Object Detection and Pose Tracking for Augmented Reality: Recent Approaches

Hideaki Uchiyama, Eric Marchand

Object Detection and Pose Tracking for Augmented Reality: Recent Approaches

Hideaki Uchiyama*
Eric Marchand**

This paper introduces recent progress on techniques of object detection and pose tracking with a monocular camera for augmented reality applications. To visually merge a virtual object onto a real scene with geometrical consistency, a camera pose with respect to the scene needs to be computed. For this issue, many approaches have been proposed in the literature. In this paper, we classify and summarize the recent trends of the solutions as a survey.

Keywords: Augmented Reality (AR), Detection, Tracking, Pose Estimation, Survey

1. Introduction

Augmented reality (AR) is an increasingly-recognized paradigm of information visualization that merge a computer-generated virtual object onto a real scene to visually augment reality [25]. Azuma has summarized three requirements in AR applications as follows [5]:

(1) combines real and virtual,
(2) interactive in real time,
(3) registered in 3-D.

To achieve these requirements, numerous technologies on computer vision and image processing have made immense contributions. With a monocular camera, 3D structure of a real scene is sensed to enable overlaying virtual objects with geometrical consistency in real-time. Virtual objects are typically visualized with one of three devices: a head-mounted display (HMD), a handheld display [87] or a projector [11].

Recently, the algorithms of object detection and pose tracking have been incorporated in AR applications. In such systems, a number of objects are first registered on a database. When one of the registered objects is captured, it is recognized, tracked and localized to overlay its corresponding virtual object on it. For target objects, various types of size and shape can be used such as a small planar paper, a large room and so on. In the literature, many approaches of localization have been proposed.

In this paper, we classify and summarize recent techniques on monocular camera based object detection and pose tracking for AR applications as a survey. So far, some survey papers on AR have been published. Teichrieb, et al. focused on visual tracking based on model based tracking and structure from motion [75]. Ong, et al. introduced research issues for developing applications in manufacturing [55]. Krevelen and Poelman summarized general research issues on AR such as display, camera tracking and user interaction [82]. Our objective is similar with a survey of object tracking by Lepetit and Fua [40] and Yilmaz, et al. [89].

* INRIA Rennes, Lagadic, Hideaki.Uchiyama@inria.fr
** Université de Rennes 1, IRISA, Lagadic, Eric.Marchand@irisa.fr

2. Definition

We first define detection and tracking with a monocular camera in the context of AR. Also, we summarize the meaning of some terminology frequently used in the literature.

2.1 Detection Detection has several meanings depending on research issues. In the context of AR, detection is a process that localizes a target object in a captured image and computes a camera pose with respect to this object.

Detection is sometimes reworded with several terms. *initialization* is used because detection is an initial process for tracking [37]. Detection is also considered as *registration* that finds best transformation parameters for fitting two different models because pose estimation involves alignment between a captured image and a reference object [66]. Both initialization and registration are usually used when the number of target objects is only one.

If multiple target objects are registered on a database, object *identification* is included in object detection such that a target object is identified and localized in a captured image [66]. This process is also called *recognition* [74] or *retrieval* [62].

2.2 Tracking In contrast to detection that estimate a camera pose for one image, tracking is camera pose estimation in temporal image sequence. For seamless augmentation of target objects, tracking is an important issue.

Tracking is reworded according to the configuration of a camera and an object. If a camera is fixed and captures a moving object in a scene, this is called *outside-in* tracking because the object is observed from outside. If a camera is movable such as a handheld device and capture a fixed/moving object, this is called *inside-out* tracking.

A technical issue on both cases is common such that a relative camera pose with respect to a target object is computed.

The technical framework of tracking is divided into two types as follows.

2.2.1 Tracking by Detection Computing a camera pose with detection in every frame is named *tracking by detection* or *tracking by matching*. In other words, a camera pose is always computed by matching between an input image and a reference with a detection technique. No previous camera pose is used for the estimation of current camera pose. Pose tracking with randomized trees [41] or ferns [56] are examples of such approaches.

2.2.2 Tracking by Tracking Computing a camera pose using its previous pose is named *tracking by tracking*, *frame-by-frame tracking*, *frame-to-frame tracking* or *recursive tracking*. Most of these approaches are based on the minimization of camera displacement between two successive images. In the optimization, previous camera pose is used as an initial camera pose. In Section 5, we introduce the details of these approaches.

3. Pose Estimation

Before starting the detailed explanation of object detection and pose tracking, we introduce the overview of projection models and pose estimation [30]. A projection model is basically selected depending on the shape of a target object. The solution of pose estimation has been proposed in the literature known as direct linear transformation (DLT) or perspective-n-point (PnP) problem.

3.1 Planar Object The geometrical relationship between a reference planar object and an input image is described with a homography. Homography is a member of $\mathbb{SL}(3)$ (special liner group). With 2D-2D point correspondences, a homography matrix $\mathbf{H}$ that satisfies the following expression is computed:

$$\min_{\mathbf{H}} \sum_i \left\| \mathbf{x}_i^r - P\left(\mathbf{H}\tilde{\mathbf{x}}_i^t\right) \right\| \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

where $\mathbf{x}_i^r$ is an image coordinate of a point on a 2D reference, $\mathbf{x}_i^t$ is that of its corresponding point on a target, $\tilde{}$ means a homogeneous coordinate and $P(\cdot)$ returns an image coordinate computed from the dehomogenisation of an input homogeneous coordinate. Basically, this expression represents the minimization of reprojection error in an image. Note that rotation and translation components of a camera can be extracted from a homography for augmentation in 3D [30].

3.2 Non-Planar Object For a camera pose with respect to a non-planar object, following 6 degree-of-freedom (DoF) transformation matrix $\mathbf{M}$ is used;

$$\mathbf{M} = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array}\right) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2)$$

where $\mathbf{R} \in \mathbb{SO}(3)$ (special orthogonal group) and $\mathbf{t} \in \mathbb{R}^3$. Therefore, $\mathbf{M}$ is a member of $\mathbb{SE}(3)$ (special Euclidean group). With 2D-3D point correspondences, $\mathbf{M}$ that satisfies following expression is computed:

$$\min_{\mathbf{M}} \sum_i \left\| \mathbf{x}_i - P\left(\mathbf{A}\left(\mathbf{I}|0\right)\mathbf{M}\tilde{\mathbf{X}}_i\right) \right\| \cdots\cdots\cdots\cdots (3)$$

where $\mathbf{X}_i$ is a 3D coordinate of a point on a 3D reference, $\mathbf{x}_i$ is an image coordinate of its corresponding point on an image and $\mathbf{A}$ is a camera calibration matrix.

3.3 SLAM Simultaneous localization and mapping (SLAM) enable wide-area tracking in unknown environment [19]. Especially, Klein and Murray proposed parallel tracking and mapping (PTAM) as a novel SLAM framework specialized for AR applications [35]. In PTAM, the processes of tracking a camera pose and updating new maps are performed in different threads on CPU such that a camera pose is tracked in all incoming images on one thread while new map is updated with batch processing on the other thread only when needed.

In SLAM, both 3D map $\mathbf{X}_i$ and camera motion $\mathbf{M}_j$ that satisfy following expression are computed:

$$\min_{\mathbf{X}_i,\mathbf{M}_j} \sum_{i,j} \left\| \mathbf{x}_{ij} - P\left(\mathbf{A}\left(\mathbf{I}|0\right)\mathbf{M}_j\tilde{\mathbf{X}}_i\right) \right\| \cdots\cdots\cdots (4)$$

where $\mathbf{x}_{ij}$ is a tracked keypoint of i-th keypoint in j-th image.

The key technology in SLAM is structure from motion (SfM), bundle adjustment and extended kalman filter (EKF) [30]. With SfM, camera motion between two images is estimated by computing a fundamental matrix with keypoint correspondences, and then 3D map is estimated by triangulating the correspondences. Estimated camera motion and 3D map are finally optimized using bundle adjustment described in Equation 4. EKF is used for pose estimation with temporal data.

3.4 Robust Estimator In established keypoint correspondences between two objects, outliers may be included because of noise or other factors and degrade the accuracy of pose estimation. In practice, it is necessary to estimate a precise camera pose even in the presence of outliers. As a solution, a robust estimator is incorporated into pose estimation [73].

The simplest method is to minimize the least-mean-square (LMS) of the reprojection error for all correspondences. However, all correspondences including outliers are equally dealt and outliers cannot be removed with this method. Thus, outliers with huge error degrade the accuracy. M-estimator is considered as an alternative to LMS. In M-estimator, a weighting function according to error magnitude is incorporated to reduce the influence of outliers with huge error. Random sample consensus (RANSAC) is a method that iteratively and randomly samples some keypoints and compute a camera pose with LMS [24]. Random sampling is iteratively performed to find a camera pose that includes least outliers. PROSAC is considered as RANSAC with an efficient sampling strategy. In PROSAC, keypoints are sampled from most reliable correspondence such as a correspondence with minimum distance of descriptors [13]. Least-median-of-squares (LMedS) algorithm is to find a camera pose with the smallest value of the median of squared residuals for all correspondences.

4. Detection

In this section, we classify the detection approaches as follows: fiducial markers, natural keypoints and natural edges. Especially, we focus on the techniques of recognition and retrieval that can deal with multiple rigid objects.

4.1 Fiducial Marker 2D fiducial markers have been used as a reference object for a long time. Compared with natural feature based approaches, fiducial markers can stably be extracted because its color and shape are defined beforehand.

The world's first square marker system was developed by Rekimoto in 1998 [66]. The marker has a black bold frame for camera pose estimation. Inside the frame, there is a coded square black and white pattern for marker identification. This type of ID-embedded markers has been developed with different shapes and identification techniques [23] [51] [80] [84] [86].

ARToolKit is another type of marker system that needs the database of patterns included inside the frame [33]. As well as other square markers described above, the marker has a black bold frame. Any texture pattern can be inserted inside the frame and identified with simple template matching.

Even at the present, new types of fiducial markers are still being developed to relax the constraint of the marker shape and improve the recognition stability and scalability. Uchiyama and Saito developed random dot markers that use randomly scattered dots as fiducial [79]. The detection of the marker is based on keypoint matching with geometrical features described by LLAH [52]. Compared with square markers, the shape of a random dot marker is flexible and the robustness against occlusion is high. RUNE-Tag also uses dots as fiducial such that dots are distributed on a circle with a certain pattern coding [10]. This work evaluated the accuracy of pose estimation with four corners in ARToolKit and that with the centers of dots in RUNE-Tag. In the presence of noise and blur, camera pose computed from RUNE-Tag is more accurate than that of ARToolKit because RUNE-Tag can use more points for pose estimation.

4.2 Keypoint Natural keypoint based approaches have been well-investigated for a long time. The basic framework of these approaches is divided into extraction, description and matching.

4.2.1 Extraction Extraction consists in finding pixels which have different appearance from other pixels. Extraction is also reworded as *detection* such as keypoint detection [77].

Harris corner detector is the most famous detector that is compared with other detectors as a benchmark [29]. FAST corner detector select a pixel that has higher or lower value than neighbors with a machine learning technique [68]. Mair, et al. improved FAST by building an optimal decision tree [47]. In some approaches, a local extrema of scale space is extracted as a keypoint. Scale space is built with differences of Gaussian (DoG) in SIFT [45], approximated Laplacian filters in CenSurE [1] and fast Hessian in SURF [8].

4.2.2 Description A vector that describes the feature of a keypoint is computed for the comparison between two keypoints. The approaches of local texture based description are widely investigated and divided into two types: histogram of gradient and binary test.

Histogram of gradient is computed from quantizing gradients within a local region and putting into bins. In SIFT [45], a local region is divided into sub-regions and histogram of gradient is computed in each sub-region. This approach is used in several methods [2] [8] [85] that improved sampling strategy of gradient and computational efficiency.

A binary test is a simple comparison of the intensity of two pixels and produces a binary result that represents which pixel is brighter. Hundreds of binary tests are performed to compute a feature vector because a binary test is not enough discriminative. The main research issue of this approach is the efficient sampling of two pixels [12] [42] [56] [69].

Instead of using local texture, local geometry of keypoints is used to compute a feature vector [78]. In this approach, neighbor keypoints are first extracted and geometrical invariant is computed from neighbor keypoints [52]. Because this approach does not use local texture, this is applicable to keypoint matching for both rich and weak texture.

4.2.3 Matching To match keypoints between an input and a reference, feature vectors of keypoints in the reference are stored in a feature database. For each keypoint in the input, the most similar feature vector in the database is searched.

If a feature vector has large dimension such as 128 in SIFT [45], full searching is hardly performed in real-time. Instead, tree based searching is used as approximate nearest neighbor searching [4] [50]. The searching cost of this approach depends on the number of features stored in the database. Another type of searching is hashing that maps a feature vector to an integer index [18]. This approach is theoretically fast with $O(1)$ regardless of the size of the database, but is sensitive to small error. If a feature vector is described with binary string, full searching with a hamming distance is performed [12]. In keypoint matching with randomized trees [41] and random ferns [56], matching is treated as a classification problem.

4.3 Edge Compared with keypoint based approaches, edge based approaches are not much investigated. In edge based approaches, a geometrical feature is computed from edges.

Hagbi, et al. proposed a method for recognizing planar shapes and estimating a camera pose from the contour of the shapes [27]. In this method, adaptive thresholding based binarization is first applied to an image to extract shape contours. For each contour concavity, projective-invariant keypoints are extracted from bitangent lines, and projective invariant features are computed for recognition. Then, initial camera pose is estimated with those projective-invariant keypoints and refined by minimizing the reprojection error.

Donoser, et al. took a similar approach described

above [20]. To extract contours, they used MSER (maximally stable extremal regions) that extract blob regions [48]. Shape recognition is based on classification of contours that is similar with classification of keypoints [56]. Synthetic views of each template contour are first trained with random ferns. In run-time, each extracted contour is classified into one of the trained classes and recognized. Pose estimation is based on the extraction of projective-invariant keypoints from bitangent lines.

5. Tracking

Most of the available tracking techniques can be divided into two main classes: feature-based and model-based tracking. The former approach focuses on tracking 2D features such as geometrical primitives (point, segments, circles, etc.) or object contours (such as active contours) extracted from images. The latter one explicitly uses a 2D/3D model of the tracked objects. In this section, we introduce the details of these approaches.

5.1 Overview In model-based tracking, the model can be a 3D model leading mainly to a pose estimation process corresponding to a registration process between measures in the image and the forward projection of the 3D model [15] [22]. Within 2D model-based approaches, the object to be tracked can be represented by a descriptor. These descriptors can profile object histogram leading to mean shift like approaches [14] or point neighborhood leading to keypoint tracking by matching approaches [41] [45]. Such approaches are usually very robust to illumination variation, occlusions, etc.

As described in Section 5.2, it is also possible to consider that this 2D model is a reference image (or a template). In that case, the goal is to estimate the motion (or warp) between the current image and a reference template. An example of such approaches are differential tracking methods such as the LK [46] or others [6] [9] [28]. Those approaches are not limited to 2D motion estimation, considering, for example, the motion of a plane in the image space, it is indeed possible to estimate its motion back in the 3D real space.

5.2 Template Tracking A widely considered approach in AR is template tracking. In this context, a measure of the alignment between the reference image and the current image and its derivatives with respect to the motion (warp) parameters is used within a non-linear estimation process to estimate the current object motion.

5.2.1 Formulation Differential tracking is a class of approaches based on the optimization of an image registration function f. The goal is to estimate the displacement $\mathbf{p}$ of an image template I^* in a sequence of images $I_0..I_t$. In the case of a similarity function, the problem can be written as:

$$\widehat{\mathbf{p}}_t = \arg\min_{\mathbf{p}} \left(f(I^*, w(I_t, \mathbf{p}))\right) \cdots\cdots\cdots\cdots\cdots (5)$$

where we search the displacement $\widehat{\mathbf{p}}_t$ that maximizes the similarity between the template I^* and the warped current image I_t. For the purpose of clarity, the warping function w is here used in an abuse of notation to define the overall transformation of the image I by the parameters $\mathbf{p}$. Indeed, its correct formulation $w(\mathbf{x}, \mathbf{p})$ gives the function that moves a point $\mathbf{x}$ from the reference image to its coordinates in the current image.

5.2.2 Basic Alignment Function One essential choice remains the one of the alignment function f. One natural solution is to choose the function f as the standard sum of squared differences (SSD) of the pixel intensities between the reference image and the transformed current image [6] [46]:

$$\widehat{\mathbf{p}}_t = \arg\min_{\mathbf{p}} \left(SSD(I^*, w(I_t, \mathbf{p}))\right) \cdots\cdots\cdots\cdots (6)$$

$$= \arg\min_{\mathbf{p}} \sum_{\mathbf{x}\in ROI} \left(I^*(\mathbf{x}) - I_t(w(\mathbf{x}, \mathbf{p}))\right)^2 \cdots (7)$$

where the summation is computed on each point $\mathbf{x}$ of the reference template that is the region of interest (ROI) of the reference image.

5.2.3 Optimization The displacement $\widehat{\mathbf{p}}_t$ can be computed with the optimization processes. In the Lucas-Kanade (LK) algorithm, the alignment is based on a Gauss-Newton gradient descent non-linear optimization [7]. A non-linear expression is linearized by using a first order Taylor expansion. In ESM, an efficient second-order minimization was proposed [9]. ESM-Blur incorporated a blur model into the optimization of ESM [58].

5.2.4 Recent Approaches Template-based tracking approaches have been extended to region tracking [6] [9] [28]. Such approaches are considered as a standard in planar region tracking and localization for augmented reality application. However, some approaches are not effective in the case of illumination changes and occlusions [6] [46] . Several solutions have been proposed to add robustness toward those variations.

Some include the use of M-estimators to deal with occlusions or an illumination estimation [28] [72]. Others can consider local zero-mean normalized cross correlation (ZNCC) [32], or sum of conditional variance [67] to replace SSD. Another solution is to maximize the information shared between the reference image and the sequence by maximizing the Mutual Information (MI) function [70] [83]. MI has also proved to be robust to occlusions and illumination variations and can therefore be considered as a good alignment measure for tracking [21] [57]. Dame and Marchand proposed to use mutual information for measuring similarity [17]. This approach can deal with multimodality such that a reference is a 2D geographical map and an input image is its satellite image.

5.3 3D Model-Based Tracking Among the various approaches allowing to get the camera pose, model-based methods have shown growing performances in the past years. The information given by the knowledge of a template or 3D CAD model of the object allows to improve the tracking robustness.

5.3.1 Formulation When considering complex environment, the pose estimation method as defined in Equation 5 has to be reformulated. From a general point

of view, the problem is defined as the minimization of the error between contour points extracted from the images and the projection of a 3D model. This approach usually uses the distance between a point and projection of 3D lines [15] [22] [44] [81].

Assuming the camera parameters and an estimate of the pose are known, the CAD model is first projected into the image according to that pose. Formally, the projection of an edge L_i of the 3D model according to the pose ${}^c\mathbf{M}_w$ will be denoted by $E_i = L_i({}^c\mathbf{M}_w)$. Each projected edge E_i is sampled, giving a set of points $\{e_{i,j}\}$. From each sample point $e_{i,j}$, a search is performed along the edge normal to find strong gradients. In the approach of Comport, et al. [15], the point of maximum likelihood with regard to the initial point $e_{i,j}$ is selected from the exploration step. It is denoted by $e'_{i,j}$ in the following. A non linear optimisation approach is then used to estimate the camera pose which minimizes the errors between the selected points and the projected edges [15] [22] [44], that is:

$$\widehat{{}^c\mathbf{M}_w} = \arg\min_{{}^c\mathbf{M}_w} \sum_{i,j} d_\perp(E_i, e'_{i,j}) \cdots\cdots\cdots\cdots (8)$$

where $d_\perp(E_i, e'_{i,j}) = d_\perp(L_i({}^c\mathbf{M}_w), e'_{i,j})$ is the squared distance between the point $e'_{i,j}$ and the projection E_i of the linear segment L_i of the model.

5.3.2 Optimization The problem of minimizing this equation has been widely investigated in the past years [40] and different approaches have been proposed to address it. Most of these approaches can be divided into two categories:

- *Non-linear optimization methods* use non linear optimization techniques (Newton minimization, virtual-visual servoing,...) to find the pose which minimizes a given reprojection error between the model and the image edges [15] [22] [44]. The robustness of these methods has been improved by using robust estimation tools [3] [15] [22]. However, they can fail in case of large displacements or wrong edge matching, especially in cluttered environment. More constraints that uses texture along with contour information have been proposed [63]. It allows to introduce a spatial-temporal constraints in this problem.
- *Bayesian methods*, on the other hand, have been used to perform the same task by estimating the probability density associated to the pose. This can be achieved by Kalman filtering when the probability density function (p.d.f.) can be represented by an uni-modal Gaussian distribution. More recently, the improvement of computational performances has allowed to consider particle filtering approaches [34] [54] [64] [76]. Instead of going from the low level edges to retrieve the camera pose, particle filtering uses a set of hypothesis on the possible camera poses (the particles). The likelihood of each particle is then measured in the image. Since the space of all possible poses is large, the main issue is to keep a fair representation of the different modes of the state probability distribution while using few particles.

5.3.3 Recent Progress Another challenge is to consider a complete polygonal 3D model, in order to track complex object. The whole information from the geometrical shape of any kind of scene can be used and a heavy phase of a manual redesign of the model is avoided. Methods that rely on the use of the graphics process units (GPU) and of a 3D rendering engine to manage the projection of the model and to determine the visible and prominent edge from the rendered scene have been proposed [60] [65] [88]. An advantage of this technique is to implicitly handle auto occlusions.

6. AR Framework

To develop AR applications, it is necessary to select the methods for detecting and tracking objects. The main frameworks of visual tracking for AR are as follows.

Detection and Tracking Object detection is first applied to an image and then detected objects are tracked in next incoming images with a tracking by tracking algorithm [62] [85]. If tracking failed, object detection is again applied to an image for re-initialization.

Tracking by Detection Tracking by detection described in Section 2.2.1 can also be consider as a framework. In this framework, an object is tracked in all incoming images with a detection algorithm [41] [56]. In other words, an object is detected in every frame. Because only detection algorithm is used, a re-initialization problem does not need to be considered.

Unified Approach This approach has recently been proposed [74]. The basic idea of this approach is that a keypoint descriptor is used for both matching with a database and tracking between two successive images. First, a descriptor is matched with a descriptor database to detect an object. Once an object is detected, it is tracked by matching keypoints between two successive images with the descriptor. Because the descriptor is compressed into 1 dimensional integer vector, fast keypoint matching between two images is possible.

7. Additional Sensors

Recently, several sensors have been integrated with a camera to acquire various sensing data. In this section, we introduce the example uses of a accelerometer and a RGB-D camera.

7.1 Accelerometer These days, accelerometers have been integrated into many devices such as mobile phones. From accelerometers, the direction of gravity is estimated and used in several ways.

Kotake, et al. proved that pose estimation was simplified using inclination constraint because inclination component of the orientation was known [36] [37]. Lee, et al. rectified a captured image into a fronto-parallel

view as normalization of a local image patch for keypoint matching [39]. Kurz and Benhimane rectified feature descriptors to make them more discriminative instead of rectifying a local image patch [38].

7.2 RGB-D Camera Thanks to Kinect from Microsoft, RGB color images and depth images (RGB-D) are captured in real-time. Therefore, a depth camera is incorporated into AR applications.

Park, et al. refined a camera pose with edges extracted from depth images [59]. Newcombe, et al. proposed a SLAM framework with a depth camera that can densely mapped and track a camera pose only with depth images [53].

8. Dataset for Evaluation

To compare a proposed method with existing methods, a benchmarking dataset is necessary for fair comparison. In this section, we introduce some datasets for both detection and tracking.

8.1 Detection This issue is equivalent to wide-baseline feature matching because it is important to detect an object under perspective views.

8.1.1 Mikolajczyk Dataset[†] This dataset is the most widely used for the evaluation of feature matching between two views [49]. In this dataset, several sets of 6 scenes including rich texture or repetitive texture are prepared. They are captured under different conditions such as perspective transformation and blurring.

For each set, ground truth homography matrices are provided. Therefore, the ratio of false positive, false negative and the number of correspondences can be computed for evaluation.

8.1.2 CMU Grocery Dataset[††] This dataset is originally designed for evaluating 3D object recognition [31]. In this dataset, 620 images of 10 grocery items such as soda cans and boxes in kitchen environments are prepared. The images are captured under perspective views and different lighting conditions with clutter background.

Ground truth 2D segmentations are provided for 500 images. These segmented images can be used as reference objects. For 120 images, 6 DoF camera poses in rodrigues format are provided. Therefore, the accuracy of estimated camera poses can also be evaluated.

8.2 Tracking Tracking is normally evaluated with an image sequences with various camera movement. Recently, several datasets have been developed as follows.

8.2.1 Metaio Dataset[†††] This dataset is designed for the evaluation of planar template based tracking [43]. Four planar textures were captured with five camera motions to generate 40 different image sequences.

The evaluation criterion is the root-mean-square (RMS) distance of four points placed on the diagonal lines of a reference image. The distance is computed by projecting points on a test image onto those on a reference image with homography between two images.

The ground truth of homography for each image was acquired with fiducial markers and a precisely calibrated robot arm that still computed a camera pose under fast camera motion. In order to avoid the use of fiducial markers for tracking, the markers were interpolated with white. Image borders were also replaced with randomized borders.

In this dataset, the ground truth is not provided because it is necessary to prevent the optimization of parameters in a method for a specific image sequence. Therefore, the users need to submit the result to Metaio. Metaio gives the ratio of successfully tracked images. Note that the ground truth of some images is provided for pose initialization.

8.2.2 UCSB Dataset[††††] This dataset is also dedicated to the evaluation of planar template based tracking [26]. The scope of the dataset is almost similar with Metaio dataset, but more different types of image sequences are included. The dataset includes 6889 frames composed of 96 image sequences with six planar textures.

The ground truth of homography for each image and reconstructed camera poses are provided. Therefore, camera position can also be evaluated in addition to RMS distance as in Metaio dataset.

8.2.3 TrakMark Dataset[†5] Compared with the two datasets described above, this dataset focuses on the evaluation of visual tracking for a 3D scene [71]. Because a 3D model of a scene is provided in some image sequences, model based tracking can be evaluated [60]. In addition, SLAM can also be evaluated. This dataset includes image sequences of three different scenes: a small room, an outdoor environment and virtual space.

First scene was captured with a movie camera at the film studio of Japanese period drama. The ground truth provided in this sequence is the position and orientation of a camera measured with several sensors.

As second scene, an outdoor environment was captured with a handheld camera on a sunny day. In this dataset, some image sequences include the ground truth of 2D-3D point correspondences. The 3D coordinates of landmark points are measured with a total station (time-of-flight laser system) and their image coordinates in each image are manually selected.

Last scene is virtual space generated by reconstruction of a real scene. Test images are generated by simulating any camera motion with textured 3D models. Therefore, the ground truth is the transformation matrix from the world coordinate system to the camera coordinate system for each image. Also, textured 3D models are provided.

9. Conclusion

This paper presented state-of-the-art technologies of object detection and pose tracking for AR applications.

[†] http://www.robots.ox.ac.uk/~vgg/research/affine/
[††] http://www.cs.cmu.edu/~ehsiao/datasets.html
[†††] http://www.metaio.com/research/
[††††] http://ilab.cs.ucsb.edu/tracking_dataset_ijcv/
[†5] http://trakmark.net/

We first explained projection models and pose estimation depending on the shape of objects. Then, we classified and summarized the recent progress of detection and tracking techniques. Also, we introduced some evaluation datasets and evaluation procedures.

As described in this paper, the technologies of visual tracking for AR have actively been investigated. However, there is not one perfect method that outperforms others [26]. Because each method has advantages and limitations, it is important to select an appropriate method depending on applications.

Acknowledgements

We would like to thank Francisco Magalhaes and Joao Paulo Silva do Monte Lima from Federal University of Pernambuco for meaningful comments and discussion. A part of this work was supported by a grant-in-aid for the global center of excellence for high level global cooperation for leading edge platform on access spaces from the ministry of education, culture, sport, science, and technology in Japan.

References

(1) M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: center surround extremas for realtime feature detection and matching. In *ECCV*, pages 102–115, 2008.

(2) M. Ambai and Y. Yoshida. CARD: Compact and real-time descriptors. In *ICCV*, 2011.

(3) M. Armstrong and A. Zisserman. Robust object tracking. In *ACCV*, volume 1, pages 58–61, 1995.

(4) S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45:891–923, 1998.

(5) R. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.

(6) S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *CVPR*, pages 1090–1097, 2001.

(7) S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(1):221–255, 2004.

(8) H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *CVIU*, 110:346–359, 2008.

(9) S. Benhimane and E. Malis. Homography-based 2D visual tracking and servoing. *I. J. Robotic Research*, 26(7):661–676, 2007.

(10) F. Bergamasco, A. Albarelli, E. Rodolà, and A. Torsello. RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience. In *CVPR*, pages 113–120, 2011.

(11) O. Bimber and R. Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters, Ltd.

(12) M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *ECCV*, pages 778–792, 2010.

(13) O. Chum and J. Matas. Matching with PROSAC – progressive sample consensus. In *CVPR*, pages 220–226, 2005.

(14) D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24:603–619, 2002.

(15) A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *TVCG*, 12:615–628, 2006.

(16) N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

(17) A. Dame and É. Marchand. Accurate real-time tracking using mutual information. In *ISMAR*, pages 47–56, 2010.

(18) M. Datar, P. Indyk, N. Immorlica, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *S. Computational geometry*, pages 253–262, 2004.

(19) A. J. Davison, W. W. Mayol, and D. W. Murray. Real-time localisation and mapping with wearable active vision. In *ISMAR*, pages 18–27, 2003.

(20) M. Donoser, P. Kontschieder, and H. Bischof. Robust planar target tracking and pose estimation from a single concavity. In *ISMAR*, pages 9–15, 2011.

(21) N. Dowson and R. Bowden. Mutual information for lucas-kanade tracking (MILK): An inverse compositional formulation. *TPAMI*, 30(1):180–185, 2008.

(22) T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *TPAMI*, 24(7):932–946, 2002.

(23) M. Fiala. ARTag, a fiducial marker system using digital techniques. In *CVPR*, pages 590–596, 2005.

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

(25) B. Furht, editor. *Handbook of Augmented Reality*. Springer.

(26) S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *IJCV*, 94(3):335–360, 2011.

(27) N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst. Shape recognition and pose estimation for mobile augmented reality. *TVCG*, 17(10):1369–1379, 2011.

(28) G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *TPAMI*, 20(10):1025–1039, 1998.

(29) C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

(30) R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

(31) E. Hsiao, A. Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *CVPR*, pages 2653–2660, 2010.

(32) M. Irani and P. Anandan. Robust multi-sensor image alignment. In *ICCV*, pages 959–966, 1998.

(33) H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IWAR*, pages 85–94, 1999.

(34) G. Klein and D. Murray. Full-3D edge tracking with a particle filter. In *BMVC*, volume 3, pages 1119–1128, 2006.

(35) G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, pages 1–10, 2007.

(36) D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A hybrid and linear registration method utilizing inclination constraint. In *ISMAR*, pages 140–149, 2005.

(37) D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A fast initialization method for edge-based registration using an inclination constraint. In *ISMAR*, pages 239–248, 2007.

(38) D. Kurz and S. Benhimane. Gravity-aware handheld augmented reality. In *ISMAR*, pages 111–120, 2011.

(39) W. Lee, Y. Park, V. Lepetit, and W. Woo. Point-and-shoot for ubiquitous tagging on mobile phones. In *ISMAR*, pages 57–64, 2010.

(40) V. Lepetit and P. Fua. Monocular model-based 3D tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

(41) V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *TPAMI*, 28:1465–1479, 2006.

(42) S. Leutenegger, M. Chli, and R. Y. Siegwar. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, 2011.

(43) S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. Benchmarking template-based tracking algorithms. *VR*, 15:99–108, 2011.

(44) D. G. Lowe. Fitting parameterized three-dimensional models to images. *TPAMI*, 13(5):441–450, 1991.

(45) D. G. Lowe. Distinctive image features from scale-invariant

keypoints. *IJCV*, 60:91–110, 2004.

(46) B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *I. C, Artificial intelligence*, pages 674–679, 1981.

(47) E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *ECCV*, pages 183–196, 2010.

(48) J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002.

(49) K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 65:43–72, 2005.

(50) M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISSAPP*, pages 331–340, 2009.

(51) L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *ISMAR*, pages 27–36, 2002.

(52) T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *I. W. Document Analysis Systems*, pages 541–552, 2006.

(53) R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011.

(54) S. Nuske, J. Roberts, and G. Wyeth. Visual localisation in outdoor industrial building environments. In *ICRA*, pages 544–550, 2008.

(55) S. Ong, M. Yuan, and A. Nee. Augmented reality applications in manufacturing: a survey. *I. J. Production Research*, 46(10):2707–2742, 2008.

(56) M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *TPAMI*, 32:448–461, 2010.

(57) G. Panin and A. Knoll. Mutual information-based 3d object tracking. *IJCV*, 78(1):107–118, 2008.

(58) Y. Park, V. Lepetit, and W. Woo. Esm-blur: Handling & rendering blur in 3d tracking and augmentation. In *ISMAR*, pages 163–166, 2009.

(59) Y. Park, V. Lepetit, and W. Woo. Texture-less object tracking with online training using depth camera. In *ISMAR*, pages 121–126, 2011.

(60) A. Petit, G. Caron, H. Uchiyama, and E. Marchand. Evaluation of model based tracking with trakmark dataset. In *I. W. AR/MR Registration, Tracking and Benchmarking*, 2011.

(61) A. Petit, É. Marchand, and K. Kanani. Vision-based space autonomous rendezvous: A case study. In *IROS*, pages 619–624, 2011.

(62) J. Pilet and H. Saito. Virtually augmenting hundreds of real pictures: An approach based on learning, retrieval, and tracking. In *VR*, pages 71–78, 2010.

(63) M. Pressigout and E. Marchand. Real-time hybrid tracking using edge and texture information. *I. J. Robotics Research*, 26(7):689–713, 2007.

(64) M. Pupilli and A. Calway. Real-time camera tracking using known 3D models and a particle filter. In *ICPR*, pages 199–203, 2006.

(65) G. Reitmayr and T. W. Drummond. Going out: Robust model based tracking for outdoor augmented reality. In *ISMAR*, pages 109–118, 2006.

(66) J. Rekimoto. Matrix: a realtime object identification and registration method for augmented reality. In *Asia Pacific Computer Human Interaction*, pages 63–68, 1998.

(67) R. Richa, R. Sznitman, R. Taylor, and G. Hager. Visual tracking using the sum of conditional variance. In *IROS*, pages 2953–2958, 2011.

(68) E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *TPAMI*, 32:105–119, 2010.

(69) E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *ICCV*, 2011.

(70) C. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, 2001.

(71) F. Shibata, S. Ikeda, T. Kurata, and H. Uchiyama. An intermediate report of trakmark WG - international voluntary activities on establishing benchmark test schemes for AR/MR geometric registration and tracking methods. In *ISMAR*, pages 298–302, 2010.

(72) G. F. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In *CVPR*, 2007.

(73) C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Rev.*, 41:513–537, 1999.

(74) G. Takacs, V. Ch, R. S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Unified real-time tracking and recognition with rotation-invariant fast features. In *CVPR*, pages 934–941, 2010.

(75) V. Teichrieb, M. Lima, E. Lourenc, S. Bueno, J. Kelner, and I. H. F. Santos. A survey of online monocular markerless augmented reality. *I. J. Modeling and Simulation for the Petroleum Industry*, 1(1):1–7, 2007.

(76) C. Teulière, L. Eck, E. Marchand, and N. Guenard. 3D model-based tracking for UAV position control. In *IROS*, pages 1084–1089, 2010.

(77) T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

(78) H. Uchiyama and E. Marchand. Toward augmenting everything: Detecting and tracking geometrical features on planar objects. In *ISMAR*, pages 17–25, 2011.

(79) H. Uchiyama and H. Saito. Random dot markers. In *VR*, pages 35–38, 2011.

(80) S. Uchiyama, K. Takemoto, K. Satoh, H. Yamamoto, and H. Tamura. MR platform: a basic body on which mixed reality applications are built. In *ISMAR*, pages 246–253, 2002.

(81) L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3D camera tracking. In *ISMAR*, pages 48–57, 2004.

(82) D. W. F. van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *IJVR*, 9(2):1–20, 2010.

(83) P. Viola and W. Wells. Alignment by maximization of mutual information. *IJCV*, 24(2):137–154, 1997.

(84) D. Wagner, T. Langlotz, and D. Schmalstieg. Robust and unobtrusive marker tracking on mobile phones. In *ISMAR*, pages 121–124, 2008.

(85) D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *ISMAR*, pages 125–134, 2008.

(86) D. Wagner and D. Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. In *Computer Vision Winter Workshop*, pages 139–146, 2007.

(87) D. Wanger. *Handheld Augmented Reality*. PhD thesis, Graz University of Technology, 2007.

(88) H. Wuest, F. Wientapper, and D. Stricker. Adaptable model-based tracking using analysis-by-synthesis techniques. In *CAIP*, pages 20–27, 2007.

(89) A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38, 2006.