 Computing the weights: let  $\mathbf{W}$  denote the weight matrix of  $G$ , where each element  $w_{i,j}$  refers to the weight of the edge  $e_{i,j} (i = 1, \dots, m, j = 1, \dots, n)$ :

$$w_{i,j} = w_{i,j}^p \times w_{i,j}^r, \quad (4)$$



**Figure 1** An example of maximum cardinality matching. The thick edges denote maximum cardinality matching.



**Figure 2** An example of maximum weight maximum cardinality matching. The thick edges denote maximum weight maximum cardinality matching.

where  $w_{i,j}^p$  is defined by using the cross correlation value  $\text{Corr}_{i,j}$  between  $x_i$  and  $y_j$  as

$$w_{i,j}^p = \begin{cases} \exp(\text{Corr}_{i,j}), & \text{Corr}_{i,j} > 0, \\ 0, & \text{Corr}_{i,j} \leq 0, \end{cases} \quad (5)$$

The definition of  $w_{i,j}^r$  is based on a bipartite subgraph  $G_{i,j}$  of  $G$ .

We construct the subgraph  $G_{i,j}$  (the second layer in the HBGM) as follows: For a pair of  $x_i$  and  $y_j$ , the bipartite subgraph  $G_{i,j}$ , with the bipartition  $\{X_{i,j}, Y_{i,j}\}$ , is extracted from  $G$ :  $X_{i,j}$  contains all the vertices in  $X$  except  $x_i$ , and  $Y_{i,j}$  contains such vertices each of which is not only an element in  $Y$ , but also one of  $y_j$ 's  $k$ -nearest neighbors in some frame before the tracking failure. An edge is added between a vertex  $x_u$  in  $X_{i,j}$  and a vertex  $y_v$  in  $Y_{i,j}$ , if  $y_v$  is one of  $x_u$ 's  $k$ -nearest neighbors among all vertices in  $Y_{i,j}$ . Then the maximum cardinality matching  $M_{\max}$  (Figure 1 shows an example of the maximum cardinality matching), which is a matching with the maximum number of the edges of  $G_{i,j}$ , is calculated by using Edmonds' maximum cardinality matching algorithm [16]. With the obtained  $M_{\max}$ ,  $w_{i,j}^r$  is defined as

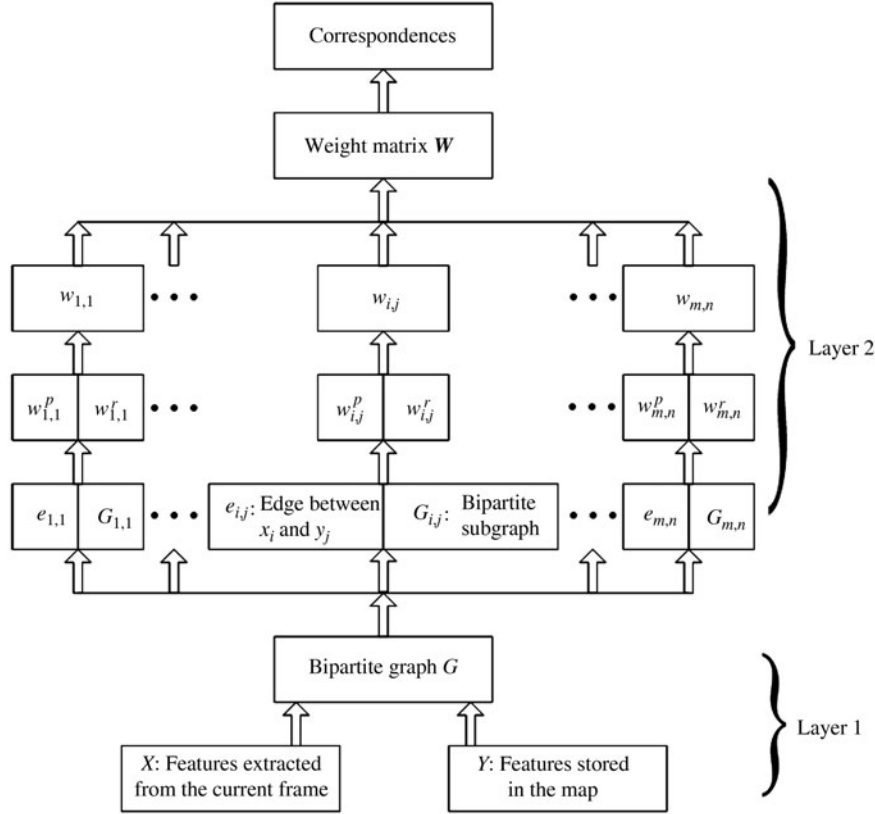
$$w_{i,j}^r = \exp\left(\frac{M_{\max}}{m-1}\right). \quad (6)$$

**Step 3** Computing the maximum weight maximum cardinality matching of  $G$ : for the weighted bipartite graph  $G$  with the weight matrix  $\mathbf{W}$ , we compute the maximum weight maximum cardinality matching of  $G$  (Figure 2 shows an example of the maximum weight maximum cardinality matching), which is a maximum cardinality matching with the greatest total weight, using the Kuhn-Munkres algorithm [17]. We consider the returned correspondences as potential correspondences between the detected corners and the map features, which we subsequently use to relocalize the system.

Figure 3 shows an HBGM flowchart. In view of Step 2, we note that the item  $w_{i,j}^p$  in (5) reflects the similarity between the corner  $x_i$  and the map feature  $y_j$ . The larger the values of  $w_{i,j}^p$ , the closer the similarities between  $x_i$  and  $y_j$ . However, since cross correlation lacks spatial-temporal constraints on features, it is usually less reliable for matching. Therefore, to enhance the reliability of the computed data association, we introduce the item  $w_{i,j}^r$  to the weight function (4), reflecting the similarity between spatial-temporal neighborhoods of  $x_i$  and  $y_j$ .

More specifically, in view of subgraph  $G_{i,j}$  for calculating  $w_{i,j}^r$ , the edges in  $G_{i,j}$  provide spatial constraints between the neighborhoods of  $x_i$  and  $y_j$ . Furthermore, since all the elements in  $X_{i,j}$  are in the same frame (i.e. in the current frame), each element in  $Y_{i,j}$  appears simultaneously with  $y_j$  in some frame. The number of the elements in  $X_{i,j}$  is  $m-1$ , the larger the ratio of the maximum cardinality matching  $M_{\max}$  of  $G_{i,j}$  to  $m-1$ , the larger the values of  $w_{i,j}^r$ , the closer the similarity between neighborhoods of  $x_i$  and  $y_j$ , the greater the likelihood of association between  $x_i$  and  $y_j$ , thus allowing for efficient exclusion of incompatible mismatches.

Although the weight matrix  $\mathbf{W}$  (in Step 2) reflects similarities between the detected corners and the map features to some extent, because of the lack of global spatial information,  $\mathbf{W}$  is not sufficiently reliable to associate a corner with a map feature based only on the weight of the linking edge in the con-



**Figure 3** Flowchart of the hierarchical bipartite graph model.

structured graph  $G$ . Therefore, within the context of image-map correspondences, we use the Kuhn-Munkres algorithm [17] (in Step 3) to calculate a maximum weight maximum cardinality matching of the graph  $G$ . These returned maximum weight and maximum cardinality correspondences generally yield better global matches.

Note that the proposed HBGM is independent of the features, i.e. the HBGM is also applicable to other visual SLAM systems where one needs to establish the correspondences of other types of features, rather than the point features in this paper.

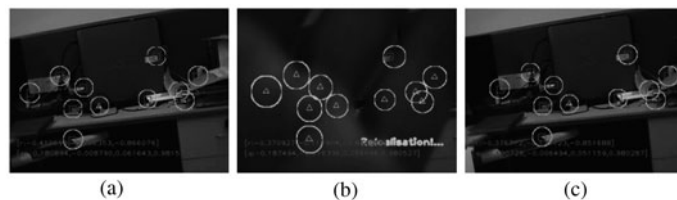
### 3.2 Camera pose recovery and SLAM system update

After obtaining potential matches, we use the method [18] to estimate the pose of the lost camera based on a set of four feature correspondences. Unlike [9–11], where RANSAC is used for seeking a consensus of potential matches, we sort potential matches in descending order of the corresponding weights. As an additional guard against mismatch, we discard potential matches with weights below a constant threshold,  $T_m$ . Hence, we use only matches with weights larger than  $T_m$  for evaluating the pose. We design an iterative scheme for evaluating the camera's pose using the fact that a match with larger weight is more reliable: first, we choose the four candidate matches with the highest relative weight sum to calculate the potential pose of the camera. Then, we use the latter to calculate the projection locations of other map features. If there is one or more computed locations satisfying the matches, we consider this pose as a good pose, thus terminating the iterative process.

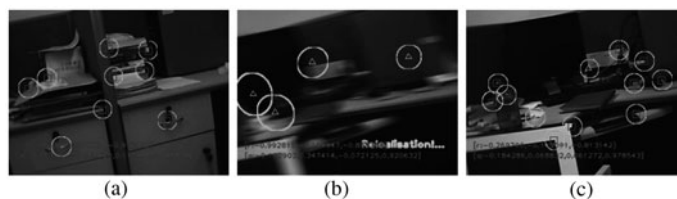
After finding a good pose, we reinitialize the EKF filter with the pose and a large artificial covariance, and use the corresponding four matches as the observations. Thereafter, the system returns to the normal loop.

## 4 Experiments

We apply the HBGM-based relocation algorithm to a multi-player game where each player's pose is



**Figure 4** An example of successful relocalization after tracking failure caused by occlusion.



**Figure 5** An example of successful relocalization after tracking failure caused by motion blur.

**Table 1** Successful relocalization rates on the three video sequences

	Number of tracking failures	Number of correct relocalization	Success rates
Sequence 1	12	11	91.7%
Sequence 2	23	21	91.3%
Sequence 3	16	15	93.8%

determined in real time to seamlessly fuse virtual objects with the real environment. We evaluate the reliability of the proposed algorithm on three captured video sequences of 5 minutes, 5 minutes and 6 minutes in an indoor environment, including tracking failures caused by motion blur or occlusion. The algorithm is implemented by running the EKF-based SLAM system on a Core 2 Duo 2.53 GHz processor. Prior to the experiment, we calibrated a handheld camera with a narrow-angled lens. The frame rate is 30 frames per second, and the captured image resolution is  $320 \times 240$  pixels. We use an image patch of size  $11 \times 11$  pixels as the descriptor of a feature point.

Figures 4 and 5 show examples of successful relocalization after tracking failures caused by occlusion and motion blur, respectively. Table 1 lists successful relocalization rates for the three video sequences. One of the reasons for not achieving correct relocalization in the subsequent frames after tracking failures could be because of insufficient detected features in these frames. Note that even for these unsuccessful cases, the algorithm is still able to successfully relocalize the lost camera when it moves from the current pose to an appropriate pose capturing sufficient map features.

Table 2 shows frame processing times during recovery of the SLAM system during the first 7000 frames of Sequence 1. It can be seen that when the size of the built map is up to 113 features, real-time relocalization can still be achieved.

Table 3 displays the time for a typical relocalization with 41 features in the map: here, the HBGM returns 20 matches, with only 9 of them larger than  $T_m$  selected as candidate matches. The algorithm chooses four matches with the largest weight sum for evaluating the camera pose [18], and two of the computed projections of the other map features satisfy the corresponding matches. It takes 18 ms to produce the correct pose. Note that, even though there are outliers among the four matches, since the number of the candidate matches is small, only little computational cost is increased for pose evaluation by using four correspondences, which does not affect the real-time performance of the proposed algorithm.

Next, we do an off-line comparison of the effectiveness of our algorithm compared with the RANSAC-based relocalization algorithm [9] using a synthetic video sequence with 50 map features. Table 4 shows relocalization results for ten randomly generated synthetic test images. The proposed algorithm achieves successful relocalization for all the ten images, whereas the RANSAC-based relocalization algorithm fails twice.

For testing the effectiveness of the algorithm for large numbers of map features, we construct a synthetic video sequence with 978 map features. Table 5 shows relocalization results for ten randomly generated synthetic test images. We see that the algorithm also works for the map with 978 features.

**Table 2** Processing times of five frames during recovery of the SLAM system

Frame number	1194	1509	2028	3397	6875
Number of the map features	19	23	34	41	113
Processing time (ms)	17	17	18	18	31

**Table 3** Time for a typical relocalization

Performance	Times(ms)
Corner detection	1
Hierarchical bipartite graph	16
Pose recovery	1
Total	18

**Table 4** Relocalization comparison on the synthetic video sequence with 50 map features

Processing time(ms)										
Relocalization number	1	2	3	4	5	6	7	8	9	10
RANSAC-based relocalization [9]	41	Failed	44	17	46	41	Failed	38	31	65
HBGM	21	22	19	22	18	19	18	20	21	20

**Table 5** Relocalization times on the synthetic video sequence with 978 map features

Relocalization number	1	2	3	4	5	6	7	8	9	10
Number of the map features	978	978	978	978	978	978	978	978	978	978
Processing time(ms)	28	30	36	31	31	30	31	Failed	35	33

## 5 Conclusions

In this paper, we propose a new real-time relocalization algorithm for automatically recovering a monocular SLAM automatically from tracking failures. When the camera is lost, we construct an HBGM for representing the similarities between the detected features in the current frame and the stored map features. Thereafter, we recover the lost camera pose based on a set of four matches by resolving the HBGM. Experimental results show that the proposed algorithm reliably relocalizes a monocular SLAM system. The main advantage of our algorithm is that the HBGM is able to achieve global data association, thus significantly increasing robustness of the relocalization module.

Future studies will focus on using the HBGM for robust relocalization within large maps with more distinctive descriptors.

## Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 60905020, 60835003).

## References

- 1 Davison A, Reid I, Molton N, et al. MonoSLAM: real-time single camera slam. *IEEE Trans Pattern Anal*, 2007, 29: 1052–1067
- 2 Eade E, Drummond T. Scalable monocular slam. In: Fitzgibbon A, Fitzgibbon C, LeCun Y, eds. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. New York: IEEE Computer Society, 2006. 469–476
- 3 Clemente L, Davison A, Reid I, et al. Mapping large loops with a single hand-held camera. In: Burgard W, Brock O, Stachniss C, eds. *Proceedings of Robotics: Science and Systems Conference*. Atlanta: MIT Press, 2007
- 4 Klein G, Murray D. Improving the agility of keyframe based slam. In: Forsyth D, Torr P, Zisserman A, eds. *Proceedings of European Conference on Computer Vision*. Marseille: Springer, 2008. 802–815
- 5 Lemaire T, Berger C, Jung I, et al. Vision-based slam: stereo and monocular approaches. *Int J Comput Vision*, 2007, 74: 343–364

- 6 Marzorati D, Matteucci M, Migliore D, et al. Monocular slam with inverse scaling parametrization. In: Everingham M, Needham C, Fraile R, eds. *Proceedings of British Machine Vision Conference*. Leeds: British Machine Vision Association, 2008. 945–954
- 7 Montemerlo M, Thrun S. Fastslam 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Gottlob G, Walsh T, eds. *Proceedings of International Joint Conference on Artificial Intelligence*. Acapulco: Morgan Kaufmann, 2003. 1151–1156
- 8 Estrada C, Neira J, Tardos D. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Trans Robot*, 2005, 21: 588–596
- 9 Williams B, Smith P, Reid I. Automatic relocation for a single-camera simultaneous localisation and mapping system. In: Siciliano B, Christensen H, Kosuge K, et al., eds. *Proceedings of IEEE International Conference on Robotics and Automation*. Roma: IEEE Computer Society, 2007. 2784–2790
- 10 Williams B, Klein G, Reid I. Real-time slam relocation. In: Metaxas D, Vemuri B, Shum H, eds. *Proceedings of IEEE International Conference on Computer Vision*. Rio de Janeiro: IEEE Computer Society, 2007. 1–8
- 11 Chekhlov D, Mayol-Cuevas W, Calway A. Appearance based indexing for relocation in real-time visual slam. In: Everingham M, Needham C, Fraile R, eds. *Proceedings of British Machine Vision Conference*. Leeds: British Machine Vision Association, 2008. 363–372
- 12 Eade E, Drummond T. Unified loop closing and recovery for real time monocular slam. In: Everingham M, Needham C, Fraile R, eds. *Proceedings of British Machine Vision Conference*. Leeds: British Machine Vision Association, 2008
- 13 Eade E, Drummond T. Monocular slam as a graph of coalesced observations. In: Metaxas D, Vemuri B, Shum H, eds. *Proceedings of IEEE International Conference on Computer Vision*. Rio de Janeiro: IEEE Computer Society, 2007. 1–8
- 14 Rosten E, Drummond T. Machine learning for highspeed corner detection. In: *European Conference on Computer Vision*. Graz: Springer, 2006. 430–443
- 15 Cheng Z B, Zhang Y, Deng Z D. A stochastic policy search model for matching behavior. *Sci China Inf Sci*, 2011, 54: 1430–1443
- 16 Edmonds J. Paths, trees, and flowers. *Can J Math*, 1965, 17: 449–467
- 17 Munkres J. Algorithms for the assignment and transportation problems. *J Soc Ind Appl Math*, 1957, 5: 32–38
- 18 Moreno-Noguer F, Lepetit V, Fua P. Accurate noniterative  $O(n)$  solution to the pnp problem. In: *Proceedings of IEEE International Conference on Computer Vision*. Rio de Janeiro: IEEE Computer Society, 2007. 1–8

## Information for authors

*SCIENCE CHINA Information Sciences (Sci China Inf Sci)*, cosponsored by the Chinese Academy of Sciences and the National Natural Science Foundation of China, and published by Science China Press, is committed to publishing high-quality, original results of both basic and applied research in all areas of information sciences, including computer science and technology; systems science, control science and engineering (published in Issues with odd numbers); information and communication engineering; electronic science and technology (published in Issues with even numbers). *Sci China Inf Sci* is published monthly in both print and electronic forms. It is indexed by Academic OneFile, Astrophysics Data System (ADS), CSA, Cabells, Current Contents/Engineering, Computing and Technology, DBLP, Digital Mathematics Registry, Earthquake Engineering Abstracts, Engineered Materials Abstracts, Gale, Google, INSPEC, Journal Citation Reports/Science Edition, Mathematical Reviews, OCLC, ProQuest, SCOPUS, Science Citation Index Expanded, Summon by Serial Solutions, VINITI, Zentralblatt MATH.

Papers published in *Sci China Inf Sci* include:

**REVIEW** (20 printed pages on average) surveys representative results and important advances on well-identified topics, with analyses and insightful views on the states of the art and highlights on future research directions.

**RESEARCH PAPER** (no more than 15 printed pages) presents new and original results and significant developments in all areas of information sciences for broad readership.

**BRIEF REPORT** (no more than 4 printed pages) describes key ideas, methodologies, and results of latest developments in a timely manner.

Authors are recommended to use *Science China's* online submission services. To submit a manuscript, please go to [www.scichina.com](http://www.scichina.com), create an account to log in <http://mco3.manuscriptcentral.com/scis>, and follow the instructions there to upload text and image/table files.

Authors are encouraged to submit such accompanying materials as short statements on the research background and area/subareas and significance of the work, and brief introductions to the first and corresponding authors including their mailing addresses with post codes, telephone numbers, fax numbers, and e-mail addresses. Authors may also suggest several qualified experts (with full names, affiliations, phone numbers, fax numbers, and e-mail addresses) as referees, and/or request the exclusion of some specific individuals from potential referees.

All submissions will be reviewed by referees selected by the editorial board. The decision of acceptance or rejection of a manuscript is made by the editorial board based on the referees' reports. The entire review process may take 90 to 120 days, and the editorial office will inform the author of the decision as soon as the process is completed. If the editorial board fails to make a decision within 120 days, please contact the editorial office.

Authors should guarantee that their submitted manuscript has not been published before and has not been submitted elsewhere for print or electronic publication consideration. Submission of a manuscript is taken to imply that all the named authors are aware that they are listed as coauthors, and they have agreed on the submitted version of the paper. No change in the order of listed authors can be made without an agreement signed by all the authors.

Once a manuscript is accepted, the authors should send a copyright transfer form signed by all authors to Science China Press. Authors of one published paper will be presented one sample copy. If more sample copies or offprints are required, please contact the managing editor and pay the extra fee. The

full text opens free to domestic readers at [www.scichina.com](http://www.scichina.com), and is available to overseas readers at [www.springerlink.com](http://www.springerlink.com).

## Subscription information

**ISSN print edition:** 1674-733X

**ISSN electronic edition:** 1869-1919

Volume 55 (12 issues) will appear in 2012

### Subscription rates

For information on subscription rates please contact:

Customer Service

**China:** [sales@scichina.org](mailto:sales@scichina.org)

**North and South America:**

[journals-ny@springer.com](mailto:journals-ny@springer.com)

**Outside North and South America:**

[subscriptions@springer.com](mailto:subscriptions@springer.com)

### Orders and inquiries:

#### China

Science China Press

16 Donghuangchenggen North Street, Beijing 100717, China

Tel: +86 10 64015683

Fax: +86 10 64016350

Email: [informatics@scichina.org](mailto:informatics@scichina.org)

#### North and South America

Springer New York, Inc.

Journal Fulfillment, P.O. Box 2485

Secaucus, NJ 07096 USA

Tel: 1-800-SPRINGER or 1-201-348-4033

Fax: 1-201-348-4505

Email: [journals-ny@springer.com](mailto:journals-ny@springer.com)

#### Outside North and South America:

Springer Distribution Center

Customer Service Journals

Haberstr. 7, 69126 Heidelberg, Germany

Tel: +49-6221-345-0, Fax: +49-6221-345-4229

Email: [subscriptions@springer.com](mailto:subscriptions@springer.com)

**Cancellations** must be received by September 30 to take effect at the end of the same year.

**Changes of address:** Allow for six weeks for all changes to become effective. All communications should include both old and new addresses (with postal codes) and should be accompanied by a mailing label from a recent issue. According to § 4 Sect. 3 of the German Postal Services Data Protection Regulations, if a subscriber's address changes, the German Federal Post Office can inform the publisher of the new address even if the subscriber has not submitted a formal application for mail to be forwarded. Subscribers not in agreement with this procedure may send a written complaint to Customer Service Journals, Karin Tiks, within 14 days of publication of this issue.

**Microform editions** are available from: ProQuest. Further information available at <http://www.il.proquest.com/uni>

### Electronic edition

An electronic version is available at [springerlink.com](http://springerlink.com).

### Production

Science China Press

16 Donghuangchenggen North Street, Beijing 100717, China

Tel: +86 10 64015683 or +86 10 64034134

Fax: +86 10 64016350

Printed in the People's Republic of China

### Jointly published by

Science China Press and Springer



## SCIENCE CHINA Series | Chinese Science Bulletin

*SCIENCE CHINA Series* and the *Chinese Science Bulletin* are academic journals supervised by the Chinese Academy of Sciences, co-sponsored by the Chinese Academy of Sciences and the National Natural Science Foundation of China, jointly published by Science China Press and Springer. *SCIENCE CHINA Series* and the *Chinese Science Bulletin* have presented the finest examples of China's development in both natural sciences and technological research to the international scientific community. In order to fully express China's achievements in fundamental scientific and engineering research, *SCIENCE CHINA Series* is published in seven journals, i.e., Mathematics, Chemistry, Life Sciences, Earth Sciences, Technological Sciences, Information Sciences, and Physics (including Mechanics and Astronomy), with the *Chinese Science Bulletin* serving as a multidisciplinary journal.

- Peer-reviewed
- Indexed by SCI, CA, EI, etc.
- Online submission
- Easy access to the electronic version

**Honorary Editor General:** ZHOU Guangzhao (Zhou Guang Zhao) | **Editor General:** ZHU Zuoyan

Mathematics	Chemistry	Life Sciences	Earth Sciences
 <p>Monthly Editor-in-Chief YANG Le (YANG Lo)</p>	 <p>Monthly Editor-in-Chief LI LeMin</p>	 <p>Monthly Editor-in-Chief WANG DaCheng</p>	 <p>Monthly Editor-in-Chief SUN Shu</p>
Technological Sciences	Information Sciences	Physics, Mechanics & Astronomy	Chinese Science Bulletin
 <p>Monthly Editor-in-Chief YAN LuGuang</p>	 <p>Monthly Editor-in-Chief LI Wei</p>	 <p>Monthly Editor-in-Chief WANG DingSheng</p>	 <p>Published three times every month Editor-in-Chief XIA JianBai</p>

[www.scichina.com](http://www.scichina.com) | [www.springer.com/scp](http://www.springer.com/scp)

Sponsored by  
Chinese Academy of Sciences (CAS)  
National Natural Science Foundation of China (NSFC)

Published by  
Science China Press & Springer

ISSN 1674-733X



9 771674 733129

129