DE_deliverable D1.2

RECOGNITION OF HAND GESTURES, FACIAL EXPRESSIONS, AND CONVERSATIONAL STATES

Maria Pateraki (FORTH), Markos Sigalas (FORTH), Panos Trahanias (FORTH)

Beneficiaries: FORTH (lead)
Workpackage: WP1: Visual Perception of Human Activity
Description: Perception of human multimodal social signals in the JAMES scenario is one of the main challenges of the project. JAMES visual perception system emphasizes the capabilities which are directly related to the development of the robot’s social interaction skills in dynamic scenes, such as (a) tracking of humans and their individual body parts as well as objects, (b) extracting pose related information from body and face and (c) recognizing communicative (waiving, pointing) and manipulative (grab /holding an object) gestures. These capabilities are in agreement with the results derived from WP8 and which refer to the nonverbal behaviour that humans utilize to order a drink and define clear visual cues to be integrated in the visual perception system.
This document reports the work done in Tasks 1.2 and 1.3. It gives details regarding the individual algorithms and methodologies that have been developed and presents the achieved results.

Version: Final
Nature: Report (R)
Dissemination level: Public (PU)
Pages: 16
Date: 2014-01-31
## Contents

1 Executive Summary 3

2 Requirements 3

3 Body Pose Estimation 4
   3.1 Torso Pose Estimation 4
      3.1.1 Model-based Torso pose estimation 4
      3.1.2 User Top View Approach 6
   3.2 Arm Pose Estimation 8

4 Head Pose Estimation 10

5 Recognition of hand gestures 11
   5.1 Recognition of communicative gestures 11
   5.2 Recognition of manipulative gestures 12

6 Visual speech recognition 13

7 Facial expression recognition 14

8 3D Object pose tracking 14

9 Conclusions 15
1 Executive Summary

JAMES visual perception system emphasizes the capabilities which are directly related to the development of the robot’s social interaction skills in dynamic scenes, such as (a) tracking of humans and their individual body parts as well as objects, (b) extracting pose related information from body and face and (c) recognizing communicative (waiving, pointing) and manipulative (grab /holding an object) gestures. These capabilities are in agreement with the results derived from WP8 on “Multimodal Data Collection” and which refer to the nonverbal behaviour that humans utilize to order a drink, and therefore defining clear visual cues to be integrated in the visual perception system.

In Task 1.2 appropriate algorithms are to be created in order to recognize a set of scenario-specific communicative and manipulative hand-gestures utilizing approaches that recognize gestures using motion patterns. Building on the work of the probabilistic tracker described in deliverable D1.1 as well as in [1, 2, 3], hand trajectories as well as arms’ poses (sec. 3.2) are forwarded to the hand gesture recognition system for communicative gestures (sec. 5.1), such as waiving and pointing. For the recognition of manipulative gestures (sec. 5.2) that act on objects both the hand and object motion patterns are combined to detect whether the user grabs or holds a bottle. Bottles as objects are tracked based on their color and shape characteristics. With respect to object tracking we have further investigated the 3D object pose tracking problem (with six degrees of freedom) from monocular sequences (sec. 8) and the research results have been shared with the EU-funded FIRST-MM project (http://www.first-mm.eu/).

The objective in Task 1.3 is to automatically analyze and recognize hand and facial motions and facial feature changes from visual information in order to convey the intention of the person in terms of communicative signs. During the course of the project, and based on the results of WP8, the focus of attention of a user approaching the bar has been of main concern, thus both torso and head pose were characterized as important attentive cues. Therefore, our efforts have been focused on extracting body pose information from torso and arms (sec. 3.1 and 3.2) of multiple users without the explicit (by having the user to stand at a specific pose, e.g. T-pose), or implicit requirement (a certain amount of frames to register the user) of an initialization phase and at the same time handling efficiently the problem of occlusions. The face pose estimation (sec. 4) follows a data-driven model-based tracking method and the inherent advantage is its ability to effectively cope with challenging illumination conditions and inter-(e.g. biological variation) and intra-personal changes (i.e. expression) without prior training. Although, facial expression recognition and visual speech detection have not been characterized as means of clear and detectable nonverbal behaviour that customers display when they want to initiate a drink ordering interaction’ relevant work has been conducted.

This document reports the work done in Tasks 1.2 and 1.3 according to the above. It gives details regarding the individual algorithms and methodologies that have been developed and presents the achieved results. Moreover, preliminary results on facial expression recognition and visual speech detection are reported. Part of the JAMES vision system has been presented and documented in conferences and journal publications [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14].

2 Requirements

The operational requirements of the vision system are most challenging. The JAMES system should operate under difficult conditions regarding occlusions, variable illumination and with a number of persons in the field of view of the robot with different intra- and inter-personal characteristics. The following list summarizes the most important operational requirements of the vision system:

- The tracker must be able to handle a number of people that may be in front of the robot, interacting with it. These people may be occluding each other very heavily.
- The vision system must handle difficult environmental conditions such as lighting variability and dynamic background.
• Both body and face pose estimation should exhibit robustness and invariance in scene- and image-changing factors and should not depend on any explicit or implicit requirement of an initialization phase to register the user and assume recovery of pose parameters.

• Take into consideration the diversity in human appearance for the extensibility of the methods beyond semi-controlled environments and persons with specific visual characteristics.

• All algorithms must be able to perform in real time.

3 Body Pose Estimation

In a large number of body pose tracking methodologies, there is the inherent requirement of an initialization phase in order to start tracking the body pose. Initialization can be done either explicitly, by having the user to stand at a specific pose (e.g. T-pose), or implicitly, requiring a certain amount of frames to register the user. However, JAMES scenario is a real life scenario where users move, act and interact freely and an initialization phase is not always possible. In such cases of naturalistic interactions, the problem of frequently occurring intra- and inter-person occlusions, namely occlusions imposed among body parts of the same user or among different users, respectively, may additionally deteriorate performance. In fact, estimating the human pose when a person is partially or heavily occluded in the scene remains a challenging, open problem.

We are interested in estimating the upper body configuration, including torso and arms, under the assumption that no initialization phase is possible and that the pose recovery and tracking should remain unaffected from partial intra- and inter-person occlusions. The employed upper body model consists of five parts; the torso and the two (left-right) upper and forearms, which are modeled and tracked independently utilizing RGB and depth information from the Kinect sensor.

3.1 Torso Pose Estimation

The basic torso pose estimation is described in the following section and the latest extension of the method is described in section 3.1.2. The method is fully integrated in the JAMES system and its latest extension will also be part of the final system.

3.1.1 Model-based Torso pose estimation

An overall schematic representation of the steps employed in our method for human-torso 3D pose recovery is given in Fig. 1. Initial face identification (Task 1.1) triggers a segmentation step that delineates the human-body area. Based on that, approximation and 3D modeling of shoulder joints is then performed, which subsequently steers modeling -via a non-linear regression schema- of the torso area as a 3D ellipsoid. The latter is used to extract the torso 3D pose parameters.

More specifically, the major steps of the proposed approach are:

• Agent Segmentation. Based on face detection and tracking, the human body silhouette is extracted for the detected users in the scene.

• Shoulder joint approximation. Given the location of the face, we select sets of points on the RGB silhouette, delineating possible shoulder or armpit areas. Selection is based on pose and scale invariant features satisfying certain geometric constraints. The selected silhouette points are used to define the 2D area of the shoulder joint and thus, using the depth information, the 3D shoulder joint point cloud. Shoulder joints are approximated by least squares fitting of 3D spheres on the selected areas on the point cloud and a set of anthropometric criteria is used to evaluate the estimation and eliminate possible outliers.
Figure 1: Methodology overview. Assuming the location of the face, the user is segmented from the rest of the scene and his silhouette is extracted. Points along the silhouette are then selected and used to estimate the location of the shoulder joints. Finally, the resulting shoulder joints define the area of the user's torso, on which the ellipsoidal torso model is fitted, in order to infer the upper body pose.

- **Body pose estimation.** Driven by the location and orientation of the shoulder joints, a set of 3D points, approximately along the user's body, is selected and used for the torso approximation. A custom iterative algorithm, resembling gradient descent optimization, is used to fit the model on the selected point cloud, during the regression step.

### Table 1: Comparative statistics.

<table>
<thead>
<tr>
<th>Test case</th>
<th>DP</th>
<th>$\mu E$</th>
<th>$\sigma E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.43%</td>
<td>88.07%</td>
<td>6.73°</td>
</tr>
<tr>
<td>2</td>
<td>97.06%</td>
<td>25.35%</td>
<td>6.69°</td>
</tr>
<tr>
<td>3</td>
<td>92.30%</td>
<td>83.81%</td>
<td>4.88°</td>
</tr>
<tr>
<td>4</td>
<td>99.47%</td>
<td>77.02%</td>
<td>6.70°</td>
</tr>
<tr>
<td>5</td>
<td>99.47%</td>
<td>0%</td>
<td>6.70°</td>
</tr>
<tr>
<td>6</td>
<td>99.47%</td>
<td>0%</td>
<td>6.70°</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>91.56%</strong></td>
<td><strong>78.38%</strong></td>
<td><strong>6.77°</strong></td>
</tr>
</tbody>
</table>

DP = percentage of frames where the orientation was estimated, $\mu E$ = mean orientation error throughout the sequence, $\sigma E$ = standard deviation of error.

The method has been evaluated in a series of experiments, with varying difficulty levels in the lab and in the JAMES environment. The experiments involved both single and multiple users, conducting several poses in various relative to the camera positions and orientations, in cluttered environments. In the lab experiments we acquired 6 sequences (5000 frames in total) and we have attached markers on the shoulders of the users, in order to obtain ground truth information for quantitative analysis purposes. We have further compared our method to the Nite skeletonization module of OpenNI [15] using the ground truth information. The quantitative analysis is presented in Table 1 and indicative results are depicted in Fig. 2. It can be seen that our method achieved a very small average error of around 7 degrees, estimated the pose after the user entered scene in less frames compared to Nite and resulted in high detection rates (> 90%). On the other hand, Nite couldn't efficiently cope with the initialization problem, which consequently led to either low detection rates (e.g. 25%) or no detection at all for the whole sequence.
Results with multiple people and different interactions as well in the JAMES environment are shown in Fig. 3. The method can achieve real-time execution, with no multi-threading or gpu programming, on a standard pc, reaching a speed of around 25 frames per second for a single user and 18 frames per second for two users in the scene. Furthermore, it has been thoroughly described in [5, 8] and has been live demonstrated in the Demo Session of CVPR 2013 [10].

### 3.1.2 User Top View Approach

We have extended the above method to increase its performance under severe occlusions. Although there are some works which deal efficiently with self-imposed occlusions, coping with inter-occlusions among multiple persons is still problematic. To cope with these shortcomings we introduced and formulated the novel concept of User Top View (UTV) as a robust indicator of the 3D body pose. Additionally, ordered rendering of each user together with the employed kinematic model, enables us to robustly cope with collisions and occlusions among
different parts or different users. The extended version of the body pose estimation, including the estimation of the arm’s poses is thoroughly discussed in [7].

Once users are detected and segmented from the scene, the depth of each face centroid is used to determine the ordering of user evaluation, and provide an initial estimation for possible existing occlusions. Then, we estimate the User Top View (UTV), a hypothesized view aligned to the axis of the user, based on the minimum projection ratio criterion. UTV is used to approximate the torso pose and, thus, the location of the two shoulders. Shoulders locations, in conjunction with the detected palms and a set of anthropometric proportions, are used to generate a set of configuration hypotheses for each arm, tracked by a separate particle filter. Ray tracing is used to render each of the body parts (particles) and to detect and handle occlusions or collisions with prior evaluated users and parts. Possible detected occlusions and collisions are further used, together with the kinematic model workspace, to constraint the hypotheses space. Finally, a hypothesized depth map is generated for each arm configuration, which is compared against the observed one in order to evaluate the hypothesis, resulting to a full upper body pose estimation. Briefly the extended methodology is depicted in Fig. 4 and can be summarized into:

- **Agent segmentation and ordering.** Based on face and palm detection, users are ordered, segmented from the scene and checked for possible occlusions. Each user will be examined according to the resulting depth order.

- **Torso pose estimation.** UTV is used to approximate the torso configuration. 2D ellipse approximation, on the hypothesized view projections, is used to estimate the torso’s vertical orientation and, thus, locate the shoulder joints.

- **Arm hypotheses generation.** Given the shoulder positions and the detected palms we generate a set of arm hypotheses, constrained by the kinematic model (see sec. 3.2).

- **User rendering and hypothesis evaluation.** Ray casting is used to render the body parts, and further constraint the arm hypotheses. For each configuration, a depth map is generated, which, in turn, is compared against the actual depth map in order to evaluate the hypothesis.

Throughout the process, specific quantitative parameters regarding the human-body are used: the torso size and the lengths of each arm part (upper and forearm). Since an initialization phase isn’t available, we rely on established adult anthropometric proportions [16, 17] to set these parameters relative to the human-body height; the latter is readily available as a by-product of detected face position (Task 1.1).

The sought virtual top-view camera, termed User Top View (UTV), has its optical axis aligned to the user’s torso axis. Based on the estimated UTV, we select the 3D points of the torso area which, in turn, are used to derive the 3D pose of the torso. To achieve this, we assume a virtual camera moving in various positions along a semi-sphere above the user, as depicted in Fig. 5. For each camera position, we select points belonging to the
hypothesized torso and re-project them on the virtual image plane. The torso is represented using an elliptic cylinder, the size of which is determined by the anthropometric measurements. To evaluate the virtual views, for each top-view re-projections of a user’s body, we employ the projection ratio criterion such as:

$$P_{ratio} = \frac{P_{proj} \cdot P_{cyl}}{P_{area}^2}$$  \hspace{1cm} (1),

where $P_{cyl}$ is the surface area of the hypothesized cylinder (and is constant), $P_{area}$ is the total number of 3D points inside the cylinder and $P_{proj}$ is the number of point projections on the image plane. In other words, assuming that the torso is the biggest part of a human body, we demand for the minimum number of point projections and the maximum body area (3D points) coverage simultaneously.

The virtual view with the minimum $P_{ratio}$ is the one obtained when the virtual camera on top of the user has its optical axis coinciding with the main axis of the human torso. This plausible and intuitive assumption has also been experimentally verified in our work.

Based on the derived UTV, torso orientation is readily available as the orientation assumed by the UTV-axis. More specifically, the UTV-axis defines the two degrees-of-freedom for the torso orientation. The third degree-of-freedom, namely torso rotation around the UTV-axis, is obtained by fitting a 2D ellipse on the torso points re-projected on UTV with center the previously approximated neck.

### 3.2 Arm Pose Estimation

Each arm is modeled using the 4-DoF kinematic model shown in Fig. 6(a), similar to the one presented in [18], and described by the Denavit-Hartenberg parameters shown in Table 2. Angles $\theta_1 \ldots \theta_3$ refer to the 3 DoFs of the shoulder joint, while angle $\theta_4$ refers to the DoF of the elbow joint. $l_u$ and $l_f$ refer to the lengths of the upper and forearm, respectively. Since we are not interested in the orientation of the palm, we practically discard the fifth angle by setting $\theta_5 = 0$.

We use a spherical system for optimization purposes and compute the mapping between the kinematic and the spherical system offline. To cope with possible data noise or errors introduced from previous computations, the
Table 2: Denavit-Hartenberg parameters for the 4-DoF model of the human arm employed in our approach.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\theta_i$</th>
<th>$\alpha_i$</th>
<th>$a_i$</th>
<th>$d_i$</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\theta_1$</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>0</td>
<td>$-90^\circ$...$90^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$\theta_2$</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>0</td>
<td>$-90^\circ$...$90^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>$\theta_3$</td>
<td>$+90^\circ$</td>
<td>0</td>
<td>$l_u$</td>
<td>$-230^\circ$...$90^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>$\theta_4$</td>
<td>$-90^\circ$</td>
<td>0</td>
<td>0</td>
<td>$0^\circ$...$145^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>$\theta_5$</td>
<td>$0^\circ$</td>
<td>0</td>
<td>$l_f$</td>
<td>$0^\circ$...$350^\circ$</td>
</tr>
</tbody>
</table>

Figure 6: (a) Kinematic model of the arm. (b) Shoulder-Palm pair sampling. Shoulders and palms are drawn from normal distributions. The elbow distribution have a torus-like shape, from which we draw samples uniformly.

detected shoulders and palms are represented as 3D gaussian distributions, with mean value $\mu$ the location of the corresponding joint. In the case of shoulders, the standard deviation $\sigma$ is the shoulder joint size, derived from the anthropometric proportions. In the case of a palm, $\sigma$ is inversely proportional to the confidence of the skin classifier, according to a predefined scale factor. A visual representation of the sampling procedure is given in Fig. 6(b).

Shoulder and palm pairs are sampled from the corresponding distributions, illustrated as spheres. For each pair, a 3D circle (depicted by red color) about the shoulder-palm axis defines the elbow allowed locations. In total, the elbow workspace forms a doughnut-like area, constrained by the kinematic model, from which samples are drawn uniformly to produce the arm hypotheses set. For each arm, the corresponding hypotheses are tracked over time by means of a separate particle filter, ensuring temporal and spatial consistency.

Illustrative results recovering torso and arm poses with multiple people in the scene, subject to intra- and inter- occlusions are shown in Fig. 7.

Figure 7: Frames out of sequences with multiple people moving freely in and out of the scene, acting and interacting and their corresponding recovered torso and arms’ poses.
4 Head Pose Estimation

Face pose estimation is extracted using a feature-based approach via Least-Squares Matching (LSM) on the RGB image and differential rotations are computed by analyzing the transformations of the facial patch across image frames [6, 4]. The problem statement is finding the corresponding part of the template image patch, in our case the face path \( f(x,y) \) in the search images \( g_i(x,y) \), \( i = 1, \ldots n-1 \).

\[
f(x,y) - e_i(x,y) = g_i(x,y) \tag{1}
\]

Equation (1) gives the least squares grey level observation equations, which relate the \( f(x,y) \) template and \( g_i(x,y) \) image functions or image patches. The true error vector \( e_i(x,y) \) is included to model errors that arise from radiometric and geometric differences in the images.

Assuming we have two images, in our case two consecutive frames, the \( f(x,y) \) and \( g(x,y) \), a set of transformation parameters need to be estimated from (1). Since (1) is nonlinear, it is linearized by expanding it into a Taylor series and keeping only zero and first order terms.

The estimation model should accommodate enough parameters in order to be able to model completely the underlying transformation. In the model only geometric parameters are included and radiometric corrections, e.g. equalization, for the compensation of different lighting conditions are applied prior to LSM in template and image. Assuming that the local surface patch of the face area is a plane to sufficient approximation (since depth variation exhibited by facial features are small enough) an affine transformation is used to model geometric differences between template or image frame \( n \) and search image or image frame \( n+1 \). The affine transformation (2) is applied with respect to an initial position \((x_0,y_0)\):

\[
x = a_0 + a_1 \cdot x_0 + a_2 \cdot y_0 \\
y = b_0 + b_1 \cdot x_0 + b_2 \cdot y_0 \tag{2}
\]

By differentiating (2) and the parameter vector being defined according to (3) the least squares solution of the system is derived.

\[
x^T = (da_0, da_1, da_2, db_0, db_1, db_2) \tag{3}
\]

To derive the above-mentioned face rotations we employ LSM by initializing the template patch, at the center of the detected blob ellipse (T1.1) and perform tracking via matching of the template across image frames and by updating the template based on the estimated parameters in each consecutive frame. To ensure that there is sufficient image signal in the mask, tracking is initiated for face blobs larger than \( 50 \times 50 \) pixels. The rotation of the head is then computed as explained in detail in \([6, 4]\), assuming that the head approximates a spherical body and using the mapping equations of the vertical perspective projection. Fig. 8 shows two frames of a JAMES sequence with superimposed results of tracking the face orientation of multiple agents.

The inherent advantage of this data-driven tracking method is its ability to effectively cope with challenging illumination conditions and inter-(e.g. biological variation) and intra-personal changes (i.e. expression) without prior training. The method is able to estimate the yaw, roll and pitch angles, however for the application at hand it is of interest only the estimation of the yaw (\( \phi \)) angle, as shown in Fig. 8. The algorithm is able to maintain significant success rates even for angles up to \( 120^\circ \), where only a small part of the facial patch is visible \([6, 4]\). However, for larger angles due the initial assumption of spherical body vertical perspective projection, there is a higher ambiguity in the estimation. In order to reduce ambiguity in the face orientation estimation, we leverage the depth point cloud information, as from the KinectTM sensor. This type of depth information, as opposed to RGB image information, has reduced dependence on scene illumination and almost no dependence on surface texturing, and therefore it can be favorably used in the discussed framework, by extending LSM to partial 3D surfaces in order to extract the 3D pose parameters directly via the least squares solution.
With depth information, the main idea is to perform matching on consecutive partial 3D facial surfaces instead of 2D facial image patches and estimate the parameters of a 3D transformation, which satisfies the Least Squares matching of the search partial surface $g(x,y,z)$ to the template partial surface $f(x,y,z)$. The observation equations relating the observations $f(x,y,z)$ to the parameters of $g(x,y,z)$ are given by:

$$f(x,y,z) - e(x,y,z) = g(x,y,z)$$ (4)

where $e(x,y,z)$ is the true error vector modeling geometric discrepancies and random errors of the sensor. A 7-parameter 3D similarity transformation (5) is used as the basic estimation model, expressing the geometric relationship between consecutive facial surface patches. This parameter space can be reduced to a rigid transformation without scaling, however we chose this extended parameter space to take into account small errors from the sensor.

$$
\begin{align*}
    x &= t_x + m(r_{11}x_0 + r_{12}y_0 + r_{13}z_0) \\
    y &= t_y + m(r_{21}x_0 + r_{22}y_0 + r_{23}z_0) \\
    z &= t_z + m(r_{31}x_0 + r_{32}y_0 + r_{33}z_0)
\end{align*}
$$ (5)

where $r_{ij} = R(\omega, \phi, \kappa)$ is the rotation matrix, $[t_x, t_y, t_z]^T$ is the translation vector and $m$ is the scale factor. For the Least Squares estimation eq. (4) is expanded using Taylor series, keeping only linear terms and eq. (5) is differentiated. A set of observation equations are formed and the unknown transformation parameters can also be treated as stochastic quantities using a-priori weights. The parameter vector $\mathbf{x}^T = [dt_x, dt_y, dt_z, dm, d\omega, d\phi, d\kappa]$ can be then estimated via the Least Squares solution of the system.

The method also considers detection of false correspondences with respect to the outliers and occlusions, that is data holes inside the model possibly well as model borders. In the course of iterations a simple weighting scheme adapted from Robust Estimation Methods is used to eliminate outliers and occluded parts. Because of the high redundancy of a typical data set, a certain amount of occlusions and/or smaller outliers do not have significant effect on the estimated parameters. Illustrative results of the 3D LSM for 3D head pose estimation are shown in Fig. 9.

5 Recognition of hand gestures

5.1 Recognition of communicative gestures

The pointing and waiving (hello) gestures comprise the communicative gesture vocabulary of JAMES. To recognize the set of gestures we perform classification of the arm trajectories (sec. 3.2) to gestures by means of a
combined Multi-Layer Perceptron (MLP)/Radial Basis Function (RBF) Neural Network structure as in [19]. Classification is achieved by buffering the trajectory of each arm (in its 4D configuration space) and feeding it to a feed-forward MLP Neural Network which is trained to recognize between five system states: idle (no gesture), preparation (hand moving towards a gesture), pointing gesture, hello (waiving) gesture, and retraction (hand retracting from a gesture). The output of the MLP is passed though an RBF which is trained as a predictor for the next state of the system and fed back to the MLP in order to improve temporal consistency and robustness of the achieved results.

For training the proposed classifier, a dataset consisting of 12 sequences was used. This dataset contains six examples of each of the two considered gestures. In each of these sequences all three phases of a gesture appear, together with cases where none of the phases is performed or when both hands are acting simultaneously. The dataset was divided into two subsets, of 6 sequences each. The first subset contained sequences from each of two gestures and it was used to train the MLP neural network while the second subset was used to train the RBF network. Using the two subsets, the training of the system has been done in two steps. Training of the MLP was performed by minimizing the mean of the squared error using the Levenberg-Marquardt algorithm. To train the RBF network, the sequences used for training the MLP cannot be used because they are known to the classifier. Thus, the second training set is used. Illustrative results are presented in Fig. 10.

5.2 Recognition of manipulative gestures

Manipulative gestures that act on objects within the JAMES framework are the grab and hold gestures. Grabbing an object is characterized as an event, whereas holding an object as a state, following the grabbing event. Bottles as objects are characterized of different color properties and are therefore tracked with the probabilistic tracker of T1.1, described in D1.1. As a result we are able to retrieve information from the tracked blobs regarding their shape characteristics and their speed. Following the classification of hands and faces, proximity of hand blobs to objects is checked to identify a grab event. The hold state is assumed by analyzing the speed and location of
the object blob with respect to the hand blob that initiated the grab event. An illustrative result of a grab event in the JAMES scenario is shown in Fig. 11.

6 Visual speech recognition

Visual speech detection relies upon detecting mouth movements on the face of the person interacting with the robot. Considering the minimum requirements with respect to the face resolution in order to extract invariant features, results can be obtained when the detected face is of dimension 70x90 or larger. Prior to detecting mouth movement, all facial feature regions, must be successfully detected and validated. Namely mouth, eyes and nose areas are detected and tracked via our method on facial feature extraction [1]. Then feature points are extracted on the boundary of the mouth contour and tracked over time. The simplest case of mouth feature points detection occurs when mouth is normally closed. Extraction becomes more complex when mouth is wide open or teeth or tongue are visible between the upper and lower lips. For mouth feature point selection a number of low level image processing operation are applied. A contrast stretching on the basis of pixels cumulative distribution is performed on the image, followed by a histogram thresholding. Then feature points are extracted at the mouth contour using the Harris operator and four of them at the left, right, top and bottom most positions are selected. Fig. 12 shows the stages of feature point selection in the mouth region. Their relative movements in x and y directions are then tracked via the Lukas Kanade feature tracker and provide input every 10 frames whether the person speaks or not.
7 Facial expression recognition

We are interested in subject-independent facial expression recognition, as well as to obtain robustness against illumination variety and image deformation. Facial expression recognition builds upon the detection and tracking of facial feature regions [1], considering also the minimum requirements of face resolution as in (sec. 6). Once the above are met we extract facial expression features and then use elastic templates matching to classify the facial expression of the user. To extract the key points within each facial feature region we lattice these regions with a predefined grid of 5 x 5 pixels and then use the gabor wavelet transformation at each lattice. A family of 18 gabor wavelets with different frequency and orientation is used to compute gabor coefficients at each lattice and form a feature vector. We select key points with large amplitude of feature vector as it is shown in Fig. 13 and generate the elastic templates for specific expressions (e.g. happy). In order to obtain subject-independent expression recognition, images containing different subjects and different typical expressions are used to construct the expression template database. The elastic templates are formed based on key-points detection and then stored in an expression template database. We then use an energy function to find the best match between the elastic templates and detected elastic graph and identify the expression of the user.

8 3D Object pose tracking

We have further shared the research results of our tracking method to the EU project FIRST-MM (http://www.first-mm.eu/) and jointly tackled the problem of 3D object pose tracking. A model based approach is formed to track the pose of an object in 3D based on 2D derived contours and edges, using a monocular camera as illustrated in Fig. 14. To enhance the performance of our method under occlusions and other artifacts, we have established a generation of multiple hypothesis, in the form of rendered objects. For this purpose, we have formulated an efficient number of hypotheses criterion within our framework’s implementation. Experimental results have demonstrated that our method achieves good pose tracking resolution at a relatively fast frame rate. These results have indicated that our tracking method exhibits better performance over the ViSP\(^1\) and BLORT\(^2\) methods [12, 13, 14].

\(^1\)http://www.irisa.fr/lagadic/visp/visp.html
\(^2\)http://users.acin.tuwien.ac.at/mzillich/?site=4
9 Conclusions

This document reported the work done within Tasks 1.2 and 1.3. During these tasks and building on Task 1.1, appropriate algorithms were created in order: (a) to extract pose related information from body and face and (b) to recognize communicative (waiving, pointing) and manipulative (grab /holding an object) gestures. We have provided information on the methodologies and the algorithms that were created, implemented and tested towards these goals.

JAMES’s operational requirements make both of these tasks very difficult in the sense that the algorithms must handle actions' gestures from multiple agents and be efficiently enough to perform in real time under specific CPU constraints imposed by the hardware of the robot.

In the remaining of the project, focus will be given to finetune the integration of the developed modalities with the rest of the JAMES system as well as integrate uncertainty measures for each module of the vision system.

References


