Object Localisation via Action Recognition

John Darby, Baihua Li, Ryan Cunningham, Nicholas Costen
School of Computing, Mathematics and Digital Technology
Manchester Metropolitan University, Manchester, M1 5GD, UK.
{*j.darby,b.li,r.cunningham,n.costen*}*@mmu.ac.uk*

Abstract

The aim of this paper is to track objects during their use by humans. The task is difficult because these objects are small, fast-moving and often occluded by the user. We present a novel solution based on cascade action recognition, a learned mapping between body- and object-poses, and a hierarchical extension of importance sampling. During tracking, body pose estimates from a Kinect sensor are classified between action classes by a Support Vector Machine and converted to discriminative object pose hypotheses using a {*body, object*} *pose mapping. They are then mixed with generative hypotheses by the importance sampler and evaluated against the image. The approach out-performs a state of the art adaptive tracker for localisation of 14/15 test implements and additionally gives object classifications and 3D object pose estimates.*

1 Introduction and Related Work

This paper attempts to recover 3D object location and orientation through sequences of images. We focus on the subset of objects that are handled by humans during use: *implements*. The problem is challenging because implements are often small, move quickly through large pose changes during use, and are heavily occluded by the user. As a result, previous work has focused on implement location (e.g. [3]) rather than full 3D pose. For humans, identifying the class and pose of implements is made easier by their ability to interpret the pose of the person using them [8]. Inspired by this fact, we generate predictions of full 3D *object pose* from estimates of the human user's current *body pose* and *action class*.

We present a *cascade* approach to action recognition (e.g. [4]), using a Support Vector Machine (SVM) to find a likely subset of current action classes from an estimate of human pose found by the Kinect sensor [7]. We then use this classification and associated body pose to inform a simultaneous search for both the object class and full 3D object pose. To make this search we describe a hierarchical extension of *importance sampling* [5] using mixed-state *action particles*. The approach combines *generative* hypotheses gradually adjusted from the previous observation with *discriminative* hypotheses conditioned on the current body pose for a *combined* approach that gives both smooth tracking and error recovery.

The closest work to our own is that of Gall et al. [3]. However, only object translations are recovered (not full 3D poses) and localisation relies on a simple heuristic: that objects always lie within 25cm of the subject's most active hand. Here we take a different approach, using an action training database labelled with object poses to learn the relative relationships between objects and *all limbs*. This allows us to track objects during more complex interactions (e.g. two-handed) and to estimate their 3D translation and rotation over time.

2 Overview

We have collected a database of 165 12-second videos showing 11 subjects performing 15 separate actions involving implements. Action names (and implied object) are given in Table 1. The database was recorded using Kinect and each observation $\underline{z}_t = [\underline{b}_t, \underline{r}_t]$ consists of a body pose estimate $\underline{b}_t$ and RGB image $\underline{r}_t$. We have also manually labelled a subset of the sequences with 3D object poses and portions of this synchronised body-object pose dataset are used to train the proposed approach and to evaluate the accuracy of results. **During training** the data is used to: i) extract body pose vectors to train an SVM for action recognition (Section 3.1); ii) extract object poses and pair them with body poses to create a set of {body, object} pose mappings (Section 3.2). **During tracking** these mappings are used to: i) produce new generative object pose hypotheses (Section 4.2); ii) produce new discriminative

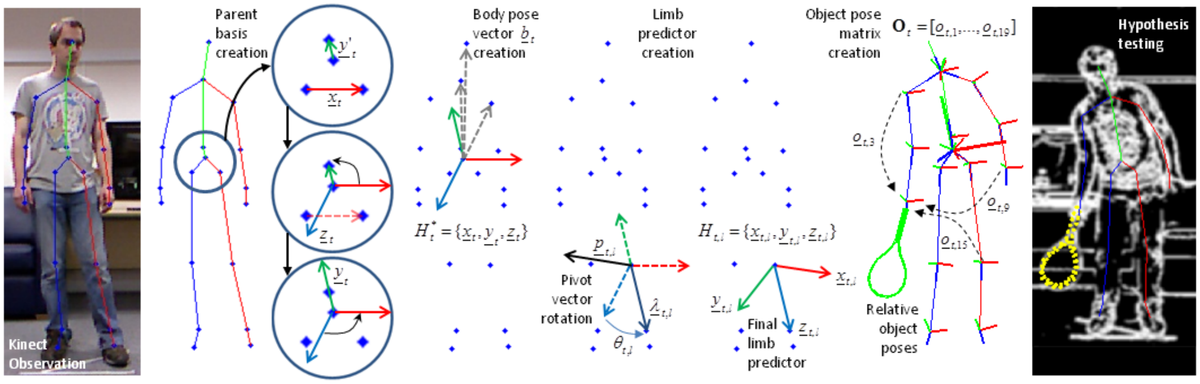

Figure 1. Body pose and object pose: (L-to-R) A parent basis H_t^* is built from joints in the pelvis and used to construct a body pose vector $\underline{b}_t$ from the relative locations of other joints (Section 3.1); Local bases are created for every limb by translating and rotating H_t^* and are used to learn a matrix of relative object poses, $\mathbf{O}_t$ (Section 3.2); During tracking these local bases, or "limb predictors", are used to produce object pose hypotheses for evaluation (Section 4).

object pose hypotheses (Sections 4.3 and 4.4).

3 Training: Body and Object Poses

3.1 Body Pose Classification

At each time t, Kinect gives 3D location estimates for 20 different joints, $\{\underline{j}_{t,i}\}_{i=1}^{20}$. For invariance to rotations and translations of the subject relative to the sensor we shift these coordinates into a new basis $H_t^* = \{\underline{x}_t, \underline{y}_t, \underline{z}_t\}$ centred on the subject's hips and calculated using the a series of simple vector crossing operations shown in Fig. 1. A body pose at a particular instant, $\underline{b}_t \in \mathbb{R}^{60}$, is then given by the concatenated locations of all 20 joints within H_t^*. We use collections of these body poses to train a multi-class SVM, $\mathcal{S}$ with a non-linear RBF kernel [2], for the classification of new body poses to action classes during tracking (Section 4.3).

3.2 $\{$Body, Object$\}$ Pose Mappings

Object pose is calculated relative to every one of the subject's limbs at every instant, see also Fig. 1. To do this we generate a new basis for each one of the $l \in [1, ..., 19]$ limbs in the kinematic tree, $\underline{\lambda}_{t,l} = \underline{j}_{t,j} - \underline{j}_{t,i}$, by translating the basis H_t^* to lie with its origin at the parent joint $\underline{j}_{t,i}$ and rotating it so the original $\underline{z}_t$-axis lies along the limb, pointing through the child joint $\underline{j}_{t,j}$. This is done by rotating H_t^* through a positive angle $\theta_{t,l}$ about a pivot vector $\underline{p}_{t,l} = \underline{\lambda}_{t,l} \times \underline{z}_t$, where $\theta_{t,l} = \cos^{-1}(\hat{\underline{\lambda}}_{t,l} \cdot \hat{\underline{z}}_t)$. The new basis $H_{t,l} = \{\underline{x}_{t,l}, \underline{y}_{t,l}, \underline{z}_{t,l}\}$ is referred to as the lth *limb predictor*.

The object's pose relative to the lth limb predictor is given by $\underline{o}_{t,l} = [\underline{a}_{t,l}, \underline{q}_{t,l}]$, where $\underline{a}_{t,l} \in \mathbb{R}^3$ is a translation and $\underline{q}_{t,l} \in \mathbb{R}^4$ is a quaternion rotation, both relative to $H_{t,l}$. At time t, we store the full set of 19 relative object poses as the matrix $\mathbf{O}_t = [\underline{o}_{t,1}, ..., \underline{o}_{t,19}]$, and denote the pose for the lth limb predictor at time t by $\mathbf{O}_{t,(:,l)}$ where $(:, l)$ denotes the lth column of the matrix.

For each action class $a \in [1, ..., 15]$ all N available $\{$body, object$\}$ training poses are paired to create a set of pose mappings $\mathcal{D}_a = \{D_1, ..., D_N\} = \{\{\underline{b}_1, \mathbf{O}_1\}, ..., \{\underline{b}_N, \mathbf{O}_N\}\}$.

4 Tracking: Particle-Based Object Pose

Particle filtering facilitates a generative approach to object tracking by maintaining an approximation to the posterior $p(\underline{o}_t|\mathcal{Z}_t)$, where $\mathcal{Z}_t = (\underline{z}_1, ..., \underline{z}_t)$, with a set of P particles, $\{(\underline{o}_t^{(p)}, \pi_t^{(p)})\}_{p=1}^{P}$. The pth particle consists of an object pose estimate, $\underline{o}_t^{(p)}$ and associated weighting $\pi_t^{(p)} \approx p(\underline{z}_t|\underline{o}_t^{(p)})$. Particles are dispersed by a dynamical model $p(\underline{o}_t|\underline{o}_{t-1})$ between observations.

4.1 Hierarchy of Mixtures

A mixed-state particle filter [6] supports changes between a number of different dynamical models over time, via a transition matrix $\mathbf{T}$. An importance sampler [5] is a particular type of mixed-state filter which mixes "normal" generative particles, conditioned on previous observations, with discriminative "importance" particles, conditioned only on the current observation.

We extend this approach to a hierarchy of mixtures by replacing "normal" particles with *action particles* which can also be propagated through two further submixtures of: i) action classes, and ii) limb predictors (Section 4.2). To do this we augment each particle with action class and limb predictor indices, $a_t^{(p)} \in [1,2,...,15]$ and $l_t^{(p)} \in [1,2,...,19]$, and we constrain hypothesis generation by replacing the unconstrained object pose $\underline{o}_{t,l} \in \mathbb{R}^7$ with an index $n_t^{(p)} \in [1,...,N]$ into the current action class's mappings, $\mathcal{D}_{a_t^{(p)}} = \{D_1,...,D_N\}$. The new particle structure is given by $((a_t^{(p)}, l_t^{(p)}, n_t^{(p)}), \pi_t^{(p)})$, where $\mathbf{O}_{n_t^{(p)},(:,l_t^{(p)})}$ gives the object's pose relative to the limb predictor.

4.2 Action Particles

Action particles are sampled from the particle set at $(t-1)$ and propagated through a sub-mixture of action classes. A transition matrix $\mathbf{A}$ where $A_{ij} = p(a_t = a_j | a_{t-1} = a_i)$ controls inter-class transitions. Limb predictors are selected randomly $l_t^{(p)} \in [1,2,...,19]$ after class changes ($a_t^{(p)} \neq a_{t-1}^{(p)}$) but remain constant otherwise, $l_t^{(p)} = l_{t-1}^{(p)}$. Local class dynamics are then given by $n_t^{(p)} = \mu_{a_t^{(p)}}(\underline{b}_{n_{t-1}^{(p)}}, \phi)$, where $\mu_a(\underline{b}, i)$ is a function that returns the ith nearest neighbour in Euclidean space to a body pose $\underline{b}$ from the set of pose mappings $\mathcal{D}_a$, and ϕ is a discrete, positive sample drawn from a Gaussian distribution, $\phi \sim \mathrm{round}(|N(0,\sigma^2)|)$.

4.3 Importance Particles

Importance particles are sampled from an *importance function* conditioned only on the current observation, $g(\underline{\breve{o}}_t | \underline{z}_t)$ where $\underline{\breve{o}}_t = (a_t, l_t, n_t)$. Importance particle weights are then multiplied by a correction factor that takes account of the dynamical model [5],

$$\frac{f(\underline{\breve{o}}_t^{(p)})}{g(\underline{\breve{o}}_t^{(p)} | \underline{z}_t)} = \frac{p(\underline{\breve{o}}_t^{(p)} | \mathcal{Z}_{t-1})}{g(\underline{\breve{o}}_t^{(p)} | \underline{z}_t)} = \frac{\sum_{i=1}^{P} \pi_{t-1}^{(i)} p(\underline{\breve{o}}_t^{(p)} | \underline{\breve{o}}_{t-1}^{(i)})}{g(\underline{\breve{o}}_t^{(p)} | \underline{z}_t)}. \tag{1}$$

To generate importance particles from a new observation $\underline{z}_t$, we submit the Kinect skeleton $\underline{b}_t$ to the SVM $\mathcal{S}$ to calculate a set of membership probabilities [2] for each action class, $\underline{m}_t = [m_{t,1}, ..., m_{t,15}]^\top$. The qth importance particle is then created as follows. First, we restrict our attentions to the set of top scoring classes $\underline{m}'_t \subset \underline{m}_t$ that together account for ≥ 0.5 of the probability mass in $\underline{m}_t$. We then select a new action class $a_t^{(q)}$ with probability proportional to the elements of $\underline{m}'_t$, a training index $n_t^{(q)} = \mu_{a_t^{(q)}}(\underline{b}_t, 0)$, and a randomly chosen limb predictor $l_t^{(q)} \in [1,2,...,19]$. The probability $g(\underline{\breve{o}}_t^{(q)} | \underline{z}_t)$, and the likelihood of generating such a particle given the previous set (see summation in Eq. 1) are then simple to calculate using $\underline{m}_t$, $p(\phi|\sigma^2)$ and $\mathbf{A}$.

4.4 Initialiser Particles

A small fraction of particles are also drawn from an initialisation prior. For this we use the importance function (Section 4.3), but do not reweight using Eq. 1. These particles aid error recovery and initialisation.

4.5 Measurement Density

To evaluate a particle's weighting $\pi_t^{(p)}$ we project the object into a chamfer image calculated from the Kinect image $\underline{r}_t$ and calculate a set of equally spaced sample points around its boundaries. These points are used to compute the sum of squared differences between the object edges and the image edges, see Fig. 1.

5 Experimental Results

All experiments were conducted on *unknown subjects*, meaning that we included no training data (body poses or object poses) from the subject being tested. The test subject's data was used once tracking was complete to evaluate localisation accuracy by computing the 2D distance in pixels between the centres of the bounding boxes (BBs) of the tracked and hand labelled implements. The tracked BB $\underline{\beta}_t$ is computed as the expected value of *all* particles' BBs, $\underline{\beta}_t = \sum_{p=1}^{P} \pi_t^{(p)} \times \underline{\beta}_t^{(p)}$. Where ≥ 0.5 of the particle set's probability mass was concentrated in a single class, we declared the object *known* and computed an expected object pose (see Fig. 2) from those particles alone, otherwise the object class was declared *unknown*.

We present initial distance results having labelled object poses for 2/11 subjects (30 sequences). The implications are thus: i) that we can only quantify errors for these two subjects' sequences; ii) that only the other labelled subject remains available for object pose training. We therefore present results for two unknown subject object-tracking scenarios, each with one subject in the training set. Note that all 10 non-test subjects' body pose data are still available to train the SVM $\mathcal{S}$.

We used $P = 100$ particles and state probabilities for {action, importance, initialiser} particles of $\{0.5, 0.4, 0.1\}$ (giving 3 identical rows of the transition matrix $\mathbf{T}$). The action transition matrix $\mathbf{A}$ was set to support only low probability inter-class transitions (equivalent to sampling $\phi > 3\sigma$) by setting $A_{ii} = 0.997$, and non-self transitions sharing the remaining

Table 1. Object tracking accuracy (pix) with classification precision in brackets.

	Proposed	[1]
`pourCream`	**47** ± **0** (83)	136 ± 108
`answerPhone`	**19** ± **3** (92)	215 ± 36
`drinkFromMug`	**16** ± **1** (89)	198 ± 1
`recordVideo`	**27** ± **4** (86)	307 ± 18
`liftWeights`	**18** ± **6** (93)	212 ± 9
`playFlute`	**12** ± **2** (90)	247 ± 13
`hitTennisBall`	**60** ± **18** (58)	157 ± 34
`hammerNail`	**46** ± **3** (63)	220 ± 5
`paintWall`	**38** ± **2** (63)	151 ± 70
`putOnGlasses`	**11** ± **1** (88)	286 ± 3
`shakeVinegar`	**19** ± **1** (78)	259 ± 16
`magnifyText`	**19** ± **0** (50)	255 ± 39
`putOnShoe`	134 ± 6 (46)	**92** ± **23**
`sweepFloor`	**33** ± **8** (96)	175 ± 28
`hitGolfBall`	**49** ± **1** (98)	162 ± 2

probability equally. Finally, we estimated a value of $\sigma = 7$ from the action database by calculating the average separation in $\mathcal{D}_a$ *in terms of Euclidean nearest neighbours* for every consecutive body pose pair from every subject. Table 1 shows a comparison with [1] based on the mean of the two average sequence errors for the two unknown subjects, and their individual separation from this mean. The typical height of the subjects is around 400 pixels. Across all sequences *known* objects were identified in 95% of observations, and the precision scores for these observations are given in brackets.

6 Discussion and Conclusions

Object localisation accuracy for the proposed approach is better than that available from the best performing adaptive tracker we have evaluated [1], although we must emphasise the test sequences are very challenging and that this baseline does not make use of any offline learning. The proposed approach can still confuse object classes where actions contain similar poses, particularly where subjects adopt resting poses after completing shorter actions e.g. `putOnShoe` and `putOnGlasses`. The fastest motions also produced motion blur on the objects, limiting the ability of the measurement density to resolve the object against background clutter, e.g. `hitTennisBall`. Future work could consider long-range temporal dynamics or more sophisticated observation densities to address these ambiguities. Completing object pose labelling on our dataset will also permit investigation into which limb predictors are most consistent across a wide range of subjects, and should therefore be favoured. In summary, we have presented a novel approach that does not restrict object pose estimates to be relative to one particular limb (e.g. hand [3]), removing sensitivity to localised errors in body pose estimation and any need for object location heuristics. It also produces full 3D object poses (see Fig. 2) and future work will evaluate other error metrics including 3D and rotational accuracy.

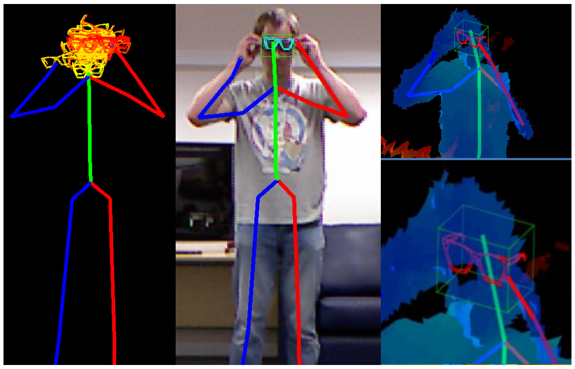

Figure 2. Example `putOnGlasses` result: (l) Object hypotheses; (c) Expected object pose; (r) Rotated view of 3D object pose.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 1–8, 2009.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. on Intell. Sys. and Technol.*, 2(3):27:1–27:27, 2011.

[3] J. Gall, A. Fossati, and L. van Gool. Functional categorization of objects using real-time markerless motion capture. In *CVPR*, pages 1969–1976, 2011.

[4] N. Ikizler and D. A. Forsyth. Searching for complex human activities with no visual examples. *IJCV*, 80(3):337–357, 2008.

[5] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *ECCV*, pages 893–908, 1998.

[6] M. Isard and A. Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In *ICCV*, pages 107–112, 1998.

[7] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1–8, 2011.

[8] A. L. Woodward. Learning about intentional action. In A. L. Woodward and A. Needham, editors, *Learning and the infant mind*, pages 227–248. OUP, 2009.