Computers and Mathematics with Applications 64 (2012) 1235–1241

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Monocular vision based 6D object localization for service robot's intelligent grasping

Yang Yang, Qi-Xin Cao *

Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Keywords:
Grasping
SIFT
Homography matrix

ABSTRACT

Intelligent grasping is still a hard problem for home service robots. There are two major issues in the intelligent grasping, i.e. the object recognition and the pose estimation. To grasp casually placed objects, the robot needs the object's full 6 degrees of freedom pose data. To deal with the challenges such as illumination changes, cluttered background, occlusion, etc., we propose a monocular vision based object recognition and 6D pose estimation method. The SIFT feature point matching and brute-force search algorithm is used to do a tentative object recognition. The object recognition result is then verified with the homography constraint. After passing the verification, the 6D pose estimation is obtained through the decomposition of the homography matrix and the result is refined using the Levenberg–Marquardt algorithm. We embed our pose estimation method in a tracking by detection framework to keep computing and refining the pose during the whole approaching procedure. To test our method, a robot arm of seven degrees of freedom was utilized for a group of grasping experiments. The experimental results showed that our approach successfully recognized and grasped a variety of household objects with decent accuracy.

1. Introduction

The home service robot is the hot spot of robotics, and the grasping of household objects is a crucial function for the home service robot. There are two preconditions for a successful grasping: object recognition and pose estimation. While grasping in a structured environment is quite mature, grasping in an unstructured environment is still a hard problem. The challenges come from illumination and viewpoint changes, cluttered environment, occlusion, etc. Li et al. [1] dealt with these challenges with QR Code. They pasted QR Code on each object and did object recognition by recognizing the QR Code. But apparently this method would cause some trouble since people need to paste QR Code on each target object. Popovic et al. [2] used a binocular vision system to extract objects' 3D contour, and generated the grasping points by analyzing the co-planarity of the 3D contour. But their method did not contain object recognition. For pose estimation, some researchers used the binocular vision system such as [2,3], but the binocular vision system is usually too expensive. Accompanying the popularity of common CMOS cameras, monocular vision based pose estimation attracts more and more attention. With the eye-in-hand composition, Speth et al. [4] converted the monocular vision to a wide-baseline stereo vision through obtaining the kinematics parameters of the robot arm. But this method had a strong requirement for the arm's movement accuracy. The popular structure from motion algorithm could recover the camera's ego-motion and compute the object's pose from multiple monocular vision images. But it needs a large database and the procedure is time consuming and hence is not applicable for the robot's grasping. Aiming at the robot's grasping, Collet et al. [5,6] used SIFT features proposed by

* Corresponding author.
E-mail addresses: yangyangcv@gmail.com (Y. Yang), qxcao@sjtu.edu.cn (Q.-X. Cao).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.03.067

Lowe [7] to recognize objects and to compute their 6D pose. In the training stage, take dozens of images of the target object from different viewpoints, extract SIFT features, match these features and build a sparse 3D model of the object with these feature point correspondences. In the online object recognition and pose estimation stage, take one single image, extract the keypoints, match them to the points stored in the training stage and estimate the object's current pose from these 2D–3D point correspondences. With this method they got quite accurate results, but they only did this looking and grasping once and did not treat it as a tracking procedure. Ekvall et al. [8] and Kragic et al. [9] proposed that for robot grasping, it is necessary to monitor the grasping process and to track the object during execution, and proposed an approach for object recognition and pose estimation based on color co-occurrence histograms and geometric models. They embedded the color co-occurrence histograms in a learning framework that facilitated a "winner–takes–all" strategy across different scales. The hypotheses generated in the recognition stage provided the basis to estimate the orientation of the object around the vertical axis. The geometric model of the object was used to estimate and to continuously track the complete 6D pose of the object. However, because it relied on the color co-occurrence histograms, its performance would degrade in cluttered environment especially when the background has similar color with the target object.

In this paper, we propose a monocular vision based object recognition and 6D pose estimation approach for the robot's grasping. The SIFT feature is used to deal with the challenges including cluttered background, viewpoint and illumination changes, occlusion, etc. Besides, since each SIFT feature point has a corresponding feature descriptor, it is easy to distinguish one feature point from the other. In the offline training phase, take a snapshot of the target object, extract the SIFT feature points and the corresponding feature descriptors, and setup a database. In the online recognition phase, firstly do a tentative object recognition with feature points matching and brute-force search, and then verify the recognition result with the homography constraint. The 6D pose is obtained by the decomposition of the homography matrix and the result is refined with the Levenberg–Marquardt algorithm. The main contribution of our work is that we illustrate the whole pipelines of the robot's intelligent grasping with monocular vision. Different from other "look and grasp" strategies such as [5,6], we embed the pose estimation in a tracking by detection framework to keep computing and refining the pose computation result while the robot's gripper approaches the target object. The rest of the paper is organized as follows. Sections 2 and 3 present the details of object recognition and pose estimation method respectively. The interest window based object tracking is introduced in Section 4. Section 5 is the intelligent grasping section. Experimental results are presented in Section 6. In Section 7 we conclude our work and give some discussions.

2. Object recognition

Various machine learning algorithms are used in object recognition such as [10–12], most of them are based on the extraction and matching of feature points. State-of-the-art feature point extraction methods include SIFT [7], SURF [13], Fern [14], to mention a few. Mikolajczyk and Schmid [15] gave a comprehensive comparison on feature points and local descriptors extracting methods and concluded that SIFT performs best. Through the extraction of extreme points in DoG (Difference of Gaussian) scale space, SIFT feature points have good invariability to scale and affine transforms. Furthermore, SIFT is insensitive to illumination changes. For an image of the size $500 * 500$, about 2000 stable SIFT feature points can be extracted [7]. Hence enough feature points could be extracted in spite of occlusions. Besides, through the analysis of the feature point's $16 * 16$ neighborhoods, a feature descriptor of the length 128 is assigned to each feature point, making the feature point more distinguishable. To speed up the matching, the KNN algorithm is used to search the two descriptors which are the closest to the current descriptor. Let the distance between the first descriptor and the current descriptor be d_{01}, and let the distance between the second descriptor and the current descriptor be d_{02}, then to reject the unreliable matches, it is required that $d_{01}/d_{02} < 0.8$. Based on the feature points matching, the brute-force search algorithm is used to recognize the object tentatively. That is, match the feature points from the current frame to those from each of the model in the offline database, and select the one with the most matches as the best candidate. Compared to the popular image search algorithms such as the vocabulary tree algorithm proposed by Nister and Stewenius [16], the brute-force search algorithm is relatively simple, but it is capable of finding the most similar image.

To get rid of the false positives in the tentative recognition result, the homography constraint is used. Let $p_1 = \begin{bmatrix} u_1 & v_1 & 1 \end{bmatrix}^T$ and $p_2 = \begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix}^T$ be a matching feature point pair from the current frame and the database respectively, they should satisfy

$$p_1 = H \cdot p_2 \tag{1}$$

where H is a 3 by 3 homography matrix. H contains 9 elements and among them 8 elements are independent. Since each matching pair can provide 2 constraints, four non-collinear matching pairs are needed to solve for H [17]. To deal with the mismatches, the RANSAC algorithm [18] is used in the computing for H. Firstly four matching pairs are selected randomly from the matching pair sets and H is computed by solving a linear equation system. Then this H and (1) are used to check how many matching pairs satisfy this H. The matching pairs which satisfy this H are called inliers. Repeat this procedure until the iteration limit is reached and the H which owns the most inliers is kept. If these H's inliers exceed the preset threshold, this match is treated as a correct match and the object is recognized. Fig. 1 shows the flow chart of the object recognition.

Fig. 2 is a snapshot from the object recognition process. The left image is the frame from the camera. The right image is the most similar image from the offline database. The yellow lines connect the matching feature points.

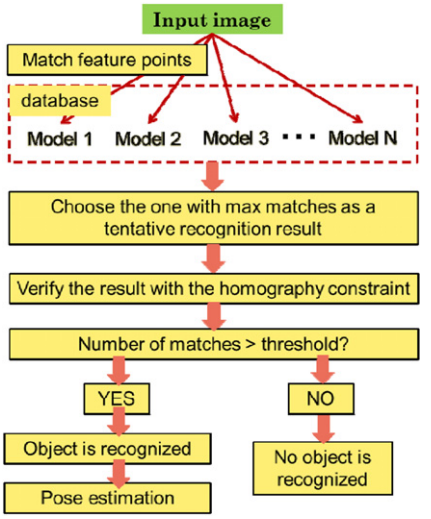

Fig. 1. Flow chart of object recognition.

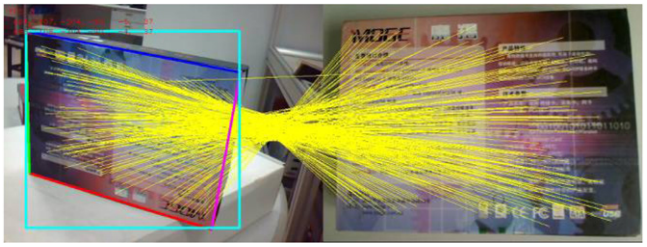

Fig. 2. Snapshot from the object recognition process.

3. 6D pose estimation

The 6D pose estimation is done through the decomposition of the homography matrix. Let $p = [u, v, 1]^T$ be a point in the image coordinate system (ICS) and $P = [x, y, z, 1]^T$ be a point in the world coordinate system (WCS). For objects with a planar surface, there always exist a WCS such that the object's planar surface overlays with the WCS's XY plane. Hence we have $P = [x, y, 0, 1]^T$. Denote the camera's intrinsic matrix as K, according to the camera's imaging model, we have

$$p = K \cdot [R \mid t] \cdot P \tag{2}$$

where R and t are the WCS's rotation and translation relative to the camera's coordinate system. Since $P = [x, y, 0, 1]^T$, we have

$$p = K \cdot \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{3}$$

Comparing (2) and (3) we get the homography matrix's expression $H = K \cdot \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$. The value of H is already obtained in the object recognition step. By multiplying the inverse of the camera's intrinsic matrix, we get $H' = K^{-1} \cdot H$. The first two columns of H' are the first two columns of the rotation matrix R. Since R is a unit orthogonal matrix, the cross product of R's first two columns is R's third column. The translation vector is the third column of H'.

To refine the result, the Levenberg–Marquardt algorithm is used to minimize the back-projection error. That is,

$$\min \left(\sum_{i=1}^{n} \left[\left(x_{1i} - \frac{h_{11}x_{2i} + h_{12}y_{2i} + h_{13}}{h_{31}x_{2i} + h_{32}y_{2i} + h_{33}} \right)^2 + \left(y_{1i} - \frac{h_{21}x_{2i} + h_{22}y_{2i} + h_{23}}{h_{31}x_{2i} + h_{32}y_{2i} + h_{33}} \right)^2 \right] \right) \tag{4}$$

where (x_{1i}, y_{1i}) and (x_{2i}, y_{2i}) are a matching pair from the input frame and the database's frame.

Until now the object's rotation and translation transform in the camera's coordinate system is obtained. For the robot's grasping, we need to go further to express the rotation as the roll–pitch–yaw angles. The transformation from the rotation

matrix to the roll–pitch–yaw angle is

$$\begin{aligned}\beta &= A\tan 2\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right)\\ \alpha &= A\tan 2\left(\frac{r_{21}}{\cos(\beta)}, \frac{r_{11}}{\cos(\beta)}\right)\\ \gamma &= A\tan 2\left(\frac{r_{32}}{\cos(\beta)}, \frac{r_{33}}{\cos(\beta)}\right)\end{aligned} \tag{5}$$

where r_{ij} is each element of the rotation matrix, and α, β, γ are the yaw, pitch and roll angles respectively. Through hand–eye calibration, the object's 6D pose is converted from the camera's coordinate system to the robot arm's coordinate system and then the arm is controlled to grasp the object by inverse-kinematics computation.

4. Interest window based object tracking

Before the robot touches and grasps the object, there is always an approaching procedure. Since the object is unmoved, the pose estimation accuracy could be improved by accumulating and filtering the pose computation results during the whole approaching procedure. With SIFT feature points used for object recognition and pose estimation, the major bottleneck comes from the large computation and the time consumed by SIFT feature points extraction and matching. To improve the real-time performance, we propose to embed the pose estimation in an interest window based tracking framework. It is conceivable that the time cost by SIFT feature point extraction has a close relationship with the image size. According to our test with Lowe's SIFT demo software (http://www.cs.ubc.ca/~lowe/keypoints/), the average time is 1.75 s for extracting SIFT feature point from an image of the size $640 * 480$ and 0.63 s for an image of the size $320 * 240$ pixels. For the object detection in the robot grasp situation, generally the target object only occupies a small part of the image. Hence it is not necessary to detect the whole image. On the contrary, we can concentrate on a small window which contains the object and update this window's property within a tracking framework using the information from the previous frame. To be specific, after recognizing the object, calculate its image position and size and setup an interest window containing the object. For the incoming frame, only extract feature points from this window and do pose estimation with these feature points. If the object is successfully detected, update the position and size of this window to continue the tracking procedure. The interest window is determined by the homography transformation between the current frame and the object frame stored in the database. The homography transformation is already calculated in the object recognition step and hence it can be used directly. To determine the object's original position in the image from the offline database, there is an extra operation. That is, label the target object with a quadrangle in the image to determine the object's four corners. Let $p_i = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T$ $(i = 1, 2, 3, 4)$ be the object's four corners in the database's image, and $q_j = \begin{bmatrix} u_j & v_j & 1 \end{bmatrix}^T$ $(j = 1, 2, 3, 4)$ be the object's four corners in the current image from the camera, then q_j is obtained with $q_j = H \cdot p_i$.

5. Intelligent grasping

The robot arm used in this research is the left arm of the robot SmartPal IV provided by Yaskawa Electrical Corporation. It has a humanoid configuration and is of seven degrees of freedom. The gripper has one degree of freedom which is a rotating thumb. The camera is fixed on the gripper. Fig. 3 shows the robot arm and its kinematics configuration. Please note that there are two cameras in Fig. 3 but only the one fixed on the gripper is used. As shown in the arm's kinematics configuration, the rotating axes of the last three joints intersect in the same point and they are used to control the gripper's roll, pitch and yaw respectively. The gripper's origin is set 200 mm away from the last three joints' intersection point along the fifth joint's rotating axis. The grasping motion can be treated as the alignment of the gripper's coordinate system to the object's coordinate system. This is fulfilled by inverse-kinematics computation with the object's 6D pose in the robot arm's coordinate system which is already obtained in Section 3. The Jacobian matrix is used to solve the inverse-kinematics problem. Since the robot arm has a redundant degree of freedom, the elbow angle is introduced as an additional constraint.

6. Experiments

Our approach is fulfilled with C++ and is applied to the robot arm mentioned in Section 5. The camera is Logitech C200. Ten different objects are placed on the desk at random and our approach is used to recognize the object and to compute its 6D pose. The result is then used to control the arm to grasp the object. In 100 trials, the correct recognition rate was 97% and the successful grasping rate was 94%. By using the GPU acceleration, an average speed of 13.5 frames per second is reached. Our CPU was an Intel Core i5-2400 at 3.1 GHz. The memory was 2 GB. The graphics card was AMD Radeon HD6350 and the GPU acceleration was realized with GLSL.

Fig. 4 shows four objects grasped in our experiments. As Fig. 1, the left image is the frame from the camera, and the right image is the recognized object from the database. As shown in Fig. 4, even though there are large viewpoint changes, our approach can still recognize the objects correctly.

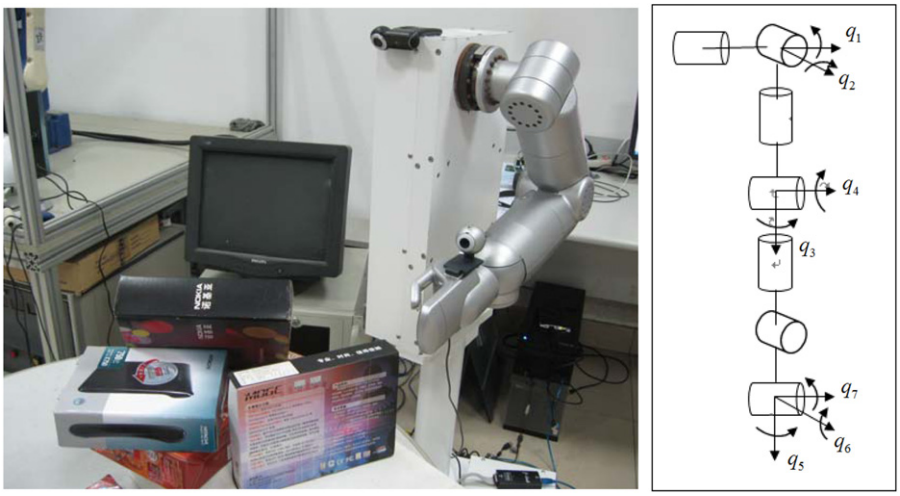

Fig. 3. The robot arm in our experiment and its kinematics configuration.

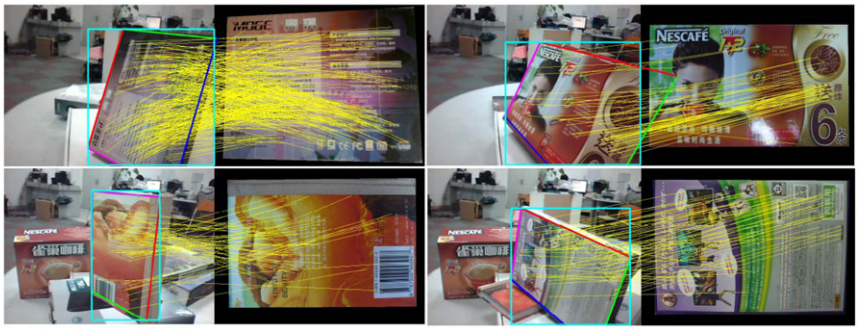

Fig. 4. The four object grasped in our experiment.

Table 1
Pose estimation errors.

Distance from object (cm)	X (cm)	Y (cm)	Z (cm)	Roll (°)	Pitch (°)	Yaw (°)
40	0.27	0.20	0.22	0.92	0.94	0.89
60	0.31	0.29	0.24	0.97	1.01	0.97
80	0.45	0.37	0.34	1.10	1.12	1.08
Average	0.34	0.29	0.27	1.00	1.02	0.98

Fig. 5 shows the four grasping states corresponding to the four objects in Fig. 4. As shown in Fig. 5, the gripper's position and orientation is adjusted according to the object's pose obtained with our pose estimation method.

To evaluate the accuracy of our pose estimation method quantitatively, the following experiment is conducted. The object is kept fixed and the gripper is moved to different positions. With our pose estimation method, the object's 6D pose is computed in each of the position. These results are denoted as $TobjTocam_k$. Since the object keeps fixed, we can get each of the camera's 6D pose relative to the camera's first position $Tcam_kTocam_1 = TobjTocam_1 \cdot inv(TobjTocam_k)$. On the other hand, the camera's relative pose can be obtained through the robot arm's kinematics computation $Tcam_kTocam_1 = Tgri_1Tocam_1 \cdot Tgri_kTogri_1 \cdot inv(Tgri_kTocam_k)$ where $Tgri_kTocam_k = Tgri_1Tocam_1 = TgriTocam$ is the transformation from the gripper's coordinate system to the camera's coordinate system and is obtained through the hand–eye calibration. Our robot arm's absolute accuracy is 0.3 mm and the repeatability accuracy is 0.1 mm and hence can be treated as a good ruler for the judgment of our method's pose estimation accuracy.

To evaluate our method's accuracy at different distances, the objects are placed at approximately 40 cm, 60 cm and 80 cm away from the camera respectively. For each distance, the gripper is moved casually to different viewpoints. We record the average translation and rotation errors between the result computed with our method and the result computed with the robot arm's kinematics parameters. Table 1 shows the pose estimation errors. From Table 1 we can see that the average translation error in each direction is less than 0.5 cm and the rotation error is about 1°. There is a trend that the errors increase with the distance between the object and the camera but they do not increase dramatically.

Fig. 5. The four grasping states in our experiment.

7. Conclusion

In this paper we proposed a monocular vision based object recognition and 6D pose estimation method for a home service robot's intelligent grasping. With SIFT feature points matching and brute-force search, the object was recognized from the offline database. The object's 6D pose was computed by the feature points matching and the decomposition of the homography matrix. To facilitate the information obtained during the approaching procedure, the pose estimation was embedded in a tracking by detection framework. An interest window based tracking method was used to improve the real-time performance. The same homography matrix was used in object recognition, pose estimation and interest window based tracking to improve the efficiency. Our method was tested using a robot arm with seven degrees of freedom. The grasping of ten household objects which were casually placed on a desk is fulfilled. Because we used the homography matrix, the current approach was only applicable to objects with a planar surface. In the future, we will explore other sensors, such as the RGB-D sensor to fulfill the grasping of arbitrary objects.

Acknowledgments

This research was supported in part by the Ministry of science and technology of the People's Republic of China under Grant 2011GB113005, and was partially funded by Yaskawa Electrical Corporation. We would like to express our grateful

thanks to Yaskawa Electrical Corporation for providing the robot arm and gripper of Smartpal IV. Special thanks go to Mr. Masaru Adachi, Dr. Rui Zhou and Dr. Xinwei Xu for all the kind help.

References

[1] G.D. Li, G.H. Tian, Y.H. Xue, Research on QR code-based visual servo handling of room service robot, J. Southeast Univ. 40 (2010) 30–36.
[2] P. Mila, K. Dirk, B. Leon, B. Emre, P. Nicolas, K. Danica, A. Tamim, K. Norbert, A strategy for grasping unknown objects based on co-planarity and color information, Robot. Auton. Syst. 58 (2010) 551–565.
[3] H. Vorobieva, M. Soury, P. Hède, C. Leroux, P. Morignot, Object recognition and ontology for manipulation with an assistant robot, in: Proceedings of the Eighth International Conference on Smart Homes and Health Telematics, 2010, pp. 178–185.
[4] J. Speth, A. Morales, P.J. Sanz, Vision-based grasp planning of 3D objects by extending 2D contour based algorithms, in: IEEE/Rsj International Conference on Robots and Intelligent Systems, 2008, pp. 2240–2245.
[5] A. Collet, D. Berenson, S.S. Srinivasa, D. Ferguson, Object recognition and full pose registration from a single image for robotic manipulation, in: IEEE International Conference on Robotics and Automation, 2009, pp. 3534–3541.
[6] A. Collet, M. Martinez, S.S. Srinivasa, The MOPED framework: object recognition and pose estimation for manipulation, Int. J. Robot. Res. 30 (2011) 1–23.
[7] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[8] S. Ekvall, D. Kragic, F. Hoffmann, Object recognition and pose estimation using color cooccurrence histograms and geometric modeling, Image Vis. Comput. 23 (2005) 943–955.
[9] D. Kragic, M. Bjorkman, H.I. Christensen, J. Eklundh, Vision for robotic object manipulation in domestic settings, Robot. Auton. Syst. 52 (2005) 85–100.
[10] Q.E. Wu, X.M. Pang, Z.Y. Han, Fuzzy automata system with application to target recognition based on image processing, Comput. Math. Appl. 61 (2011) 1267–1277.
[11] H. Han, Y.S. Ding, K.R. Hao, X. Liang, An evolutionary particle filter with the immune genetic algorithm for intelligent video target tracking, Comput. Math. Appl. 62 (2011) 2685–2695.
[12] N.K. Alham, M.Z. Li, Y. Liu, S. Hammoud, A MapReduce-based distributed SVM algorithm for automatic image annotation, Comput. Math. Appl. 62 (2011) 2801–2811.
[13] H. Bay, T. Tuytelaars, L.V. Gool, Speeded-up robust features (SURF), Comput. Vis. Image Underst. 110 (2008) 346–359.
[14] M. Özuysal, M. Calonder, V. Lepetit, P. Fua, Fast keypoint recognition using random ferns, IEEE Trans. Pattern Anal. 32 (2010) 448–461.
[15] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. 27 (2005) 1615–1630.
[16] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.
[17] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge Univ. Press, London, 2003.
[18] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.