Hybrid Feature and Template Based Tracking for Augmented Reality Application

Gede Putra Kusuma, Fong Wee Teck and Li Yiqun

Visual Computing Department
Institute for Infocomm Research
1 Fusionopolis Way, Singapore 138632, Singapore
{igpknegara, wtfong, yqli}@i2r.a-star.edu.sg

Abstract. Visual tracking is the core technology that enables the vision-based augmented reality application. Recent contributions in visual tracking are dominated by template-based tracking approaches such as ESM due to its accuracy in estimating the camera pose. However, it is shown that the template-based tracking approach is less robust against large inter-frames displacements and image variations than the feature-based tracking. Therefore, we propose to combine the feature-based and template-based tracking into a hybrid tracking model to improve the overall tracking performance. The feature-based tracking is performed prior to the template-based tracking. The feature-based tracking estimates pose changes between frames using the tracked feature-points. The template-based tracking is then used to refine the estimated pose. As a result, the hybrid tracking approach is robust against large inter-frames displacements and image variations. It also accurately estimates the camera pose. Furthermore, we will show that the pose adjustment performed by the feature-based tracking reduces the number of iterations necessary for the ESM to refine the estimated pose.

1 Introduction

The vision-based augmented reality (AR) application has been popularized by the rise of the smart-phones. Visual tracking is the core technology that enables the vision-based augmentation. Visual tracking approaches can be roughly categorized into three main groups: feature-based, template-based and hybrid tracking approaches. The feature-based tracking approaches track a set of local features across image sequence. Tracking local features in image sequence can be done by detection or by frame-to-frame tracking. The local features can be detected from salient regions of a reference image. The detected features can then be matched to the features of the input image. Features of both images that provide the best matching scores are considered as the matching pairs. The camera pose is then estimated from the pairs of features using Levenberg-Marquardt algorithm [1] or RANSAC homography [2]. Popular choice of feature extraction methods include SIFT [3] and SURF [4]. The feature locations can also be tracked between frames using a feature tracker such as Kanade-Lucas-Tomasi (KLT) feature tracker [5]. The KLT uses spatial intensity information of

the image to direct the search for the feature location that yields the best match. The KLT has been known to efficiently and robustly track the feature locations in an image sequence. The feature-based tracking approaches rely heavily on feature detection and cannot be applied to texture-less objects that do not contain reliable features to track. The camera pose estimated from a set of feature pairs is usually not so accurate. However, the feature-based tracking approach is usually robust to handle large inter-frames displacements and image variations.

The estimated camera pose can be refined through an iterative numerical method. As the projective geometry of the pinhole image formation process is non-linear, second-order numerical methods are typically used to minimize the mean squared errors between the values predicted by a pose model and those obtained from measurements. For feature-based tracking, standard methods, such as the Levenberg-Marquardt, perform well. However, its accuracy depends on the number of feature points available. This can be partially addressed by using all the available pixel intensity information directly. The Efficient Second-order Minimization (ESM) [6] is one such method, which uses the current pose to warp the current image back to the reference image, so as to minimize the resultant pixel intensity errors. As this method depends on the image gradients, it can work well for well-textured images. Fong et al. [7] added sub-grids to exclude sub-regions within the larger reference image with little textures from computation. Furthermore, the sub-grids are also used to estimate the change in illumination, as well as to handle occlusion. The template-based tracking approaches make use of image intensity information to estimate the camera pose. The pose is estimated by adjusting parameters of a pose model that minimizes an error measure based on image brightness. The pose estimated by the template-based tracking approach is usually more accurate than the one estimated by the feature-based tracking. However, the template-based approach is easier to lose track for large inter-frames displacements and image variations.

The feature-based and template-based tracking approaches are complementary in nature. Therefore, it is logical to combine both feature-based and template-based tracking approaches into a hybrid tracking approach to obtain performance gain. Their specific strengths can be exploited to improve the overall tracking performance. Combining the feature-based and template-based tracking approaches into a hybrid tracking model is not a new idea. A hybrid tracking for augmented reality application has been proposed by Ladikos et al. [8]. They adopted the extended version of the ESM as the template-based tracking and Harris points as the feature-based tracking. It is observed that the template-based tracking works well for small inter-frames displacements and feature-based tracking can deal with larger inter-frames displacements. Therefore, they implemented an adaptive switching strategy between the template-based and feature-based tracking depending on the scene condition. The template-based tracking is used as the default tracking. While, the feature-based tracking is designed to act as a backup to recover the pose in the event that the template-based tracking fails. They avoided running the template-based and feature-based tracking at the

same time, because they believed that the combined approach would increase the computational burden and the inter-frames displacements.

Our idea of combining the feature-based and template-based tracking approaches is quite the opposite of theirs. In this contribution, we propose a hybrid tracking approach that combines the feature-based and template-based tracking approaches, where the feature-based tracking is performed prior to the template-based tracking. The feature-based tracking estimates pose changes between frames using the tracked feature-points. The template-based tracking then refines the estimated pose. The feature-based tracking is used as a coarse estimate of the pose for large inter-frames displacements. In order words, the feature-based tracking has reduced the inter-frames displacements for the template-based tracking to handle. As a result, the hybrid tracking approach is robust against large inter-frames displacements and image variations. It also accurately estimates the camera pose. One may expect that it will result in slower processing speed due to larger computation burden. But, according to our experiments, the hybrid tracking is faster than the template-based tracking alone. The pose adjustment performed by the feature-based tracking has reduced the number of iterations necessary for the template-based tracking to refine the estimated pose. Therefore, the hybrid tracking approach is benefited by the strength of both feature-based and template-based tracking approaches; also it is faster than the template-based tracking approach.

2 Combining Feature-Based and Template-Based Tracking

2.1 Overview

Fig. 1 shows the overview of the proposed hybrid tracking approach. It consists of three different parts: initialization, feature-based tracking and template-based tracking. The initialization is only done once at the starting point of a tracking process. The initialization process starts with keypoints detection and features extraction. The detected keypoints are used to recognize the object contained in the image. The recognition process is repeated until a match is found (1). The initial pose is then estimated from the keypoints pairs between the input and reference images. A set of keypoints are selected for tracking and also the template image is generated from the input image using the initial estimated pose.

The location of the keypoints in the subsequent input images are tracked using Kanade-Lucas-Tomasi (KLT) feature tracker [5]. The average displacement of the keypoints between successive images is calculated to judge the movement level. If large movement is detected, the pose will be updated by the RANSAC Homography [2] using the keypoints pairs; otherwise, this step will be skipped (2). The feature-based tracking is stopped when the number of tracked keypoints falls below a certain threshold (3). The re-initialization will be invoked when a tracking failure is detected.

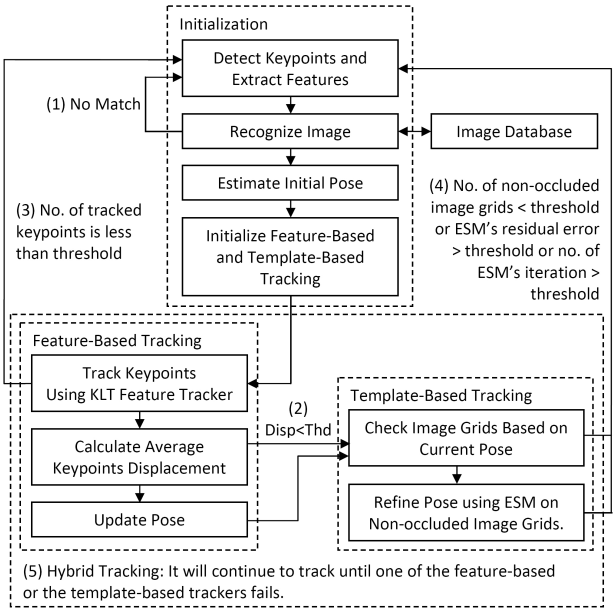

Fig. 1. Overview of the proposed hybrid tracking approach.

The template-based tracking is performed to refine the estimated pose. It divides the template image into sub-grids of smaller patches. Each sub-grid is checked individually for occlusion in the current image based on the estimated pose. A sub-grid is occluded when its average pixel error is above a defined threshold. If there are enough non-occluded sub-grids available, the pose is refined using the ESM performed on the pixels of the non-occluded sub-grids. The template-based tracking is stopped if the number of non-occluded sub-grids is lower than threshold, the residual error of the ESM is larger than threshold, or the number of ESM's iteration is larger than threshold (4).

The hybrid tracking approach combines the template-based and feature-based tracking. It will stop tracking when any of the feature-based or the template-based trackers fails (5). The details of each part will be described in the followings.

2.2 Initialization

The initialization process is started by detecting keypoints and extracting feature descriptors on the input image. Here, we adopt the Fast-Hessian keypoints detector and SURF descriptor proposed by Bay et al. [4] because of its speed and robustness. Object recognition is then performed based on the detected features to obtain the identity of the object contained in the image. We employ the appearance-based object recognition method based on weighted longest increas-

ing subsequence proposed by Kusuma et al. [9]. The features are matched to the database of images features indexed by the k-d tree data structure [10]. The matching features are then subjected to a geometric validation method based on the longest increasing subsequence [11]. The identity of the object is defined by the class in the database that has the highest similarity score to the input image.

Once the identity of the object has been known, the tracking process can then be initialized. The objective of the tracking process is to estimate a pose matrix that positions the reference coordinate frame to the current coordinate frame. The pose matrix is composed by a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{T} \in \mathbb{R}^3$. The only available information for the pose estimation are a reference image of the target object $\mathbf{I}_0$ and an image of the current scene $\mathbf{I}_t$. Assuming a planar target, the reference and current images are related by a homography. The pose matrix can then be estimated by decomposing the homography matrix using the SVD-based homography decomposition [12]. The homography decomposition requires the intrinsic parameters of the camera, which can be obtained through camera calibration [13].

The initial homography is estimated from a set of feature pairs using RANSAC homography [2]. The SURF features of the input image are matched to the features of the selected reference image. The matched features are sorted according to the matching distances. Only, the top 100 feature pairs are used to calculate the initial homography. Let us define $W(\mathbf{H})$ as a warping function of an image based on a homography $\mathbf{H}$ and $\mathbf{H}_{ij}$ is a homography that transform the i^{th} image to the j^{th} image. The initial homography $\mathbf{H}_{01}$ will transform the reference image $\mathbf{I}_0$ to the initial image $\mathbf{I}_1$ by

$$\mathbf{I}_1 = W(\mathbf{H}_{01})\,\mathbf{I}_0\ . \tag{1}$$

The feature-based tracking requires a set of initial keypoints to start the tracking. The initial keypoints are selected from the reference keypoints that are consistent with the initial homography.

Meanwhile, the template-based tracking requires an initial estimate of the pose and an image template to start the tracking. There are two ways to set the template image: set template image from a reference image or generate template image from the initial image. The reference image is an image of the target object that is prepared prior to the tracking process. Since the reference image is readily available, therefore the accuracy of the initial pose estimation is less critical for the template image set from a reference image. However, it requires more memory to store the reference image. This approach also requires that the reference image to be similar to the expected image quality and conditions of the target object in the scene. These two requirements are impractical for AR application on the phone. The phone has limited memory space and the camera quality between phones varies. Furthermore, the illumination variation alters the appearance of the target object.

In this contribution, the template image is generated from the initial image. The initial image is warped to the template using the inverse of the initial pose,

such as

$$\mathbf{I}_0^* = W\left(\mathbf{H}_{01}^{-1}\right)\mathbf{I}_1 \ . \tag{2}$$

This approach does not require storage space. Furthermore, it is easier to track the template because the appearance of the template image is similar to the appearance of the target object in the scene. However, the accuracy of the initial pose estimation is now more critical. It also adds to the processing load to generate the template image. The initial pose estimation and the template image generation can be done in a different thread to avoid jittery display. Hence, a more expensive method can be used to accurately estimate the initial pose.

The accuracy of the initial pose can be improved by implementing a loop of pose tuning process. A set of SURF features is extracted from the generated template image. These features are matched to the features of the reference image. The homography between the template and reference images is estimated from the keypoint pairs similar to the approach described above. This homography is then used to update the initial homography. This tuning process is performed until an acceptable level of pose accuracy or the maximum number of loop is achieved. The pose accuracy is measured from the number of consistent keypoints to the estimated homography. Since the initialization is done in a separate thread, the user will not really notice it. It will just increase the latency to start tracking, which is usually only a fraction of a second.

2.3 Feature-Based Tracking

The locations of the initial keypoints are tracked throughout the image sequence using Kanade-Lucas-Tomasi (KLT) feature tracker [5]. The KLT has been known to efficiently and robustly track the feature locations in an image sequence. It uses spatial intensity information of the image to direct the search for the feature location that yields the best match. In the current implementation, we employ the pyramidal implementation of the Lucas-Kanade feature tracker [14]. It calculates the optical flow for sparse features using the iterative version of the Lucas-Kanade method in pyramids.

The average displacement of the keypoints between successive images is calculated to judge the movement level. If large movement is detected, the pose will be updated using the between-frames homography. The between-frames homography is estimated from the between-frames keypoint pairs using RANSAC Homography. The current homography is then defined by multiplying the between-frames homography to the previous homography, such as

$$\mathbf{H}_{0t} = \mathbf{H}_{t-1t} * \mathbf{H}_{t-2t-1} * ...\mathbf{H}_{12} * \mathbf{H}_{01} \ . \tag{3}$$

The homography will only be updated by the feature-based tracking if the average displacement of the keypoints between successive images is larger than threshold. The feature-based tracking may lose some of the keypoints during tracking due to occlusions or image variations. The set of tracked keypoints is constantly updated to the last set of trackable keypoints. The feature-based tracking is stopped when the number of tracked keypoints falls below threshold.

The re-initialization will be invoked when the feature-based tracking failure is detected.

2.4 Template-Based Tracking

The template-based tracking is performed to fine-tune the estimated homography. The homography is refined using the efficient second order minimization (ESM) such that to minimize the pixel error between the current and template images. The ESM is developed as an iterative method for second-order minimization of image errors. Compared to the widely-used iterative methods, such as Gauss-Newton and Levenberg-Marquardt, the ESM is shown to have a higher convergence rate [6]. In general, ESM requires a model of the transformation of the image of a surface due to camera motion. The current image is transformed to match the template image using the current camera pose. With a suitable parameterization of small motion about the current camera pose, the ESM can iteratively converge to the camera pose that gives the minimal image error between the template and warped images. For the tracking of planar surfaces, homography is used to correctly model the perspective transformations due to camera motion. As a large number of pixels are used in an efficient manner, the end result is highly accurate and jitter-free pose estimation.

In this contribution, we adopt the modified version of ESM proposed by Fong et al. [7] as the template-based tracking. The template image is divided into sub-grids. The average image gradient within each sub-grid is computed, and only those sub-grids where the gradient is above 10 grey levels per pixel are used in tracking. The sub-grids filtering are performed based in the image gradient as it is used to construct the Jacobian matrices used in the ESM. Experimental observation shows that image regions with low gradients do not contribute additional information for ESM convergence, and in certain cases causes convergence towards the wrong minima.

In the current implementation, the template image is resized to 160x120 pixels and then divided into 12x9 sub-grids with the size of 12x12 pixels per grid. Some of the remaining pixels around the image borders are ignored. The formulation of ESM tracking in terms of sub-grids improves the tolerance to illumination changes and partial occlusion. For illumination changes, both the mean and standard deviation of the pixel intensities within each warped sub-grid is adjusted to match those of the corresponding template sub-grid. As the initial pose is close to the actual pose, the compensation required for illumination changes can be directly computed using the warped and template sub-grids. As both the transformation and illumination models are accurate, the occlusion of a sub-grid can be simply detected when its average pixel error is above a predefined threshold, which is set to 25 in the current implementation. The average pixel error is defined as the mean of absolute differences between corresponding pixels in gray scale.

The ESM pose estimation is then performed based on a set of pixels from the non-occluded sub-grids. The average pixel error of the set of pixels is calculated

based on the ESM's estimated pose. The template-based tracking failure is detected when the average pixel error is larger than 10. The tracking will also be stopped when the number of non-occluded sub-grids is less than 12 grids or the number of ESM's iterations is larger than 10. The modified ESM [7] is observed to converge within five iterations.

2.5 Hybrid Tracking

The feature-based and template-based tracking are combined to form a hybrid tracking. Based on our observation, the feature-based tracking is more robust against large inter-frames displacements and image variations than the template-based tracking. On the other hand, the template-based tracking is more accurately estimate the pose than the feature-based tracking. Therefore, the feature-based tracking is performed prior to the template-based tracking. The feature-based tracking updates the homography based on the between-frames keypoint pairs, such as

$$\mathbf{H}_{0t^*} = \mathbf{H}_{t-1t^*} * \mathbf{H}_{0t-1} \ , \tag{4}$$

where $\mathbf{H}_{0t-1}$ is the homography up to the previous frame, $\mathbf{H}_{t-1t^*}$ is the between-frames homography estimated by the feature-based tracking, and $\mathbf{H}_{0t^*}$ is the updated homography. The hybrid tracking approach also checks the average displacement of the keypoints between successive images in order to reduce the computation burden. The homography will only be updated by the feature-based tracking if the average displacement is larger than threshold.

The template-based tracking is then performed to refine the estimated homography, such as

$$\mathbf{H}_{0t} = ESM\left(\mathbf{H}_{0t^*}\right) \ , \tag{5}$$

where $ESM\left(\mathbf{H}\right)$ indicates the ESM iterative refining process. It is also observed that refining the homography from $\mathbf{H}_{0t^*}$ requires less number of iteration than directly from $\mathbf{H}_{0t-1}$. Thus, the hybrid tracking approach is not only combining the strength of both feature-based and template-based tracking approaches; it is also reducing the number of ESM iteration necessary to refine the pose.

3 Experiments

3.1 Methodology

To evaluate the performance of the proposed hybrid tracking approach, we perform experiments on public benchmarking datasets presented by Lieberknecht et al. [15]. We perform our experiments on the normal textured targets: a car (Isetta) and a cityscape (Philadelphia), as shown in Fig. 2. There are five image sequences of different motion patterns for each target. The motion patterns include "Angle", "Range", "Fast Far", "Fast Close", and "Illumination". Therefore, there are ten different image sequences for the experiments and each image sequence contains 1200 image frames.

Fig. 2. The target objects: Isetta (car) and Philadelphia (cityscape).

The image frame is resized to 320x240 pixels for the feature-based tracking and 160x120 pixels for the template-based tracking. The feature-based, template-based and hybrid tracking are performed separately on the image sequences. Their performances are measured by the ratio of the successfully tracked images in the sequence. The root mean square (RMS) of the pixel distance is also defined for each image in the sequence based on four reference points. The RMS of the pixel distance (err) for an image frame (i) in the sequence is computed as:

$$err_i = \sqrt{\frac{1}{4}\sum_{j=1}^{4} \left\|\mathbf{x}_j - \mathbf{x}_j^*\right\|^2}\ , \tag{6}$$

where $\mathbf{x}_j$ and $\mathbf{x}_j^*$ are the reference point in the current frame and the ground truth of the reference point respectively. All frames with $err_i > 10$ pixels are considered to be unsuccessfully tracked. Hence, the ratio of the successfully tracked images is calculated based on the filtered results.

Additional experiments are also performed to compare between the performances of the template-based tracking based on a template image set from a reference image and a template image generated from the initial image. The first approach is adopted by Ladikos et al. [8] where a reference image is used as a template image. These additional experiments are also performed on the same datasets and evaluated using the same performance measure as described above.

3.2 Experimental Results

Table 1 shows the ratio of successfully tracked images for the feature-based, template-based and hybrid tracking approaches without err_i thresholding. These results are based on the performances of maintaining tracking throughout the image sequence without considering their accuracies. The results show that the feature-based tracking outperforms the template-based tracking on all image sequences. They also show that the hybrid-based tracking is benefited by the performance of the feature-based tracking. The results of the hybrid tracking are slightly lower than the results of the feature-based tracking. Overall, the feature-based and hybrid tracking achieved much higher ratio of successfully tracked images (without err_i thresholding) than the template-based tracking.

Table 1. Ratio of successfully tracked images (without err_i thresholding): feature-based, template-based, and hybrid tracking.

Target	Image Sequence	Feature-Based Tracking	Template-Based Tracking	Hybrid Tracking
Isetta	Angle	100.00%	79.67%	100.00%
	Range	99.92%	55.33%	99.92%
	Fast Far	78.00%	7.33%	78.00%
	Fast Close	93.42%	81.00%	93.25%
	Illumination	99.75%	96.58%	99.75%
Philadelphia	Angle	100.00%	71.50%	100.00%
	Range	99.58%	62.17%	91.83%
	Fast Far	94.58%	14.67%	74.92%
	Fast Close	81.08%	49.83%	81.83%
	Illumination	99.92%	88.17%	99.92%
Average		94.63%	60.63%	91.94%

Meanwhile, Table 2 shows the ratio of successfully tracked images for feature-based, template-based and hybrid tracking approaches with err_i thresholding. These results are based on the performances of maintaining tracking throughout the image sequence as well as their accuracies in estimating the camera pose. Image frames with err_i more than 10 pixels are considered to be unsuccessfully tracked and removed from the performance calculation. We have also added the baseline ESM performance extracted from [15] in the last column.

From the results in Table 1 and Table 2, we can clearly see significant drops in the ratio of successfully tracked images for the feature-based template. These indicate that the pose estimated by the feature-based tracking is not accurate, even though it can maintain its tracking throughout the image sequence. On the other hand, smaller drops are observed for the feature-based and hybrid tracking approaches. They accurately estimate the camera pose in the majority of the tracked images. It also shown in Table 2 that the ratios of successfully tracked images for hybrid tracking are much higher than the feature-based and template-based tracking approaches on all image sequences. These results prove that the hybrid tracking has combined the strengths of the feature-based and template-based tracking approaches.

The performance of our hybrid tracking is better than the baseline ESM in 6 out of 10 image sequences. On average, our hybrid tracking approach achieved 3.14% higher ratio of successfully tracked images. Unfortunately, our template-based tracking is not as good as the baseline ESM. This opens up a room for further improvement.

We also show the box-plots of the err_i for Isetta and Philadelphia datasets in Figs. 3 and 4 respectively. Please note that these box-plots are based on the ratio of successfully tracked images with err_i thresholding shown in Table 2. Figs. 3 and 4 show that, out of the successfully tracked images, the template-based tracking estimates the camera pose more accurately than the feature-based

Table 2. Ratio of successfully tracked images (with $err_i < 10$ pixels): feature-based tracking, template-based tracking, hybrid tracking, and baseline ESM.

Target	Image Sequence	Feature-Based Tracking	Template-Based Tracking	Hybrid Tracking	Baseline ESM
Isetta	Angle	0.40%	43.40%	99.10%	95.42%
	Range	0.80%	17.60%	97.00%	77.75%
	Fast Far	5.30%	5.40%	54.30%	7.50%
	Fast Close	8.10%	29.10%	38.40%	67.08%
	Illumination	9.80%	91.50%	99.60%	76.75%
Philadelphia	Angle	12.80%	35.90%	96.00%	99.58%
	Range	4.90%	34.70%	66.40%	99.00%
	Fast Far	5.20%	6.20%	56.00%	15.67%
	Fast Close	14.60%	36.60%	40.90%	86.75%
	Illumination	13.20%	50.70%	99.90%	90.67%
Average		7.51%	35.11%	74.76%	71.62%

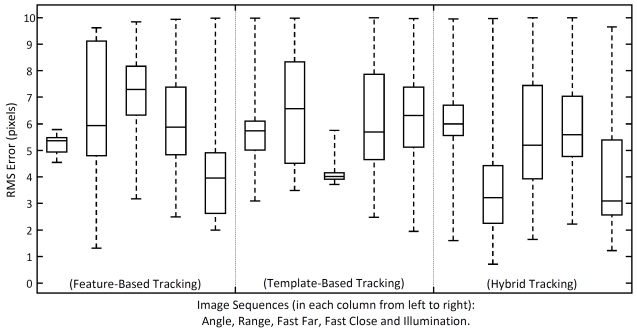

Fig. 3. Box-plots of RMS error (err_i) for Isetta dataset.

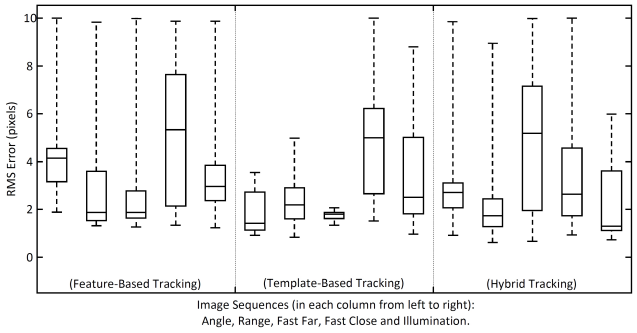

Fig. 4. Box-plots of RMS error (err_i) for Philadelphia dataset.

Table 3. Ratio of successfully tracked images (with $err_i < 10$ pixels) for template-based and hybrid tracking: template image set from reference image vs. template image generated from initial image.

		Template-Based Tracking		**Hybrid Tracking**	
Target	**Image Sequence**	**Reference Image**	**Initial Image**	**Reference Image**	**Initial Image**
Isetta	Angle	20.10%	43.40%	64.70%	99.10%
	Range	20.80%	17.60%	12.20%	97.00%
	Fast Far	5.00%	5.40%	5.20%	54.30%
	Fast Close	20.20%	29.10%	29.20%	38.40%
	Illumination	49.40%	91.50%	82.60%	99.60%
Philadelphia	Angle	8.20%	35.90%	29.00%	96.00%
	Range	7.80%	34.70%	7.90%	66.40%
	Fast Far	2.20%	6.20%	5.20%	56.00%
	Fast Close	18.10%	36.60%	46.30%	40.90%
	Illumination	16.10%	50.70%	24.20%	99.90%
Average		16.79%	35.11%	30.65%	74.76%

tracking. The distributions of err_i values for the hybrid tracking is within the range of the err_i values for the template-based and feature-based tracking, even though the number of tracked images for the hybrid tracking is much higher than the template-based and feature-based tracking.

We have also compared the tracking performance of the template image set from a reference image and template image generated from the initial image for the template-based and hybrid tracking. Their ratios of successfully tracked images with err_i thresholding are shown in Table 3. Please refer to Section 2.2 for the details of the template image initialization. These results clearly show that template image generated from the initial image is easier to track than the template image set from the reference image. The template image generated from the initial image is similar to the appearance of the target object in the current scene. Also, generating template image from the initial image is more practical for mobile augmented reality application since it does not require storage space to store the reference images.

All of the experiments are performed on a personal computer powered by Intel® Core™ i7-2600 3.40 GHz processor, 3.16 GB of RAM, and Windows XP operating system. The entire codes are written in C++ programming language. The average frame rates for the feature-based, template-based and hybrid tracking approaches are 64.0 fps, 46.5 fps and 59.0 fps respectively. The feature-based tracking is faster than the template-based tracking. There is a slight drop in frame rates for the hybrid tracking compared to the feature-based tracking. However, the hybrid tracking is faster than the template-based tracking alone. It proves that the pose adjustment performed by the feature-based tracking has

Table 4. Average number of ESM iteration for template-based and hybrid tracking.

Target	Image Sequence	Template-Based Tracking	Hybrid Tracking
Isetta	Angle	3.84	2.58
	Range	3.46	2.96
	Fast Far	3.84	2.88
	Fast Close	4.16	3.00
	Illumination	3.78	2.02
Philadelphia	Angle	3.44	2.02
	Range	3.24	2.02
	Fast Far	3.78	2.02
	Fast Close	3.54	2.02
	Illumination	3.52	2.02
Mean of Average		3.66	2.35

reduced the number of iteration necessary for the template-based tracking to refine the estimated pose.

To back-up our claim, we also measured the average number of ESM iteration for the template-based and hybrid tracking for all image sequences. Their average numbers of ESM iteration are shown in Table 4. It clearly shows that the hybrid tracking requires less average number of ESM iteration than the template-based tracking for all image sequences.

The obvious drawback of the proposed hybrid tracking compared to the feature-based or template-based tracking is its higher memory footprint. The memory requirements for feature-based, template-based and hybrid tracking during run-time are summarized in Table 5. It is shown that the feature-based tracking requires slightly more memory than the template-based tracking; and the hybrid tracking requires memory slightly less than the combined memory requirements for the feature-based and template-based tracking. The memory requirement for the hybrid tracking is mainly for storing the template and current frames required by the template-based tracking and the image pyramids of two successive frames required by the feature-based tracking. This memory requirement is still very low compared to the memory space available in a smartphone. Current smart-phones available in the market are usually equipped with at least 1GB of RAM. As a demo, we have created a sample mobile AR application using the proposed hybrid tracking method. It can run on iPad Air at around 60 fps.[1]

[1] Demo video can be found at `http://scholar-milk.i2r.a-star.edu.sg/demo/imev14_videos.html`

Table 5. Run-time memory requirements for feature-based, template-based and hybrid tracking.

Methods	Run-Time Memory
Feature-Based Tracking	1.85 MB
Template-Based Tracking	1.66 MB
Hybrid Tracking	3.25 MB

4 Conclusion

We have presented in this paper a hybrid tracking approach that combines the feature-based and template-based tracking approaches. The feature-based tracking is performed prior to the template-based tracking. The feature-based tracking estimates the pose changes between successive frames using feature points; while the template-based tracking refines the estimated pose. It has been shown that the proposed hybrid tracking has combined the strength of both feature-based and template-based tracking approaches. It is robust against large inter-frames displacements and image variations and also accurately estimates the camera pose. The pose adjustment performed by the feature-based tracking has also been shown to reduce the number of iteration required by the template-based tracking to refine the estimated pose. Therefore, the hybrid tracking is faster than the template-based tracking alone.

References

1. Lourakis, M.I.A.: homest: A c/c++ library for robust, non-linear homography estimation. [web page] `http://www.ics.forth.gr/~lourakis/homest/` (Jul. 2006) [Accessed on 17 Dec. 2011].
2. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. of Computer Vision **60** (2004) 91–110
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proc. European Conf. on Computer Vision. (2006) 404–417
5. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conf. on Computer Vision and Pattern Recognition. (1994) 593 – 600
6. Benhimane, S., Malis, E.: Homography-based 2d visual tracking and servoing. Int. J. of Robotics Research **26** (2007) 661–676
7. Fong, W.T., Ong, S.K., Nee, A.Y.C.: Computer vision centric hybrid tracking for augmented reality in outdoor urban environments. In: Proc. of the Int. Conf. on Virtual Reality Continuum and Its Applications in Industry. VRCAI '09, New York, NY, USA, ACM (2009) 185–190
8. Ladikos, A., Benhimane, S., Navab, N.: A real-time tracking system combining template-based and feature-based approaches. In: VISAPP. (2007) 325–332
9. Kusuma, G.P., Szabo, A., Li, Y., Lee, J.A.: Appearance-based object recognition using weighted longest increasing subsequence. In: Proc. Int. Conf. on Pattern Recognition, Tsukuba, Japan (2012) 3668–3671

10. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18** (1975)
11. Fredman, M.L.: On computing the length of longest increasing subsequences. Discrete Mathematics **11** (1975) 29–35
12. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision: From Images to Geometric Models. SpringerVerlag (2003)
13. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 1330–1334
14. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker. Intel Corporation, Microprocessor Research Labs (2000)
15. Lieberknecht, S., Benhimane, S., Meier, P., Navab, N.: A dataset and evaluation methodology for template-based tracking algorithms. In: Proc. of IEEE Int. Symp. on Mixed and Augmented Reality. ISMAR '09, Washington, DC, USA, IEEE Computer Society (2009) 145–151