TRACKING PLANAR-TEXTURED OBJECTS
On the Way to Transport Objects in Packaging Industry by Throwing and Catching

Naeem Akhter
Vienna University of Technology, Vienna, Austria

Keywords: Efficient, Flexible, Robust, Pose tracking, Planar-textured objects.

Abstract: In manufacturing systems transportation of objects can be optimized by throwing and catching them mechanically between work stations. There is a need to track thrown objects using visual sensors. Up to now only ball-shaped objects were tracked under controlled environment, where no orientation had to be considered. This work extends the task of object tracking to cuboid textured objects considering industrial environment. Indeed, tracking objects with respect to the robotics tasks to be achieved in a not too restricted environment remains an open issue. Thus, this work deals with efficient, flexible, and robust estimation of the object's pose.

1 INTRODUCTION

Transporting objects within production systems by throwing and catching is a new approach that aims at future prospect. The basic advantages of the throw-catch approach are: high speeds are possible, flexibility can be achieved, and fewer resources are required (Frank et al., 2008). Functionally the approach is divided into four subtasks. A throwing machine throws object towards a target where it needs to be grasped. Since flight of such an object is non-deterministic in general, a catching mechanism has to be located before object reaches the target. This is achieved by predicting current trajectory. Visual sensors are employed to find trajectory. At each measurement interception is updated, consequently catching mechanism moves to the predicted interception. Figure 1 provides schematic description of the approach. Scope of this work is restricted to trajectory measurement.

There are four classes in logistic chains in which the throw-catch approach can be realized (Frank et al., 2009). Table 1 summarizes these. So far feasibility of the approach is tested with spherical objects in fact tennis ball (Barteit et al., 2009). Objects of different nature behave aerodynamically differently. Their appearance also varies. Among these spherical objects are least vulnerable to change their trajectory, appearance, and grasping. Non-spherical objects change their view and hence project differently on image plane. A change in their view results into a change in their area across airstream that changes its aerodynamic behavior. Moreover, grasping of the object would also need its orientation information. Therefore, task of trajectory measurements in that case becomes task of pose tracking.

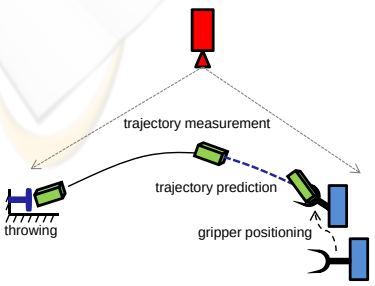

Figure 1: A schematic description of the throw-catch approach.

Table 1: Classification of throwing tasks and objects.

Object	workpiece	packaging	assembly	food
Shape	ball	cuboid	axial symmetric	irregular
Function	sorting	transportation	separation	commissioning

Work done so far in the throw-catch approach is not only based on simplified nature of object but also the environment. While trajectory measurement, background is supposed to be static, a high contrast between object and background is set, and diffused lighting is assumed. This is contradiction to the claim of flexible transportation. In production environments, pose tracking may present challenging situations. The first is dynamic background. This is due to a number of activities going on in parallel. These include staff movement, motion of assembly units and

Akhter N. (2012).
TRACKING PLANAR-TEXTURED OBJECTS - On the Way to Transport Objects in Packaging Industry by Throwing and Catching.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 316-321
DOI: 10.5220/0003731503160321

accessories, displacement of tools, chairs, tables etc., and objects of multirouting. The second is appearance of object and background may change. A new appearance may be introduced. Maintaining high contrast and constraint of diffused light impose unnecessary conditions. The third is object can be partially occluded. This could be due to moving entities of the scene, on the way object may partially leave field of view, or illumination may blur part of the object. Finally, interframe displacement may be large. This could be due to high speed of object, low temporal resolution of camera, or frame drop.

Indeed, flexibility not only depends on overcoming the challenging situations but also on adaptivity to change in object and background nature. This can be achieved by decoupling foreground from background. An approach that provides 6-DoF pose without offline learning and/or image segmentation becomes important. Rather than independently performing object detection and pose estimation, integration of these tasks may reduce computational cost. A single camera and exploiting minimum structural detail of object may reduce information density. In brief, objective of this research is not only to handle challenges of tracking in industry, but also to make the throw-catch approach flexible in real sense. Scope of the work is restricted to packaging industry. Packaging objects are labeled with text and graphics such as product information, brand name, brand logo etc. Therefore, it is reasonable to assume they have sufficient texture.

Rest of the paper is divided into four sections. Section 2 reviews state of the art in pose tracking, based on the outcome, a hybrid approach is presented in Section 3. Evaluation of the approach is described in Section 4. Finally, Section 5 concludes the paper.

2 POSE TRACKING

On a broad level, approaches of pose tracking for textured-planar targets can be divided into two groups: pose tracking by detection and pose tracking by modeling. The first type of approaches (Bjorkman and Kragic, 2004; Ekvall et al., 2005; Lepetit et al., 2004) build 2D-3D mapping using training data consisting of several views of the target. The constructed mapping is then used to find pose of a given 2D target image, which makes them rigid to learned scenarios. Scenes in which targets are easy to detect are assumed (Azad, 2009). Although suitable for large interframe displacement as strong prior on the pose is not required, they are less accurate and more computationally intensive than the second type of approaches (Lepetit and Fua, 2005).

The second type of approaches pre-assumes a 3D model of target. They require a strong prior on the pose to iteratively evolve to actual pose. Typically, they recover pose by first establishing 2-3D feature correspondence and then solving for the pose using a pose estimation technique. Based on the type of feature, they are further divided into template based and keypoint based approaches. Template based approaches (Mei et al., 2008; Baker and Matthews, 2004; Jurie and Dhome, 2002) estimate pose of a reference template by minimizing an error measure based on image brightness. In general, they work under diffused lighting, no occlusion, and small interframe displacement (Ladikos et al., 2009; Lepetit and Fua, 2005). Keypoint based approaches (Ladikos et al., 2009; Vacchetti et al., 2004b; Collet et al., 2009) exploit local appearance of targets. They work opposite to template based approaches but relatively computationally expensive (Lepetit and Fua, 2005). As they require sophisticated feature model of complete object, they are less flexible to adapt new object. A common problem with the second type of approaches is pose drift due to error accumulation over long sequences (Lepetit and Fua, 2005).

Approaches (Choi and Christensen, 2010; Ladikos et al., 2009; Rosten and Drummond, 2005; Vacchetti et al., 2004a) also exist that combine more than one type of approaches with intention to increase accuracy and/or achieve robustness. There is none that simultaneously addresses large interframe displacement and flexibility to adapt change in scene. Moreover, rather than fusing they work either by feeding output of one approach to second approach or by switching between the two approaches. The proposed approach intrinsically assimilate template based and keypoint based tracking due to their complementary role in achieving the goal. In contrast to estimate pose from pre-learned samples, deformation in the template is used. In place of intensity, point based error measure is defined to find the deformation. Tasks of detection and pose estimation are performed simultaneously without imposing constraints on background. The approach intrinsically delays pose drift.

3 FUSING POINT AND TEMPLATE INFORMATION

To fuse point information into template, template based tracking introduced by Mei et al. (Mei et al., 2008) is chosen. This is for its high accuracy and better convergence. Pseudocode of the algorithm with point information incorporated is given in figure 2.

Let I_1 be a reference image of a monocular sequence I_k, $k = 1...K$, such that a region I_{ref} (reference patch) of this contains projection of the planar target. Given an approximate transformation $\tilde{T}$ consisting of rigid motion (rotation $\tilde{R}$, translation $\tilde{t}$) in terms of camera motion, features F_{ref} extracted from I_{ref}, and a set of thresholds, the algorithm returns the actual T. Theoretically, it is equivalent to map I_{ref} to desired region defined by T in the current image I_k that minimizes sum of square distance (SSD) over all feature points.

```
Input: I1, Iref, Ik, Fref, T̃, thresholds(maxIter, num, err)
Output: T

Iter = 0

While(iter < maxIter)

    Compute R̃, t̃
    H = R̃ + t̃n_d'
    Icur = definePatch(Ik, Iref, H)
    Fcur = extractFeatures(Icur)
    matches = matchFeatures(Fref, Fcur)
    removeOutliers(matches)

    x = -2J⁺D(0)

    if  ||x|| < err
        T = T̃
        break
    else
        T̃ = T(x)T̃
    end

end
```

Figure 2: Pseudocode of the ESM algorithm fused with point information.

Algorithm starts by computing transformation of the target in image plane using homography H associated to $\tilde{T}$. Such that

$$\tilde{T} = \begin{bmatrix} \tilde{R} & \tilde{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

$$H = (\tilde{R} + \tilde{t} n_d^{'}) \quad (2)$$

$$p = \pi(w(H(\tilde{T})))\pi^{-1}(p^*) \quad (3)$$

where n_d is a vector defined as $\frac{n}{d}$ consisting of normal n and distance d of the target plane from camera. w is a warping function that defines a coordinate transformation between points on a unit plane (normalized plane). π is a projection function that defines projection of a point on the unit plane to image plane. Practically, this is to find the new position p in the current image of a pixel p^* in the reference image. With this a patch I_{cur} (current patch) in the current image is defined. This leads to four benefits. One region to search the target in image is confined. Second explicit detection or segmentation of the target is avoided which saves computation on run time. Third likelihood of correspondence with background is reduced. Fourth background is intrinsically ignored which in turn makes background dynamics irrelevant.

In the next step features F_{cur} are extracted from the defined patch. The Scale Invariant Feature Transform (SIFT) is an approach for extracting local features that are reasonably invariant to scaling, translation, rotation, illumination changes, image noise, affine distortion, occlusion, and viewpoint change (Sangle et al., 2011). Further motivation comes from its use in real-time tracking on mobile phones (Wagner et al., 2008). Therefore, this work uses SIFT. Extracted features are then matched with the F_{ref} using K-d tree. False correspondence is avoided by first removing points with multiple correspondences. Then further removing whose Euclidian distance and slope exceeds a specific range. Based on the number of features, empirically determined two strategies are employed. If the number exceeds 40, Gaussian distribution is assumed and the range is defined by equation 4. Otherwise, it is defined by equation 5.

$$Mean\ \{slope,\ distance\} \pm 1.5 \times its\ Standard\ deviation \quad (4)$$

$$Median\ \{slope,\ distance\} \pm 0.66 \times Median\ \{slope,\ distance\} \quad (5)$$

Once outliers are removed, cost of matching is then computed between the corresponding points. Let the corresponding points are $\{l_{i,j}\}$ and $\{m_{i,j}\}$ in the reference and current patches respectively. Let D_q be the distance between q^{th} pair of corresponding points, the cost is defined as:

$$\forall i \in 1,2,...,q \quad D_i = l_i - m_i \quad (6)$$

If the SSD value of vector D approaches to zero, the estimated pose becomes equal to the actual pose. Tracking jumps to the next image. Otherwise, there is a need to update $\tilde{T}$. Let the update is denoted by $T(x)$. Where x is a parameter vector that consists of coefficients of base elements: three for translation B_1-B_3 and three for rotation B_4-B_6 such that

$$T(x) = exp(\sum_{i=1}^{6} x_i B_i) \quad (7)$$

$$B_1 = \begin{bmatrix} 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix} B_4 = \begin{bmatrix} 0&0&0&0 \\ 0&0&-1&0 \\ 0&1&0&0 \\ 0&0&0&0 \end{bmatrix}$$
$$B_2 = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix} B_5 = \begin{bmatrix} 0&0&1&0 \\ 0&0&0&0 \\ -1&0&0&0 \\ 0&0&0&0 \end{bmatrix} \quad (8)$$
$$B_3 = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \end{bmatrix} B_6 = \begin{bmatrix} 0&-1&0&0 \\ 1&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix}$$

More precisely, the problem of pose estimation is to minimize the cost of matching which in terms of the parameter vector can be described as

$$\forall i \in 1,2,...,q \ D_i(x) = \pi(w(H(T(x)\tilde{T})))\pi^{-1}(l_i) - m_i \quad (9)$$

Minimizing this expression is a nonlinear optimization task. Let cost function $D(x)$ be the vector $[D_1(x)\ D_2(x)\ D_3(x)\ ...\ D_q(x)]'$ that corresponds to the distance over all points at the given parameter vector x. By the second order approximation of $D(x)$ about $x = \mathbf{0}$ using Taylor series and simplification (Mei et al., 2008)

$$D(x) \approx D(\mathbf{0}) + \frac{1}{2}Jx \tag{10}$$

$$J = J_\pi J_w J_H J_T \tag{11}$$

where J is jacobian of the D with respect to the x. In contrast to the original Jacobian which is composed of jacobians of each of image J_I, image projection function J_π, image warping function J_w, homography J_H, and transformation J_T. In this work, it is composed of the other four except J_I. This is to reduce non-linearity in the cost function. In the former case there are two factors that introduce non-linearity in the cost function. First corresponds to non-linear projection and second corresponds to intensity information. In fact pixel values are essentially un-related to pixel coordinates (Baker and Matthews, 2004), therefore, J_I is ignored. This allows using fewer details, regions in spite of the complete reference patch. Moreover, impact of non-linearity should be reduced as the cost function is better defined. Outcome of testing with simulated sequence confirms this. Expression of each of the jacobian for each feature point l_i normalized to the unit plane is

$$J_\pi = \nabla_P \pi(P)|_{P=l_i} \tag{12}$$

$$J_w = \nabla_H (w(H))(P)|_{H=H(0)=I} \tag{13}$$

$$J_H = \nabla_T H(\tilde{T})^{-1} H(T\tilde{T})|_{T=T(0)=I} \tag{14}$$

$$J_T = \nabla_x T(x)|_{x=0} \tag{15}$$

Solution to the problem lies in finding a parameter vector x_0 such that $D(x_0) = \mathbf{0}$. This is obtained by iteratively solving the cost function such that for a vector $x = x_0$

$$\nabla D(x)|_{x=x_0} = \mathbf{0} \tag{16}$$

At each iteration an updated x is calculated as follows

$$x = -2J^+ D(\mathbf{0}) \tag{17}$$

where $D(\mathbf{0})$ is the cost at $x = \mathbf{0}$, and J^+ means pseudo-inverse of J. Once convergence ($\|x\| < err$) is achieved in the current image, the optimal transformation T_k between the reference I_1 and the current image I_k is obtained. The algorithms finishes with this image and restarts with the next image I_{k+1}. Let $T_k(x_0)$ be the relative transformation between the last two consecutive frames I_{k-1} and I_k. Pose estimation starts in the next image with the following approximation

$$\tilde{T}(k+1) = T_k(x_0) T_k \tag{18}$$

Tracking continues till the last image I_K is reached and a total transformation T_K without pose drift is found.

4 RESULTS AND DISCUSSION

Evaluation of the proposed approach is made using both the simulated and real sequences. In the first case, it is made with reference to the Mei et al. using the same simulated sequence on which the referenced approach was tested. Figure 3 shows three images of the simulated sequence.

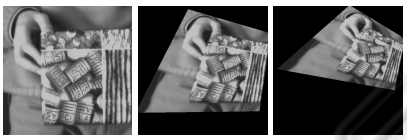

Figure 3: Images 1, 50, and 100 respectively in the simulated sequence.

Figure 5 shows how do the two approaches behave on average bases, in terms of absolute translational error, absolute rotational error, and number of iterations elapsed to converge, with the increase in interframe displacement. The interframe displacement is increased by skipping multiple images at regular intervals from the original sequence. Started by skipping alternate images and ending with two images in the sequence. Figure 4 elaborates skipping procedure. One can see by fusing point information into the pure template based tracking both the errors remain more than half below. The errors oscillate in the beginning for the reason of small baseline effect while stabilizes later. In the case of number of iterations, although difference between the two is small in the beginning but immediately that is after skipping just two images raises dramatically. The proposed approach showed consistent behavior. The most considerable fact is that the referenced approach fails tracking beyond 10 number of images skipped. This is due to its reliance on strong prior on the pose.

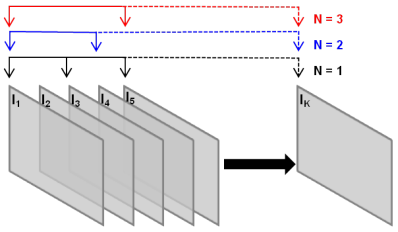

Figure 4: Three instances of skipping alternate images. N corresponds to the number of images skipped. Arrow points to the selected image.

In the second case, the approach is tested with real sequences. These sequences consist of flight of ten cubical objects thrown horizontally across the principal axis of camera. For each object 50 sequences are collected. They are thrown at a distance of $1.6_{\pm 0.45}$ m from camera with their largest plane exposed to the

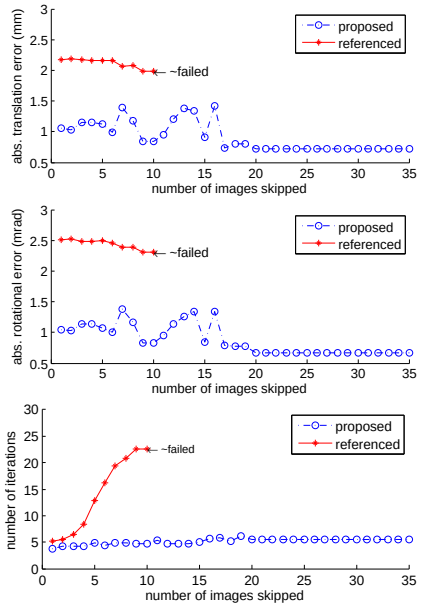

Figure 5: Comparison based on interframe displacement.

camera. Figure 6 shows the planes. Their sizes and number of features extracted from each plane at this distance are given in Table 2. Horizontal field of view at this distance is 1.2 *m*. Before leaving field of view, they lie at 1.44±0.45 and 0.07±0.21 *m* from camera along Z and Y axis respectively. Another calibrated camera is used to find the distances and normal to the plane using stereo vision. The range of estimated rotation along each of the X, Y, and Z axis is -37.93 to 35.08, -30.50 to 50.90, and -15.50 to 44.68 degrees respectively.

Figure 6: Planes of the objects and their assigned names. Top to bottom followed by left to right: (a) Daisy, (b) Garment, (c) Donuts, (d) Monster, (e) Rice, (f) Chicken, (g) China, (h) Biscuit, (i) Juice, (j) and Bravo.

Evaluation is made using the methodology introduced in (Lieberknecht et al., 2010). Tracking error is defined as root mean square (RMS) distance between estimated corner p of the plane and its ground truth p^* which is generated manually. Such that

Table 2: Sizes and feature amount of the planes.

Plane	**Number of features**	**Size** (mm × mm)
Daisy	155	300 × 160
Garment	177	
Donuts	99	285 × 120
Monster	130	
Rice	185	250 × 160
Chicken	114	
China	188	200 × 175
Biscuit	107	
Juice	69	240 × 115
Bravo	102	

$$tracking\ error = \sqrt{\frac{1}{4}\sum_{i=1}^{4} \|p_i - p_i^*\|^2} \qquad (19)$$

Figure 7(a) shows tracking error on average and extreme bases. For each plane the average is taken per image over all the 50 throws. One can see the approach performs equally well in all the cases except Juice. This is due its much lower amount of texture (number of features) relative to the rest. A common trend among all the planes is that the error increases with the increase in image number. This is partially due to error accumulation and partially due to loss in features. The loss is due to throwing objects in front of the camera. So in the subsequent frames fine texture loses. Figure 7(b) shows decrease in feature amount on average bases with the increase in image number. The interframe displacement was large enough that in no case the referenced approach is able to track the plane.

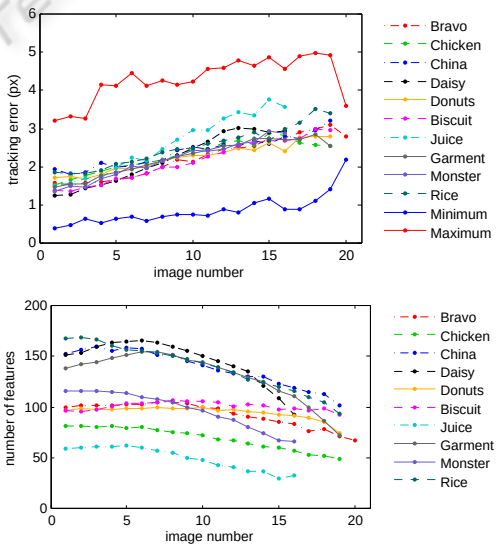

Figure 7: Testing with real sequences: (a) tracking error, (b) feature decay.

Sequences are acquired without diffused lighting. Figure 8 confirms this. Having success with this shows robustness of the approach against illumination changes. To further show robustness of the approach

against partial occlusion, Figure 9 presents two instances of tracking under extreme occlusion for each plane before it leaves field of view.

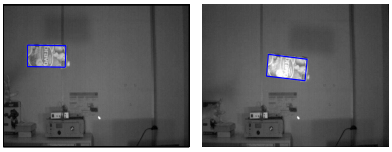

Figure 8: Two images of a sequence in which appearance of the plane changes due to non-diffused lighting.

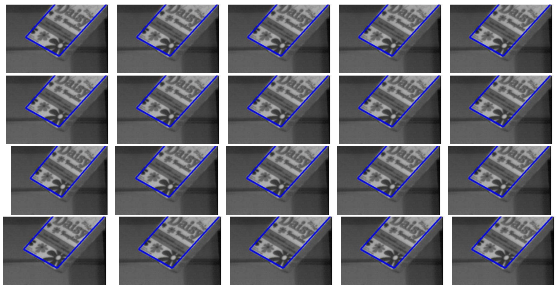

Figure 9: Two instances of tracking each plane under extreme occlusion. Top to bottom followed by left to right: (a) Daisy, (b) Donuts, (c) Rice, (d) China, (e) Juice, (f) Garment, (g) Monster, (h) Chicken, (i) Biscuit, (j) and Bravo.

5 CONCLUSIONS

A hybrid approach by fusing point and template based tracking to track planar-textured targets with large interframe displacement is introduced. The approach is flexible to adapt change in scene. It makes an efficient use of object and scene detail. Its evaluation is made using both the simulated and real sequences. In the first case, the approach performs better in terms of accuracy, convergence, and interframe displacement. In the second case, a consistent behavior is seen with the change in target. Robustness of the approach against partial occlusion and illumination changes is also shown. One may argue the approach is computationally expensive in terms of feature employed. To compensate this, a part of image is exploited. Moreover, faster convergence further weakens the argument, particularly, when the interframe displacement is large. At the application level, scope of trajectory measurement is extended to packaging industry considering industrial environment.

REFERENCES

Azad, P. (2009). State of the art in object recognition and pose estimation. *Cognitive Systems Monographs: Visual Perception for Manipulation and Imitation in Humanoid Robots*.

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *IJCV*.

Barteit, D., Frank, H., Pongratz, M., and Kupzog, F. (2009). Measuring the intersection of a thrown object with a vertical plane. In *IEEE INDIN*.

Bjorkman, M. and Kragic, D. (2004). Combination of foveal and peripheral vision for object recognition and pose estimation. In *IEEE ICRA*.

Choi, C. and Christensen, H. I. (2010). Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *IEEE ICRA*.

Collet, A., Berenson, D., Srinivasa, S. S., and Ferguson, D. (2009). Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE ICRA*.

Ekvall, S., Kragic, D., and Hoffmann, F. (2005). Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. *Image and Vision Computing*.

Frank, H., Barteit, D., and Kupzog, F. (2008). Throwing or shooting - a new technology for logistic chains within production systems. In *IEEE TePRA*.

Frank, H., Mittnacht, A., and Scheiermann, J. (2009). Throwing of cylinder-shaped objects. In *IEEE/ASME AIM*.

Jurie, F. and Dhome, M. (2002). Hyperplane approximation for template matching. *IEEE TPAMI*.

Ladikos, A., Benhimane, S., and Navab, N. (2009). High performance model-based object detection and tracking. *incollection-Theory and Applications: Computer Vision and Computer Graphics*.

Lepetit, V. and Fua, P. (2005). Monocular model-based 3d tracking of rigid objects. *FTCGV*.

Lepetit, V., Pilet, J., and Fua, P. (2004). Point matching as a classification problem for fast and robust object pose estimation. In *IEEE CVPR*.

Lieberknecht, S., Benhimane, S., Meier, P., and Navab, N. (2010). Benchmarking template-based tracking algorithms. *Virtual Reality*.

Mei, C., Benhimane, S., Malis, E., and Rives, P. (2008). Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE TRO*.

Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *IEEE ICCV*.

Sangle, P., Kutty, K., and Patil, A. (2011). A method for generation of panoramic view based on images acquired by a moving camera. *IJCA*.

Vacchetti, L., Lepetit, V., and Fua, P. (2004a). Combining edge and texture information for real-time accurate 3d camera tracking. In *IEEE/ACM ISMAR*.

Vacchetti, L., Lepetit, V., and Fua, P. (2004b). Stable real-time 3d tracking using online and offline information. *IEEE TPAMI*.

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2008). Pose tracking from natural features on mobile phones. In *IEEE/ACM ISMAR*.