Wide area camera localization

Valeria Garro[1], Maurizio Galassi[1], and Andrea Fusiello[2] *

[1] Department of Computer Science,
University of Verona,
Strada Le Grazie 15, 37134 Verona (Italy)
[2] Department of Electrical, Mechanical and Management Engineering,
University of Udine,
Via Delle Scienze 206, 33100 Udine (Italy)

Abstract. In this paper we describe a mobile camera localization system that is able to accurately estimate the pose of an hand-held camera inside a known urban environment. The work leverages on a precomputed 3D structure obtained by a hierarchical Structure from Motion pipeline to compute the 2D-3D correspondences needed to orient the camera. The hierarchical cluster structure, given by the SfM, guides the localization process providing accurate and reliable features matching. Experiments in outdoor challenging environments demonstrate the effectiveness of the method compared to a standard image retrieval approach.

Keywords: Localization, Camera pose, Structure from motion

1 Introduction

The problem of providing a precise localization of a portable camera has been widely investigated in computer vision. A particular aspect of this issue is *image-based localization*, i.e., computing the camera pose estimation of the device using only the information given by the image or video itself. This topic is included in a wide range of applications such as video surveillance and robot localization, augmented reality application for eHeritage and gaming.

In these particular scenarios a very accurate level of localization is needed, hence positioning systems employing only GPS or Wi-Fi sensors are not sufficient. For example GPS signal is missing in indoor environment and even if available outside, its accuracy could be affected by atmospheric conditions and natural and artificial barriers, furthermore these type of sensors provide only the 3D position of the hand-held device but not the camera orientation. For these reasons further techniques based directly on image processing must be included in the system in order to provide a complete and precise camera pose estimation. We propose a complete image-based system that provides an accurate camera pose estimation of compact devices like smartphones, surveillance and consumer

* This work has been carried out while a.f. was with the University of Verona

cameras in a urban environment. Image-based techniques usually request as input only a set of unordered images of the scene where one wants the positioning to take place. Thanks to the recent improvement in computer vision research on Structure from Motion (SfM) [1–5], in addition to this image archive we can also rely on the 3D reconstruction of the environment. Implementations of different SfM algorithms are available online[3], furthermore in the last years several new SfM techniques have been presented to improve scalability exploiting large scale photo collections [2, 3] and to augment efficiency and precision using hierarchical methods [4, 5].

A variety of approaches that exploit both 3D and 2D data for location recognition has been presented in the computer vision literature. In [6] the authors present a complete system integrating SfM and image-base technique for fast location recognition. They propose the creation of a set of synthetic views in addition to the initial dataset of images used for SfM reconstruction in order to cover as much as possible the corresponding area and be able to compute also the camera pose of query images taken far from the original dataset. In [7] a typical computer graphics approach for visibility estimation is applied in order to reduce the dataset of images to process during the retrieval step. The authors divide the 3D points cloud into view cells and pre-compute a cell-to-cell visibility data. These Potentially Visible Sets (PVS) determine the subset of 3D points and related descriptors that have to be considered according to the current cell.

Recent works focus on a direct 2D-to-3D registration that omits the conventional image retrieval step. In [8] a prioritized feature matching algorithm that matches a limited set of representative 3D scene features to features in the query image is proposed. In [9] the authors devised a direct matching procedure based on visual vocabulary quantization of the 3D features and a prioritized correspondence search. A further step has been introduced by [10] and [11], where a unified formulation of searching strategies has been explored that includes both 2D-to-3D and 3D-to-2D matching on large scale datasets.

In this paper we present a localization system leveraging on spatial 3D information, that combined with an efficient image retrieval technique, provides a fast and precise camera pose estimation of a single image or a video frame capture with an hand-held device. Our algorithm relies on a hierarchical SfM pipeline [4, 5] that besides the 3D points cloud creation provides a hierarchical cluster structure that guides the reconstruction process. It computes a sparse set of 3D points endowed with features descriptors (the "model") by processing a unordered set of images of the scene (the "images archive"). A set of 2D-3D point correspondences between the current frame and the model is needed in order to compute the camera pose estimation. Since typically the 2D points visible in one image are a small subset of the whole reconstruction, it is highly advisable to deploy pruning strategies to limit the set of 3D candidates. Our technique is based on retrieving a small set of the most similar images to the current frame from the archive and then limiting the candidates to those 3D points that are visible in the retrieved images. The retrieval procedure follows a Bag-of-Words

[3] `http://homes.cs.washington.edu/~ccwu/vsfm/` or `http://www.3dflow.net/`

(BoW) approach with tf-idf weighting [12, 13] in order to give a compact representation of each image of the archive. Additionally this last step exploits also the hierarchical organization (called "dendrogram" or binary clustering tree) of the images archive produced by the clustering stage of the SfM algorithm.

More in details, the leaves of the binary cluster tree are associated with the single images of archive, the inner nodes represent a cluster of two or more images created during the reconstruction process of the SfM algorithm. The proposed approach leads to a fast and precise search algorithm. It is more efficient than the classic indirect image retrieval approach because the BoW vectors comparison is limited to a particular set of the inner nodes of the dendrogram avoiding a comparison procedure with the complete image archive. At the same time it guarantees more coherent retrieval results preventing the retrieval of single "outlier" image.

We test the performance of the algorithm in four different urban scenarios. In particular on the last more challenging dataset we have built also an handmade ground-truth in order to compute quantitative results on a video frame sequence.

2 System Overview

The system involves two main stages (see Fig.1):

- An "off-line" stage that runs the SfM pipeline and indexes each node of the dendrogram according to the Bag-of-words approach;
- An "on-line" stage during which the video stream captured from the mobile camera is transmitted over Wi-Fi connection to a server that processes each frame accordingly in order to orient the camera.

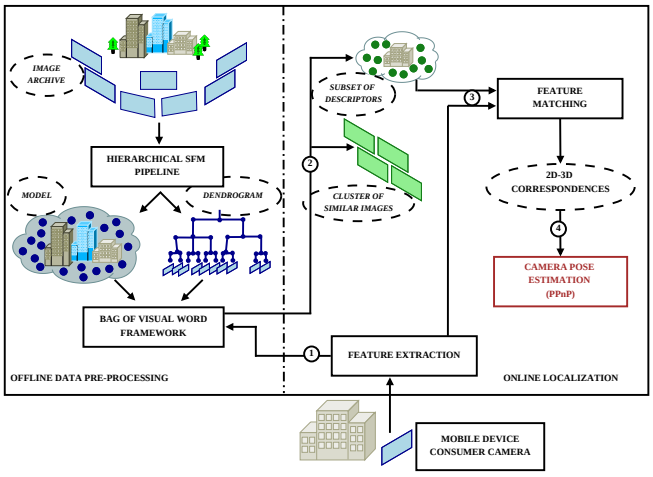

Fig. 1. System Overview. The "off-line" data pre-processing step are represented on the left of the image, the "on-line" stages are outlined on the right.

2.1 Offline data pre-processing

The off-line stage is devoted to compute a 3D reconstruction of the environment from the images archive. After running the SfM pipeline [4, 5] we obtain a 3D points cloud where each 3D point is endowed with a set of SURF [14] features descriptors, and a binary cluster tree (dendrogram) from which we can infer a hierarchical clustering of images.

Indexing and retrieval follows the well-known Bag-of-Words (BoW) framework. With the BoW approach images are represented by an histogram of occurrences of visual words from the codebook. These visual words are usually provided by quantization of the entire set of feature descriptors associated with the 3D points of the model. Additionally, we represent also each inner node of the dendrogram by a BoW histogram. Each inner node is a cluster of images and its BoW vector is computed using the feature descriptors related to the subset of 3D points visible from the images belonging to that cluster. The root of the binary cluster tree (level 0) identifies the whole 3D reconstruction. If the dendrogram is well balanced, its first levels are associated with big portions of the 3D points cloud, therefore their BoW histograms are not very discriminative and can be excluded from the retrieval step.

Different approaches can be employed for the descriptors quantization depending on the size of the dataset: for a relatively small dataset k-means clustering can be sufficient, however, in the pursue of scalability, more complex data structures like vocabulary tree [15] or random forest [16] must be used. We apply also the "term frequency - inverse document frequency" (tf-idf) weighting scheme: given a visual word t in an image d, its weight is given by: $\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$. The term frequency (tf) is simply the (normalized) occurrence count of a visual word in the image: $\text{tf}_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}}$ where $n_{t,d}$ is the number of occurrences of the visual word t in the image d. The inverse document frequency (idf) evaluates the general importance (or rarity) of the visual term: $\text{idf}_t = \log \frac{|M|}{1+|\{i:n_{t,i}\neq 0\}|}$ where M is the set of all images and $\{i : n_{t,i} \neq 0\}$ is the set of images where the visual word t appears at least one time.

2.2 On-line processing

The on-line phase consists first in retrieving from the archive the most similar images to the current one, in order to limit the 3D matching candidates to those points that are visible in the retrieved images. Then the 2D-3D matches are used to orient the camera by solving an exterior orientation or Perspective-n-Point problem camera pose (PnP) problem. In particular, the on-line stage consists of the following steps, as illustrated in Fig. 1:

1. Fast-Hessian features detection and SURF descriptor extraction [14];
2. Retrieval of the most similar images exploiting the dendrogram and recover of SURF descriptors related to the 3D points viewed by the retrieved images;
3. Descriptors matching;
4. Camera orientation (or pose estimation) from 2D-3D correspondences.

First, keypoint features are detected and descriptors are extracted from the current frame (query image), in the specific case Fast-Hessian features and SURF descriptors [14] have been chosen, then each feature is assigned to a visual word of the codebook using a kd-tree structure, and the BoW histogram (H_q) of the query image is computed. Then, we run the retrieval step where the similarity between H_q and the BoW histograms related to the nodes of the dendrogram is computed by using the cosine similarity function: $\mathrm{sim}(H_q, H_i) = \frac{H_q \cdot H_i}{\|H_q\| \|H_i\|}$ for each node i belonging to a specific level of the dendrogram D.

Suppose having a balanced dendrogram, the similarity check can be applied only to a particular subset of inner nodes D, reducing the number of comparisons. Nodes of the dendrogram with small depth (i.e. near the root of the tree) are associated to a large portion of the reconstruction and therefore their BoW histograms can be not so discriminative. For this reason we compare the BoW histogram of the query image only with the inner nodes whose subtrees contain a limited number of leaves (dataset images) (e.g. $6-10$). The most similar inner node $\tilde{D}$ is now determined, the leaves of the subtree with root $\tilde{D}$ are the subset of most similar images, defined $\tilde{M} \subset M$.

The second step consists in selecting the points of the 3D model visible from the cluster related to $\tilde{D}$ and $\tilde{M}$. Finally, a set of tentative correspondences between 2D query points and 3D model points is obtained with nearest-neighbor matching between the descriptors extracted from the query image and the descriptors of the 3D points just selected.

Given a number of 2D-3D point correspondences $\mathbf{m}_j \leftrightarrow \mathbf{M}_j$ and the intrinsic camera parameters K, the exterior image orientation problem requires to find a rotation matrix R and a translation vector $\mathbf{t}$ (which specify attitude and position of the camera) such that:

$$\zeta_j \tilde{\mathbf{m}}_j = K[R|\mathbf{t}]\tilde{\mathbf{M}}_j \quad \text{for all } j \tag{1}$$

where ζ_j denotes the depth of $\mathbf{M}_j$, and the $\tilde{}$ denotes homogeneous coordinates (with a trailing "1").

In literature there are many algorithms that solve this problem [17–19]; we adopted the PPnP approach [20], a simple and efficient solution that formulates it in terms of an instance of the anisotropic orthogonal Procrustes problem. In the remaining of this section we will briefly summarize this approach. After some rewriting, (1) becomes:

$$\underbrace{\begin{bmatrix} \mathbf{M}_1^T \\ \vdots \\ \mathbf{M}_n^T \end{bmatrix}}_{S} = \underbrace{\begin{bmatrix} \zeta_1 & 0 & \dots & 0 \\ 0 & \zeta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \zeta_n \end{bmatrix}}_{Z} \underbrace{\begin{bmatrix} \tilde{\mathbf{p}}_1^T \\ \vdots \\ \tilde{\mathbf{p}}_n^T \end{bmatrix}}_{P} R + \underbrace{\begin{bmatrix} \mathbf{c}^T \\ \vdots \\ \mathbf{c}^T \end{bmatrix}}_{\mathbf{1}\mathbf{c}^T}. \tag{2}$$

where $\tilde{\mathbf{p}}_j = K^{-1}\tilde{\mathbf{m}}_j$, $\mathbf{c} = -R^T\mathbf{t}$, and $\mathbf{1}$ is the unit vector. Therefore, the previous equation can be written more compactly in matrix form:

$$S = ZPR + \mathbf{1}\mathbf{c}^T \tag{3}$$

Fig. 2. Sample images of dataset *Piazza del Santo*(top) and *Piazza Brà* (bottom).

This is an instance of an *anisotropic* orthogonal Procrustes problem with *data* scaling [21]. The solution of this problem finds Z, R and $\mathbf{c}$ in such a way to minimize the sum of squares of the residual matrix $\Delta = S - ZPR - \mathbf{1}\mathbf{c}^T$.
This can be written as

$$\min \|\Delta\|_F^2 \text{ subject to } R^T R = I, \tag{4}$$

which can be solved with Lagrangian multipliers, yielding (the derivation of the formulae is reported in [20]):

$$R = U \text{diag}\left(1,1,\det(UV^T)\right)V^T \text{ with } UDV^T = P^T Z \left(I - \mathbf{1}\,\mathbf{1}^T/n\right) S \tag{5}$$

$$\mathbf{c} = \left(S - ZPR\right)^T \mathbf{1}/n \tag{6}$$

$$Z = \text{diag}\left(PR(S^T - \mathbf{c}\mathbf{1}^T)\right) \text{diag}\left(PP^T\right)^{-1}. \tag{7}$$

It turns out that – as opposed to the isotropic case – here the unknowns are entangled in such a way that one must resort to a *block relaxation* scheme, where each variable is alternatively estimated while keeping the others fixed. Empirically, the procedure always converges to the correct solution starting from a random initialization. In order to cope with outliers we use PPnP as minimal solver ($n = 3$) and MSAC [22], as customary. A further processing of camera pose can be done applying a non-linear refinement on inliers that minimizes the reprojection error. This is however discretionary, subject to the time budget.

3 Experiments

In this section we describe the two different experiments performed to evaluate the proposed method. The first experiments has been run testing two outdoor scenarios, *Piazza del Santo* and *Piazza Brà*. The first archive is composed by 105 images (2592×1944) of a big city square outside an historical church, the second archive represents a bigger and more articulated square with much more repetitive structures, with 320 images (1504×1000).

Fig. 2 shows some image examples. A quantitative evaluation of camera pose estimation accuracy is based on leave-one-out tests.

Each camera provided by the SfM pipeline has been first removed from the image archive and consequently the related set of feature descriptors; then the proposed algorithm has been run on the updated archive. In this way we can consider the registered camera obtained with the SfM pipeline as our ground-truth data.

In order to test the performance of our retrieval method involving the dendrogram of the Structure from Motion reconstruction we compare it with the classic approach that measures the cosine distance between the BOW vector of the query image and the BOW vectors of each dataset image.

Table 1. Leave-one-out validation results on the two datasets. Values within the parenthesis indicate the errors after the non-linear refinement.

	# Images	# Features	# 3D points	Success Rate	Translation Error [m]	Rotation Error [deg]	Reprojection Error [px]
Piazza del Santo	105	45k	30k	95%	0.114	0.103	1.016
classic retrieval					(0.080)	(0.074)	(0.812)
Piazza del Santo	105	45k	30k	96%	0.050	0.091	0.951
dendrogram approach					(0.037)	(0.051)	(0.791)
Piazza Brà	320	233k	50k	92%	0.192	0.378	0.586
classic retrieval					(0.176)	(0.338)	(0.481)
Piazza Brà	320	233k	50k	92%	0.077	0.154	0.602
dendrogram approach					(0.058)	(0.114)	(0.486)

Table 1 shows the results for the leave-one-out tests, where the success rate is the percentage of images localized having a set of correspondences inliers larger than 20 after the camera pose estimation using MSAC. The accuracy of our algorithm is shown in terms of mean euclidean distance of the camera center with respect to the ground-truth, the mean reprojection error of the 3D points visible from the specific camera and the mean residual rotation angle given by the geodesic distance in $SO(3)$:

$$d_g(R_{gt}R_l) = \left\|\log R_{gt}^T R_l\right\|_F \tag{8}$$

where R_{gt} is the rotation component of the camera matrix of the ground-truth data $P_{gt} = K_{gt}\left[R_{gt}|T_{gt}\right]$ and R_l is the rotation component of the camera matrix $P_l = K_l\left[R_l|T_l\right]$ computed by our algorithm. In both experiments our approach clearly outperforms the classic retrieval in terms of accuracy with comparable registration rate results.

Furthermore, for the *Piazza del Santo* dataset three different video sequences of a person walking on the area have been acquired with a simple consumer camera, each video sequence is formed by 900 frames out of which only 38 have not been successfully localized. A qualitative evaluation of the localization is reported in Fig 3.

The second test has been run on a challenging outdoor environment consisting of a parking space located in between several buildings with repetitive structures. The image archive is composed by 543 images (2048×1536), the 3D model

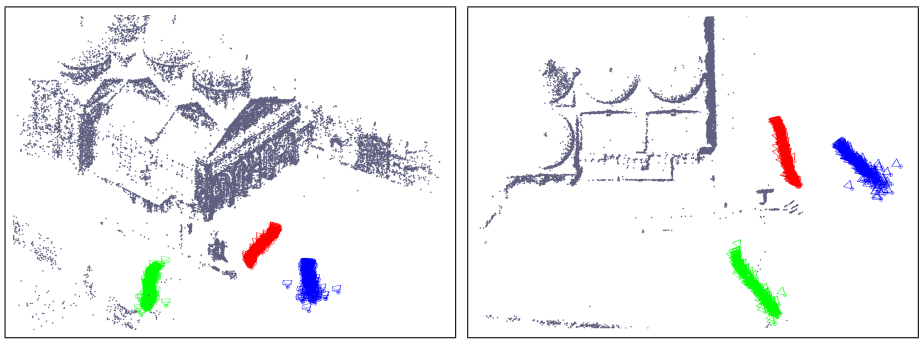

Fig. 3. The colored blobs represent the estimated camera pose for each frame of three different video sequences, perspective (left) and top (right) views.

of the scene is described by a set of 32k 3D points and 203k SURF descriptors. Furthermore the experiment involved four static cameras slightly overlapped, installed on the parking area corners. These cameras were connected with a server that stored the 30 fps images from the cameras, synchronizing them and giving a common time stamp. The four cameras have been calibrated with respect to the reference system of the 3D model, achieving a coherent system. A sample of the four camera views can be seen in Fig. 4. The test consisted of an agent equipped with a proprietary device[4] fixed on the shoulder, walking in the area in a wide closed loop while recording the scene. Analyzing the video sequences recorded by the static cameras we computed the ground-truth for each frame, estimating the 3D position of the agent on the ground floor using a suitable homography transformation.

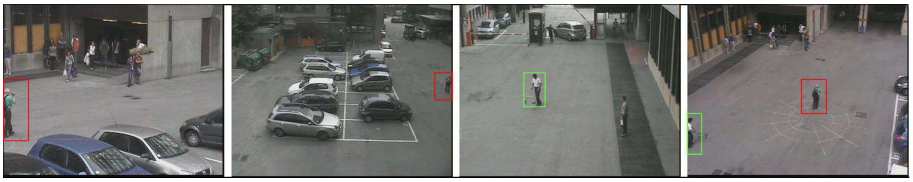

Fig. 4. Snapshots taken at the same time by the four static cameras.

Due to the hardware setup the chain delays induced a variable frame rate transmission of the mobile device data, therefore it was not possible to couple frame by frame the agent and the static cameras views. In order to overcome this synchronization problem over the all frames we decided to evaluate the trajectory of the agent instead of comparing each single position. Each ground-truth path extracted by the four cameras has been approximated by fitting a polynomial curve to the data, generating a set of four segments representing the path. The trajectory directions, the original ground-truth positions and the

[4] This device have been specifically designed for mobile surveillance and provides high quality recording of time-stamped audio-video sequences. It has a ARM Cortex A8 core processor running at 720 MHz and an integrated Microsoft LifeCam Studio webcam with a resolution of 1280×720.

fitted segments are shown in Fig. 5. The average distance error is 2.82 m and the success rate is 45% We run our tests on a Intel QuadCore with 2.4Ghz, the C++ implementation of the algorithm takes less than 2 seconds. More in details, the feature and descriptor extraction requires 0.55 seconds, the retrieval 0.30 seconds, for the feature matching step the time is 0.65 seconds and finally the camera pose estimation with PPnP and MSAC takes 0.17 seconds.

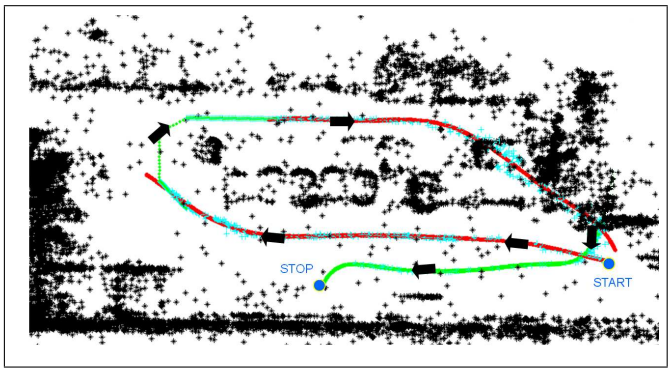

Fig. 5. Sequence trajectory. The ground-truth positions are marked in cyan, the four segments are indicated in alternated colors, green and red. Dashed lines indicate the absence of ground-truth data, as the path falls out of the field of view of the cameras.

4 Conclusions and Future Works

We described a mobile camera localization system in a known urban environment. Localization occurs via 2D keypoint matching against a 3D points cloud obtained by a hierarchical SfM pipeline, leveraging in the image retrieval step on the additional hierarchical structure given by the SfM. Future work will aim at achieving real-time processing of the video frames by GPU implementation and exploitation of motion constraints (presently, each frame is localized independently from the previous ones).

Acknowledgments. This work has been funded by the EU SAMURAI Project.

References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques. (2006) 835–846
2. Frahm, J., *et al.*: Building rome on a cloudless day. In: Proc. of the European Conference on Computer Vision. (2010) IV: 368–381
3. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8

4. Farenzena, M., Fusiello, A., Gherardi, R.: Structure-and-motion pipeline on a hierarchical cluster tree. In: Proc. of the IEEE International Workshop on 3-D Digital Imaging and Modeling. (2009) 1489–1496
5. Gherardi, R., Farenzena, M., Fusiello, A.: Improving the efficiency of hierarchical structure-and-motion. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. (13-18 June 2010) 1594 – 1600
6. Irschara, A., Zach, C., Frahm, J., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. (2009) 2599–2606
7. Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D.: Wide area localization on mobile phones. In: Proc. of the 8th IEEE International Symposium on Mixed and Augmented Reality. (2009) 73–82
8. Li, Y., Snavely, N., Huttenlocher, D.: Location recognition using prioritized feature matching. In: Proc. European conference on Computer vision. (2010) 791–804
9. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: Proc. of the International Conference on Computer Vision. (2011) 667–674
10. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3d point clouds. In: Proc. European Conference on Computer Vision. (2012) 15–29
11. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: Proc. of the European conference on Computer Vision. (2012) 752–765
12. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. of the International Conference on Computer Vision. (2003) 1470–1477
13. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag-of-visual-words representations in scene classification. In: Proc. of the International Workshop on Multimedia Information Retrieval. (2007) 197–206
14. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110**(3) (2008) 346–359
15. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. Conference on Computer Vision and Pattern Recognition. (2006) 2161–2168
16. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. (2007)
17. Moreno-Noguer, F., Lepetit, V., Fua, P.: Accurate non-iterative o(n) solution to the pnp problem. In: Proceeding of the IEEE International Conference on Computer Vision. (October 2007)
18. Fiore, P.D.: Efficient linear solution of exterior orientation. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(2) (2001) 140–148
19. Schweighofer, G., Pinz, A.: Globally optimal o(n) solution to the pnp problem for general camera models. In: Proc. of the British Machine Vision Conference. (2008) 55.1–55.10
20. Garro, V., Crosilla, F., Fusiello, A.: Solving the pnp problem with anisotropic orthogonal procrustes analysis. In: Proc. Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission. (2012)
21. Bennani Dosse, M., Ten Berge, J.: Anisotropic orthogonal procrustes analysis. Journal of Classification **27**(1) (2010) 111–128
22. Torr, P.H.S., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding **78** (2000)