Model Based Pose Estimation Using SURF

Peter Decker and Dietrich Paulus

Active Vision Group
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz, Germany
{decker,paulus}@uni-koblenz.de

Abstract. Estimation of a camera pose (position and orientation) from an image, given a 3d model of the world, is a topic of great interest in many current fields of research. When aiming for a model based pose estimation approach, several questions arise: What is the model? How do we acquire a model? How is the image linked to the model? How is a pose computed and verified using the latter information? In this paper we present a new approach towards model based pose estimation based solely on SURF features. We give a formal definition of our model, show how to build such a model from image data automatically, how to integrate two partial models, and how pose estimation for new images works.

1 Introduction

Computing the pose of a camera given an image and a model of the world is an important task in computer vision. There are many different approaches using all kind of different models and matching techinques. Most are feature based, some use features which provide a descriptor for easier matching. SURF[1] is popular because of its invariance properties and high distinctiveness of the descriptor, as well as its speed. We present an approach towards model based pose estimation based solely on SURF features.

The rest of the paper is organized as follows: In the next section we give an overview of related work. We define our model in section 3 and show how to generate such a model automatically from images in section 4. In section 5, we demonstrate how a camera pose is computed from the model and a query image. Section 6 describes an algorithm to integrate two models which partially overlap into a single model. Evaluation takes place in section 7. Section 8 concludes the paper.

2 Related Work

Zhang and Kosecka discuss pose estimation in urban environments [2]. They store a number of GPS localized images and extracted SIFT features. Then the images most similar to a query image are identified and possible motion models

R. Koch et al. (Eds.): ACCV 2010 Workshops, Part II, LNCS 6469, pp. 11–20, 2011.
© Springer-Verlag Berlin Heidelberg 2011

are computed. For a final pose triangulation the two best fitting views are taken into account. Schindler et. al. focus on large scale databases and present an approach which is able to handle over 100 million SIFT features using vocabulary trees [3]. More recently, Wu et. al. introduced a new method of matching so called VIP features, which greatly increased the number of correct matches from query images [4]. The system described by Snavely et al. [5] covers much more aspects than pose estimation of images alone. Not only do they show how to compute structure and camera position from a large number of unstructured, uncalibrated images, but they also cover means of how to visualize and navigate the result. Irschara et al. introduce the idea of synthetic views to handle images taken from new viewpoints [6].

In our approach, we reconstruct 3d data directly from the images using SURF, thus skipping the search for the best fitting image as for example in [2]. We establish a connection between the features from a query image and the model directly, thus enabling direct pose estimation.

3 Model Definition

A model $\mathcal{M}$ is defined as a tuple

$$\mathcal{M} = (\mathcal{P}, \mathcal{F}, \mathcal{S}, g, h) \ . \tag{1}$$

It consists of a set of *world points* $\mathcal{P}$, *frames* $\mathcal{F}$, *SURF features* $\mathcal{S}$ and the relations $g \subseteq \mathcal{P} \times \mathcal{S}$ as well as $h \subseteq \mathcal{S} \times \mathcal{F}$. A world point $\boldsymbol{p}^{\mathrm{w}}$ is a simple point in three dimensional Euclidian space: $\boldsymbol{p}^{\mathrm{w}} \in \mathbb{R}^3$. A frame $\boldsymbol{f}$ represents a three dimensional Euclidian transformation. It describes the position of a camera by the rotation and translation applied to a world point before projection to the image plane, thus $\boldsymbol{f} \in SE(3)$. A surf feature $\boldsymbol{s} = (x, y, \sigma, \theta, \boldsymbol{d})$ consists of its location (x, y) in the image, detection scale σ, orientation θ and a 64 dimensional descriptor $\boldsymbol{d}$, as described in [1]. The relation g holds information, which SURF feature $\boldsymbol{s}$ is connected to which world point $\boldsymbol{p}^{\mathrm{w}}$. h connects the surf features to the frames from which they originate. For easier notation, we define the set of features $\mathcal{S}_{\boldsymbol{f}_i}$ extracted from frame $\boldsymbol{f}_i \in \mathcal{F}$ as

$$\mathcal{S}_{\boldsymbol{f}_i} := \{\boldsymbol{s}_j \in \mathcal{S} \mid (\boldsymbol{s}_j, \boldsymbol{f}_i) \in h\} \ . \tag{2}$$

The set of all world points $\mathcal{P}_{\boldsymbol{f}_i}$ visible in frame $\boldsymbol{f}_i \in \mathcal{F}$ is defined using both relations g and h:

$$\mathcal{P}_{\boldsymbol{f}_i} := \{\boldsymbol{p}_j^{\mathrm{w}} \in \mathcal{P} \mid \exists \boldsymbol{s} \in \mathcal{S} : (\boldsymbol{p}_j^{\mathrm{w}}, \boldsymbol{s}) \in g \land \boldsymbol{s} \in \mathcal{S}_{\boldsymbol{f}_i}\} \ . \tag{3}$$

With these relations a number of queries to the model are possible, e.g.

- In which frames has world point $\boldsymbol{p}^{\mathrm{w}}$ been recognized?
- Which surf descriptors are connected to a given world point $\boldsymbol{p}^{\mathrm{w}}$?
- Is there already a world point associated with feature $\boldsymbol{s}$?

These are important during the model building process.

4 Automatic Model Generation from Images

Automatic model generation from images is split into two phases. The first phase initializes the model using stereo geometry, while the second phase iteratively expands and improves the model. We assume the images to be in an order in which the first two images have a sufficient overlap for stereo processing. All further images have to be taken roughly from a direction of any preceding image, so they show a detail of the world which has already been covered to some extent.

We assume a geometrically calibrated camera with known intrinsic parameters. All images are undistorted beforehand. This allows us to use image coordinates directly, which simplifies the structre from motion process.

4.1 Initialization

For the two initial frames, we extract SURF features and compute possible correspondences using nearest neighbour matching of the descriptors along with a distance ratio threshold as proposed by Lowe [7]. RANSAC [8] with an adaptive termination criterion [9] is used to estimate the best fitting epipolar geometry using Nistér's five-point algorithm [10]. We then extract the camera movement from the essential matrix to get the pose of the second camera [9]. Next, all inlier to the epipolar constraint are triangulated and tested for their reprojection error in both images (which can in fact differ). For triangulation, we used the sum of the squared reprojection errors as minimization criterion. We also found it neccessary to test if the triangulated point is in front of both cameras, since with very many extracted SURF features it eventually happens, that outlier correspondences are inlier to the epipolar geometry by chance but reconstruct a point behind the cameras. This is also known as the cheirality constraint [10]. Fig. 1 shows an initial stereo pair and the reconstructed world points.

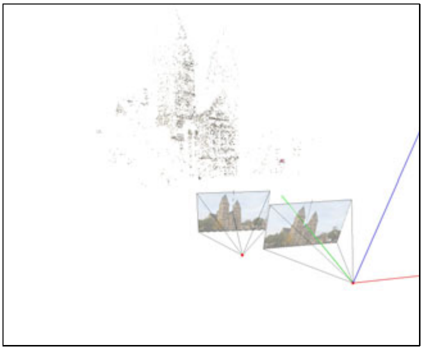
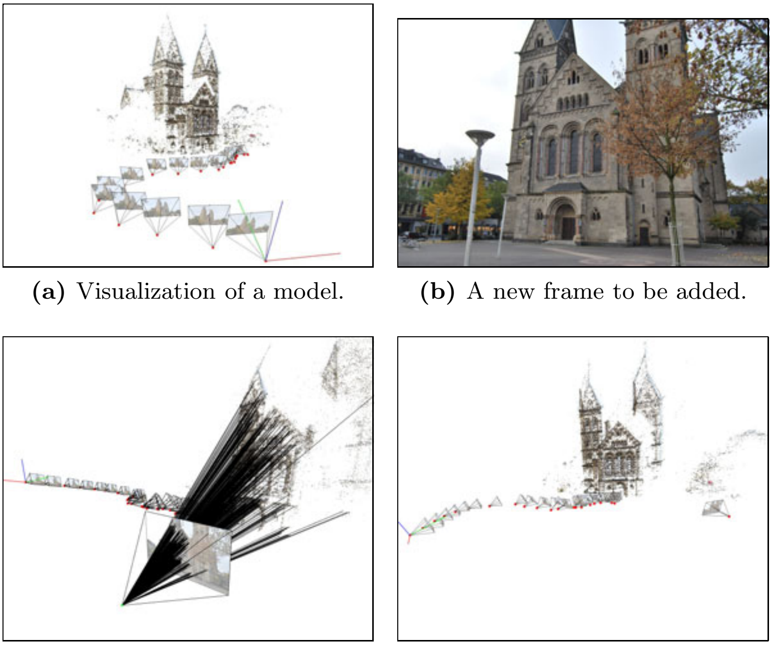


Fig. 1. Initialization of the model with a stero image pair

4.2 Incremental Expansion of the Model

The model is expanded frame by frame. First, the pose of the new frame $\boldsymbol{f}_{n+1}$ with respect to the models coordinate system is computed as will be shown in section 5. The new frame is added to $\mathcal{F}$, features extracted in the frame are added to $\mathcal{S}$. The inlier correspondences from existing world points to features are then added to g.



(a) Visualization of a model.

(b) A new frame to be added.

(c) Connections from world points to features in the new frame.

(d) The new frame was integrated at the computed position.

Fig. 2. Integration of a new frame. Features are extracted and correspondences to descriptors connected to world points of the model are computed. From these 2d/3d correspondences, the pose of the new frame is computed.

After adding a frame, we apply global bundle adjustment to all estimated world points and frames. We do so by optimizing these with respect to the reprojection error using the sparse bundle adjustment software [11] based on a Levenberg Marquard implementation [12], which both are publicly available. We parametrize our world points as $\boldsymbol{p}^{\mathrm{w}} \in \mathbb{R}^3$ and the camera location as the vector of three parameters representing translation and another three parameters representing the rotation axis. The length of the rotation axis defines the amount of rotation. Jacobians are computed using finite differences.

After global structure and motion optimization, new world points are created by computing the epipolar geometry between $\boldsymbol{f}_{n+1}$ and any other frame we want to consider. It makes sense to restrict the choice of these frames using constraints concerning their relative pose to each other, thus ommiting frames which are too far away from each other or have too different viewing directions.

Assuming $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ to be the rotation and translation of frame i from its projection matrix, an essential matrix between two frames $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ can be computed by

$$\boldsymbol{E}_{ij} = [\boldsymbol{R}_j(\boldsymbol{R}_i{}^{\mathrm{T}}(-\boldsymbol{t}_i) - \boldsymbol{R}_j{}^{\mathrm{T}}(-\boldsymbol{t}_j))]_{\times}\boldsymbol{R}_j\boldsymbol{R}_i{}^{\mathrm{T}} \ . \tag{4}$$

Inlier matches to the epipolar geometry between these frames can create new world points, if they also satisfy geometric and reprojection constraints as in section 4.1. These are added to $\mathcal{P}$, their connections are added to g. If an inlier contains a feature from the model which is already connected to a world point, no new world point is created. Instead, the new feature is connected to the existing world point by adding the relation to g, thus increasing the number of SURF features describing the particular world point.

5 Model Based Pose Estimation

We can compute a camera pose given the model and SURF features extracted from a new frame directly.

5.1 Matching

First, the descriptors from the new image are matched against all descriptors from the model which are connected to world points. For each feature we consider the two best matches. If they pass the distance ratio threshold as in [7] *or* they are connected to the same world point, we create a 2d/3d correspondence. Passing the distance ratio threshold means, that the first match is destinct enough from the second best. If both matches are connected to the same world point it means, that the feature from the new image describes this particular world point very well. There is still the possibility that several descriptors connected to the same world point are matched to different features in the query image. In that case we would create contradicting 2d/3d correspondences, where a world point is projected to different points in the image - we therefore consider these matches unstable and drop them all.

For faster nearest neighbour queries on the descriptors, we use the Fast Library for Approximate Nearest Neighbors FLANN [13], which is publicly available.

5.2 Pose Estimation

The resulting 2d/3d correspondences are passed to a RANSAC procedure encapsulating Fiore's linear pose estimation algorithm [14]. Since we work in image coordinates at this point, the result is an Euclidian transformation in $\mathbb{R}^3$, describing the pose of the new frame with respect to the model's world coordinate system.

6 Model Integration

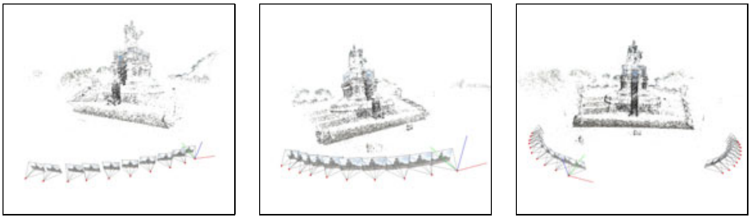
We developed a method to integrate a model $\mathcal{M}_n$ into an existing model $\mathcal{M}_e$. The models need to have some overlapping areas to do so. We do not need an initial guess of the position or correspondences. The result is a model $\mathcal{M}_{\hat{e}}$, which includes world points, frames, features, and connection information from both models.

First, we need to identify the set of features in each model which are connected to a world point: $\mathcal{S}_{\boldsymbol{p}_n^{\mathrm{w}}}$ and $\mathcal{S}_{\boldsymbol{p}_e^{\mathrm{w}}}$. These sets are matched against each other, again taking into account the distance ratio and discarding contradicting correspondences as in section 5.1. The result is a set of 3d/3d correspondences. These correspondences are passed to a RANSAC procedure encapsulating an absolute orientation estimation, which estimates the Euclidian transformation $\boldsymbol{T} \in SE(3)$ between the points of the models, as well as an overall scale $\sigma_{\boldsymbol{T}}$ between the models. We use the algorithm proposed by Umeyama [15] to do so. After the transformation $\boldsymbol{T}$ and scale $\sigma_{\boldsymbol{T}}$ between the models have been determined, all frames from $\mathcal{M}_n$ are transformed into the coordinate system of $\mathcal{M}_e$:

$$\begin{aligned} \mathcal{F}_{\hat{e'}} &= \mathcal{F}_n \boldsymbol{T}^{-1} \\ \mathcal{F}_{\hat{e}}^{\boldsymbol{t}} &= \mathcal{F}_{\hat{e'}}^{\boldsymbol{t}} \sigma_{\boldsymbol{T}} \\ \mathcal{F}_{\hat{e}}^{\boldsymbol{R}} &= \mathcal{F}_{\hat{e'}}^{\boldsymbol{R}} \ , \end{aligned} \tag{5}$$

where $\mathcal{F}^{\boldsymbol{t}}$ denotes the translation part of the frame's transformation, $\mathcal{F}^{\boldsymbol{R}}$ the rotation. Feature positions do not need to be transformed, they stay in the coordinate system of the according frame. h_e is joined with h_n:

$$h_{\hat{e}} = h_e \cup h_n \ . \tag{6}$$



(a) Model $\mathcal{M}_n$ (24486 world points), to be integrated into $\mathcal{M}_e$.

(b) Existing model $\mathcal{M}_e$ (33954 world points).

(c) Resulting model $\mathcal{M}_{\hat{e}}$ after the integration step. 664 world points were unified.

Fig. 3. Model integration. We built two models from frames $1-10$ ($\mathcal{M}_n$) and $21-30$ ($\mathcal{M}_e$) of the Deutsches Eck sequence and automatically integrated them to a single model ($\mathcal{M}_{\hat{e}}$) afterwards.

World points $\boldsymbol{p}^{\mathrm{w}}$ and the relation g need to be treated differently, depending on whether a world point was an inlier to the result of RANSAC or not. If it was an outlier, it is transformed and added to the model:

$$\boldsymbol{p}^{\mathrm{w}}_{\hat{e}} = \sigma_{\boldsymbol{T}}(\boldsymbol{T}\boldsymbol{p}^{\mathrm{w}}_{n}) \ . \tag{7}$$

If it was an inlier, it needs to be *unified* with its corresponding world point. The unification of two world points $\boldsymbol{p}^{\mathrm{w}}_{e}$ and $\boldsymbol{p}^{\mathrm{w}}_{n}$ creates a new world point $\boldsymbol{p}^{\mathrm{w}}_{\hat{e}}$ at the position of $\boldsymbol{p}^{\mathrm{w}}_{e}$. All connections in g_n and g_e from the specific world point are then inserted into $g_{\hat{e}}$, with $\boldsymbol{p}^{\mathrm{w}}_{\hat{e}}$ substituting $\boldsymbol{p}^{\mathrm{w}}_{e}$ or $\boldsymbol{p}^{\mathrm{w}}_{n}$.

Fig. 3 shows two models and the result of the integration. A run of the global bundle adjustment should always follow the model integration step to minimize possible errors resulting from unified world points.

7 Evaluation

In this section we evaluate our model generation and pose estimation approach. We first show some exemplary models, before we take a look at model accuracy and the robustness of the pose estimation algorithm. All images presented here were taken with a 10 megapixel consumer camera.

7.1 Exemplary Models

We considered three different image sequence:

Deutsches Eck: An image series of the Deutsches Eck monument. Images are taken in regular intervalls while circeling the monument, focusing the same point approximately.

ATM: An image series of the ATM building. Images are taken in regular intervalls while circeling the building, focusing the same point approximately.

Herz Jesu: An image series of the Herz Jesu church. Images are taken in irregular intervalls while approaching the church, focusing on different parts of the building.

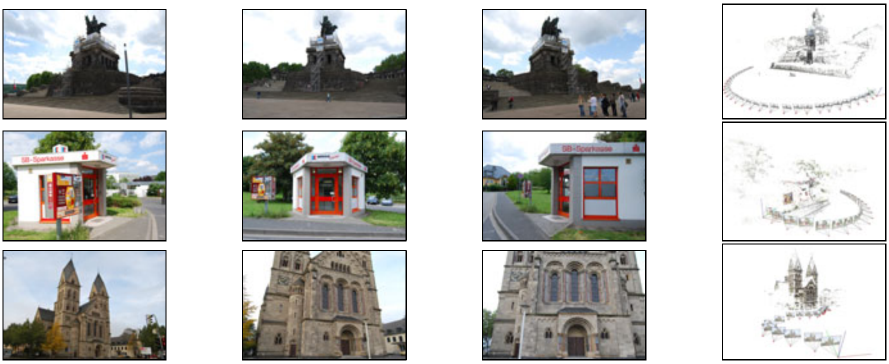


Fig. 4. Examples from the Deutsches Eck, ATM and Herz Jesu image sequences and the resulting models.

Table 1. Number of frames, world points and the mean reprojection error for all models

model	$\Vert\mathcal{F}\Vert$	$\Vert\mathcal{P}\Vert$	$\mu(\Delta\boldsymbol{p}^{\mathrm{p}})$
Deutsches Eck	31	41403	1.51057
ATM	18	53367	1.19294
Herz Jesu	20	62886	1.76852

7.2 Model Accuracy

To analyze the accuracy of our models, we consider the reprojection error $\Delta\boldsymbol{p}^{\mathrm{p}}$ of world points, that is the error between the feature location and the corresponding world point reprojected to the image plane. Table 1 lists the mean reprojection errors $\mu(\Delta\boldsymbol{p}^{\mathrm{p}})$ of all world points and their corresponding features, measured in pixels. Note that a 2 pixel error is less then 0.05% of the image's diagonal.

7.3 Robustness of the Pose Estimation

To test the robustness of our approach towards changes in the environment, we took a second sequence of images from the Deutsches Eck. The second sequence includes another 31 images, taken from very different positions and viewing angles than the first sequence. They were taken several weeks later, when the scaffolding was removed from the monument and the weather condition was very different. The main challenges are a difficult lighting situation with backlighting, the missing scaffolding which contributed many features to the model, as well as the very wide baseline of several images towards the first sequence. Fig. 5 shows two exemplary images from the second sequence, for the first sequence see Fig. 4.

The results of our pose estimation applied to the images of the second sequence is visible in Fig. 6a. Note that the images of the second sequence were *not* used to enhance the model iteratively.

Since there was no ground truth of the image sequences, we had to determine manually if the computed pose was correct or not. From the 31 images, 23 times the pose was computed correctly, in 3 cases there was only a small error and for 5 images the pose estimation produced erroneous results. In most cases, this

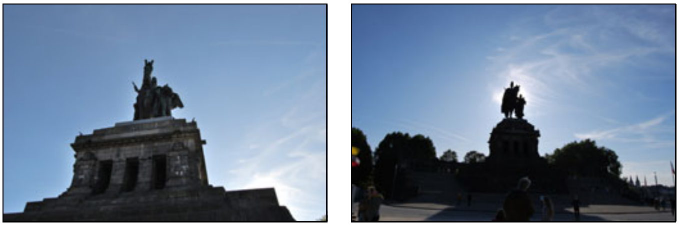
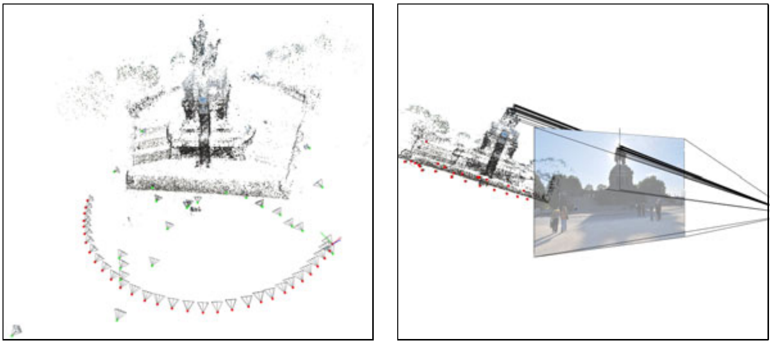


Fig. 5. Examples from the second image sequence of the Deutsches Eck



(a) Green cameras mark the positions of frames from the second sequence, red cameras mark the positions of frames from the first sequence which was used to build the model.

(b) An example of a wrong estimated pose due to degenerate data and a single outlier.

Fig. 6. Pose estimation applied to the second Deutsches Eck sequence

was due to some degenerate point configuration which we do not detect and handle correctly yet, as in Fig. 6b. There, a degenerate set of points allows the pose estimation to include a single outlier (at the bottom of the image) and still appear valid.

8 Conclusion

In this paper we presented a formalism for a model suitable for image based pose estimation. The model uses SURF features solely. We showed how to create a model from images automatically, and how pose estimation on a model works. We also formulated an algorithm to integrate two models which partially overlap into a single model.

Our evaluation revealed a high accuracy of the automatically generated models, with a mean reprojection error of world points less than 0.05% of the image's diagonal. We showed that the proposed model can be used for pose estimation even for images taken under different, more difficult lighting situations with large changes in viewpoint and a partially changed world. It is therefore suitable for initialization of pose tracking or similar applications in changing outdoor environments.

In future work we would like to address the scaleability of our approach and refine the detection of degenerate configurations to make pose estimation even more robust. We would also like to test our model building and pose estimation on images with ground truth information.

This work was supported by the DFG under grant PA 599/7 and PR 161/12.

References

1. Bay, H., Ess, A., Tuytelaars, T., van Gool, L.: Speeded-up robust features (surf). Journal of Computer Vision 110, 346–359 (2008)
2. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT 2006: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2006), pp. 33–40. IEEE Computer Society, Washington, DC (2006)
3. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2007)
4. Wu, C., Fraundorfer, F., Frahm, J.M., Pollefeys, M.: 3d model search and pose estimation from single images using vip features. In: Computer Vision and Pattern Recognition Workshop, pp. 1–8 (2008)
5. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vision 80, 189–210 (2008)
6. Irschara, A., Zach, C., Frahm, J., Bischof, H.: From structure-from-motion point clouds to fast location recognition, pp. 2599–2606 (2009)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395 (1981)
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
10. Nistér, D.: An efficient solution to the five-point relative pose problem. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 195–202 (2003)
11. Lourakis, M., Argyros, A.: Sba: A software package for generic sparse bundle adjustment. ACM Trans. Math. Software 36, 1–30 (2009)
12. Lourakis, M.: levmar: Levenberg-marquardt nonlinear least squares algorithms in c/c++ (2004) (accessed on January 31, 2005)
13. Muja, M.: Flann, fast library for approximate nearest neighbors (2009), `http://mloss.org/software/view/143/`
14. Fiore, P.D.: Efficient linear solution of exterior orientation. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 140–148 (2001)
15. Umeyama, S.: Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 376–380 (1991)