REAL-TIME MULTIPLE HEAD TRACKING USING TEXTURE AND COLOUR CUES

Vasil Khalidov Jean-Marc Odobez

Idiap-RR-02-2017

FEBRUARY 2017

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Real-time Multiple Head Tracking Using Texture and Colour Cues.

Vasil Khalidov and Jean-Marc Odobez
Idiap Research Institute
Centre du Parc, Rue Marconi 19, PO Box 592
CH-1920 Martigny, Switzerland
vasil.khalidov, odobez@idiap.ch

Abstract

We address the task of monocular visual head tracking in the context of applications that involve human-robot interactions, where both near field and far field tracking settings could occur and real-time constraints are imposed. The original contribution of this paper is a real-time multi-person tracking model that combines a priori texture and colour models for different head poses with face detectors for different face orientations. We show that such a combination improves tracker performance significantly. At the same time the proposed model takes into account major difficulties that are related to real-time data processing (non-uniform observations, processing time restrictions). The model is evaluated on a set of realistic scenarios recorded on a humanoid robot that involve interactions between the robot and the participants with robot motion, unconstrainted displacement of the participants, lighting variations etc. The algorithm runs in real-time and shows significant improvement of performance.

1. Introduction

Monocular tracking of multiple persons is a problem that is often addressed in various contexts: video surveillance, video conferencing, human-computer interaction, human-robot interaction, etc. In this paper we consider human-robot interaction (HRI) applications in which a robot interacts with a single person or with a group of persons. Such HRI scenarios possess the following properties, which make multiple person tracking a challenging task:

- **Moving persons** - persons are not seated as in video conferencing setting, they can move around, turn their head, gesticulate;
- **Moving robot** - the robot can gesticulate, turn its head and walk when holding a conversation;
- **Unconstrainted environment** - people can interact with robot at different (but reasonable) distance and in different lighting conditions (e.g. according to the time of the day).

These properties of HRI scenarios imply the following requirements to the tracking system:

- *Robustness* against lighting variations, appearance changes (that could be due to head rotations, change of the view angle of the robot, etc), human motion, robot motion and occlusions;
- *Initialization* and *destruction* of trackers should be accurate and timely;
- *Real-time* performance meaning that tracker guarantees certain performance and is able to work with video stream sampled irregularly and at speed, at which the tracker processes it.

There are not many state of the art trackers that would satisfy all of the mentioned requirements in the context of human-robot interaction scenarios. Among multiple face tracking systems that perform *tracking* or simultaneous *tracking and detection*, the one that probably suits the requirements the best is [3]. It provides a reasonable strategy for initialization and destruction of trackers, partly satisfies real-time constraints and is somewhat robust to occlusions and head rotations. However, this system was tested on sequences with controlled lighting conditions, stationary camera and seated people. For an overview of other multiple face tracking systems we refer to [3].

The state of the art single object trackers that adopt *tracking by detection* [6, 9] strategy could be used for multiple heads tracking by employing several instances of such trackers. However, their initialization, robustness to head rotations and real-time performance are questionable.

The approach presented in this paper adopts the *tracking and detection* paradigm. It formulates the tracking problem as continuous time discrete model and considers it in the *particle filtering* framework. This way we solve the problem of irregularly sampled video streams in the presence of human and robot motion. Our approach makes use of trained prior models of colour and testure features for various head poses to track in variable lighting conditions and tackle appearance changes. We employ tracker management techniques proposed in [3] to aid track creation and removal processes and handle occlusions.

This paper contains the following original contributions:

- *Continuous time discrete modelling* allows the tracking algorithm to work in systems where data arrives at irregular time instants;
- *Coupled detection and tracking* is achieved through the use of estimated trained detector statistics in the proposal distribution of the particle filter;
- *Coupled head pose estimation and tracking* allows the algorithm to perform head pose estimation at the same time as tracking;
- *Real-time performance* of the tracking algorithm is verified on the Vernissage database [5] that contains sequences recorded by a humanoid robot when giving a quiz scenario to a couple of people in a realistic setting.

The rest of the paper is organized as follows. The tracking model in formulated in Section 2. Details on particle filter implementation, likelihood definition and track management are given there. Section 3 summarizes the proposed approach in an algorithm. Section 4 describes the conducted experiments and Section 5 concludes the paper.

2. Tracking Using Colour and Texture Cues

We formalize the task of monocular multiple head tracking and formulate our approach based on colour and texture cues. We start with the description of various elements for single person tracking and track management in Sections 2.1 and 2.2, and proceed with extensions to multiple persons tracking in Section 2.3.

2.1. Tracking a Single Person

We consider the task of real-time head tracking based on a set of consecutive images originating from a single camera. There are several difficulties related to real-time data processing as opposed to offline processing in general, these are often ignored when addressing visual tracking tasks:

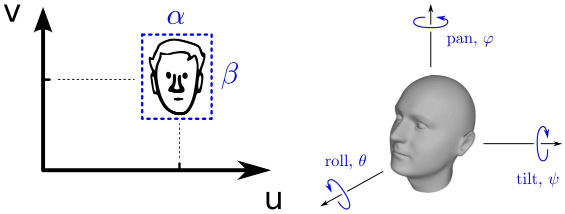

Figure 1. Head parameters $\boldsymbol{s}$ used in the tracking model: location (u, v), scale α and excentricity β of a head in a 2D image, and head pose defined by pan φ, tilt ψ and roll θ.

1. *Non-uniform observations*: time intervals between two consecutive observations may be irregular;
2. *Processing time influence*: with the increase of tracker processing time, difference between two consecutive observations starts to be more and more significant, their irregularity becomes even more tangible.

To explicitly account for the fact that camera images can arrive at non-uniform time intervals, we assume that observations $\boldsymbol{Y}_0, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n, \ldots$ are independent given the object states and are available at time instants $t_0, t_1, \ldots, t_n, \ldots$ respectively. The discussion of processing time influence is left till Section 3 where we discuss the complexity of the proposed tracking algorithm. We note that here and in what follows we use capital letters for random variables and small letters for their realizations.

Tracker state $\boldsymbol{s}$ is defined as

$$\boldsymbol{s} = \{u, v, \alpha, \beta, \varphi, \psi\}, \tag{1}$$

where (u, v) is head *location* on the 2D image plane, α and β are *scale* and *excentricity* parameters that determine 2D head shape and φ and ψ are *pan* and *tilt* head rotation angles. Figure **??** illustrates the state variables. We note that parameters $\{u, v, \alpha, \beta\}$ define a bounding box B associated with the tracker state.

We assume that tracker evolution is described by the following Ito stochastic differential equation (SDE):

$$\mathrm{d}\boldsymbol{S}(t) = \boldsymbol{\mu}(\boldsymbol{S}(t), t)\mathrm{d}t + \boldsymbol{\Sigma}(\boldsymbol{S}(t), t)\mathrm{d}\boldsymbol{W}(t), \tag{2}$$
$$0 \le t \le t_0, \quad \boldsymbol{S}(0) = \boldsymbol{s}_0, \tag{3}$$

where $\boldsymbol{W} = (\boldsymbol{W}(t))_{t\ge 0}$ is standard Brownian motion independent of the initial condition $\boldsymbol{s}_0$. Functions $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are assumed to be infinitely differentiable and bounded together with all their derivatives.

This formalization is known as continuous-discrete time model [7]. The tracking task is then formulated as optimal filtering density estimation

$$P(\boldsymbol{S}_n, t_n \mid \boldsymbol{y}_0, t_0; \boldsymbol{y}_1, t_1; \ldots, \boldsymbol{y}_n, t_n). \tag{4}$$

Since functions $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in (2) are not known in advance, we cannot derive the exact solution for (4).

We consider discretized version of (2) and apply *particle filtering* [1] to approximate the optimal filtering density (4) by a set of weighted particles. More precisely, we adopt the sequential importance sampling strategy and take the importance density $\pi(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{0:n-1}, t_{0:n-1}, \boldsymbol{y}_{0:n})$ as a mixture of the dynamic model $P_{1|1}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1})$ and image-based proposal distribution $P_{\text{I}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1})$:

$$\begin{aligned}\pi(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{0:n-1}, t_{0:n-1}, \boldsymbol{y}_{0:n}) = \\ \gamma_{\text{D}} P_{1|1}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}) + \\ \gamma_{\text{I}} P_{\text{I}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1}),\end{aligned} \tag{5}$$

where $\gamma_{\text{D}} + \gamma_{\text{I}} = 1$ and functions $P_{1|1}$ and P_{I} are defined below. Hence, the weight w_n^i of a particle i with state $\boldsymbol{s}_n^i$ at time t_n after sampling is computed as

$$w_n^i = w_{n-1}^i \frac{P(\boldsymbol{y}_n|\boldsymbol{s}_n^i) P_{1|1}(\boldsymbol{s}_n^i, t_n|\boldsymbol{s}_{n-1}^i, t_{n-1})}{\pi(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{0:n-1}, t_{0:n-1}, \boldsymbol{y}_{0:n})}. \tag{6}$$

The rationale behind such a choice of the importance density is that we try to use both, dynamics and the observed data in order to approximate the optimal proposal distribution $P(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{s}_{n-2}, t_{n-2}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1})$.

Dynamic Model As prior on the state process, the choice of dynamics is important to constrain the estimation and avoid tracking failure. In our tracking framework, it is even more important since it used as a part of the proposal distribution to explore the state space during optimization. However, people's motion is difficult to predict: they may remain relatively static when interacting with other people or a robot. When they move around, they can have a constant speed. Finally, we can also observe abrupt motion changes at motion transitions, or due to sudden and fast motion of the robot. Accordingly, to handle all these situations, we have defined the dynamical model as a mixture of two elements:

$$\begin{aligned}P_{1|1}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}) = \\ \gamma_{\text{RS}} P_{\text{RS}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}) + \\ \gamma_{\text{SB}} P_{\text{SB}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}),\end{aligned} \tag{7}$$

where $\gamma_{\text{RS}} + \gamma_{\text{SB}} = 1$ and the distributions P_{RS} and P_{SB} are defined as follows. The first one is a *random search* which accounts for a no-motion situation:

$$P_{\text{RS}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}) = \mathcal{N}(\boldsymbol{S}_n;\ \boldsymbol{s}_{n-1}, \Delta t_n \Sigma), \tag{8}$$

and the second one is a *random search with state-based velocity estimates*:

$$P_{\text{SB}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}) = \mathcal{N}(\boldsymbol{S}_n;\ \boldsymbol{s}_{n-1} + \Delta t_n \boldsymbol{\mu}_n, \Delta t_n \Sigma), \tag{9}$$

that accounts for constant speed motion, where $\Delta t_n = t_n - t_{n-1}$ is a time interval between the two states, $\mathcal{N}$ is a probability density of a multivariate Gaussian distribution and $\boldsymbol{\mu}_n$ is an estimate of the drift function $\boldsymbol{\mu}$ at t_n taken as $\boldsymbol{\mu}_n = (\boldsymbol{s}_{n-1} - \boldsymbol{s}_{n-2})/(t_{n-1} - t_{n-2})$, for the state estimates $\boldsymbol{s}_{n-1}$ and $\boldsymbol{s}_{n-2}$ obtained at t_{n-1} and t_{n-2} respectively.

Image-based Proposal Distributions Dynamic models defined above are entirely based on random search and state statistics. However, as mentionned above, there are also abrupt speed changes that are difficult to predict based only on past information, and which are the situations that often lead to failure. Indeed, in these cases, it is more appropriate to directly exploit the information contained in the images, and which are of two different natures: instantaneous observations reflecting the presence of the object, as produced by a *face detector*; and sequential observations reflecting observed *image-based motion* between frames. Thus, the image based proposal distribution is defined as

$$\begin{aligned}P_{\text{I}}(\boldsymbol{S}_n, t_n|\boldsymbol{y}_n, \boldsymbol{y}_{n-1}) = \\ \gamma_{\text{ID}} P_{\text{ID}}(\boldsymbol{S}_n, t_n|\boldsymbol{y}_n) + \\ \gamma_{\text{IM}} P_{\text{IM}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1}),\end{aligned} \tag{10}$$

where $\gamma_{\text{ID}} + \gamma_{\text{IM}} = 1$ and face detector-based $P_{\text{ID}}(\boldsymbol{S}_n, t_n|\boldsymbol{y}_n)$ and motion-based $P_{\text{IM}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1})$ proposal distributions are defined as follows. Face detector-based proposal distribution is given by

$$P_{\text{ID}}(\boldsymbol{S}_n, t_n|\boldsymbol{y}_n) = \mathcal{N}(\boldsymbol{S}_n;\ \boldsymbol{s}_{\text{ID}}(\boldsymbol{y}_n), \Gamma_{\text{ID}}), \tag{11}$$

where $\boldsymbol{s}_{\text{ID}}(\boldsymbol{y}_n)$ denotes the closest face detection associated with the track (when it exists) and Γ_{ID} is the corresponding covariance matrix. Proximity of face detection bounding box B_{ID} to tracker bounding box B_n is evaluated based on their F-measure:

$$F(B_1, B_2) = \frac{2a(B_1 \cap B_2)}{a(B_1) + a(B_2)}, \tag{12}$$

where $a(\cdot)$ denotes the area operator. This measure computes the intersection area as a fraction of average area of the two bounding boxes. In what follows we also apply F-measure to tracker states $F(\boldsymbol{s}_1, \boldsymbol{s}_2)$ assuming that it is calculated for the corresponding bounding boxes $F(B_1, B_2)$. The covariance matrix Γ_{ID} is evaluated offline based on face detector statistics.

Motion-based proposal distribution is given by

$$\begin{aligned}P_{\text{IM}}(\boldsymbol{S}_n, t_n|\boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1}) = \\ \mathcal{N}(\boldsymbol{S}_n;\ \boldsymbol{s}_{\text{IM}}(\boldsymbol{s}_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1}), \Gamma_{\text{IM}}),\end{aligned} \tag{13}$$

where $\boldsymbol{s}_{\text{IM}}(\boldsymbol{s}_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1}) = \boldsymbol{s}_{n-1} + \boldsymbol{\mu}_n^{\text{IM}}(\boldsymbol{y}_n, \boldsymbol{y}_{n-1})$ is a state predicted from the image motion with the corresponding covariance matrix Γ_{IM}. This motion is measured using

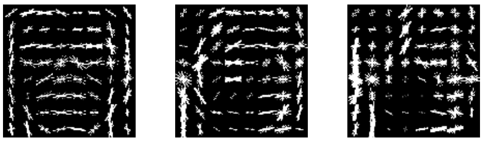

Figure 2. Examples of trained texture models corresponding to pan (φ) angle values of 0, 45 and 90 degrees and tilt (ψ) angle value 0. The representation is based on histograms of oriented gradients (HOGs).

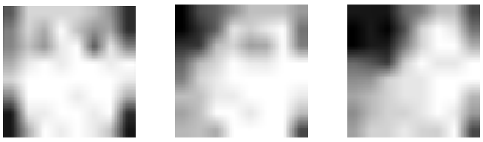

Figure 3. Examples of trained colour patterns corresponding to pan (φ) angle values of 0, 45 and 90 degrees and tilt (ψ) angle value 0.

a robust parametric motion model [8] estimator applied to an image patch around the estimated head location in the previous frame to calculate the displacement field at every pixel and derive $\boldsymbol{\mu}_n^{\text{IM}}$.

Thus the dynamics distribution now takes into account both, the low-level information on correlated motion and high-level information on the detected template, which makes sampling more efficient.

Likelihood Distributions We adopt a classification based approach to head pose estimation, i.e. feature templates are trained based on head pose database [4] that contains images of 15 persons taken at different discretized pan (φ) and tilt (ψ) angles, 13 pan and 7 tilt angle values in ranges $[-90, 90]$ and $[-60, 60]$ respectively, are taken. We use two kinds of features in our experiments to characterize tracker's state: *texture* $\boldsymbol{y}^{\text{tex}}$ and *colour* $\boldsymbol{y}^{\text{col}}$.

Texture likelihood. Texture features $\boldsymbol{y}^{\text{tex}}$ are computed using multiscale descriptors based on histograms of oriented gradients (HOG) [2]. They are trained using a head pose database [4] that contains images of 15 persons taken at different discretized pan (φ) and tilt (ψ) angles. Some examples of the set of trained texture descriptors $\bar{\boldsymbol{y}}^{\text{tex}}(\varphi, \psi)$ are shown in Figure 2. The likelihood function of an observation $\boldsymbol{y}^{\text{tex}}$ given the state $\boldsymbol{s}$ is then defined as

$$P(\boldsymbol{y}^{\text{tex}}|\boldsymbol{s}) \;=\; \exp\left\{-\lambda^{\text{tex}} d_{\text{tex}}^2(\boldsymbol{y}^{\text{tex}}, \bar{\boldsymbol{y}}^{\text{tex}}(\varphi, \psi))\right\}, \quad (14)$$

where λ^{tex} is the texture modality weight and we take $d_{\text{tex}}(\boldsymbol{y}_1^{\text{tex}}, \boldsymbol{y}_2^{\text{tex}})$ as a componentwise thresholded L_2 distance function.

Skin likelihood. Features based on skin colour $\boldsymbol{y}^{\text{col}}$ are also used to characterize image patches. We applied colour models trained on frontal face images [10] to classify pixels as skin or non-skin. Again, the descriptors were trained on a head pose database [4] for different discretized pan (φ) and tilt (ψ) angles. Some examples of the set of trained colour descriptors $\bar{\boldsymbol{y}}^{\text{col}}(\varphi, \psi)$ are shown in Figure 3. We employ the following likelihood function of an observation $\boldsymbol{y}^{\text{col}}$ given the state $\boldsymbol{s}$:

$$P(\boldsymbol{y}^{\text{col}}|\boldsymbol{s}) = \exp\left\{-\lambda^{\text{col}} d_{\text{col}}^2(\boldsymbol{y}^{\text{col}}, \bar{\boldsymbol{y}}^{\text{col}}(\varphi, \psi))\right\}, \quad (15)$$

where λ^{col} is the colour modality weight and we take $d_{\text{col}}(\boldsymbol{y}_1^{\text{col}}, \boldsymbol{y}_2^{\text{col}})$ as an empirical score function based on L_2 distance between the features.

2.2. Single Person Track Management

To achieve proper tracker management, we rely on the long term tracking framework [3]. We introduce long-term tracker manager that helps to detect conditions under which a tracker should be initialized or suppressed. We use face detector to initialize tracks and the tracker manager makes the initialization process more robust to false detections. In case of tracking failure it helps to detect and remove the problematic tracker from the system.

2.3. Multiple Person Tracker

So far the model has been described for the case of a single person tracker. In the case of multiple trackers, one cannot simply assume independent state evolution processes (2), since persons may occlude each other and two different trackers may end up tracking the same person.

Thus following [3] we introduce an additional interaction term to the overall system dynamics. Let's denote $\boldsymbol{X}_n = \{\boldsymbol{S}_n^{(1)}, \ldots, \boldsymbol{S}_n^{(K)}\}$ the overall system's state consisting of K trackers. Then

$$P_{1|1}(\boldsymbol{X}_n, t_n \mid \boldsymbol{X}_{n-1}, t_{n-1}) \propto \prod_{k_1 \neq k_2} \exp(-\lambda F(\boldsymbol{S}_n^{(k_1)}, \boldsymbol{S}_n^{(k_2)})) \times \prod_{k=1}^{K} P_{1|1}(\boldsymbol{S}_n^{(k)}, t_n \mid \boldsymbol{S}_{n-1}^{(k)}, t_{n-1}), \quad (16)$$

where $F(\boldsymbol{S}_n^{(k_1)}, \boldsymbol{S}_n^{(k_2)})$ is the F-measure function defined by (12).

We note that using the same arguments as in [3], we assume that observations are conditionally independent given states, which seems a reasonable approximation in our scenarios in the presence of tracker manager.

3. Tracking Algorithm

Given the model described in Section 2, we formulate the tracking algorithm. Suppose that at time instant t_{n-1}

the model contains $K \geq 0$ trackers. Then at the next time instant t_n when an image $\boldsymbol{y}_n$ from the camera is available, we perform the following iteration:

1. Detect faces on image $\boldsymbol{y}_n$ and estimate motion between $\boldsymbol{y}_{n-1}$ and $\boldsymbol{y}_n$ to define face detector-based $P_{\text{ID}}(\boldsymbol{S}_n, t_n | \boldsymbol{y}_n)$ and motion-based $P_{\text{IM}}(\boldsymbol{S}_n, t_n | \boldsymbol{s}_{n-1}, t_{n-1}, \boldsymbol{y}_n, \boldsymbol{y}_{n-1})$ proposal distributions given by equations (11) and (13);
2. Assign faces to trackers based on overlap measure (12);
3. Sample particles using the proposal distribution (5);
4. Assign weights to particles following (6) and using system dynamics (16) and likelihood functions (15);
5. Apply tracker manager (Section 2.2), update tracker statistics, keep or remove existing trackers based on these statistics, select candidates for tracker initialization;
6. Select one candidate for tracker initialization (if candidates were proposed by the manager) and initialize the tracker using distribution over the state space $P_{\text{FD}}(\boldsymbol{S}_n, t_n)$ associated with the detector that generated the candidate;
7. Adapt colour models for those trackers, for which the associated detector results are available.

4. Experimental Results

The goal of this paper is to show how colour models based on semantic segmentation can be used to perform head tracking by a mobile robot in challenging scenarios involving human-robot interactions. The typical requirements to the tracking system in such scenarios are:

1. *Robustness* against lighting variations, appearance changes (that could be due to head rotations, change of the view angle of the robot, etc), human motion, robot motion and occlusions;
2. *Initialization* and *destruction* of trackers should be accurate and timely;
3. *Real-time* performance meaning that tracker guarantees certain performance and is able to work with video stream sampled at speed, at which the tracker processes it.

We analyzed the performance of our tracker on Vernissage database [5] that contains a set of realistic human-robot interaction scenarios where the robot Nao[1] gives a quiz to a couple of persons. The sequences are recorded with robot's camera, every video frame is timestamped. External view of the Vernissage database setting, Nao robot's head with sensors and an example of video frame recorded with the left camera are shown in Figure 4. Four sessions were taken for evaluation: "slot09", "slot19", "slot24" and "slot30". They contain monocular (left camera) recordings of about 10 minutes long each, thus the total length of the considered data is about 40 minutes. Head locations for both persons in each session were annotated. Some statistics of the annotated data are given in Table 1.

	sequence length (s)	# annotated frames	# annotated heads
slot 09	667.52	1122	2150
slot 19	767	737	1427
slot 24	651	636	1216
slot 30	702	692	1367
Overall	2787.52	3187	6160

Table 1. Statistics on evaluation data sequences from the Vernissage database used in the experiments: sequence duration is seconds, number of annotated frames and number of annotated objects (heads) are provided.

Sequences from the Vernissage database present several challenges to any tracking algorithm. Firstly, they were shot at different times of the day and in different lighting conditions. Some contain more of diffused sunlight, some have mostly indoor lighting. Secondly, persons appear at different depths and move their heads which results in high appearance variations. Finally, the robot performs motions, nods and turns its head, so that captured images become blurry, persons' positions in an image can change rapidly, persons often disappear from the field of view of the robot and can stay out of the field of view for several seconds. This way tracker performance on Vernissage database reflects well its expected performance in real life scenarios involving human-robot interactions.

Our evaluations are based on well-established metrics for tracker performance estimation [11]. We would like to track the target as long as possible and stop tracking as soon as object disappears or tracking failure occurs. Thus we adopt the *recall* and *false positive* (FP) metrics:

$$R = |TP|/|GT| \quad \text{and} \quad FP = (N - |TP|)/|GT|, \quad (17)$$

where $|TP|$ is the number of true positives (i.e. cases where reported tracked heads matched ground truth), $|GT|$ is the amount of ground truth heads in frames and N is the total amount of reported tracked heads in those frames. Thus recall measures how much of GT is covered by estimated

[1] www.aldebaran-robotics.com

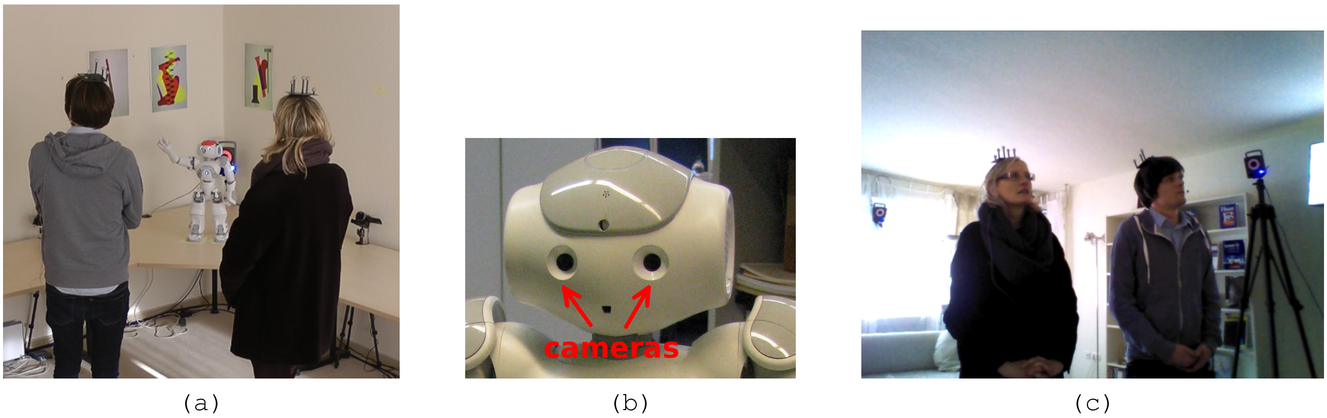

Figure 4. Vernissage database setting. (a) External view of the scene, Nao robot gives a quiz to a couple of persons; (b) Nao robot's cameras that were used to make the recordings; (c) A typical image from Nao robot's cameras.

tracks and FP rate shows the amount of unassociated tracks / track parts. An estimate bounding box B_{est} is considered to match the ground truth bounding box B_{GT} if their overlap given by the F-measure function (12) exceeds 0.1: $F(B_{\text{est}}, B_{\text{GT}}) > 0.1$.

We report on several versions of tracking algorithm: the tracking algorithm presented during the year 2 review (Y2) and a version of the tracking algorithm that corresponds to this report (Y3). Tracking results are given in Table 2. For each sequence we give statistics (mean and standard deviation values) on recall, false positive rate (FP rate) and the number of interruptions (# int). The statistics are based on 10 runs of the tracking algorithm. We notice that due to high variability of lighting conditions and person behaviours and appearances, the results differ much from one sequence to another.

First, we note that the results for qVGA resolution videos are partially absent. This is because face detections in slot 19 and slot 24 are not frequent at this image resolution given the lighting conditions. Trackers do not get initialized at all or get removed almost immediately because of robot's motion. Noticeable improvement from 42% to 82% of recall has been achieved on slot 30 at the cost of 9 additional interruptions. However, on slot 09 recall measures stayed the same and the number of interruptions increased significantly.

The VGA version of the algorithm shows average improvement of 10% in terms of recall for all the slots at the cost of slight increase of false positive rate and number of interruptions.

The computations for qVGA and VGA versions of Y2 tracker were done at 14-16 FPS and 3-5 FPS respectively. These processing rates include all the computation done within one algorithm iteration, i.e. face detection, motion estimation and tracking. Real-time processing could be made only with the Y2 qVGA version.

The processing rates for qVGA and VGA versions of Y3 tracker were significantly improved: 27-30 FPS and 13-15 FPS respectively. Thus real-time processing can now be performed on the VGA version as well.

5. Conclusion

We introduced an approach to real-time multiple head tracking using texture and colour features in the context of human-robot interaction scenarios. This approach follows the *tracking and detection* paradigm, efficiently combining information from face detectors with the appearance cues and performing Bayesian filtering. Y2 and Y3 versions of the proposed model were tested on sequences from the Vernissage database that contains a set of realistic human-robot interaction scenarios recorded by robot's camera. Tracker performance in challenging conditions of these recordings (lighting variations, persons' motion, robot's motion, variations in person's appearances) reflects its potential performance in real-life human-robot interaction scenarios.

The results show that tracking quality depends on lighting and appearance conditions and does not fluctuate much between different runs on the same sequence. Tracker performance on VGA resolution sequences increased from 79% in average to 90% in average with small increases in false positive rate (2%) and nuisance interruptions (12). VGA tracker speed was improved from 3-5 FPS to 13-15 FPS, so that it became possible to use VGA tracker in real-time.

Tracker running on qVGA images showed speed improvement from 14-16 FPS to 27-30 FPS. However, the tracking quality is significantly lower than that of VGA

		Y2 VGA		Y2 qVGA		Y3 VGA		Y3 qVGA	
		mean	stddev	mean	stddev	mean	stddev	mean	stddev
slot 09	Recall	0.761	0.0178	0.65	0.0115	**0.859**	0.0062	0.641	0.0124
	FP rate	0.020	0.0055	0.01	0.0061	0.068	0.0043	0.053	0.0844
	# int	44	3.7683	38.6	4.1521	51.333	2.1602	188.2	9.368
slot 19	Recall	0.871	0.0085	–	–	**0.901**	0.0057	–	–
	FP rate	0.028	0.0043	–	–	0.051	0.0044	–	–
	# int	44.9	1.6401	–	–	63.444	3.7450	–	–
slot 24	Recall	0.605	0.0225	–	–	**0.849**	0.0089	–	–
	FP rate	0.004	0.0006	–	–	0.07	0.0056	–	–
	# int	57.1	3.8846	–	–	80.1	5.1078	–	–
slot 30	Recall	0.94	0.0073	0.418	0.0248	**0.977**	0.0034	0.823	0.0272
	FP rate	0.102	0.187	0	0.0007	0.043	0.0112	0.002	0.001
	# int	35	2.3238	14.2	2.4819	34.9	1.6401	23.4	2.7276
Overall	Recall	0.794	0.1275	0.267	0.2796	**0.897**	0.0518	0.39	0.3485
	FP rate	0.039	0.1009	0.003	0.0053	0.058	0.0134	0.014	0.0478
	# int	45.25	8.4343	13.2	15.9534	57.447	17.2683	55.425	77.1799

Table 2. Evaluation of head tracking algorithms Y2 and Y3 running at different video resolutions. For each algorithm mean and standard deviation of recall, false positive rate (FP rate) and number of interruptions (# int) are given.

tracker. Thus VGA tracker stays the best option for real-time tracking.

6. Acknowledgements

The authors gratefully acknowledge the financial support from the HUMAVIPS project, funded by the European Commission Seventh Framework Programme, Theme Cognitive Systems and Robotics, Grant agreement no. 247525

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, pages 174–188, 2002.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] S. Duffner and J.-M. Odobez. Track creation and deletion framework for long-term online multiface tracking. *IEEE Trans. on Image Processing*, 22(1):272–285, 2013.

[4] N. Gourier, D. Hall, and J. L. Crowley. Estimating Face Orientation from Robust Detection of Salient Facial Features. In *Pointing 2004, ICPR international Workshop on Visual Observation of Deictic Gestures*, pages 183–191, 2004.

[5] D. B. Jayagopi, S. Sheikhi, D. Klotz, J. Wienke, J.-M. Odobez, S. Wrede, V. Khalidov, L. S. Nguyen, B. Wrede, and D. Gatica-Perez. The vernissage corpus: A conversational human-robot-interaction dataset. Tokyo, Japan, 2013.

[6] Z. Kalal, J. Matas, and K. Mikolajczyk. Online learning of robust object detectors during unstable tracking. In *Proc. on IEEE Int. Conf. on Comp. Vision Workshops*, pages 1417–1424, 2009.

[7] S. Lototsky and B. Rozovskii. Recursive nonlinear filter for a continuous discrete-time model: separation of parameters and observations. *IEEE Trans. on Automatic Control*, pages 1154 – 1158, 1998.

[8] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Visual Communication and Image Representation*, 6(4):348–365, 1995.

[9] F. Pernici. Facehugger: The alien tracker applied to faces. In *ECCV Workshops*, pages 597–601, 2012.

[10] C. Scheffler and J.-M. Odobez. Joint adaptive colour modelling and skin, hair and clothing segmentation using coherent probabilistic index maps. In *British Machine Vision Conference*. British Machine Vision Association, Sept. 2011.

[11] K. Smith, D. Gatica-Perez, S. Ba, and J.-M. Odobez. Evaluating Multi-Object Tracking. In *CVPR Workshop on Empirical Evaluation Methods in Computer Vision*, San Diego, June 2005.