3D Tracking in Unknown Environments Using On-Line Keypoint Learning for Mobile Augmented Reality

Gerhard Schall Helmut Grabner Michael Grabner Paul Wohlhart
Dieter Schmalstieg Horst Bischof

Graz University of Technology
Institute for Computer Graphics and Vision

{schall, hgrabner, mgrabner, wohlhart, schmalstieg, bischof}@icg.tugraz.at

Abstract

In this paper we present a natural feature tracking algorithm based on on-line boosting used for localizing a mobile computer. Mobile augmented reality requires highly accurate and fast six degrees of freedom tracking in order to provide registered graphical overlays to a mobile user. With advances in mobile computer hardware, vision-based tracking approaches have the potential to provide efficient solutions that are non-invasive in contrast to the currently dominating marker-based approaches. We propose to use a tracking approach which can use in an unknown environment, i.e. *the target has not be known beforehand. The core of the tracker is an on-line learning algorithm, which updates the tracker as new data becomes available. This is suitable in many mobile augmented reality applications. We demonstrate the applicability of our approach on tasks where the target objects are not known beforehand,* i.e. *interactive planing.*

1. Introduction

Augmented Reality (AR) is a powerful user interface for mobile computing and location-based systems. AR superimposes registered 3D graphics on the user view of the real world, allowing the user to share the computers perception of the environment. Mobile augmented reality systems provide this service without constraining the users whereabouts to a specially equipped area. In recent years mobile computing devices have seen immense progress in miniaturization and performance. With the advent of smaller mobile and even handheld computing devices the challenge of developing computationally efficient algorithms for these devices increases.

One of the major technological issues in order to create mobile AR solutions is registration. The rendered graphics need to be aligned accurately with the real world view of

(a) (b) (c)

Figure 1. An augmented reality scene with a fiducial marker tracking approach (a) and our marker-less tracking (b) for an indoor application. In the applications we assume that we do not know the target object in advance, which is the case for many planing task, especially for outdoor applications (c).

the user. Consequently, the demands for robust and high-accuracy tracking are strong in order to achieve satisfying registration with limited computational resources. In this context, the applied tracking approach is the key enabler for high-quality AR applications. Tracking approaches for mobile augmented reality are subjected to challenging conditions, such as high variability of the environment (tracking targets), blur, changes in illumination, sudden motion, reflection and occlusion.

In particular, we consider the application in the mobile AR project Vidente[1] for visualization and planing of underground GIS infrastructure in 3D [26]. The tasks within Vidente include table-top models, which can be used for AR urban planning as well as outdoor activities. The goal of this application is to provide egocentric visualization of 3D underground infrastructure data from geospatial databases providing a variety of dynamic visualization styles, like magic lenses for showing excavations [17]. Figure 1(a) shows the superimposed underground infrastructure rendered on the map using fiducial markers and whereas in Figure 1(b) only natural features are used for vision-based tracking. Especially, this is a requirement for outdoor applications as depicted in Figure 1(c).

In this paper we focus on an extension of this application in order to apply the system for planning purposes in unknown environments. Due to the large amount of data and variability no prior model as well as no markers can be assumed to be given. While approaches exist to track natural feature (see Section 2 for a review of methods) the are either very slow or have the need of a (long) pre-training phase. To overcome these limitations, we use a recently proposed tracker based on on-line boosting for key-point matching [9] where robust and accurate tracking results can be obtained in real-time. Despite these properties the major advantage of the tracker is that it does not require any a priori training phase and that it can adapt to several appearance changes of the object (like different lightning condition). Additionally, the approach can deal with any kind of objects (if they have at least some texture) and track them robustly even when they are heavily occluded or move very fast. Furthermore, we can cope naturally with a large database (many different tracking targets) since the on-line learning focuses on a specific sub-problem.

The rest of the paper is organized as follows. Section 2 discusses mobile augmented reality and shows different approaches for tracking a mobile computer. In Section 3 we review the natural feature tracking approach based on on-line boosting and the applicability for tracking in our mobile augmented reality scenario. Especially, we demonstrate the apporaoch within the Vidente framework by experiments in Section 4. Section 5 concludes the paper and gives an outlook for further work.

[1] `http://www.vidente.at`, 2008/03/14

2. Mobile Augmented Reality

Particularly mobile AR is challenged by a variety of factors: For example, hardware and tracking equipment needs to be ergonomic, lightweight enough to carry and at the same time sufficiently powerful for rendering 3D models. Additionally, the platform needs to be resistant to indoor and outdoor conditions as well as functional across a wide spectrum of environmental conditions including illumination, temperature and humidity. Prototype systems for outdoor augmented reality include the MARS backpack setup from Höllerer et al. [12] introducing mobility and Piekarski [22] with the Tinmith navigation system. However, these backpack head-mounted display (HMD)-based setups continue to be inconvenient for mobile AR. By contrast smaller computers, such as ultra-mobile PCs, can be used as a see-through AR device [25]. There is a continuous trend towards these more mobile, lightweight and socially acceptable devices for AR. In general, these systems are mainly focused on presenting information to the user.

Other topics to be addressed in mobile AR include input and interaction technologies to enable the user to interact with the augmented world. Furthermore, wireless networking capabilities need to be considered for accessing remotely stored data. But, most importantly for a satisfying AR experience the digital content needs to be aligned with the real world accurately. Therefore, highly efficient tracking methods are necessary.

Tracking: AR requires 3D real-time tracking, which aims at continuously recovering all six degrees of freedom that define the camera position and orientation relative to the scene, or, equivalently, the 3D displacement of a target object relative to the camera. Many mobile AR applications rely on planar, textured target objects, typically maps and table-top models. Usually these target objects contain artificial information to enable tracking (*e.g.* fiducial markers). There is a wealth of literature on marker-based tracking approaches, like ArtoolkitPlus [33] or Artag [4]. The markers can be designed in such a way that they can be easily detected and identified with an ad hoc method. Illustrative examples of AR applications that rely on marker-based tracking are Invisible Train [32]. The main advantage of traditional fiducial marker tracking lies in robustness. Additionally, marker tracking is not computationally intensive and therefore performs well on mobile devices. But otherwise the major disadvantage is that the markers obscure parts of the valuable map space and therefore represent an invasive tracking approach. Several strategies can be applied for addressing these issues, *e.g.* by using smaller markers or markers with map content such as a north arrow.

Ideally, only natural features are used for tracking without deploying any fiducial markers. Vision-based methods offer a potential for accurate, non-invasive, and low-cost pose tracking. Various approaches are applied to real-time vision-based localization of a mobile device in the real world, indoors and outdoors. A majority of vision-based systems implemented in AR rely on marker based tracking, where the environment needs to be prepared beforehand. But there is a clear shift towards tracking natural features such as corresponding points or edges, statring by *e.g.*, [30, 29, 23].

Tracking using natural features: Recently for dealing with tracking in larger environments model-based approaches became popular for achieving real-time tracking by re-projecting features of the given 3D model into the 2D image. Pose estimation can for example be done by least-squares minimization of an objective function [24, 2]. The main advantage of model-based methods is that the a-priori knowledge allows improving the robustness and performance by being able to predict hidden movement of the object. It is obvious that this approach needs more computational power since the task is more demanding than simple marker tracking approaches. But, since it is often not possible to provide the relevant 3D models, other tracking approaches are needed.

However, it is possible to simultaneously estimate both camera motion and scene geometry, without any such model. These methods use on-the-fly techniques based on simultaneous location and mapping (SLAM) approaches. In [34] Williams *et al.* presented a real-time system based on monocular SLAM which automatically detects and recovers from tracking failure while preserving map integrity. By extending recent advances in keypoint recognition the system can quickly resume tracking. In [18] Nister showed a real-time pose estimation approach in completely unknown scenes. However, long-term stability of that approach is not optimal, because the algorithm tends to drift. Using some of the absolute information is a way to eliminate the drift problem. In order to increase accuracy and robustness Chen *et al.* [1] use a priori knowledge, in the form of a small number of calibrated keyframes and a rough 3D model, for their natural feature algorithm.

Tracking of objects or natural image patches is nowadays often formulated as a classification problem (*e.g.* [7, 14]). The approaches can be summarized as tracking by (fast re-) detection. In contrast to methods using a fixed metric for keypoint description (*e.g.* SIFT [15]), discriminative learning of keypoint descriptions allows incorporating scene information (*e.g.* FERNS [21]). In order to overcome the speed limitations (even with fast implementations, like [8]), efficient keypoint matching using classifiers have been proposed (*e.g.* [14]). Even though the on-line tracking phase can be done very fast, the approach has some limitations. The objects have to be learned off-line (including all possible appearance changes [16, 14]) and no information of the current background can be taken into account.

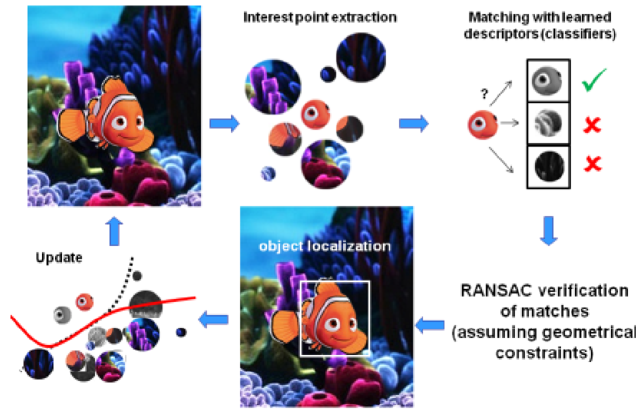


Figure 2. Tracking framework: From a given image the keypoints are extracted, which were matched by a set of classifier. To establish correct matches a robust verification is performed and the object is detected (tracked). In order to adapt to changes in the scene (the object as well as the background) the classifiers (which are describing the object keypoints) gets updates. Positive updates are from the matching patch, negative are from any other detected keypoint.

3. Tracking by Learning Keypoints

We use a tacker recently proposed by Grabner *et al.* [9] which continuously updates its representation in an on-line manner. The tracker, reviewed in the following, performs in real-time, has low memory requirements (as opposed to FERNS [21] and randomized trees) and is able to robustly track a large variety of different objects. The idea is to formulate tracking as a classification problem between successive frames. In fact, this is done by establishing matches over successive frames. A set of classifiers is learned and updated continuously in an on-line manner in order to solve the matching task. The overall workflow is depicted in Figure 2 and described in the following. Keypoints are extracted by some detector (*e.g.* Harris corners [10] or DoG points [15]) from the whole image. In order to start tracking, the assumption is an initial definition of the object region is given. This can be done either manually or by an object detector (*e.g.* using SIFT [15]). Note, when tracking proceeds the region is propagated automatically. The detected keypoints, can be separated into object keypoints and background keypoints. The surrounding region which is covered by the patch for the keypoint is used to train the initial classifiers. Thus, a classifier is connected with a keypoint and should distinguish it from all the other keypoints. In other words, this yields to a multiclass classification problem with a simple one vs. all partition.

Now the tracking loop starts and in order to detect the object again we perform the following steps. When a new image frame is available, first, a set of keypoints K points are extracted. Second we establish possible matches by finding for each classifier its best possible match

$$m_i = \underset{k_j \in K}{\operatorname{argmax}} C_i(\mathbf{p_j}) \quad (1)$$

by evaluation all the classifiers C_i on the image patch p_j which corresponds to the j-th keypoint. Afterwards a verification step for the proposed matches is needed to remove mismatches. This is done by robust estimation of the homography using RANSAC [11] over the set of suggested matches (in our application we assume that the tracked object is planar). Thus, a subset of correct matches verified by the homography is achieved. In case the number of inliers exceeds a threshold, it is assumed to have correctly determined the homography and successfully tracked the object. Therefore, for each classifier its corresponding patch in the actual frame can be calculated which is then used for making a positive update of the classifier. For negative updates patches extracted from any other keypoint are used. If the homography can not be established between two consecutive frames no detection is achieved and no updates are applied. This means, that if the verification is not robust (the object may have disappeared or the geometric constrained are violated) no updates of the classifier are made which limits drifting.

As a result, our goal is to find a set of classifiers such that the probability of a correct match is high with respect to the probability of an incorrect match. For learning and updating the keypoint descriptors on-line boosting for feature selection [6] is used. The machine learning algorithm boosting selects a small subset of simple image features from a large pool of possible candidates. Haar-like features are used which can be fast evaluated using integral images [31]. It allows to generate classifiers that can be efficiently updated by incrementally applying samples. In addition the classifier provides a confidence measure, which is needed in order to determine the best match (see Equation 1). Since the problem of discriminating one keypoint from the others in the current frame is quite simple, the complexity of the classifiers can be low, the number of used feature can be small (we use 20). Therefore, the tracker is applicable to real-time tasks.

Summarizing, in order to track a specific object within the current scene, the current object keypoints have to be distinguished from the ones detected in the background. The usage of on-line classifiers allows to collect samples over time for improving generalization and in addition, to adapt keypoint descriptions even if the scene changes. As a result discriminative classifiers allow to incorporate scene information by considering them as negative samples for the keypoint descriptions. Hence, the object can be tracked robustly even when the appearance changes or under different illumination conditions.

3.1. Pose Estimation

For tracking a mobile device the pose of the camera is of interest. The recovery of a 3D camera pose from a set of object keypoints, for which 3D world coordinates and their projection onto an image plane are given, is known as the Perspective-n-Point (PnP) problem [5]. Among others the POSIT algorithm [3] provides good performance at reasonable computational cost. POSIT iteratively approximates the pose by calculating the scaled orthographic projection (SOP) with respect to a preliminary pose. Then a pose is searched that better maps the object to the SOP, thus providing a better estimation of the pose. We apply an extension of POSIT for planar object point configurations [20]. Since POSIT cannot deal with outliers the algorithm is often used in combination with RANSAC [11]. Since the keypoint matches coming from the tracker are already checked for geometric consistency by calculating a stable homography with RANSAC, we can use them directly as input to only one POSIT run. Furthermore, Schweighofer and Pinz [27] show an efficient method for receiving stable pose estimations for planar objects. Recently Moreno-Noguer *et al.* presented a non-iterative solution to the PnP problem running much faster than and as accurate as iterative methods [19]. By solving the PnP problem the task of tracking a mobile device reduces to calculating world coordinates for keypoints found in the image.

For initialization and aligning the real and the virtual coordinate system we manually mark four points in the first frame. If we know the corresponding world coordinated of these points and the initial camera pose this is sufficient. Otherwise, we mark a rectangle and apply similar techniques as proposed by Simon *et al.* [28]. During runtime world coordinates for image keypoints that were found online by the tracker can be calculated by back-projection onto the object plane. Therefore the pose calculated from the keypoints with known world coordinates from the last step is used.

4. Experiments

We demonstrate the natural feature tracking approach via on-line boosting in two scenarios of an augmented reality application. For the tracker we used 25 object keypoints. Each of them is described the local image patch by selection of 30 features out of a shared feature pool of 50 features (weak classifier). The basic features as well as the object keypoints are continuously adapted as described above. With these parameters and our non-optimized C++ implementation we track with about 5 frames per second, including the visualization. Details where the time is spent can be seen in Table 1 for a typical tracking sequence. We chose an ultra-mobile PC (standard Sony Vaio UX, Intel Core Solo 1.1 GHz, Windows XP, 0.5 kg, Camera resolu-

Figure 3. Setup of the Vidente planning application, where the user is equipped with an ultra-mobile PC that accurately renders 3D structures and information directly on top of the orthographic photo of Jakominiplatz in Graz (first row) and setup for the interactive outdoor planning tool (second row).

tion 640×480) as the core hardware platform, running the Studierstube[2] software.

Task	Time (ms)	Percent
Image Capture & Preperation	15	9.49%
Interest Point Extraction	45	28.48%
Object Feature Matching	19	12.03%
Pose estimation	21	13.29%
On-line Feature Update	8	5.06%
Rendering	50	31.65%

Table 1. Timings per frame.

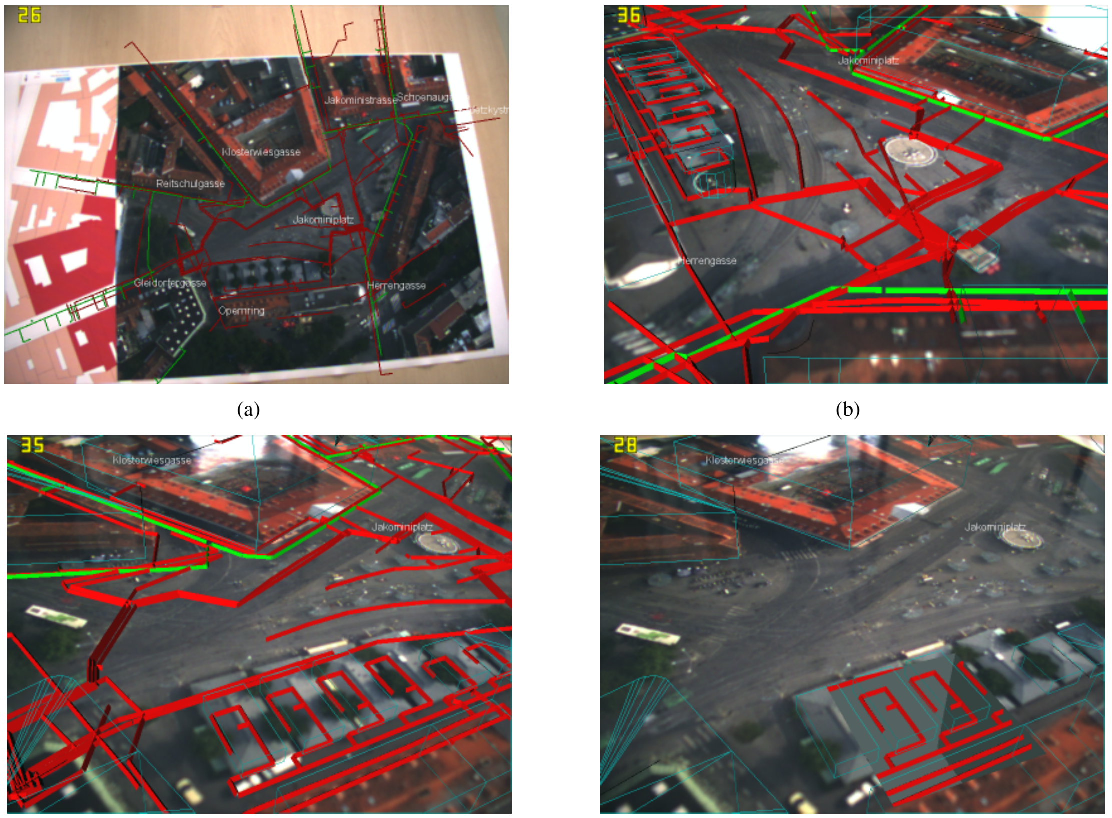
We apply this tracker in the Vidente application that aims to superimpose underground infrastructure on the real world to support planners and field workers. They are aided by this kind of egocentric visualization in tasks such as contractor instruction, outage management and network planning of underground infrastructure. First, we show a planning task, where an arbitrary map can be overlaid with computer graphic models. We illustrate such a planning task with an aerial photo on which we superimpose a 3D model of underground infrastructure. That high-resolution aerial photo was obtained by Vexcel Imaging using an UltraCamX camera [13]. Second, we show how the proposed tracker can be used in unknown environments to perform planning tasks outdoors.

[2] http://www.studierstube.org, 2008/03/14

4.1. Augmented Reality Planning on Maps

This mobile AR application allows the user to superimpose graphical content on arbitrary maps. The first row of Figure 3 depicts the table-top setup. Note that, no prior learning or any information about the target to be tracked needs to be known in advance. The digital content overlaid graphically on the map consists of a 3-dimensional model of the buried assets and pipe networks. Filtering of objects such as pipes, trenches or canals can be determined based on semantic attributes during runtime. This indoor planning task can for example be used for superimposing planned architectural structures and street furniture onto the map or for redesigning existing architecture. For our experiments we chose aerial photos that we track via on-line keypoint learning, but of course the tracking approach is not limited to a specific kind of maps. Figure 4 depicts an aerial map as the target object. We need an initial object region for initialization of the tracker or an initial pose of the object and its world coordinates for the pose estimation respectively. We apply the following procedure. First, by pointing the built-in camera of an ultra-mobile PC towards the orthographic aerial photo, the user determines the object he wants to track from the video background. Afterwards the video is frozen which allows the user to select the target object on the still image. This is done by selecting a quadrangle area on the image via the touch screen of the device. Having performed this procedure the target object is known and the user unfreezes the video background. In addition we know the relative position of the corners in world coordinates. The system allows for tracking the target immediately. The pose of the mobile users camera is determined in real time which leads to a precise registration of the 3D model with the map.

This tracker is ideally suited for interactively choosing the target object and start tracking immediately. Consequently the major advantage of the proposed tracker especially evolves in unknown environments by learning online. Classically, this is the case when going outdoors where it is not possible to collect sufficient information needed by an off-line trained tracker. In this section, some example screenshots of our mobile AR application are shown and described. Figure 4(a) illustrates a top view onto the orthographic map overlaid with the gas and electricity pipe network in "X-Ray Vision" to be able to see what is underneath the ground. Figure 4(b) and Figure 4(c) depict an augmented view of the mobile user from a closer distance and different perspectives onto the map. Figure 4(d) shows the user view from the same perspective as shown in Figure 4(c). Here the visualization style "Magic Lens" is used, yielding a movable virtual excavation. Usually, the rendering style of objects inside the lens is changed and objects are displayed differently. In this example buried utilities are only shown inside the lens. The images in Figure 1(a) and



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

Figure 4. Rendering the full 3D model of underground assets at the Jakominiplatz site in X-Ray Vision. Additionally the visualization style Magic Lens is used, yielding a movable virtual excavation.

Figure 1(b) at the front page show the AR user view from a similar perspective using the two visualization techniques “X-Ray Vision” and “Magic Lens” in combination.

4.2. Augmented Reality Planning in Outdoor Environments

The major advantage of on-line learning for mobile augmented reality is that for tracking planar, textured objects no previous preparation of the target object or environment is necessary. In this example scenario we show that the same tracker used for the indoor planning tool is perfectly suited to be applied in outdoor environments. The procedure the user performs is analog to the one described previously. By pointing the camera of the ultra-mobile PC at the facade of the building, the user selects four points on the still video image and can start tracking immediately. The second row in Figure 3 depicts an outdoor user assisted by the AR planning application. Figure 5(a) illustrates a planned 3D pipe object overlaid on the outdoor environment. The user is able to move the pipe object with the integrated joystick to the desired position. The object is then fixed to this position and can be viewed from different view points. Figure 5(b) shows a “Magic Lens” that is superimposed on the ground next to the building faade. In contrast to model-based tracking approaches used in AR outdoor applications the major advantage of the on-line learning tracker is that no textured models of the environment are necessary.

Summarizing, the experiments illustrate the applicability of the presented tracker for dealing with tracking planar, textured objects in both indoor and outdoor environments. Consequently a mobile device used for AR applications is localized even under difficult environment conditions.

5. Conclusions and Further Work

Augmented reality benefits from applying non-invasive natural feature tracking methods in contrast to currently dominating marker-based approaches. Towards this aim we presented a natural feature tracking algorithm based on on-line boosting used for localizing a mobile computer. We demonstrated this fast marker-less vision-based tracking approach within an augmented reality application. We also built an outdoor prototype within the Vidente project. Therefore our mobile platform is equipped with both position and orientation sensors for outdoor tracking (*i.e.*, real-time kinematics GPS and inertia measurement unit). In-

(a) (b)

Figure 5. A planned pipe object overlaid on the outdoor environment(a) and a "Magic Lens" representing a virtual excavation that is superimposed on the ground next to the building facade.

door AR applications already can benefit from tracking natural features on planar, textured objects. But also in outdoor environment, where conditions are more challenging, the deployment of natural feature tracking seems applicable and promising in future. For example a hybrid tracking approach by combining traditional outdoor trackers with an edge-based tracker for more accurate localization was demonstrated [24]. This vision tracking approach is based on textured 3D models, which are more commonly available. Possible future data sources include servers for virtual globe browsers such as Microsoft Virtual Earth[3] or Google Earth[4]. Traditionally, outdoor environments are subjected to continuous change. Therefore, a priori knowledge used by *e.g.* model-based trackers cannot be guaranteed to be up to date in many situations. Thus leading to deficiencies of off-line trackers. Note, that in this context the tracker based on on-line boosting is especially suited for performing well in such unknown outdoor environments, since new keypoints are learned on-line. Our work represents a step towards confluence of computer vision and graphics helping to produce robust wide-area augmented realities.

Acknowledgements This work was sponsored by the Austrian Research Promotion Agency FFG under contract no. BRIDGE 811000 and no. APAFA 813397, the European Union under contract no. FP6-2004-IST-427571, and the Austrian Science Fund FWF under contract no. Y193 and W1209-N15. Additionally, it has been supported by the Virtual Earth Academic Research Collaboration funded by Microsoft and by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04. We also thank GRINTEC GmbH for providing the relevant geospatial data.

[3] `http://www.virtualearth.com`, 2008/03/14
[4] `http://earth.google.com`, 2008/03/14

References

[1] J. Chen, Y. Wang, Y. Li, W. Hu, and X. Zang. An improved real-time natural feature tracking algorithm for ar application. In *In Proc. Int. Conf. on Artificial Reality and Telexistence Workshop*, pages 119–124, 2006.

[2] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Trans. on Visualization and Comp. Graphics*, 12(4):615–628, 2006.

[3] D. F. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *IJCV*, 15:123–141, 1995.

[4] M. Fiala. Artag, a fiducial marker system using digital techniques. In *Proc. CVPR*, pages 590–596, 2005.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[6] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. CVPR*, volume 1, pages 260–267, 2006.

[7] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proc. BMVC*, volume 1, pages 47–56, 2006.

[8] M. Grabner, H. Grabner, and H. Bischof. Fast approximated SIFT. In *Proc. ACCV*, pages 918–927, 2006.

[9] M. Grabner, H. Grabner, and H. Bischof. Learning features for tracking. In *Proc. CVPR*, 2007.

[10] C. Harris and M. Stephens. A combined corner and edge detection. In *Proc. Alvey Vision Conf.*, pages 147–151, 1988.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Comp. Vision*. Cambridge University Press, second edition, 2004.

[12] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring mars: Developing indoor and outdoor user interfaces to a mobile augmented reality system. In *Comp.s & Graphics*, page 23(6):779785, Zurich, 1999.

[13] F. Leberl, M. Gruber, M. Ponticelli, S. Bernoegger, and R. Perko. The ultracam large format aerial digital camera system. In *Proc. of the ASPRS Annual Convention*, 2003.

[14] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. CVPR*, volume 2, pages 775–781, 2005.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] J. Matas, K. Zimmermann, T. Svoboda, and A. Hilton. Learning efficient linear predictors for motion estimation. In *In Proceedings Indian Conference on Computer Vision, Graphics and Image Processing*, pages 445–456, 2006.

[17] E. Mendez, D. Kalkofen, and D. Schmalstieg. Interactive context-driven visualisation tools for augmented reality. In *Proc. of Int. Symposium on Mixed and Augmented Reality*, 2006.

[18] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. on PAMI*, 26(6):756–777, 2004.

[19] F. M. Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o(n) solution to the pnp problem. In *Proc. ICCV*, pages 1–8, 2007.

[20] D. Oberkampf, D. F. DeMenthon, and L. S. Davis. Iterative pose estimation using coplanar feature points. *Comput. Vis. Image Underst.*, 63(3):495–511, 1996.

[21] M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *CVPR*, 2007.

[22] W. Piekarski and B. Thomas. Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable comp. In *Proc. Int. Semantic Web Conf.*, pages 31–38, Zurich, 2001.

[23] S. Prince, K. Xu, and A. Cheok. Augmented reality camera tracking with homographies. *IEEE Comput. Graph. & Appl.*, 22(6):39–45, 2002.

[24] G. Reitmayr and T. W. Drummond. Going out: Robust tracking for outdoor augmented reality. In *Proc. of Int. Symposium on Mixed and Augmented Reality*, pages 109–118, 2006.

[25] J. Rekimoto. Navicam: A palmtop device approach to augmented reality. In *In Proc. Fundamentals of Wearable Comp.s and Augmented Reality*, 2001.

[26] G. Schall, B. Reitinger, E. Mendez, S. Junghanns, and D. Schmalstieg. Handheld geospatial augmented reality using urban 3d models. In *Workshop on Mobile Spatial Interaction in conj. with ACM Int. Conf. on Human Factors in Computing Systems*, 2007.

[27] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Trans. on PAMI*, 28(12):2024–2030, Dec. 2006.

[28] G. Simon and M. Berger. Pose estimation for planar structures. *IEEE Comp. Graphics and Applications*, 22(6):46–53, 2002.

[29] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. of Int. Symposium on Mixed and Augmented Reality*, pages 120–128, 2000.

[30] D. Stricker and T. Kettenbach. Real-time and markerless vision-based tracking for outdoor augmented reality applications. In *ISAR*, pages 189–190, 2001.

[31] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, volume 1, pages 511–518, 2001.

[32] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Proc. Int. Conf. on Pervasive Computing*, 2005.

[33] D. Wagner and D. Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In *Proc. Comp. Vision Winter Workshop*, 2007.

[34] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. In *Proc. ICCV*, pages 1–8, 2007.