# EyeWatchMe—
# 3D Hand and Object Tracking for Inside Out Activity Analysis

Li Sun
School of Electronic and Information Engineering
Xi'an Jiaotong University, Xi'an, China
sunli@mailst.xjtu.edu.cn
Ulrich Klank, Michael Beetz
Intelligent Autonomous Systems Group, Department of Informatics
Technische Universität München, Munich, Germany
klank@in.tum.de, beetz@in.tum.de

## Abstract

*This paper investigates the "inside-out" recognition of everyday manipulation tasks using a gaze-directed camera, which is a camera that actively directs at the visual attention focus of the person wearing the camera. We present* EyeWatchMe*, an integrated vision and state estimation system that at the same time tracks the positions and the poses of the acting hands, the pose that the manipulated object, and the pose of the observing camera. Taken together,* EyeWatchMe *provides comprehensive data for learning predictive models of vision-guided manipulation that include the objects people are attending, the interaction of attention and reaching/grasping, and the segmentation of reaching and grasping using visual attention as evidence.*

*Key technical contributions of this paper include an ego view hand tracking system that estimates 27 DOF hand poses. The hand tracking system is capable of detecting hands and estimating their poses despite substantial self-occlusion caused by the hand and occlusions caused by the manipulated object.* EyeWatchMe *can also cope with blurred images that are caused by rapid eye movements. The second key contribution is the of the integrated activity recognition system that simultaneously tracks the attention of the person, the hand poses, and the poses of the manipulated objects in terms of a global scene coordinates. We demonstrate the operation of* EyeWatchMe *in the context of kitchen tasks including filling a cup with water.*

## 1. Introduction

Inside-out activity recognition, in particular the visual interpretation of everyday manipulation tasks using head-mounted gaze directed cameras, is an essential tool for understanding vision-guided action. Thus, inside-out activity observation is a new challenge in the research field of computer vision. On one hand, these recognition capabilities will enable activity recognition systems to better segment actions into meaningful subevents because the gaze is proactively directing the visual attention with respect to the current and subsequent motion and action goal — in other words: *gaze leads action*. On the other hand, understanding the proactive task-directed visual attention of the gaze informs us about how to design visual attention mechanisms for autonomous robots performing everyday manipulation tasks in human environments.

In this paper we investigate the problem of 3D hand and object tracking for inside-out activity recognition. Thus, we focus on the following computational problem: given a image stream recorded by a gaze-directed camera, compute the position and the pose of the hands (whenever they are sufficiently visible), the object including its position and orientation that the gaze is directed to, and other features related to visual attention and hand manipulation such as the grasp type and the pregrasp.

Observing manipulation activities is a challenging visual task because saccadic eye movements are fast and frequent and therefore tracking is the repeated interleaved execution of two steps: (1) visual popout of the visually attended objects and the hands and (2) the incremental tracking of objects and hands as long as the gaze stays focused on the same object. Because of the fast eye movements many captured images are highly blurred. Also, because of the specific view during object manipulation the hand tracking has to be performed in the context of substantial self occlusions and occlusions caused by the manipulated object.

In this paper we propose a markerless hybrid 2D-3D ap-

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

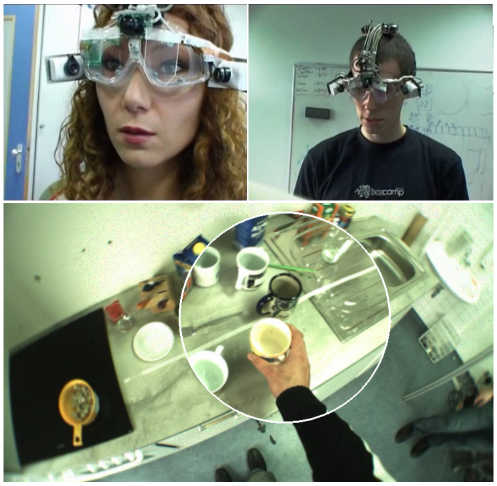


Figure 1. The gaze directed camera developed by [1] and its output. The camera itself is shown in the upper images. The lower image shows a superimposed camera view generated by the gaze-directed camera. The wide angle view of a camera mounted at the head and pointed towards the context of the scene. The inner picture depicts the focus of attention of the gaze in higher resolution.

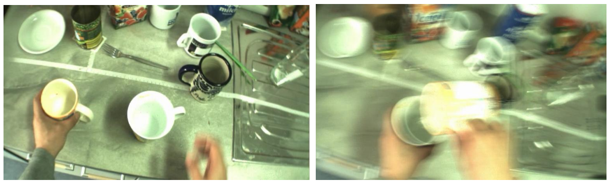


Figure 2. The Figure shows challenges in the interpretation of gaze-directed image streams. The upper picture shows the self occlusion of the hand and the occlusion caused by the held object, which are typical for the head-mounted gaze-directed cameras. The lower picture shows the blur generated through following the rapid eye movements.

proach for hand tracking which is specialized to working for images from ego view of a person wearing a gaze directed camera. It is based on the 2D hand model for initialization and 3D hand model for particle filtering. The 2D model has 9 degrees of freedom (DOF) which describes the 2D position (include length and width) of palm and thumb. The 3D model has 27 DOF which describe the global position of hand and the angle of each joint. A basic form of particle filter SIR (Sampling importance Resampling) is used for the 3D tracking of hand.

For each particle, we use edges and skin color to evaluate the fitness between the particle and the image. Because hands are often grasping some objects in gaze-directed video, how to handle occlusion between hand and grasping object is a crucial problem. We combine the result of object tracking into the process of hand tracking. For each particle of possible hand configuration, we check crucial points on it to see whether it is occluded by objects. If it is occluded, we just eliminate the occlusion region when we calculate the measurement of particle.

The key contributions of this paper are the following ones:

1. simultaneous object, hand and gaze tracking in absolute coordinates for inside-out activity analysis

2. the integrated object occlusion calculation that allows to improve the tracking results of the hand and is giving at the same time the interactions with those objects

3. the hybrid 2D-3D hand tracking method itself, especially the fast 2D model that allows an initial guess of the 3D model

In the remainder of this paper we proceed as follows: We will first shortly introduce related work. Then an overview of our method and its three major parts are given in detail. Finally, we will show results of the analysis of two simple actions.

## 2. Related Work

Visual interpretation of hand gestures imposes several problems that are discussed in a review by Pavlovic *et al.* [9]. They give an introduction to several approaches dealing with the problems like partial occlusions and the high degrees of freedom of an articulated hand.

A common approach is tracking of a hands in 3D using cameras, for example proposed by Tomasi *et al.* in [13]. They create a large database of hand gestures appearances and classify new hand appearances after a normalization using this database into a certain 3D hand position. A different also 3 dimensional approach was presented recently by Romero *et al.* in [10] that is based on two cameras and extract a contour based a stereo camera setup. Other work like [12, 16] are also tracking hand in 3D. A second, different approach is a 2D tracking, like for example Kölsch and Turk, who showed a method for tracking hands in 2D from an ego perspective in [6]. This work was enhanced to a 3D version by Guan *et al.* in [2]. Similar to our approach, Tsubuku *et al.* included object tracking into the hand pose estimation in [14]. Their work was more directed towards detecting the object under occlusions induced by the hand, while we are assuming to have a detection already working under partial occlusions for the object and aiming for the hands. Like all previously mentioned methods this method only applies to a well defined and static work space with defined camera position, while our approach applies to a fixed observer-hand relation and an unconstrained observer-workspace relation. Inside-out activity analysis was introduced by Land *et al.* in [7]. For the task of preparing tea they observed the gaze direction of the acting person and derived that even in such a simple task, a high degree of attention to all manipulated objects before and during the manipulation act is necessary. This fact is also used by Mayol and Murray in [8] for activity recognition. For this, they detect hands and variable objects in 2D.

## 3. An Overview of EyeWatchMe

Figure 3 depicts the function and the internal operation of EyeWatchMe. EyeWatchMe receives as its input the image stream generated by the gaze-directed camera and computes the hand poses for the time intervals where the hands are sufficiently visible in the camera image and the poses of the (known) objects that the visual attention of the acting person is directed to. Thus the output are 3D movements of the hands and fingers, trajectories for the gaze relative to the global scene, and trajectories for the manipulated objects. To perform these computations EyeWatchMe employs a library of precomputed hand poses and models of the objects of interest and the environment as resources. Trajectories can be found in Results section. In Figure 9(a) , Figure 9(b) and Figure 9(c), Trajectories for gaze, object and hand are shown respectively.

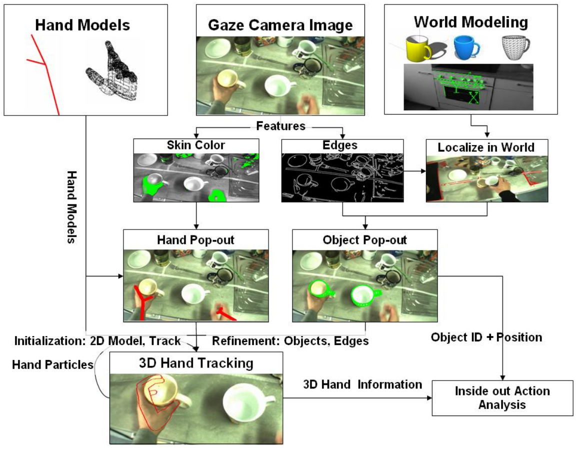


Figure 3. Our algorithm from the image to the final inside out action analysis.

EyeWatchMe organizes the computational process, which is also depicted in Figure 3 along two dimensions: a phase-oriented view, which deals with the detection of entities to be tracked and the entity-type specific computations that deal with the different entities — the hands, the objects, and the scene. The processing consists of a preprocessing and popout phase. As long as EyeWatchMe is not tracking the respective entity stays in the popout phase until it has detected the hands, the object, and landmarks of the environment. It transitions from the popout into the incremental tracking phase as soon as the entity detection is stable. The following three sections describe the entity-type specific computations of hand (Section 4), object (Section 5) and scene tracking (Section 5).

## 4. Hand Tracking

Let us now consider the computational process of hand tracking in more detail.

### 4.1. Hand Pop-out

The processing pipeline for hand tracking consists of the preprocessing/popout phase, in which hypotheses for hands are generated in a data-driven manner, using skin color as the central feature. EyeWatchMe filters these hypotheses using domain knowledge and constraints. Thus, it makes a strong bias on which parts of the image could possibly exist hands and arms. Upon having a hand hypothesis sufficiently validated, the hypothesis is passed to the incremental tracking process step.

To strongly bias the search space for the hand pose in the high-dimensional parameter space, EyeWatchMe applies a two step process. First, a 2D hand model with only 9 degrees of freedom , including the position, angle and length of the arm, palm and thumb is fitted to the arm-hand hypothesis using morphological operations. This 2D fitting can be performed both reliably and efficiently compared to the full 3D pose fitting. Then the 2D pose is used in the second step to distribute the particles for the 3D hand pose tracking.

#### 4.1.1 Skin Color Region Segmentation

EyeWatchMe segments skin colored regions in the image using the method proposed by Hsu in [4], which detects skin color tone in a transformed Y'Cb'Cr' color space which can reduce the effect of lighting condition to some extent.

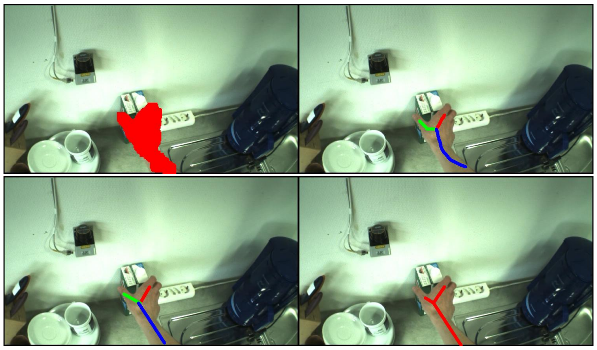


Figure 4. 2D model parameter estimation based on skeleton.

The result of skin color tone detection is based on pixels, and we extract the part of the image containing arms and hands. First, we fill the holes and then perform opening on it to remove some small part in order to reduce noisy regions. Finally, we select the separate region whose area is larger than the minimal hand region that could give a reasonable result later on. If still more than two regions are selected, we can apply the previously mentioned bias and select the two at the bottom.

### 4.1.2 Hand Parameter Estimation in 2D

Given a segmented arm hand region, we use simple heuristics about the most probable appearances of hands in an image. Figure 4 depicts the steps: First, the skeleton of the segmented regions are extracted. Then we represent the skeleton with straight line segments. This is achieved by the standard least squares line fitting algorithm. The parameters of the line segments are used to fit the parameter of 2D hand model. The bottom line is considered as arm skeleton. Thumb and finger skeleton are selected from other branches. Here we hold the assumption that these two skeleton should be long enough and the thumb is on the right side for the right hand. These two assumptions are normally satisfied in gaze-directed video.

## 4.2. 3D Hand Tracking

With an initial parameter set we can set up a initial hand gesture for the 3D tracking. Given this initial guess, we can start tracking of the 3D hand using a particle filter.

### 4.2.1 3D Hand Model

We are using the hierarchical 3D hand model proposed by Stenger *et al*. [11]. The model consists of 39 truncated quadrics as building blocks, approximating the anatomy of a real human hand and its application is depicted in Figure 5. The model provides 27 DOF: 6 for the global hand position, 4 for the pose of each finger and 5 for the pose of the thumb. The DOF for each joint correspond to the DOF of a real hand. Joints between the bones are named according to their location on the hand as metacarpophalangeal (MCP) (joining fingers to the palm), interphalangeal (IP, PIP, DIP, ...) (joining finger segments) and carpometacarpal (CMC) (connecting the metacarpal bones to the wrist). Additionally, it also offers 19 parameter to adapt the shape of model to the hand of a certain person. These parameters should be calibrated for different people before the tracking process.

Because joint angles for real hands are highly correlated with each other and bounded within a small region, some constraints need to be adopted to avoid unrealistic poses. Based on the studies in biomechanics, certain closed-form constraints can be derived. An important constraint is the relationship $\theta_{DIP} = \frac{2}{3}\theta_{PIP}$ between the PIP and DIP angles that helps us decrease the DOF by 4. Other constraints which limit the joint angles are also incorporated in the hand model.

After hand modeling, the pose of the hand can be fully determined by a vector $\mathbf{x_t}$ that comprises 21 joint angle parameters and 6 parameters that specify the global position and orientation of the palm with respect to the camera's coordinate frame. We have an efficient way to project such a hand given a certain parameter set and camera parameters. Even so, we speed up our approach by building up a database of already projected hands, because loading from a file even faster.

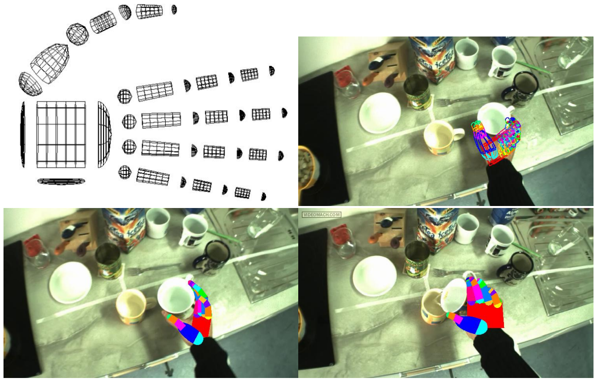


Figure 5. The 3D hand model, [11], particles and matches.

### 4.2.2 Particle Filter

Visual tracking can be regarded as the estimation of the system state which changes over time given a sequence of noisy observations. The essence of hand tracking is to determine the best pose $\mathbf{X}$ based on the observation image $\mathbf{Z}$. The best pose is the minimization over a state vector $\mathbf{X}$ evaluating an error function $E(\mathbf{X}, \mathbf{Z})$:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} E((\mathbf{X}), \mathbf{Z}) \tag{1}$$

According to the Bayes rule, the hand pose of the current frame $\mathbf{x_t}$ depends from prior hand pose $\mathbf{x_{t-1}}$ by following relation:

$$p(\mathbf{x_t}|z_t) \propto p(z_t|\mathbf{x_t})p(\mathbf{x_t}|\mathbf{x_{t-1}}) \tag{2}$$

where $p(\mathbf{z_t}|\mathbf{x_t})$ is the probability given the handstate of the image and it is proportional to our distance metric.

**Generating Samples:** One important part of particle filtering is to generate samples, which is also known as condensation ([5]). A time stamped sample set $\mathbf{s_t}^{(n)}, n = 1, ..., N$, which is weighted by their observations in image $\pi_t^{(n)} = p(z_t|\mathbf{x_t} = \mathbf{s_t}^{(n)})$, is used to represent a posteriori $p(\mathbf{x_t}|z_t)$. Here $\pi_t^{(n)}$ is normalized as $\sum_{n=1}^{N} \pi_t^{(n)} = 1$. So for every frame we can get a weighted sample set$\{\mathbf{s_t}^{(n)}, \pi_t^{(n)}\}$. The sample set propagates from $\{\mathbf{s_{t-1}}^{(n)}, \pi_{t-1}^{(n)}\}$ which represents $p(\mathbf{x_{t-1}}|z_{t-1})$ for previous frame to $\{\mathbf{s_t}^{(n)}, \pi_t^{(n)}\}$ which represents $p(\mathbf{x_t}|z_t)$ for current frame. The prediction is applied on $\{\mathbf{s_{t-1}}^{(n)}, \pi_{t-1}^{(n)}\}$ according to Equation 3 and 4. In 3a and 3b, the center position of the hand along X and Y axis is predicted. $X_t$ and $Y_t$ is the position of a particle center for current frame. $\lambda^X$ and

$\lambda^Y$ is the predicting moving distance between current frame and previous frame. We use the position of the 2D model to compute $\lambda^X = x_t^{2D} - x_{t-1}^{2D}$ and $\lambda^Y = y_t^{2D} - y_{t-1}^{2D}$. Here we change the 2D distance into 3D based on the previous depth $Z_{t-1}$ and focal length $f$. Other state parameters are generated as in Equation 4 . Here we hold assumption that these states do not change too much between frames, especially when people have already grasped something.

$$X_t^{(n)} \sim p(X_t|X_{t-1}) = N(\lambda_t^X + X_{t-1}|\sigma_X) \quad (3a)$$

$$Y_t^{(n)} \sim p(Y_t|Y_{t-1}) = N(\lambda_t^Y + Y_{t-1}|\sigma_Y) \quad (3b)$$

$$\mathbf{s_t}^{(n)} \sim p(\mathbf{s_t}|\mathbf{s_{t-1}}) = N(\mathbf{s_{t-1}}|\sigma) \quad (4)$$

### 4.2.3 Distance Measurement

Our method employs a combination of different distance measurements including color and edges:

- Quantity of overlapping of the projected arm-hand region with the corresponding skin color region.
- Distance Integration over edges in the image and the projected hand contour with corresponding directions

The overall error function $E(Z_t|\mathbf{X_t})$ consists of multiplication result of four values:

$$E(Z_t|\mathbf{X_t}) \propto e_{colorSI} \cdot e_{colorMI} \cdot e_{edgeHM} \cdot e_{edgeEM} \quad (5)$$

where $e_{colorSI}$ and $e_{colorMI}$ are calculated using the skin color region and $e_{edgeHM}$ and $e_{edgeEM}$ are based on the edges.

For the first item, we calculate areas of the skin color region $A_S$, hand model region $A_M$ and their intersection $A_I$ by counting the pixel number in the region. We use the following three measurements to measure the fitness of a particle according to the skin color information:

$$e_{colorSI}(z_t|\mathbf{x_t}) \propto \exp - \frac{(A_S - A_I)^2}{2\sigma_{SI}^2} \quad (6a)$$

$$e_{colorMI}(z_t|\mathbf{x_t}) \propto \exp - \frac{(A_M - A_I)^2}{2\sigma_{MI}^2} \quad (6b)$$

For the measurement of edge information, we need to count the point number of hand model contour $N_H$, edge pixel $N_E$ and matched contour $N_M$. First we apply canny edge detection on the image. Then we check along normal direction of every point of the projected model contour in a ten-pixel width neighborhood. If an edge pixel with high enough amplitude and similar enough direction is found in the neighborhood, we increase the number of $N_M$. So the measurement of edges is calculated as follows:

$$e_{edgeHM}(z_t|\mathbf{x_t}) \propto \exp - \frac{(N_H - N_M)^2}{2\sigma_{edge}^2} \quad (7a)$$

$$e_{edgeEM}(z_t|\mathbf{x_t}) \propto \exp - \frac{(N_E - N_M)^2}{2\sigma_{edge}^2} \quad (7b)$$

### 4.2.4 Object-Specific Improvements of the Hand Tracking

With a current guess of the hand and a current position estimation of several manipulated objects, possible occlusions of fingertips and parts of the palm are easily to detect. For performance reasons we only check 5 points of fingertip per hand candidate if they are in the area which lays inside the projection of objects. Only if possible occlusions occur we perform a z-buffering to check which parts of the hand are in front or beside and which parts behind an object.

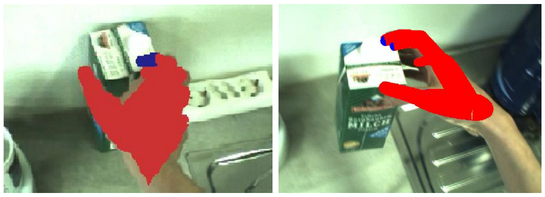


Figure 6. The index is occluded by the milk box.

**Determining of Occlusions**: Because the hand model is composed of quadrics, the position $(X_c, Y_c, Z_c)$ of the 3D center point of every fingertip quadric can be obtained easily given a specific hand pose. It is projected onto current image with a 2D coordinate $(x_c, y_c)$. If $(x_c, y_c)$ is in the 2D region of an object. We need to further check whether $(X_c, Y_c, Z_c)$ is occluded by the object or not. A ray going through camera can be determined by the point $(X_c, Y_c, Z_c)$. So every 3D point on this ray is represented with a coordinate $(X_c/s, Y_c/s, Z_c/s)$. Here $s$ is a scalar value which is larger than 0. This ray should have a intersection point $(X_c/s_0, Y_c/s_0, Z_c/s_0)$ with the object and $s_0$ is computable if we know the 3D shape of the object. If $s_0 < 1$, the intersection point is closer to the camera, which means the point $(X_c, Y_c, Z_c)$ is occluded by the object. Here we need to make a further assumption that we consider the whole quadric to be occluded if the center point it is occluded. Figure 6 shows an example for an image containing such an occlusion and we are successful in detecting the occlusion area on the tip of index.

**Influences on the distance measure:** The measurement of a particle should be changed if some part of the hand candidate is occluded by object. If $A_O$ denotes the area of occlusion hand region, $p_{colorMI}$ in Equation 6 is changed as follows:

$$p_{colorMI}(z_t|\mathbf{x_t}) \propto \exp - \frac{(A_M - A_I - A_O)^2}{2\sigma_{MI}^2} \quad (8)$$

$N_O$ denotes the point number of hand model contour which is occluded by object. $p_{edgeHM}$ in Equation 7 is changed like following.

$$p_{edgeHM}(z_t|\mathbf{x_t}) \propto \exp - \frac{(N_H - N_M - N_O)^2}{2\sigma^2_{edge}} \quad (9)$$

## 5. Object Tracking

Since our current application of EYEWATCHME is the inside-out recognition and interpretation of everyday manipulation tasks we can in many cases restrict ourself to dealing only with known objects.[1] Thus, in our setting we have equipped EYEWATCHMEwith a library of 3D object models that it is to detect, recognize, and localize in the image sequences. In our experiments we use the objects depicted in Figure 7.

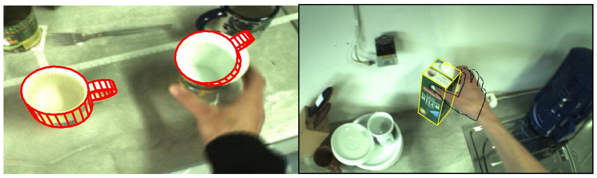


Figure 7. Examples for detected object models, cups and a milk-box.

Given a 3D model of an object EYEWATCHME can localize objects relatively fast and accurately. To this end, EYEWATCHME applies a state-of-the-art matching method proposed by Wiedemann *et al.* in [15] for finding this model. This method applies to a calibrated camera image, and uses a pyramid visual approach to match a projection of an object in an edge image. It is an exhaustive search in a previously defined search space of possible locations relative to the camera. The search space can be determined for a scenario given the position in the world that we get by camera localization. The search space consists mainly of the possible distances from the camera to the object and a range of possible viewing angles.

## 6. Scene Tracking

Tracking the focus of attention, the position of the hands and the objects in the global scene requires EYEWATCHME to estimate the pose of the gaze directed camera with respect to the global scene continually.

To this end, the system learns planar edge structures of the working environment beforehand. Figure 8 shows the planar edge structure learned for the top of the counter and how it is used for tracking the pose of the camera. Using the method proposed by Hofhauser *et al.* [3] EYEWATCHME can detect and track these planar edge structures in real time with a high accuracy. Given any reference point such as the position on of these edge structures and having them localized in the image, we can then directly infer the pose of the gaze-directed camera. In our experiments we are use a calibration tag fixed to a wall as a reference point to our world model that allows us to easily pick planar structures.

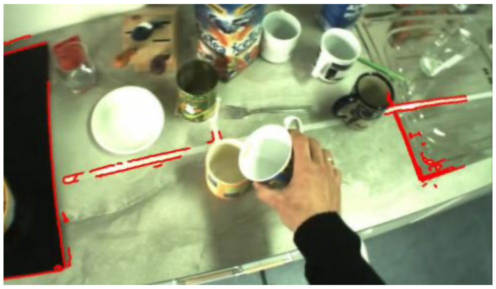


Figure 8. The contour on the table is localized.

## 7. Results

To validate and evaluate the proposed algorithms, we perform experiments based on several real sequences in normal kitchen environment captured by the gaze-directed camera. We want to show two kind of results. First a quality analysis of a pouring task for the single components, second, complete and integrated results for one sequence of pouring milk into a cup. Several test persons performed the first pouring task, which contained to fill water from one cup into another.

### 7.1. Quality analysis

In order to test the effect of particle number on tracking result, we perform hand tracking using a different number of particles such as 50, 200, 800, 1000 and 2000 per frame. The calculation of one particle takes about 10ms. The measurement of the best particle is recorded frame by frame. Then we calculate the mean value of the measurement. Table 1 shows the results. According to the objective measurement and direct visual evaluation, we found that 800 particles per frame are needed for relatively good hand tracking for simple grasp movement. But if we are only interested in the center position of hand, the number can be reduced to 200. There is no significant improvement while increasing particle number from 1000 to 2000. The measurement for video clip B is higher than it is for clip A, but this does not mean that the tracking result is better for clip B. Because the absolute value can be affected by many factors like the total area of skin color region $A_S$ and total number of edge pixel $N_E$.

In Table 2, we list the measurement of first frame (initializing frame), end frame (failure frame) and mean and deviation values for different persons. For every person we

[1] This is, however, on our agenda for future research including recognition and localization of novel objects and even deformable objects.

| Video Clip | 50 | 200 | 800 | 1000 | 2000 |
|---|---|---|---|---|---|
| A | 0.0012 | 0.0037 | 0.0115 | 0.0121 | 0.0124 |
| B | 0.0026 | 0.0055 | 0.0132 | 0.0146 | 0.0149 |

Table 1. Measurements under different particle number

| Test Person | First Frame | End Frame | Mean Value | Deviation |
|---|---|---|---|---|
| P1 | 0.00425 | 0.00823 | 0.0115 | 0.0138 |
| P2 | 0.0179 | 0.0209 | 0.0233 | 0.0114 |

Table 2. Mean measurements for different people performing the same pouring task

| Test Person | Frames | Model Update | Failed Frames |
|---|---|---|---|
| P1 | 250 | 26 | 2 |
| P2 | 200 | 35 | 3 |

Table 3. Times of model updating and the number of failure frames for gaze detection

have two video clips which perform the same pouring task. We can see that the measurement for the first frame is not so high as for other frames. The reason is that on the first frame we can only get the pose information from 2D model and this is not enough to find a better pose. But we can recover the pose a few frames later if the initialization is not too bad. It is similar in the case of blurring frame.

## 7.2. Trajectory samples

For a short sequence of a pouring action we want to show extracted trajectories of the gaze, the manipulated object and the right hand. The requested action that is performed in the sequence was to pour water from the right cup into the left cup.

### 7.2.1 Gaze Focus during Pouring Action

The first trajectory we can extract is the movement of the eyes and the gaze. Figure 9(a) shows results for one person pouring water from one cup to another. The trajectory starts on the right side, concentrates shortly on the left cup that was lifted, observes then the pouring and follows the cup set back on the table. This trajectory includes a compensation of the person's movement. The gaze-scene relation is logged by the camera. We calculated the person's movement with the proposed method in an average time of 125 ms per frame taking 2 planar models into account, namely a table and a wall. The test scenarios imply that one of the models is always in the field of view of the scene camera. Table 3 describes the results of our experiments in more detail. The counter updates refer to recalculations of visible models using the current image. This allows a smaller relative changes of the model and therefore a faster search. For the core frames of two test persons performing the pouring action, we measure the success rate if we could localize the camera. A match with a reliable score is counted as a success, no or a bad matches annotate a failure.

| Test Person | Frames | Missed Right | Missed Left |
|---|---|---|---|
| P1 | 250 | 25 | 16 |
| P2 | 200 | 35 | 26 |

Table 4. The object localization and the number of overall frames and the number of frames without a good match for the two cups

### 7.2.2 Object Movement

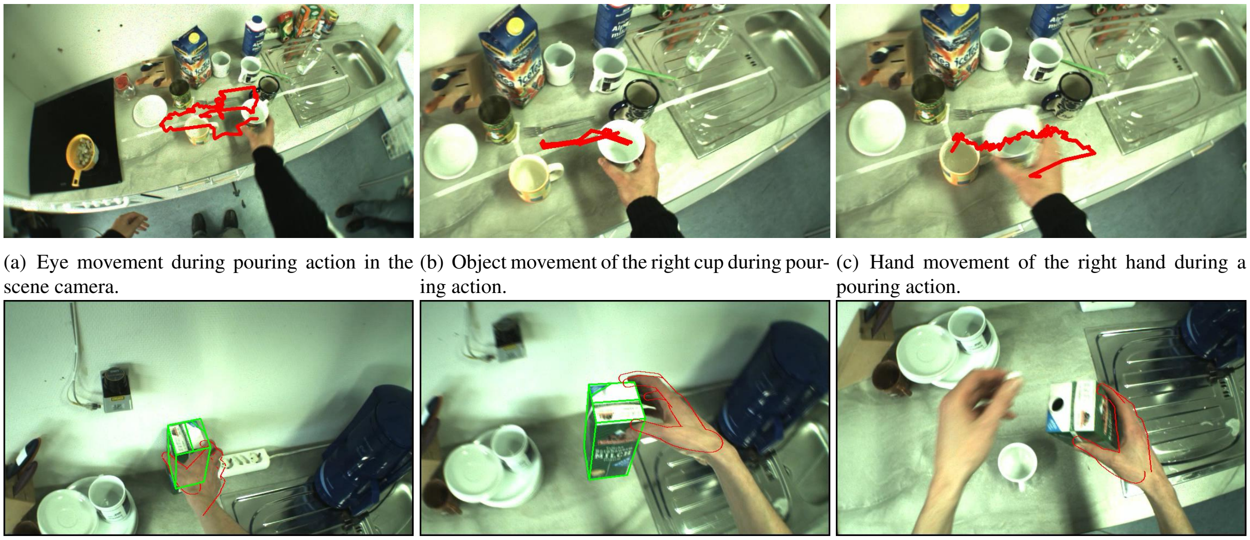
We visualize in Figure 9(b) the movement of the right cup in the pouring action. It can be seen that the object moves first up then to the left side and back to its old place. Table 4 shows the frequency of successful object localization in two scenes. Again those data were observed and manually decided over a success or failure. As soon as the object orientation or its position gets obviously wrong, a missed frame was counter. Respective occurring occlusions the frequency of localization is good. The right cup is occluded partially by the hand of the test person during manipulation, while the left cup is occluded by the right cup during the pouring itself.

## 7.3. Complete Sequence Data

We test both hand and object tracking in a sequence which shows that a person reaches his hand for milk and pours milk from a milk box to a cup. Here is some images about the tracking results on his right hand and milk box. There still some problems on the result of hand tracking. Because thumb is totally occluded during the process of pouring, we can not get thumb position correctly after pouring. Some of the most important frames are visualized in Figure 9.

# 8. Conclusion

In this paper we present EyeWatchMe, an integrated vision and state estimation system for a gaze-directed camera that at the same time tracks the positions and the poses of the acting hands, the pose that the manipulated object, and the pose of the observing camera. EyeWatchMe is an enabling technology for inside-out activity recognition, interpretation and analysis. The key contributions of our work are the tight coupling of pose estimation, hand and object tracking and visual popout mechanisms that reliably and quickly reinitialize the system after rapid eye movements. EyeWatchMe is also interesting for its hand pose estimation techniques that solve a high DOF estimation problem even in the case of partial occlusions of the hand caused by itself as well as the manipulated object. Our current work focuses on achieving more reliability and accuracy in hand and object tracking even in cases where most of the tracked entities are occluded and on the reduction of the computational resources required. This as well as automatic self-monitoring and reinitialization are key steps towards auto-



(a) Eye movement during pouring action in the scene camera.

(b) Object movement of the right cup during pouring action.

(c) Hand movement of the right hand during a pouring action.

(d) Pregrasp.

(e) Get the object.

(f) Start pouring, the object detection failed here.

Figure 9. Results for hand tracking and the inside-out action analysis.

matic long term activity recognition. Another thread is the broadening towards more comprehensive observation models and towards additional application domains. Thus we intend to extend the system to recognize important aspects in computational motor control such as contact events, the pregrasp pose, and the focus on attention at object levels. These capabilities will turn EYEWATCHME into a powerful tool for the investigation of human everyday manipulation tasks in the cognitive, neural, and medical sciences.

## References

[1] T. Brandt, S. Glasauer, and E. Schneider. A third eye for the surgeon. *J Neurol Neurosurg Psychiatry*, 77(2):278, Feb 2006. 2

[2] H. Guan, R. Feris, and M. Turk. The isometric self-organizing map for 3d hand pose estimation. In *Proc of the IEEE Conf on Automatic Face and Gesture Recognition, Southampton, UK*, 2006. 2

[3] A. Hofhauser, C. Steger, and N. Navab. Edge-based template matching and tracking for perspectively distorted planar objects. In *International Symposium on Visual Computing*, 2008. 6

[4] R. Hsu, M. Abdel-Mottaleb, and A. Jain. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):696–706, 2002. 3

[5] M. Isard and A. Blake. CONDENSATION: Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 4

[6] M. Kolsch and M. Turk. Robust hand detection. In *Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition*, 2004. 2

[7] M. Land, N. Mennie, and J. Rusted. The roles of vision and eye movements in the control of activities of daily living. *PERCEPTION-LONDON-*, 28(11):1311–1328, 1999. 2

[8] W. Mayol and D. Murray. Wearable hand activity recognition for event summarization. In *Ninth IEEE International Symposium on Wearable Computers, 2005. Proceedings*, pages 122–129, 2005. 3

[9] V. Pavlovic, R. Sharma, and T. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997. 2

[10] J. Romero, D. Kragic, V. Kyrki, and A. Argyros. Dynamic Time Warping for Binocular Hand Tracking and Reconstruction. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2289–2294, 2008. 2

[11] B. Stenger, P. Mendonca, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. In *Proc. British Machine Vision Conference*, volume 1, pages 63–72, 2001. 4

[12] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-Based Hand Tracking Using a Hierarchical Bayesian Filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1372–1384, 2006. 2

[13] C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. *Computer Vision, IEEE International Conference on*, 2:1441, 2003. 2

[14] Y. Tsubuku, Y. Nakamura, and Y. Ohta. Object Tracking and Object Change Detection in Desktop Manipulation for Video-Based Interactive Manuals. *Lecture Notes in Computer Science*, pages 104–112, 2004. 2

[15] C. Wiedemann, M. Ulrich, and C. Steger. Recognition and tracking of 3d objects. In G. Rigoll, editor, *Pattern Recognition*, volume 5096 of *Lecture Notes in Computer Science*, pages 132–141, Berlin, 2008. Springer-Verlag. 6

[16] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *International Conference on Computer Vision*, pages 426–432, 2001. 2