Learning Gestures for Customizable Human-Computer Interaction in the Operating Room

Loren Arthur Schwarz*, Ali Bigdelou*, and Nassir Navab

Computer Aided Medical Procedures, Technische Universität München, Germany
{schwarz,bigdelou,navab}@cs.tum.edu
http://campar.cs.tum.edu/

Abstract. Interaction with computer-based medical devices in the operating room is often challenging for surgeons due to sterility requirements and the complexity of interventional procedures. Typical solutions, such as delegating the interaction task to an assistant, can be inefficient. We propose a method for gesture-based interaction in the operating room that surgeons can customize to personal requirements and interventional workflow. Given training examples for each desired gesture, our system learns low-dimensional manifold models that enable recognizing gestures and tracking particular poses for fine-grained control. By capturing the surgeon's movements with a few wireless body-worn inertial sensors, we avoid issues of camera-based systems, such as sensitivity to illumination and occlusions. Using a component-based framework implementation, our method can easily be connected to different medical devices. Our experiments show that the approach is able to robustly recognize learned gestures and to distinguish these from other movements.

1 Introduction

Computerized medical systems, such as imaging devices, play a vital role in the operating room (OR). At the same time, surgeons often face challenges when interacting with these systems during surgery. Due to sterility requirements, control terminals are in many cases spatially separated from the main operating site. A typical resulting situation is that a less skilled assistant controls the computer using keyboard and mouse, guided by verbal communication with the surgeon [1,2]. This indirection can be inefficient and cause misundertandings. In addition, surgeons often prefer to have manual control over computerized systems for immediate feedback and, thus, higher precision [1].

We propose a method that allows surgeons to interact with medical systems by means of gestures. Based on the circumstances and the workflow of a particular interventional scenario, a surgeon can define a set of gestures that are most suitable. After demonstrating each gesture to the proposed system, our method learns prior gesture models from the training data (Section 2). These models, termed gesture manifolds, efficiently capture the underlying structure of the movements for each gesture and provide a low-dimensional search space

* Joint corresponding and first authors.

G. Fichtinger, A. Martel, and T. Peters (Eds.): MICCAI 2011, Part I, LNCS 6891, pp. 129–136, 2011.

for gesture recognition. We model a gesture as a sequence of multiple, smoothly varying body poses, allowing surgeons to adjust continuous parameters. While recognizing gestures, the prior models enable us not only to determine which gesture is performed (categorical control), but also to infer the particular pose within one gesture (spatio-temporal control). Instead of using video cameras to capture gestures, we rely on the data of a few wireless inertial sensors on the surgeon's body. This way, our method can easily handle the challenging conditions of operating room environments, where lighting is highly variable and personnel and equipment cause complex occlusions. Additionally, gestures are recognized regardless of the surgeon's position and orientation. As each inertial sensor can be identified uniquely, gestures can also be assigned to multiple persons, e.g. a surgeon and an assistant, for distributing the interaction workload. By building on a component-based framework implementation, our system enables an easy and dynamic association of learned gestures to the properties of arbitrary intra-operative computer-based systems (Section 3).

We evaluate our gesture-based interaction approach in the scenario of controlling a medical image viewer. Quantitative experiments show that the proposed method can simultaneously recognize up to 18 gestures to a high accuracy from as little as four inertial sensors. The promising results of the usability study encourage a practical application of our method in the operating room.

Related Work. Several authors have recently proposed gesture-based interaction systems for the OR [3,2,4,5]. Graetzel et al. [2] use a stereo camera setup for tracking a surgeon's hand and controlling the mouse pointer on a screen. In [3], hand gestures are recognized from two cameras for controlling a medical image viewer. A similar functionality is presented in [4] using a time-of-flight camera. Guerin et al. [6] use gestures for controlling a surgical robot. Vision-based approaches, such as all above methods, require that gestures are performed in a restricted region seen by a camera. We argue that using wireless inertial sensors for capturing a surgeon's gestures alleviates restrictions of visual systems, e.g. dependence on illumination and line of sight. Inertial sensors have been used for activity [7,8] and gesture recognition [9,10], but the OR has not been addressed. Several authors emphasize the inter-person variability of human gestures, e.g. [7], and propose methods that adapt to person-specific variations in performing a given set of gestures [7]. To provide most flexibility to surgeons, our system allows defining a completely arbitrary set of gestures by only demonstrating one example per gesture. While existing methods typically treat gestures as single commands, such as "click the mouse", the proposed gesture manifold model enables us to automatically recognize the performed gesture and to track the movements within a gesture for fine-tuning continuous parameters. Manifold learning techniques have shown to provide compact, low-dimensional representations of human motion data [11] and have been used for human pose tracking [12,8]. We combine multiple, gesture-specific manifold models and subdivide the embeddings into phases allowing us to assign particular poses to arbitrary parameter settings. Our method also naturally handles the problem of gesture segmentation by means of a predictive confidence measure.

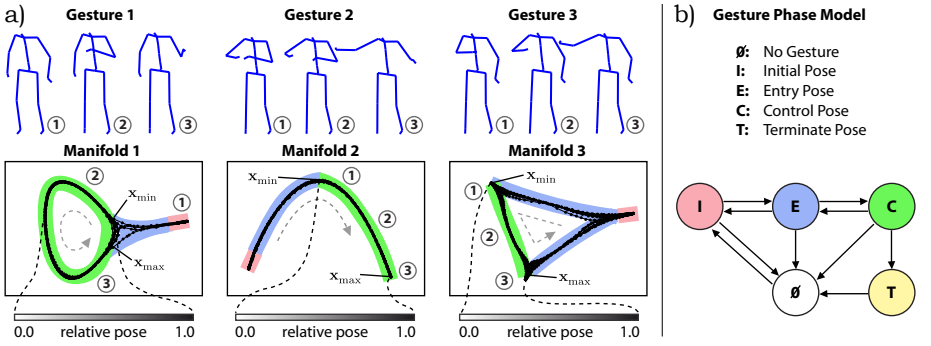

Fig. 1. a) The proposed method recognizes multiple user-defined gestures (*top*) and tracks the relative pose within a gesture by means of learned gesture manifolds (*middle*). The relative gesture pose, given by a value in the interval $[0, 1]$, is used for smooth parameter adjustment (*bottom*). b) Gesture phase model, colors correspond to a).

2 Gesture Recognition Method

The proposed gesture recognition method consists of a training phase, where gesture models are learned from example sensor data for each considered gesture, and a testing phase, where the models are used to recognize gestures from previously unseen sensor data.

Learning Gesture Manifolds. In the training phase, we learn prior models of gestures from sample sensor data. Let N be the number of considered gestures and let $\mathbf{S}^c = [\mathbf{s}_1^c, \ldots, \mathbf{s}_{n_c}^c]$, $1 \leq c \leq N$, be a dataset of n_c labeled sensor measurements. Each vector $\mathbf{s}_i^c \in \mathbb{R}^{d_s}$ consists of four quaternion values per sensor. To obtain a compact parameterization of feasible sensor values, we use Laplacian Eigenmaps, a manifold learning technique [13]. In particular, we map the training data $\mathbf{S}^c$ for each gesture c to a low-dimensional representation $\mathbf{X}^c = [\mathbf{x}_1^c, \ldots, \mathbf{x}_{n_c}^c]$, such that $\mathbf{x}_i^c \in \mathbb{R}^{d_x}$ and $d_x \ll d_s$. Figure 1.a) shows exemplary two-dimensional manifold embeddings for three gestures. The crucial property of the manifold embeddings is that the local spatial distribution of vectors in the original, high-dimensional representation is preserved. In particular, similar sensor measurements will map to close-by embedding points, even if they occur at different times within a gesture. In the testing phase, this makes our gesture recognition method invariant to movement speed.

We relate the space of sensor measurements and the low-dimensional manifold embeddings using kernel regression mappings. The mappings allow projecting new sensor values $\mathbf{s}^*$ to points $\hat{\mathbf{x}}$ in embedding space (*out-of-sample mapping*) and predicting sensor vectors $\hat{\mathbf{s}}$ from given embedding points $\mathbf{x}^*$ (*reconstruction mapping*). Following [8], we define the *out-of-sample mapping* for gesture c as $\hat{\mathbf{x}} = f_c(\mathbf{s}^*) = \frac{1}{\phi^c(\mathbf{s}^*)} \sum_{i=1}^{n_c} k_s^c(\mathbf{s}^*, \mathbf{s}_i^c)\mathbf{x}_i^c$, where $\phi^c(\mathbf{s}^*) = \sum_{j=1}^{n_c} k_s^c(\mathbf{s}^*, \mathbf{s}_j^c)$. We use a Gaussian kernel k_s^c with a width determined from the variance of the training sensor data. The mapping is a weighted average of all manifold embedding points

$\mathbf{x}_i^c \in \mathbf{X}^c$, with the largest weights attributed to points projected from sensor values $\mathbf{s}_i^c$ which are similar to $\mathbf{s}^*$. By interchanging the roles of sensor values and manifold embedding points, we obtain the *reconstruction mapping* $\hat{\mathbf{s}} = g_c(\mathbf{x}^*)$.

Parameterizing Gesture Manifolds. After training, any pose corresponding to one of the learned gestures can be represented as a pair $(c, \mathbf{x})$, where c identifies one of the N manifold embeddings and $\mathbf{x}$ is a point in that embedding. As we allow gestures to have a temporal extent, we also introduce a simple phase model, shown in Figure 1.b). Each manifold embedding is subdivided into three phases $\{I, E, C\}$. Phase I indicates the beginning and end of a gesture, phase E contains introductory movements, such as raising a hand, and phase C is actually used for controlling a target system. The points $[\mathbf{x}^c_{\min}, \mathbf{x}^c_{\max}]$ indicating the boundaries of phase C, mapped to a minimal and maximal parameter setting, can be defined by the user in the training phase, e.g. by holding the respective poses for several seconds. The additional phase $\emptyset$ represents poses that do not belong to any of the learned gestures. Phase T is used to terminate parameter adjustment. Our use of the phase model is explained below.

Recognizing and Tracking Gestures. In the testing phase, we employ a particle filter [14] that continuously explores the gesture manifolds to find the manifold index $\hat{c}_t$ and point $\hat{\mathbf{x}}_t$ that best explain the sensor measurements $\mathbf{s}_t$ at any time t. Every particle $\mathbf{p}_t^i = (c_t^i, \mathbf{x}_t^i)$, $1 \leq i \leq n$, represents one pose hypothesis. Initially, all n particles are randomly distributed across the manifold embeddings. In every iteration of the particle filter, we let particles propagate through the manifold embeddings, ensuring that only positions close to the learned embedding points are sampled (see Figure 3). With a certain probability, particles are allowed to switch between different manifold embeddings. We model this probability to be high in embedding space regions that correspond to an idle pose separating gestures. To evaluate the fitness of a particle, we define the observation likelihood $p(\mathbf{s}_t | c_t^i, \mathbf{x}_t^i) \propto \mathcal{N}(g_{c_t^i}(\mathbf{x}_t^i); \mathbf{s}_t, \mathrm{cov}(\mathbf{S}^{c_t^i}))\, \mathcal{N}(f_{c_t^i}(\mathbf{s}_t); \mathbf{x}_t^i, \mathrm{cov}(\mathbf{X}^{c_t^i}))$. The first normal distribution is centered around the observation $\mathbf{s}_t$, giving a high weight to a particle if the sensor value predicted from its position $\mathbf{x}_t^i$ is close to $\mathbf{s}_t$. In the second normal distribution, centered around $\mathbf{x}_t^i$, particles are favored that are close to the projection of $\mathbf{s}_t$ into embedding space. The final estimated gesture index $\hat{c}_t$ is selected as the most frequent index c_t^i among the best-scoring particles. Among these particles, those with $c_t^i = \hat{c}_t$ are used to compute the final position $\hat{\mathbf{x}}_t$ in embedding space.

Temporal Gesture Segmentation. Having estimated $\hat{c}_t$ and $\hat{\mathbf{x}}_t$, we determine the corresponding phase in our gesture phase model (Figure 1.b). Phase $\emptyset$, indicating an unknown (or non-gesture) movement, is activated when the prediction confidence $\lambda_t^{\emptyset} = -\log(k_s^{\hat{c}_t}(\mathbf{s}_t, g_{\hat{c}_t}(\hat{\mathbf{x}}_t)))$ falls below a preset threshold. This measure evaluates how closely the estimated state $\hat{\mathbf{x}}_t$, projected to sensor space, matches the true sensor observation $\mathbf{s}_t$. When in phase $\emptyset$, the gesture prediction will likely be incorrect and can be disregarded. Note that our phase model permits transitions into phase $\emptyset$ from any other phase, allowing the user to exit gesture recognition at any time. To identify the *initial* phase I, and thus the beginning of

a gesture, we define $\lambda_t^I = k_s^{\hat{c}_t}(\mathbf{s}_t, \mathbf{s}_0)$, which evaluates the similarity between the sensor measurements $\mathbf{s}_t$ and the idle pose $\mathbf{s}_0$ used for separating gestures. While λ_t^I is above a preset threshold, we assume the idle pose is taken, implying the onset of a gesture. If neither of the phases $\{\emptyset, I\}$ are active, we assume the current phase is one of $\{E, C\}$ and compute a relative pose value $\hat{a}_t \in [0, 1]$ from the manifold position $\hat{\mathbf{x}}_t$ (see Figure 1.a). To this end, we transfer the Cartesian coordinate $\hat{\mathbf{x}}_t$ into a polar representation (r_t, θ_t), such that the pole is at the centroid of the embedding points $\mathbf{x}_i^{\hat{c}_t}$, and keep the angular component θ_t. Since the angular representation $[\theta_{\min}^c, \theta_{\max}^c]$ of the points $[\mathbf{x}_{\min}^c, \mathbf{x}_{\max}^c]$ labeled in the training phase is known, we can compute the desired relative pose value as

$$\hat{a}_t = \begin{cases} (\theta_t - \theta_{\min}^{\hat{c}_t})/(\theta_{\max}^{\hat{c}_t} - \theta_{\min}^{\hat{c}_t}) & \text{if} \quad \theta_{\min}^{\hat{c}_t} \leq \theta_t \leq \theta_{\max}^{\hat{c}_t}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We define phase C to be active when $\hat{a}_t \neq 0$. By changing the pose within the boundaries of phase C, the user can fine-tune parameters. When a suitable parameter setting has been found, the user can trigger a transition from phase C to the *termination* phase T. The current parameter value is then stored and movements are ignored for a certain amount of time.

3 Component-Based Implementation

In a complex domain such as the OR, it is important to use a proper underlying architecture to achieve a usable gesture-based user interface. In order to practically control parameters of an arbitrary intra-operative device with recognized gestures, we developed a specialized component-based framework. In this model, components encapsulate features of the target systems and expose them to the framework. Such a modular design provides extensibility and therefore a wide range of computer-based devices can be controlled with the proposed gesture-based user interface. Furthermore, to freely customize the behavior of the system for a specific scenario, we implemented the framework according to the data streaming pipeline model. With this model, users can adapt the user interface response to recognized gestures by altering an underlying pipeline graph. This graph contains a set of components as well as the data flow connections (Figure 2). Using visual editing environments and thanks to late binding, the graph can be defined at runtime without further programming.

As shown in Figure 2, we have created separate components for the inertial sensors, the gesture recognition method and a medical image viewer as an exemplary target device. If the gesture recognition component detects the control phase C, a demultiplexer component forwards the relative pose value $\hat{a}_t$ to a separate output signal based on the gesture index $\hat{c}_t$, which further can be bound to any property of the target system. This removes the dependency on other interaction techniques, such as mouse or voice commands, for switching between different properties to control. Additionally, the demultiplexer component blocks further data flow when the current phase is one of $\{\emptyset, I, E, T\}$. This ensures that no modification in the target system will happen when an unknown movement is performed or when a gesture has been terminated.

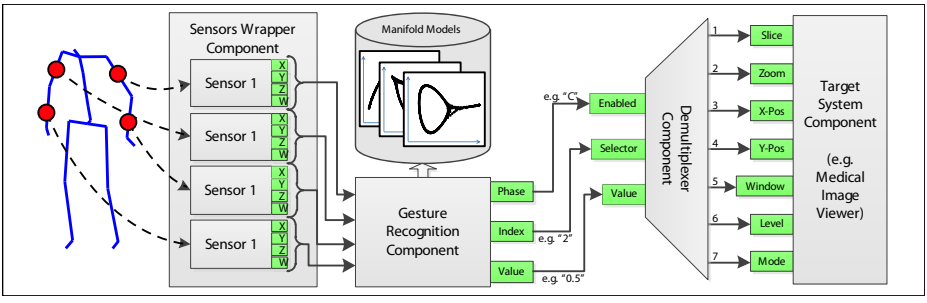

Fig. 2. Component graph for the proposed system. User movements, captured by the inertial sensors, are forwarded through the gesture recognition component, relying on learned manifold models, to properties (green) of the medical target system.

4 Experiments and Results

To evaluate our gesture-based interaction method for the OR, we conducted qualitative and quantitative experiments. We used two to six Colibri Wireless orientation sensors (Trivisio GmbH) attached to the arms of our testing persons. Gestures we defined for evaluation included moving an arm horizontally or vertically, tracing out circles, or other movements with one or both arms. Our implementation with Matlab and C++ components runs in realtime.

User Study. Tests with 10 subjects were performed to assess the usability of the proposed method with the medical image viewer as an exemplary target system. We asked each person to localize a stent bifurcation within a volumetric CT dataset using 6 personalized gestures (Figure 3). The gestures were assigned to the main parameters of the image viewer, such as scaling, contrast, slice number, etc. Average user answers in a questionnaire consisting of five questions are given in Figure 4.a). Very positive feedback was given to the wearability of the sensors, the responsiveness and the achieved precision of the system.

Gesture Recognition. The gesture recognition accuracy of our method was evaluated systematically on a dataset of 18 different gestures, each recorded four times. We created labeled sequences of multiple gestures in a row, each between 1000 and 5000 frames. Experiments were performed in a cross-validation manner, each time using one of the sequences for training and one of the others for testing. While measuring the percentage of frames with correctly recognized gestures, we varied the number of simultaneously trained gestures and the number of inertial sensors. As shown in Figure 4.b), best gesture recognition rates were achieved with six inertial sensors. In this case, 95% of all frames were correctly recognized with a set of four gestures, and 88% when the method was trained on 18 gestures. Although the recognition rates decrease for less than six sensors, even only two sensors (one on each arm) yield correct recognition rates above 80% for all considered numbers of gestures. Our method thus scales well with respect to the number of gestures to be recognized simultaneously. However, we expect that less than 10 gestures need to be distinguishable in practical applications.

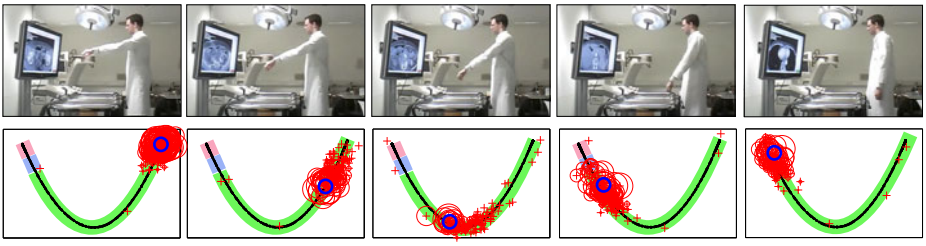

Fig. 3. *Top row*: Sample images of a test subject performing one of the learned gestures. *Bottom row*: Manifold embedding for the same gesture with distribution of particles, shown in red, for each of the above images. The point $\hat{\mathbf{x}}_t$ is given by a dark circle.

Gesture Segmentation. It is crucial for a gesture-based interaction method to distinguish instances of learned gestures from arbitrary other movements. Using six inertial sensors, we varied the threshold associated with the confidence measure $\lambda_t^{\emptyset}$ and measured the percentage of frames with gesture movements recognized as such (true positive rate, TPR) and the percentage of non-gesture frames wrongly identified as gestures (false positive rate, FPR). We trained the method on different numbers of gestures and randomly created sequences where only one of multiple gestures was known to the method. Figure 4.c) shows the resulting ROC curves. The best combination of high TPR and low FPR was achieved in the setting with four gestures. In this case, above 90% of gestures were detected with less than 10% of false positives. Although distinguishing non-gesture movements becomes more difficult when many gestures are learned simultaneously, detection results remain reasonable, even for 18 learned gestures.

5 Discussion and Conclusion

In this paper, we proposed a novel gesture-based interaction method for the OR using wireless inertial sensors. During a training phase, manifold models

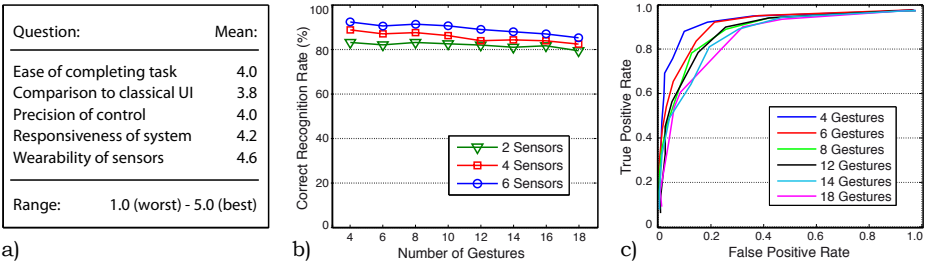

Fig. 4. a) Average questionnaire results after qualitative user study with 10 test subjects. b) Correct gesture classification rates for various settings. c) ROC curves for automatic differentiation of learned gestures from non-gesture movements.

are learned for each considered gesture based on the sensor data. This allows customizing gestures for each individual user and for different workflow phases, considering different constraints. Our evaluation shows promising results of using this method in our experimental setup. While the automatic differentiation between learned gestures and other movements achieves true positive rates above 90%, an additional tool, e.g. a pedal or voice command, could be used to activate and deactivate automatic gesture-based control within critical workflow stages. We will concentrate on these options in our future work.

References

1. Johnson, R., O'Hara, K., Sellen, A., Cousins, C., Criminisi, A.: Exploring the potential for touchless interaction in image-guided interventional radiology. In: ACM Conference on Human Factors in Computing Systems, pp. 1–10 (January 2011)
2. Graetzel, C., Fong, T., Grange, S., Baur, C.: A non-contact mouse for surgeon-computer interaction. Technology and Health Care 12(3), 245–257 (2004)
3. Kipshagen, T., Graw, M., Tronnier, V., Bonsanto, M., Hofmann, U.: Touch-and marker-free interaction with medical software. In: World Congress on Medical Physics and Biomedical Engineering 2009, pp. 75–78 (2009)
4. Soutschek, S., Penne, J., Hornegger, J., Kornhuber, J.: 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In: Computer Vision and Pattern Recognition Workshops (April 2008)
5. Wachs, J.P., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M., Handler, J.: A real-time hand gesture interface for medical visualization applications. Applications of Soft Computing, 153–162 (2006)
6. Guerin, K., Vagvolgyi, B., Deguet, A., Chen, C., Yuh, D., Kumar, R.: ReachIN: A modular vision based interface for teleoperation. In: SACAI Workshop (2010)
7. Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uWave: Accelerometer-based personalized gesture recognition and its applications. Pervasive and Mobile Computing 5(6), 657–675 (2009)
8. Schwarz, L.A., Mateus, D., Navab, N.: Multiple-activity human body tracking in unconstrained environments. In: Perales, F.J., Fisher, R.B. (eds.) AMDO 2010. LNCS, vol. 6169, pp. 192–202. Springer, Heidelberg (2010)
9. Hartmann, B., Link, N.: Gesture recognition with inertial sensors and optimized DTW prototypes. In: IEEE Conference on Systems Man and Cybernetics (2010)
10. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., Marca, S.: Accelerometer-based gesture control for a design environment. Pers Ubiquit Comput. 10(5), 285–299 (2006)
11. Elgammal, A., Lee, C.S.: The role of manifold learning in human motion analysis. In: Rosenhahn, B., Klette, R., Metaxas, D. (eds.) Human Motion. Computational Imaging and Vision, vol. 36, pp. 25–56. Springer, Netherlands (2008)
12. Jaeggli, T., Koller-Meier, E., Gool, L.V.: Learning generative models for multi-activity body pose estimation. International Journal of Computer Vision 83(2), 121–134 (2009)
13. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
14. Isard, M., Blake, A.: Condensation—conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)