A Facade Tracking System for Outdoor Augmented Reality

José F. Martins[1,2]   Jorge A. Silva[1,3]   A. Augusto de Sousa[1,4]

[1]FEUP

[2]ISMAI, [3]INEB, [4]INESC Porto
R. Dr. Roberto Frias
4200-465 Porto, Portugal

{jfmm, jsilva, aas}@fe.up.pt

ABSTRACT

We describe a real-time facade tracking system that uses, as setup information, only two images of a facade, captured on the moment. No more previous information is needed, such as a facade 3D model, dimensions or aspect ratio. Feature points and their local descriptors are extracted from that pair of images and used during the detection and tracking of the facade. Additionally, parallax and topological information is also used in order to increase the overall robustness of the tracking process. Experiments show that the system can detect and track a wide variety of facades, including those that are not entirely planar, partially occluded or have few distinguishable visual landmarks. The reliance on on-the-spot information, alone, makes this system useful for Outdoor Augmented Reality applications, in an Anywhere Augmentation urban context.

Keywords

Outdoor Augmented Reality, Anywhere Augmentation, Facade Tracking, Computer Vision.

1. INTRODUCTION

The main purpose of Augmented Reality (AR) is to add, in real-time, virtual objects to real world images in such way that they appear to naturally belong to that world [Bar01].

Adding virtual objects to a real image is only visually convincing if they are perfectly registered with the real world. In order to do that, it is necessary to render them from a virtual camera that, ideally, should have the same pose as the real one. There are several methods that use a GPS, a gyroscope, an accelerometer or other sensors to determine the camera pose [Rol01]. However, the methods that appear to have the greatest potential are those that use captured images. One way to determine the camera pose is to find a set of, at least, four matches between points in an image of a reference plane, taken with a known camera pose (usually, a frontal one) and points from the same plane in a image whose camera pose is to be determined [Lep05].

The reference plane must be continuously tracked. For this purpose, Lepetit et al describe several methods.

In indoor scenes, the reference plane is usually an easily identifiable synthesized pattern, as happens with the well known ARToolkit [Kat99, Art09].

In outdoor scenes, the use of synthesized patterns is not feasible for several reasons, namely, the need for a large pattern and the probable difficult positioning of the pattern in the surrounding environment. However, in an outdoor urban environment, the facades of the buildings can act as a suitable reference pattern.

In the AR domain, some important contributions to facade tracking have been made during the current decade [Sim02, Jia04, Rei06, Xu08]. Generally, they can use a building facade as a reference pattern for achieving the registration between the virtual objects and the real world scenario. The affordability and overall good image quality of off-the-shelf digital cameras make the use of captured images a very common approach [Sim02, Xu08].

However, using only captured images as input data has some drawbacks: the captured image can be blurred by jerky or fast camera movements and the reference pattern may become partially visible or not visible at all because of occlusion. These drawbacks make facade tracking very difficult or even impossible to achieve. To overcome them, some systems combine captured images with other types of data, taken from sensors like GPS and/or inertial sensors [Jia04, Rei06, Rei07].

The first step of a facade tracking system is to detect the facade in the captured frames. This will allow the AR system to do the camera pose initialization (CPI). After that, most systems track the facade, based on what happened on previous frames, until the tracking fails, when the CPI must be redone. However, several systems cannot solve the CPI problem automatically; some demand for user interaction [Sim02, Jia04] while others need to know the initial camera position in the real world [Rei06, Xu08].

When the tracking fails, the system must be able to detect the facade again. Although most systems can perform this task automatically, some of them need the camera pose to be similar to the last known pose (prior to tracking failure) [Sim02, Jia04] or to one of the previous known poses [Rei06].

To achieve a more robust tracking, some systems require a 3D model of the facade. It can be a wireframe model [Jia04], or a more elaborated textured one [Rei06]. Unfortunately, creating the model is not an easy task and may take considerable time. Additionally, since it must be created before the tracking phase, anywhere augmentation is an impossible goal for these systems. Such a goal is possible for systems that only need on-the-spot information to start tracking [Sim02, Xu08]. This information is limited to a single image of the facade or just a few ones.

The determination of the camera pose can be based on the matching of feature points, as in [Sim02, Xu08], edges [Jia04] or a combination of the above [Rei06]. Reference image based systems try to match feature points in the captured image with those previously detected in the reference image, while 3D model based systems try to match edges in the image with the rendered 3D model edges.

During the augmentation phase, reference image based systems, like [Sim02, Xu08], usually operate at relatively long distances from the tracked facade (distances superior to the overall dimensions of the facade).

This paper presents a tracking system that can detect and track a building facade in a video stream, provided that the facade has a dominant planar surface and that two images of it, previously taken from different viewpoints, are available. This system overcomes some of the limitations of other systems exclusively based on captured images, namely: it only needs information gathered on-the-spot; the CPI problem is solved without user intervention; when the tracking fails, the camera pose does not need to be similar to a previously known one, for the facade to be detected again; and, finally, the camera does not need to be far from the facade.

The paper is organized as follows: section 2 presents a brief overview of the system; sections 3 and 4 describe the most significant steps of its implementation; section 5 presents the results of the system evaluation and, finally, section 6 enumerates the most important conclusions retrieved from this work.

2. OVERVIEW OF THE SYSTEM

The proposed tracking system has two operating phases: a setup phase and a working phase. In the setup phase, the system acquires and prepares all the necessary information for the next phase, namely a set of feature points, coplanar with the dominant plane of the facade, and their topological relations.

In the working phase, the system will try to detect the facade in a captured image, by matching the feature points detected in the video sequence images with those detected in the setup phase, by using feature descriptors and topological information. If the detection is successful, it will start tracking the facade, by using a conventional sparse optical flow technique. When the tracking fails, the system reverts to the detection phase and the process is repeated.

To detect the facade in an image, it would be convenient to have a reference image, in a frontal pose. In most cases, this pose is impossible to obtain. In addition, since most facades are not planar, a single image, even in a frontal pose, may be insufficient for a correct detection of the facade. Instead, using two reference images of the facade, taken from different positions, will help to identify the coplanar points that, in the proposed way for determining the camera pose, are used in the matching process.

The matching process between one of the reference images of a facade and an arbitrary image of the same facade, captured with an unknown pose, is not a simple task, for various reasons: first, because they have different poses; then, because facades rarely are truly planar; finally, because the two images can be significantly different due to lighting variations, reflections, occlusions or even changes of the facade visual appearance (e.g. when a window blind is closed, between the captures).

Several matching methods are based on feature point extraction from images. Feature point detectors, like SIFT [Low03], SURF [Bay06], FAST [Ros06] or FIRST [Bas08] have been used because of their robustness to changes in scale, view point, luminance and even to partial occlusions. For each extracted feature point there is an associated local descriptor. The comparison of the descriptors is the basis of the matching process. However, since there are many repeated visual landmarks in a facade (e.g. windows), a straightforward comparison would, most likely, produce a large number of incorrect matching pairs. Therefore, additional processing is needed to remove those false matches. In the proposed system, topological constraints are used for this purpose. Similar constraints have been used by other authors [Fer03] to help solving the matching problem between images acquired from different poses.

In the following, the two above mentioned phases are described.

3. SETUP PHASE

The setup phase has the following steps: (a) capture of two facade reference images; (b) delineation of the facade region of interest (ROI); (c) detection of feature points; (d) identification of all the feature points that belong to the facade dominant plane and (e) topological characterization of the feature points.

The information resulting from this phase is: the facade ROI in both reference images; the detected feature points that belong to the facade dominant plane, Π (a plane coplanar with a major part of the facade surface), and the characterization of the feature points. This characterization is achieved in two ways: by a local descriptor that will be used in the matching process and by two types of topological information, associated with each feature point: the collinearity information which tells if it is collinear with other feature points, making up a "line" of feature points and the sidedness information which tells on which side of each of these "lines" the point lies.

Reference Images and ROI

Two reference images of the facade are needed. These images, I_L and I_R, must be taken from different positions and preferably from opposite sides of the facade (Figure 1). This will increase the parallax effect which will help identifying the feature points that do not belong to the dominant plane, Π.

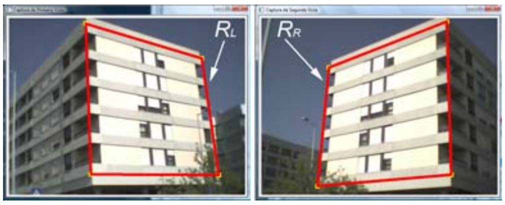

Figure 1. ROIs delineation on I_L and I_R.

The facade ROIs, R_L and R_R, must be delineated on I_L and I_R (Figure 1). This is the only step in the whole process requiring human intervention. This step is necessary to restrict the feature point detection to R_L and R_R and also to identify Π. R_L and R_R are manually delineated by the user, through the selection of the corners of the quadrilaterals, on I_L and I_R, corresponding to a rectangle in the facade. This rectangle does not need to visually exist; only some of its edges and corners are needed, as a visual aid for the user. However, it must be coplanar with Π and cover a major part of the facade (ideally, all of it).

Feature Point Detection

To detect the facade in an image, I_U, captured with an arbitrary pose, it is necessary to match feature points between I_U and the reference images (I_L and I_R). In I_L and I_R, feature detection is restricted to R_L and R_R.

Feature points are detected using the SURF algorithm [Bay06], which is one of the fastest available and has a great potential for AR applications. This algorithm returns the coordinates of each detected point, its feature strength (Hessian value), size, orientation and Laplacian value, as well as a descriptor of its neighborhood in the image. This descriptor is an array of 128 elements that describes the distribution of Haar-wavelet responses within the feature point neighborhood and has the important properties of being invariant to rotation, scale and luminance. These properties help the matching process to produce a higher rate of correct matches between different poses. The detected SURF point sets in R_L and R_R will be named, respectively, S_L and S_R. Only points with a Hessian value greater or equal to 500 are retained. For a better matching performance, both sets are filtered, in order to assure that all the remaining features have a minimum size and a minimum distance between each other.

Identification of Points Belonging to Π

In the working phase, the facade will be detected through the matching of two sets of coplanar points. Therefore, only points belonging to Π should, ideally, be used; all the other points should be removed from S_L and S_R.

The identification of the points non-coplanar with Π has two important steps. In the first one, R_L and R_R are transformed into frontal pose (Figure 2), F_L and F_R. In order to rectify both R_L and R_R, the true dimensions of these regions, or at least their aspect ratio, must be known. Usually, those dimensions are unknown, but the aspect ratio of a rectangle can be calculated from a non-frontal pose image of itself, through the use of the image vanishing points, as described in [Cip99].

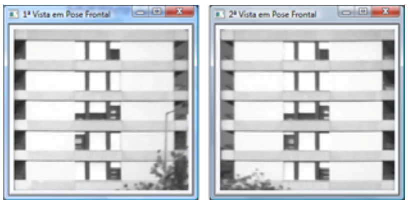

Figure 2. ROIs in a frontal pose (F_L and F_R).

In the second step, S_L and S_R are projected into a frontal pose. After this, the feature points, in R_L and R_R, which belong to Π can be easily identified, since their positions should be the same in F_L and F_R. However, it is also possible that a facade landmark can generate a feature point in one of the reference images, but not in both of them.

So, in order to identify the largest possible number of points belonging to Π, another approach, that does not require a feature point to be detected in both reference images, was adopted instead. Each feature point, in either R_L or R_R, is projected into both F_L and F_R. The neighborhoods of each projected point, in F_L and F_R, are compared using normalized cross correlation and if they are similar enough (correlation factor superior or equal to 85%) then the point is identified as belonging to Π. A feature point resulting from a landmark, like a lamppost, that is away from Π, will usually have different neighborhoods, in F_L and F_R (Figure 3), being identified as non-coplanar with Π.

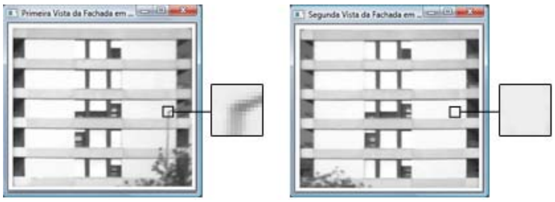
Primeira Vista da Fachada em ...
Segunda Vista da Fachada em ...

Figure 3. Detection of feature points that do not belong to Π.

Topological Characterization of the Feature Points

Facades frequently have a large number of very similar visual landmarks. This contributes to a large rate of false matches. A possible solution for this problem would be to use only singular feature points (points whose descriptor is unique). However, in the case of a facade, this solution would probably reduce the number of useful points to a few.

In this particular case, the topological characterization of the points is a better solution to make each point more easily distinguishable from the others. In many facades, visual landmarks are usually concentrated along horizontal/vertical "lines". In both F_L and F_R, these "lines" can be easily identified, since they are approximately coincident with the rows/columns of the image. Rows/columns of F_L and F_R that have at least 12 feature points within a threshold distance, T_l=5 pixels (Figure 4) are retained as horizontal/vertical "lines". All feature points that do not belong to any retained "line" are removed.

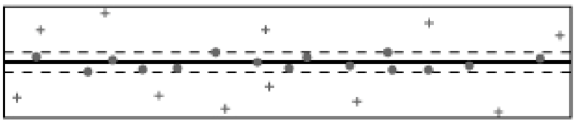

Figure 4. Identification of feature points concentrated near an image row, in F_L or F_R.

The topological characterization process associates the following two types of information with each point, in F_L and F_R: collinearity and sidedness. Collinearity identifies the horizontal and/or vertical "line(s)" of near-collinear feature points that the point belongs to. Sidedness identifies the side where the point lies relatively to each other "line".

4. WORKING PHASE

During the working phase, an image sequence is acquired and processed. This phase is divided into two subphases: detection and tracking. In order to start tracking, the facade must be detected in one of the incoming frames. If the detection is successful, the facade is tracked in the following frames until the tracking fails. Then, the process is repeated.

Facade Detection

The detection subphase consists of the following steps: (a) capture of a frame, I_U; (b) detection of feature points; (c) matching between the feature points detected in I_U and those detected in I_L and I_R; (d) elimination of false matches and (e) calculation of the homography between the matched feature points. Using this homography, it is possible to delineate the facade ROI in the captured frame and determine the camera pose.

4.1.1 Feature Point Detection and Matching

The first step is to capture a frame I_U in which the facade will be either detected or tracked, whatever the case may be. The camera pose and the existence of the facade in I_U are both unknown. Facade detection requires a matching process between the SURF feature points detected in I_U and those detected in I_L and I_R (during the setup phase). In I_U, feature point detection is applied to the whole image. The resulting set of detected points in I_U will be named S_U.

In order to maximize the number of correct matching pairs, two matching processes, $m(S_U,S_L)$ and $m(S_U,S_R)$, are applied to the feature sets S_U, S_L and S_R. To accelerate the matching process, each set is divided into two subsets, based on the Laplacian signs, thus avoiding unnecessary comparisons between local descriptors of points with different signs.

The Best-Bin-First (BBF) method [Bei97] is used for matching. This method uses local feature descriptors in order to find, for each point of the first image, the best match (nearest neighbor), in the second image. A k-d tree [Fri77] is used to store the local descriptors of the points from one of these sets. The nearest neighbor of a given descriptor, on the other set, can be found, very efficiently, by searching only a relatively small part of this tree. The returned nearest neighbor is considered valid, if the Euclidean distance between both descriptors is inferior to certain threshold (a value of 3 was used in our experiments).

Two sets of matching pairs, P_L=$m(S_U,S_L)$ and P_R=$m(S_U,S_R)$, result from this matching process. If neither P_L nor P_R has enough pairs (at least 20), the detection is considered unsuccessful. Otherwise the set with the largest number of matching pairs, named P_S, is selected.

4.1.2 Elimination of False Matches

P_S usually has some incorrect pairs whose number can be affected by several factors: images acquired with different camera poses; luminance variation; the used local descriptor; the matching method; and the visual content of both images. In order to obtain a more robust facade detection and a precise camera pose, it is imperative to reduce, as much as possible, the number of incorrect matches.

A common method to solve this problem [Har03] uses the RANSAC algorithm [Fis81] to calculate, in a robust form, the homography that relates two images of the same plane. This method starts by randomly selecting a set of matching pairs. This set is used to calculate a homography that will be used to project each point of the first image into the second one. The distance between the projected point and its matched pair is calculated. If the distance is too large, the pair is labeled as an incorrect one (outlier). This process is repeated until the number of correct matching pairs (inliers) is acceptable or a maximum number of iterations is reached. If a minimum number of inliers is not achieved, the facade detection is declared as failed. Usually, this method produces good results even in the presence of a large number of outliers. However, in a facade, its straightforward use may generate poor results.

Usually, inliers display a common and uniform behavior that set them apart from outliers, which show a behavior of a random nature. In the case of a facade, outliers may display a common and uniform behavior, for example, when feature points belonging to one building floor match with similar points from another floor. If the number of matches is substantial, RANSAC may wrongly assume that they are inliers.

The proposed solution for outlier elimination is based on the application of the collinearity and sidedness constraints to the selected set of matching pairs (P_S).

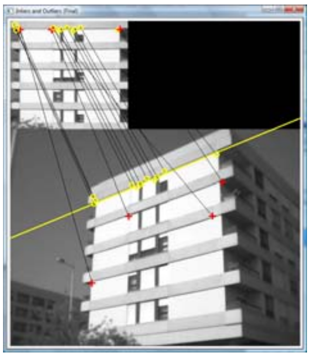

Figure 5. Extracted horizontal line with some outliers (represented by the symbol "+").

The collinearity constraint states that coplanar landmarks that are collinear in F_L or F_R should also be collinear in I_U. Feature points that do not verify this constraint are most likely outliers and must be removed from P_S (Figure 5).

This constraint is applied in the following way: a "line" of feature points in S_L or S_R is taken; these feature points are matched with points in S_U (the resulting subset is named L_U); if L_U has a minimum number (8) of matched points, a line is fitted to these points, using RANSAC; the resulting fitted line is considered valid if it fits a minimum number of points in L_U (40%). The points in L_U that do not belong to the line are marked as outliers and removed.

In order to fit a line to the points in L_U, it is necessary to define a threshold, T_1, for the distance between each point and the line, beyond which the point will be labeled as an outlier. Since the size of the facade in I_U depends on its distance to the camera, T_1 must be updated by a scaling factor; this factor is calculated as the ratio between two distances d_1 (measured between the two horizontal/vertical lines in F_R or F_L, having the largest sets of matched points in P_S) and d_2 (measured between the corresponding lines in I_U) as shown in Figure 6.

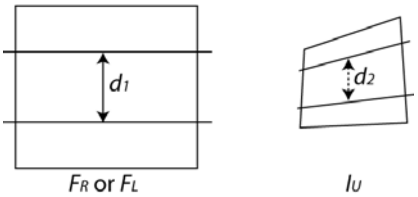

Figure 6. Distances d_1 and d_2 used to update the threshold T_1.

After the application of the collinearity constraint, some incorrect matches may still remain. To remove them, the sidedness constraint is applied.

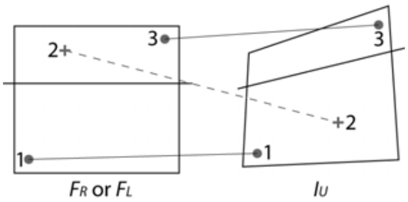

Figure 7. Verification of the sidedness constraint (point 2 violates the constraint).

Let us consider a line fitted to L_U and the corresponding line in F_L or F_R; a point A, in I_U, not belonging to L_U; the point B, in F_L or F_R, matched with A. The sidedness constraint states that A and B must be on the same side of the line fitted to L_U and the corresponding line in F_L or F_R (Figure 7). Ideally, a single mismatch would be sufficient to mark a point as an outlier. However, since there are occasionally some poorly fitted lines, a single mismatch may be insufficient to assume that conclusion. To overcome this problem, for each matched point, the number of times that sides match and mismatch, in both images, relatively to each fitted line, are totaled. If the first total is inferior to the double of the second one, then the point is marked as an outlier and removed.

4.1.3 Facade Delineation

If the number of matches remaining in P_S is above a threshold (8 pairs), the facade is declared as being present in I_U and the last step of the facade detection consists of delineating the contour of the facade in I_U. In order to delineate the facade on I_U and to enable the AR system to perform the CPI, a homography between the feature points of the reference image (I_L or I_R) and those of I_U is calculated, using RANSAC, and refined, using the Levenberg-Marquardt method [Har03]. Using the final homography, the corner points of the facade ROI (R_L or R_R), in the reference image are projected into I_U, thus delineating the ROI in this image. As a final validation step, the ROI shape must also pass a convexity test. Finally, the homography can also be used by the AR system to determine the initial camera pose.

Facade Tracking

The tracking subphase consists of the following steps: (a) capture of the next frame, I_{U+1}; (b) tracking of a selected set of feature points via sparse optical flow; (c) replacement of the lost tracked points and positional correction of the ones with high tracking error, if feasible, and (d) calculation of the homography between the feature points tracked in I_{U+1} and those detected previously in I_L or I_R..

4.2.1 Feature Point Tracking

After the facade has been successfully detected in the current frame, the system will try to track it in the next one. To do that, a number of feature points must be selected for tracking.

The selected points will be tracked by sparse optical flow between two consecutive frames (I_U and I_{U+1}). The optical flow implementation is based on the Bouguet sparse variant [Bou99] of the Lucas-Kanade optical flow algorithm [Luc81]. This implementation demands that the tracked points are good features to track which, in this case, means that they must be corner points. Unfortunately, not all SURF points are corner points so, during the setup phase, all the SURF feature points, coincident or near (within the feature size) to a Harris corner point [Har88], were labeled as good features to track. The position, in F_L or F_R, of the neighboring Harris corner point relative to the SURF point is stored, since the first one will be the point effectively tracked.

The subset P_K of matching pairs with good features to track is extracted from the current set of matching pairs P_S. From P_K, the system will select the points that are near the corners and to the center of the ROI. The maximum number of tracked points depends on the performance of the hardware platform.

The optical flow algorithm is applied to all the neighboring Harris Corner points of the selected SURF points in the current captured frame (I_U) and the next one (I_{U+1}). The algorithm returns the new estimated position for the tracked points in I_{U+1}.

4.2.2 Feature Point Replacement or Correction

Some points can be lost or have a high tracking error. The system tries to replace the lost points with new ones and correct the position of the points that have a high tracking error. In order to do both, the SURF detector is applied to a very small region centered on the estimated point position in I_{u+1}. The local descriptors of the detected points are matched with the single local descriptor of the replacement point or the point with high tracking error and if the match is successful, the point is replaced or its position is corrected, whatever the case may be.

If the number of remaining points is greater than 8, the homography is calculated by RANSAC, making it possible to delineate the ROI, and the tracking process will continue. Additionally, the AR system will be able to determine the camera pose in I_{U+1}. Otherwise, if the number of remaining points is insufficient, the tracking process is considered as failing and the system will try to detect the facade in the next frames, until the detection is successful.

5. SYSTEM EVALUATION

Since the proposed method depends on a large number of parameters, it would not be possible to evaluate its performance for each parameter combination. So, the parameters were selected through several experimental tests related to the particular result desired for each one. The global parameter setting was evaluated through a set of experiments described below.

The evaluation of the system was divided into two parts for evaluating separately the detection subphase only and the whole working phase. The evaluation was done on a computer running at 2.4 GHz using images with a 640×480 resolution.

In the first part, the system was forced to detect a facade in image sequences of four different buildings (Figure 8). These facades have some properties that can make the detection difficult, namely: landmarks non-coplanar with the dominant facade plane (all the facades, particularly no. 2), rounded lines (no. 4), small occlusions from trees, lampposts and traffic signs (nos. 1, 2 & 4). The used reference images were, for each sequence: 1.7 and 1.11, 2.2 and 2.9, 3.1 and 3.2, and, 4.5 and 4.8.

In all the four sequences, the facade was delineated with an average reprojection error inferior to 2 pixels, respectively: 1.98 pixels, 1.83 pixels, 1.16 pixels and 1.45 pixels. The reprojection error in each image was calculated as the average distance between the four reprojected corners and the real ones, these being manually selected.

The detection was robust even when the facade had a large number of visually similar landmarks (facades no.1 & no. 2) or non-coplanar ones (no. 2) and in the presence of small occlusions (nos. 1, 2 & 4). In spite of the large number of round visual landmarks and the existence of few horizontal/vertical lines on the facade the robustness of the detection (no. 4) did not decrease. The system was also able to detect a facade from viewpoints at larger distances than those used to capture the reference images (no. 3). As expected, the system is unable to robustly detect facades that do not have a dominant planar surface and those that have few visual landmarks or large occlusions.

In the second part, a captured video was used to evaluate the working phase (Figure 9). The building has a large planar surface with some non-coplanar landmarks (the balconies) and there is also a small occlusion (the tree). The video sequence has 200 frames. The facade was detected in the first frame and tracked, without failure, until the last frame. The two images in Figure 1 were used as reference images.

In order to evaluate the camera pose determination, an augmented wireframe box was registered to the tracked facade. By visual inspection of the augmented video, it is possible to conclude that the augmented box was satisfactorily registered with the facade.

In the setup phase, the identification of points belonging to the facade dominant plane took on average 334 ms to complete and the topological characterization was executed with an average time of 356 ms. During the tracking subphase, it was possible to achieve a real-time frame rate (superior to 25 fps). However, during the detection subphase, the frame rate dropped to a value between 3 to 10 fps (depending on the visual content of the frame). The drop on the frame rate was caused mainly by the feature point detection and by the two matching processes. On the other hand, the elimination of false matches by the application of the topological constraints took a median time of 52 ms to complete.

Using a smaller local descriptor (with 64 elements) and a faster but more limited version of SURF, the U-SURF [Bay06], it is possible to boost the detection frame rate. Unfortunately, in this case, the detection phase generates a higher reprojection error that has a negative impact in the overall robustness of the tracking system.

Many other factors should be considered for a more complete evaluation of the system, such as: the number of coplanar landmarks available, the camera pose of the reference images, the accuracy of the selection of the ROIs corners, the occlusions and lighting variations in the captured images, etc. However, such an evaluation is beyond the scope of this paper.

6. CONCLUSION

This paper has presented a tracking system that can detect and track a facade in a video sequence. The main contribution of this work is the use of parallax and topological information, namely the collinearity and sidedness constraints, to increase the overall robustness of the facade detection.

This tracking system can be very useful for an Anywhere Augmentation AR system, operating in an urban environment. To start operating, it only needs two reference images of a facade, taken on-the-spot.

Several types of facades can be detected and tracked, provided that they have a dominant planar surface and enough visual landmarks. Facades having many landmarks that are non-coplanar with its dominant plane or a large number of visually similar landmarks and those suffering from small occlusions were successfully handled.

Future work will be focused on two main developments: first the tracking of several facades and, as a generalization, the tracking of a building considered as an inter-related group of facades; second, the use of all the detected feature points in a facade, including the, presently discarded, non-coplanar ones. Additionally, a more complete and precise evaluation of the proposed tracking system will be done.

7. REFERENCES

[Art09] <http://www.hitl.washington.edu/artoolkit>

[Bar01] Barfield, W. and Caudell, T. Basic Concepts in Wearable Computers and Augmented Reality. Fundamentals of Wearable Computers and Augmented Reality, LEA Pub., pp.3-26, 2001.

[Bas08] Bastos, R. and Dias, J.M.S. Automatic Camera Pose Initialization, using Scale, Rotation and Luminance Invariant Natural Feature Tracking. WSCG' 08, pp.97-104, 2008.

[Bay06] Bay, H., Tuytelaars, T. and Gool, L.V. SURF: Speeded Up Robust Features. 9th ECCV, pp.404-417, 2006.

[Bei97] Beis, J.S. and Lowe, D.G.. Shape Indexing Using Approximate Nearest-Neighbor Search in High-Dimensional Spaces. IEEE CVPR, pp.1000-1006, 1997.

[Bou99] Bouguet, J.-Y., Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm, Intel Corp., Microprocessor Research Labs., 1999.

[Cip99] Cipolla, R., Drummond T. and Robertson, D. Camera Calibration from Vanishing Points in Images of Architectural Scenes. BMVC, pp.382-391, 1999.

[Fer03] Ferrari, V., Tuytelaars, T and Van Gool, L. Wide-baseline Multiple-view Correspondences. IEEE CVPR, pp.718-725, vol.1, 2003.

[Fis81] Fischler, M.A. and Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. ACM 24, pp.381-395, 1981.

[Fri77] Friedman, J., Bentley J. and Finkel, R. An Algorithm for Finding Best Matches in Logarithmic Expected Time. ACM Trans. Math. Software, vol.3, is.3, pp.209-226, 1977.

[Har88] Harris, C. and Stephens, M. A Combined Corner and Edge Detector. 4th Alvey Vision Conference, pp.147-151, 1988.

[Har03] Hartley, R. and Zisserman, A. Multiple View Geometry in Computer Vision (Second Edition). Cambridge University Press, 2003.

[Jia04] Jiang, B., Neuman U. and You, S. A Robust Tracking System for Outdoor Augmented Reality. IEEE VR, pp.27-31, 2004.

[Kat99] Kato, H. and Billinghurst, M. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. IWAR, pp.85-94, 1999.

[Lep05] Lepetit, V. and Fua, P. Monocular Model-based 3D Tracking of Rigid Objects: A Survey. Foundations and Trends® in Computer Graphics and Vision. Vol. 1, No.1, pp.1-89, 2005.

[Low03] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. IJCV, pp.91-110, 2003.

[Luc81] Lucas, B. and Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision, 7th IJCAI, pp.674-679, 1981.

[Rei06] Reitmayr, G. and Drummond, T.W. Going out: Robust Model-based Tracking for Outdoor Augmented Reality. ISMAR’06, pp.109-118, 2006.

[Rei07] Reitmayr, G. and Drummond, T.W. Initialization for Visual Tracking in Urban Environments. ISMAR’07, pp.161-172, 2007.

[Rol01] Rolland, J.P., Davis, L.D. and Baillot, Y. A Survey of Tracking Technology for Virtual Environments. Fundamentals of Wearable Computers and Augmented Reality, LEA Publishers, pp.67-112, 2001.

[Ros06] Rosten, E. and Drummond, T. Machine learning for high-speed corner detection. 9th ECCV, pp.430-443, 2006.

[Sim02] Simon, G. and Berger, M-O. Reconstructing While Registering: a Novel Approach for Markerless Augmented Reality. ISMAR’02, pp.285-293, 2002.

[Xu08] Xu, K., Chia, K.W. and Cheok, A.D. Real-time Camera Tracking for Marker-less and Unprepared Augmented Reality Environments. Image and Vision Computing, vol.26, is.5, pp.673-689, 2008.

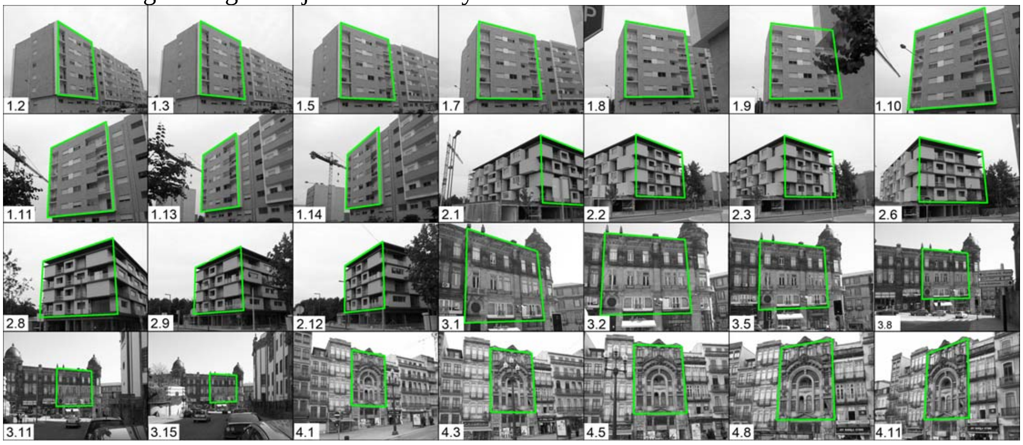

Figure 8. Samples of some of the image sequences used for the evaluation of the detection subphase.

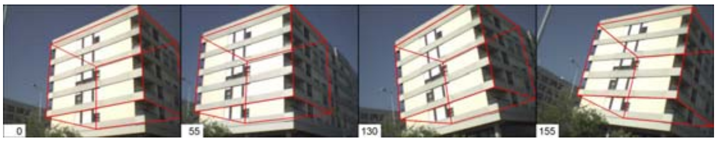

Figure 9. Frames of the augmented video of a facade used in the evaluation of the working phase.