

# A Method for Fast Initialization of Markerless Object Tracking

Jens Maiero<sup>†1</sup> and Andre Hinkenjann<sup>‡1</sup>

<sup>1</sup> University of Applied Sciences Bonn-Rhein-Sieg  
Sankt Augustin, Germany

---

## Abstract

*In Mixed Reality (MR) Environments, the user's view is augmented with virtual, artificial objects. To visualize virtual objects, the position and orientation of the user's view or the camera is needed. Tracking of the user's viewpoint is an essential area in MR applications, especially for interaction and navigation. In present systems, the initialization is often complex. For this reason, we introduce a new method for fast initialization of markerless object tracking. This method is based on Speed Up Robust Features and paradoxically on a traditional marker-based library. Most markerless tracking algorithms can be divided into two parts: an offline and an online stage. The focus of this paper is optimization of the offline stage, which is often time-consuming.*

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis —Tracking

---

## 1. Introduction

The idea of measuring arbitrary 2D objects based on common markers to track them will be described in this paper/poster. The developed feature-based markerless system tracks textured flat objects. This paper/poster is focused on a fast and uncomplicated initialization step. This step contains an object detection followed by the estimation of the relative position between the structured object and the user's view or the camera. In the offline stage the user has the choice between two provided methods:

1. Initialization with reference: In this case the textured object will be measured by a traditional marker-based system. Hence, the estimation between the real object and the user's viewpoint is exact.
2. Initialization without reference: If there is no reference or the reference cannot be placed next to the object, it is still possible to initialize the tracker. In this case, a marker will be generated automatically after the user has selected the real object. The estimation between the structured object and the user's viewpoint can be computed up to a scale

factor. However the object can also be augmented without any problems.

Both the result of the first and the second method are called keyframe. This keyframe, the extracted features and the camera position and orientation calculated by the marker system are stored together as one reference frame.

In the online stage, features (Speeded Up Robust Feature, SURF [BTV06]) are extracted every frame. With the detected features and the corresponding features from the reference frame the transformation (homography) between the current camera image and the reference frame can be calculated. Subsequently, the camera position and orientation can be computed by a marker-based library with the warped keyframe. In a final step, the environment can be augmented with virtual objects (compare figure 1d) or the tracked object can be used for interaction and navigation.

## 2. Related Work

There are two different categories of tracking algorithms based on computer vision: marker-based and markerless systems. With marker-based systems the camera position and orientation is calculated by a predefined marker [WLS08]. The second class, called markerless systems, computes the

---

<sup>†</sup> jens.maiero@h-brs.de

<sup>‡</sup> andre.hinkenjann@h-brs.de

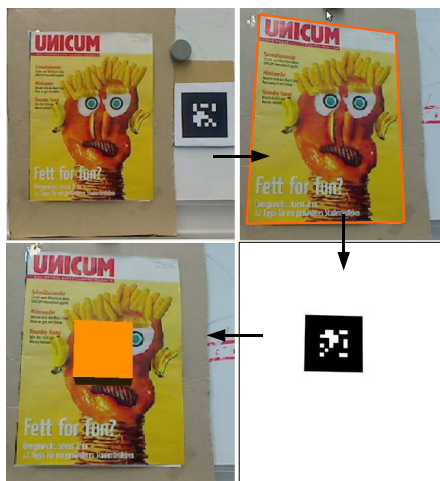
position between the camera/user and the real world without any reference, just using natural feature points (edges, lines or complete 3D-models). Markerless methods often require a manual a priori step, a correct model or a reference image for initialization [LF05, RD06]. This a priori step is often time-consuming or expensive.

### 3. Implementation

The method introduced in this paper/poster only needs an image of the scene with or without a common marker for initialization. Based on this reference image, position and orientation of the user's view relative to the real object are computed. In this section the algorithm will be described.

#### 3.1. Offline stage

In the offline stage, a reference frame has to be captured, which may, but does not have to include a marker. Additionally, the user has to select the 2D object in the current frame. Then SURF features will be extracted and stored automatically. For stable tracking, features need to have special properties (scale and rotation invariance). Also, the recognition rate of the features is important [CXH07]. An additional im-



**Figure 1:** Schematic view with reference: (a) keyframe, (b) feature-based object detection, (c) marker generation, (d) augmented scene

age to calculate the objects's pose by a marker based tracking library will also be generated automatically (see figure 1c). The marker is placed at the center of the selected object - this image is used for background processing and is not visible to the user.

#### 3.2. Online stage

In the online stage the SURF features of each frame are extracted. For matching of these features with the reference

frame, the user can select two methods. The first method is the Random Sample Consensus (RANSAC) and the second method is Least Median of Squares [FB87]. Both algorithms can solve an equation which contains a huge number of outliers. With the resulting corresponding features a transformation (homography) is calculated. The homography is an invertible transformation which describes a plane to plane mapping:  $x'_i = H \cdot x_i$ , where  $x'_i$  are the matched feature points in the current frame and  $x_i$  are the points of the reference frame, while  $H$  describes the desired transformation. The automatically generated marker image (see figure 1c) is now warped using the determined homography. The estimation of position and orientation between the object and the warped image is calculated by a conventional marker library.

### 4. Results

The major impact on performance is caused by feature extraction and matching of their descriptors. Therefore, two implementations were evaluated. On a single dualcore CPU (Intel Core2Duo T7200), the tracker achieves 4-5 fps depending on the number of extracted features. In the second implementation, a GPU-based library for feature detection has been used. This configuration improves performance up to 14-16 fps (nVidia GeForce 8800 GTX). Concerning accuracy and radius of movement, the tracker equals the common marker based tracking libraries, except some jitter and infrequent fails of position estimation. The introduced system is derived from a marker-based system, but in the online stage no marker is visible or needed (compare figure 1). The next step would be to not only track a selected object, but measure/track also unknown environments like other Structure from Motion methods do.

### References

- [BTV06] BAY H., TUYTELAARS T., VAN GOOL L.: Surf: Speeded up robust features. In *9th European Conference on Computer Vision* (Graz Austria, 2006).
- [CXH07] CHENG D., XIE S., HIMMERLE E.: Comparison of local descriptors for image registration of geometrically-complex 3d scenes. In *M2VIP 2007. 14th International Conference on* (2007), pp. 140–145.
- [FB87] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. 726–740.
- [LF05] LEPETIT V., FUA P.: Monocular model-based 3d tracking of rigid objects. *Found. Trends. Comput. Graph. Vis.* (2005), 1–89.
- [RD06] REITMAYR G., DRUMMOND T.: Going out: robust model-based tracking for outdoor augmented reality. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on* (Oct. 2006), pp. 109–118.
- [WLS08] WAGNER D., LANGLOTZ T., SCHMALSTIEG D.: Robust and unobtrusive marker tracking on mobile phones. In *ISMAR '08: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2008), IEEE Computer Society, pp. 121–124.