Visual estimation of pointed targets for robot guidance via fusion of face pose and hand orientation

Maria Pateraki†, Haris Baltzakis†, Panos Trahanias†‡
†Institute of Computer Science
Foundation for Research and Technology – Hellas (FORTH)
Heraklion, Crete, GR
‡Department of Computer Science, University of Crete
Heraklion, Crete, GR
{pateraki,xmpalt,trahania}@ics.forth.gr

Abstract

In this paper we address an important issue in human-robot interaction, that of accurately deriving pointing information from a corresponding gesture. Based on the fact that in most applications it is the pointed object rather than the actual pointing direction which is important, we formulate a novel approach which takes into account prior information about the location of possible pointing targets. To decide about the pointed object, the proposed approach uses the Dempster-Shafer theory of evidence to fuse information from two different input streams: head pose, estimated by visually tracking the off-plane rotations of the face, and hand pointing orientation. Detailed experimental results are presented that validate the effectiveness of the method in realistic application setups.

1. Introduction

In the emergent field of social robotics, human–robot interaction via gestures is an important research topic.Pointing gestures are especially interesting for interaction with robots, as they open up the possibility of intuitively indicating objects and locations and are particularly useful as commands to the robot. Some of the most important challenges are related to the requirement for real time computations, the accuracy of the computations and the operation in difficult cluttered environments with possible occlusions, variable illumination and varying background. Another common requirement is that pointing gestures must be recognized regardless of scale, referring to large pointing gestures performed with full arm extend and small pointing gestures reduced to forearm and hand movement only [11].

Based on the fact that, for most applications, it is the pointed target rather than the actual pointing direction which is important, we formulate a novel approach which, in contrast to existing pointing gesture recognition approaches, also takes into account prior information about the location of possible pointing targets. Assuming the most common type of deictic gesture, i.e., the one that involves the index finger pointing at the object of interest and the user's gaze directed at the same target [13], we formulate an approach which uses a monocular-only camera setup to track off-plane face rotations and at the same time recognize hand pointing gestures. These two input streams are combined together to derive the pointing target using a formulation which is based on the Dempster-Shafer theory of evidence. The later allows the approach to elegantly handle situations when one or both of the input streams are missing (e.g. the hand pointing direction is not visible due to self-occlusions), achieving impressive results which could not have been derived with contemporary probabilistic fusion approaches.

2. Related work

Recent vision-based techniques for gesture recognition have been comprehensively reviewed [9] and there are several approaches using stereo or multi-camera systems and which focus only on the hands/arms [5, 6] or both hands/arms and face [7, 10, 8, 11]. Some of the above approaches suffer from delayed recognition,e.g. [10, 6], limited accuracy assessment, e.g [8, 6] and most (with an exception here [11]) do not support the scale of gesture.

Unlike the above approaches we use a single camera, that can be placed on a moving robotic platform. Single camera systems, were examined by different authors, e.g. [16, 14] and although good initial results were reported, further efforts are necessary to deal with the demanding cases of real world environments.

The method presented in this paper supports scale of ges-

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00

ture, obtuse angle of pointing gesture beyond the range of $[-90\,°, 90\,°]$, considering at the same time the real-time response of the system and the pointing accuracy. In this work, face orientation is effectively fused with hand gesture recognition, to accurately estimate the pointed target. The Dempster–Shafer theory [15] is utilized to formulate fusion as a belief estimation problem in the space of possible pointing directions. Even in cases that the system is not able to recognize hand pointing gestures or face orientation (or neither), this information provides a piece of evidence which in most cases is enough to significantly limit the number of possible solutions.

3. Target scenario and proposed methodology

The target scenario we address is a robot operating in a public space, such as an exhibition or a museum, interacting with humans and providing information about specific Points Of Interest (*"POIs"*, e.g. exhibits). It is assumed that all pointing gestures refer to POIs and the task at hand regards the accurate estimation of the POI that the user points to.

The overall approach consists of the estimation of the pointed POI by integrating information of face orientation with information from a hand gesture recognition system which is able to robustly recognize two different hand gestures: "point left" and "point right". The face and the hands are detected and tracked in 2D using a 2D skin-color blob tracker and detected skin-colored blobs are further classified into left hand, right hand and face. The occurrence of a hand pointing gesture is detected based on a rule-based technique taking into account the motion of the hand and the number and relative location of the distinguishable hand fingertips. Face pose estimation is based on Least-Squares Matching (LSM) and differential rotations are computed via patch deformations across image frames. Finally, combine evidence from the two different information sources (hand pointing gesture and face orientation), we make use of the Dempster's rule of combination. The proposed approach is depicted in Fig. 1 and in the following sections we describe these components in detail.

4. Hand pointing gesture recognition

The first step of our approach is to detect skin-colored regions in the input images. For this purpose we use a technique similar to the one described in [2]. Initially, the foreground area of the image is extracted by the use of a background subtraction algorithm. Then, foreground pixels are characterized according to their probability to depict human skin and then grouped together into solid skin color blobs using hysteresis thresholding and connected components labeling. The location and the speed of each blob is modeled as a discrete time, linear dynamical system which is tracked

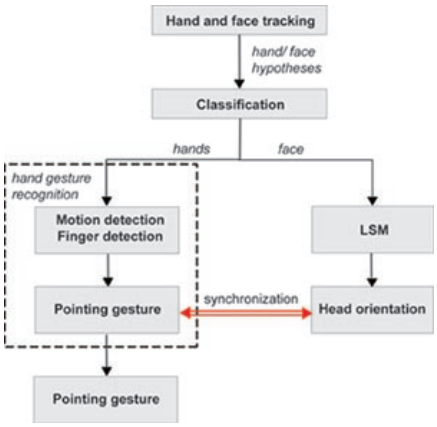


Figure 1. Block diagram of the proposed method for pointing direction estimation.

using the Kalman filter equations, according to the propagated pixel hypotheses algorithm as in [1]. Information about the spatial distribution of the pixels of each tracked object (i.e. its shape) is passed on from frame to frame using the object's current dynamics, as estimated by the Kalman filter. The density of the propagated pixel hypotheses provides the metric, which is used in order to associate observed skin-colored pixels with existing object tracks in a way that is aware of each object's shape and the uncertainty associated with its track.

The second step is to further classify blobs as left hand, right hand and face, as well as maintain and continuously update the belief about the class of each tracked blob. For this purpose we employ an incremental probabilistic classifier, as in [3], using as input the speed, orientation, location and contour shape of the tracked skin-colored blobs. This classifier permits identification of hands and faces of multiple people and is able to maintain hypotheses of left and right hands, even in cases of partial occlusions.

For the actual hand pointing recognition, an important aspect is the detection of the effective time a pointing gesture occurs. According to [13], the temporal structure of hand gestures can be divided in three phases: preparation, peak and retraction, with an exception to this rule the so called "beats" (gestures related to the rhythmic structure of the speech). "Preparation" and "retraction" are characterized by the rapid change in position of the hand, while in the "stroke", the hand remains, in general, motionless. Taking into account the trajectory of the moving hand and a number of relevant criteria, as in [18], we detect the "stroke" phase, i.e. the phase at which the pointing gesture takes place.

In order to recognize hand pointing gestures among the set of gestures that comprise the gesture vocabulary of the robot, and, additionally, to classify them as "point left" and "point right" gestures, we employ a rule-based technique, similar to the one presented in [18]. According to this tech-

nique, gesture recognition is performed based on the number and the posture of the distinguishable fingers of the hand performing the gesture, i.e., the number of visible fingertips and their relative location with respect to the centroid of the hand's blob.

5. Face pose estimation

To estimate the POI that the user is looking at while performing a pointing gesture in a non-intrusive way, we employ a technique that tracks the orientation of the user's head. This is achieved by tracking off-place facial rotations via a feature-based face tracking approach based on Least-Squares Matching (LSM).

5.1. The Least Squares approach

LSM is a matching technique able to model effectively radiometric and geometric differences between image patches, also considered as a generalization of cross-correlation, since, in its general form, it can compensate geometric differences in rotation, scale and shearing, whereas cross-correlation can model geometric differences only by translation and radiometric differences only due to variations in brightness and contrast.

In our context LSM is used for inter-frame calculations, over a long time span in tracking, to derive the rotation of the user's face while performing the pointing gesture. The problem statement is finding the corresponding part of the template image patch $f(x, y)$ in the search images $g_i(x, y), \quad i = 1, ...n - 1$.

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

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), which 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 image formation process. Similar efforts in modeling a region include Hager and Belhumeur's work [4] that explicitly modeled the geometry and illumination changes with low parametric models. 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, as 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 $n + 1$. Instead of a conformal set of parameters [12], we utilize an affine transformation to track the face patch during off-plane face rotations. The affine transformation (2) is applied with respect to an initial position (x_0, y_0):

$$\begin{aligned} x &= a_0 + a_1 \cdot x_0 + a_2 \cdot y_0 \\ y &= b_0 + b_1 \cdot x_0 + b_2 \cdot y_0 \end{aligned} \tag{2}$$

By differentiating (2) and the parameter vector being defined according to (3) the least squares solution of the system is given by (4).

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

$$\hat{x} = (A^T P A)^{-1} (A^T P l) \tag{4}$$

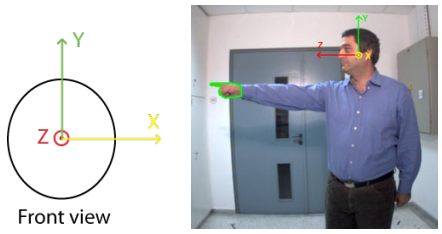
where $\hat{x}$ is the vector of unknowns, A is the design matrix of grey level observation equations, P is the weight matrix, and l is the discrepancy vector of the observations. The number of grey level observations relates to the number of pixels in the template and a weighting scheme is adopted, to reduce contribution for grey level observation equations that correspond to pixels close to the border.

The method requires that the change from frame to frame is small, considering the speed of the object and the framerate of the acquired image sequence, for the solution to converge. To improve performance and handle cases of fast motions we operate the algorithm at lower resolution levels.

5.2. Estimating the head orientation

Considering the human head as a rigid body in a three-dimensional space, head orientation can be derived by analyzing the transformations of the facial patch (frontal part of the head). The rotations of the head can be in-plane rotations of the face around the head's z-axis, off-plane rotation in vertical direction around the head's x-axis and off-plane rotation in horizontal direction around the head's y-axis (see Fig. 2). The latter, which corresponds to an off-plane rotation of the face towards the pointing direction, mainly deforms the facial patch in x-shift and x-scale.

To derive the above-mentioned face rotations we employ LSM by initializing the template patch, at the center of the detected blob ellipse at the occurrence of the preparation phase of gesture at image frame n and assuming frontal view of the face.In practice the initial position of the frontal view of the face can be derived via available face detectors as [17]. The template is updated in image frame $n + 1$ based on the estimated affine parameters and matched to the next image frame. A number of criteria used to evaluate the matching result, also ensure that the error propagation from frame to frame is minimized. E.g. these are the number of



(a) (b)

Figure 2. (a) Rotation axes of head in three dimensional space and (b) off-plane facial patch rotation around the y axis of the head.

iterations (assuming fast convergence should be achieved), the alteration of the size of parameters in each iteration and the size of parameters, variations in the parameter values (magnitude and sign) in each iteration have to be observed for the stability of the solution.

The rotation between the initial position of the template and the final matched position is computed by accumulating the differential rotation angles derived by matching each consecutive template and patch. In Fig. 3, p_x denotes the initial patch width and p'_x is the patch width, occurring at different rotations of the face around the y-axis of the head. x_1 and x'_1 are the projections of the minimum and maximum patch width in frontal view of the face with respect to the camera. Similarly x_2 and x'_2 are the projections of the minimum and maximum patch width at horizontal rotation of the head.

To compute the rotation angle we assume the mapping equations of the vertical perspective projection, given by the transformation equations

$$\begin{aligned} x &= k' cos\phi \cdot sin(\lambda - \lambda_0) \\ y &= k'[cos\phi_1 \cdot sin\phi - sin\phi_1 \cdot cos\phi \cdot cos(\lambda - \lambda_0)] \end{aligned} \tag{5}$$

where P is the distance of the point of perspective in units of sphere radii

$$k' = (P-1)/(P - cosc) \tag{6}$$

and

$$cosc = sin\phi_1 \cdot sin\phi + cos\phi_1 \cdot cos\phi \cdot cos(\lambda - \lambda_0) \tag{7}$$

Under the assumption that the head approximates a spherical body, we compute the horizontal angle λ_h from $\lambda_0 = 0\,^\circ$ with $\phi = 0\,^\circ$ and $\phi_1 = 0\,^\circ$ according to (8) and (9)

$$\begin{aligned} \lambda_2 &= \arcsin(\frac{x_2 - \mu_x}{k'}) \\ \lambda'_2 &= \arcsin(\frac{x'_2 - \mu_x}{k'}) \end{aligned} \tag{8}$$

$$\lambda_h = \frac{\lambda_2 + \lambda'_2}{2} \tag{9}$$

Fig. 4 illustrates temporal matching results with LSM, tracking the face's off-plane horizontal rotation.

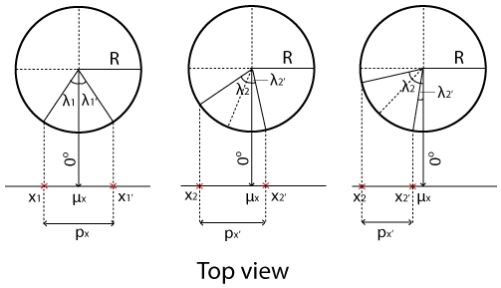


Figure 3. Horizontal face orientation computation.

(a) (d)

Figure 4. Derivation of face orientation via LSM with varying distance from the camera.

6. Deriving evidence of pointing direction

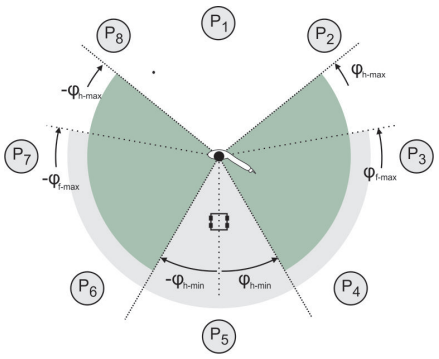


Figure 5. Fusion of information from two input sources.

Assume the setup depicted in Fig. 5. The user (at the center of the figure) and the robot stand in a scene which contains N points of interest $P_1, P_2, \ldots, P_N$, ($N = 8$ in this specific example) surrounding the user and the robot. Each point of interest constitutes a possible pointing direction.

Let X be the universal set: This is the set that contains all visible POIs, also called *the frame of discernment* (10).

$$X = \{P_1, P_2, \ldots, P_N\} \tag{10}$$

The power set 2^X is the set that comprises of all subsets of X, including the empty set $\varnothing$ and the whole set

X. The elements of the power set can be taken to represent propositions about the pointing direction. Each proposition contains the POIs for which the proposition holds true and it is assigned an amount of belief by means of a function $m : 2^X \rightarrow [0, 1]$ which is called a *basic belief assignment* and it has two properties:

- The mass of the empty set is zero:

$$m(\varnothing) = 0 \tag{11}$$

- The masses of all the members of the power set add up to a total of 1:

$$\sum_{A \in 2^X} m(A) = 1 \tag{12}$$

In the task at hand, the user raises his hand to point to an exhibit $P_i, 1 \leqslant i \leqslant N$ and simultaneously rotates his/her face to look at the direction of the exhibit, as described in section 3. Hence, two sources of information exist: information from the hand pointing gesture and information from the face orientation. Each of the two sources of information has an independent basic belief assignment. Let m_f represent the belief from face pose estimation and m_h represent the belief from the hand pointing direction. To combine evidence from these two sources of information we make use of Dempster's rule of combination.

According to the Dempster's rule of combination, the joint mass $m_{f,h}$ can be computed as the orthogonal sum (commutative and associative) of the two masses, as follows:

$$m_{f,h}(\varnothing) = 0 \tag{13}$$

$$m_{f,h}(A) = (m_f \oplus m_2)(A) = \frac{1}{K} \sum_{B \cap C = A \neq \varnothing} m_f(B) m_h(C) \tag{14}$$

where K is a normalization coefficient which is used to evaluate the amount of conflict between the two mass sets, given by:

$$K = 1 - \sum_{B \cap C = \varnothing} m_f(B) m_h(C) \tag{15}$$

Equation (14) provides a combined belief mass for every POI A as a function of all pieces of evidence $m_f(B)$ and $m_h(C)$ that agree on A. The POI with the larger combined mass is selected as the one pointed by the user.

It is to be noted at this point that the assumption behind this work that the user is simultaneously looking and pointing at the same exhibit eliminates the cases of conflicting evidence which, according to Zadeh's criticism [19] for Dempster's rule of combination, may lead Equation 14 to produce counter intuitive results.

In the next two sections we will elaborate on the actual calculation of m_h and m_f.

6.1. Computation of the belief mass m_h

For hand pointing gesture recognition, we assume that the system knows when a gesture takes place but can only recognize the pointing direction if it is within the intervals $[\phi_{h-min}, \phi_{h-max}] \cup [-\phi_{h-max}, -\phi_{h-min}]$. If a "point left" gesture is recognized, we assume that the user is pointing to a POI within $[-\phi_{h-max}, -\phi_{h-min}]$ (P_6 or P_7 in the example of Fig. 5). Similarly, if a "point right" gesture is recognized, we assume that the user is pointing to a POI within $[\phi_{h-min}, \phi_{h-max}]$ (either P_3 or P_4). If there is no recognized gesture, we assume that the user pointed to a POI outside these two intervals ("point center") with a belief mass of $m_{h-0} = m_h(\{P_1, P_2, P_5, P_8\})$. For the "point center" case, in the example of Fig. 5, the user might have pointed to any of P_1, P_2, P_5 or P_8. Depending on the recognized hand pointing gesture ("point left", "point right") or the fact that the gesture "invisible", different belief masses are assigned for the exhibits on the left, the exhibits on the right and the exhibits in the front and in the back of the user for which a pointing gesture cannot be recognized.

For the example of Fig. 5, the above belief masses are defined as follows:

$$\begin{aligned} m_{h-L} &= m_h(\{P_6, P_7\}) \\ m_{h-R} &= m_h(\{P_3, P_4\}) \\ m_{h-C} &= m_h(\{P_1, P_2, P_5, P_8\}) \end{aligned}$$

with $m_{h-L} + m_{h-R} + m_{h-C} = 1$.

To appoint the sets that correspond to "point left", "point right" and "point center" directions and define the masses m_{h-L}, m_{h-R} and m_{h-C} accordingly, we use specific values for ϕ_{h-min} and ϕ_{h-max}. These values have been experimentally calculated as $\phi_{h-min} = 140^o$ and $\phi_{h-max} = 40^o$ and roughly correspond to the angle limits beyond which the hand pointing gesture is not recognizable.

Let G be the actual gesture performed by the user and let G_O be the gesture recognized (or not recognized) by the system. G takes values in $H_G = \{\text{"}point\,left\text{"}, \text{"}point\,right\text{"}, \text{"}pointcenter\text{"}\}$. and G_O takes values in $H_O = \{\text{"}point\,left\text{"}, \text{"}point\,right\text{"}, \text{"}invisible\text{"}\}$.

To assign masses to m_{h-L} m_{h-R} and m_{h-C}, we calculate the probabilities $P(G = \text{"}point\,left\text{"}|G_O)$, $P(G = \text{"}point\,right\text{"}|G_O)$ and $P(G = \text{"}point\,center\text{"}|G_O)$, respectively which are computed using the Bayess rule as:

$$P(G|G_O) = P(G_O|G) \frac{P(G)}{\sum_{h \in H_O} P(G_O|G = h) P(G = h)} \tag{16}$$

Table 1. Confusion matrix for hand pointing gesture

G_O \ G	Point left	Point center	Point right
"$point\,left$"	0.90	0.05	0.00
"$invisible$"	0.10	0.90	0.10
"$point\,right$"	0.00	0.05	0.90

In the above equation $P(G)$ is computed as the number of visible POIs that fall within G divided by the total number of visible POIs. The likelihoods $P(G_O|G)$ are obtained off-line and correspond to the percentage of times a pointing gesture was recognized as G_O given that the actual gesture was G. The actual values computed during our experiments are summarized in the confusion matrix in Table 1.

6.2. Computation of the belief mass m_f

For face orientation, we assume that it is recognizable only within the range $[-\phi_{f-max}, \phi_{f-max}]$. If the user is looking at a POI which lays within this range then the face orientation can be computed using the algorithm described in section 5.2 and, additionally, this information can be employed to identify the target exhibit P_i, with a belief $m_f(\{P_i\})$. If the face orientation cannot be computed, we assume with a belief $m_{f-O} = m_f(\{P_1, P_2, P_8\})$ that the face is looking at a POI outside $[-\phi_{f-max}, -\phi_{f-max}]$. Since m_f is a basic belief assignment, we make sure that the following equation holds.

$$m_{f-O} + \sum_{k=1..N} m_f(\{P_i\} = 1 \qquad (17)$$

Similarly to hands, to assign masses to m_f, we use the conditional probabilities for the user looking at each POI P_i given the perceived face orientation ϕ_O, calculated as:

$$P(P_i|\phi_O) = P(\phi_O|P_i)\frac{P(P_i)}{\sum_{k=1}^{N} P(\phi_O|P_k)P(P_k)} \qquad (18)$$

In the absence of any prior information, $P(P_i)$ are assigned equal values for all P_i. The likelihood $P(\phi_O|P_i)$ is computed according to the relative angle of P_i with respect to the user. The exact values are found by interpolation to data gathered off-line, stored as a confusion matrix (Fig. 6).

7. Experimental results

7.1. Ground truth data

Quantitative evaluation of pointing direction is considered difficult because of the lack of dependable ground truth. In our case, we have setup a series of experiments which involve a user standing in front of the robot and pointing in predefined directions (specified POIs) using both his hand and face. The POIs were defined in the range of $0\,° \pm 180\,°$ with an angular step of $10\,°$.

For each pointing gesture, the system identifies the orientation of the face and classifies the hand gesture as a left pointing gesture or a right pointing gesture. Sample recognition results for face orientation are shown in Fig. 4. The confusion matrix of Fig. 6 relates the absolute (both left and right) estimated head orientations to the intended head orientations in the range of $0\,° \pm 180\,°$. The percentages were derived from image sequences of a total of 7000 image frames. As can be easily seen, the algorithm achieves high success rates for low angles (user looks in directions close to the direction of the camera) which are decreased for higher angles. The algorithm maintains significant success rates (more than 50%) for angles up to $120\,°$, where only a small part of the facial patch is visible. The hand pointing gesture has always been recognized correctly for pointing orientations within the range $[30\,°, 130\,°]$ and $[-30\,°, -130\,°]$.

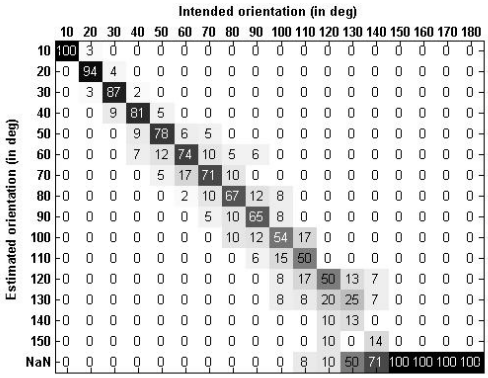


Figure 6. Confusion matrix encoding perceived face orientations (rows) for intended face orientations (columns) in the range of $0\,° \pm 180\,°$. The matrix contains data for both left and right pointing directions.

7.2. Simulated environment

Evidently the performance of an algorithm that identifies pointed POIs instead of pointing directions depends on the structure of the environment and the distribution of the POIs within it. To evaluate the performance of the proposed methodology under different environment arrangements, we have conducted a series of experiments in three different simulated environments, depicted in Figs. 7 and 8. The first environment, shown in Fig. 7(a), consists of a single rectangular room with four POIs positioned on its walls. The second environment, depicted in Fig. 7(b), is similar to the first but contains eight POIs instead of four. Finally, the last environment, depicted in Fig. 8, contains five rooms connected together via a corridor. Within the rooms there are eight exhibits in total but they are arranged in a way that

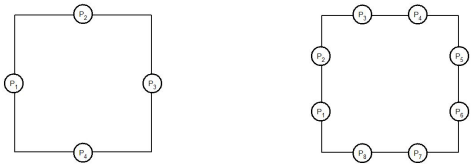
Table 2. Simulation results obtained by the proposed approach for the three environments depicted in Figs. 7 and 8.

	Experim. A (Fig. 7a)	Experim. B (Fig. 7b)	Experim. C (Fig. 8)
N	1000	1000	1000
n_{av}	1.06	1.25	1.13
Cor_m	996 (99.6%)	970 (97.0%)	997 (99.7%)
Cor_s	961 (96.1%)	819 (81.9%)	927 (92.7%)
$False$	4 (0.4%)	30 (3.0%)	3 (0.3%)

Table 3. Simulation results obtained by using a single source of information only

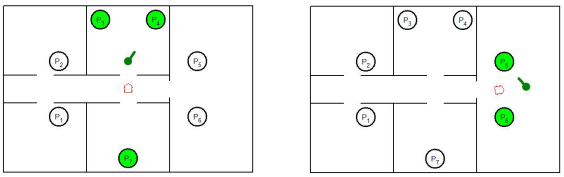
	Experiment A (Fig. 7a)	Experiment B (Fig. 7b)	Experiment C (Fig. 8)
V_h	550 (55.00%)	558 (55.80%)	535 (53.50%)
C_h	245 (24.50%)	47 (4.70%)	336 (33.60%)
V_f	731 (73.10%)	730 (73.00%)	796 (79.60%)
C_f	722 (72.20%)	703 (70.30%)	776 (77.60%)

no more than three exhibits are simultaneously visible by both the robot and the user, for any user-robot arrangement.



(a) (b)

Figure 7. Two simulated environments, used to evaluate the performance of the proposed algorithm.



(a) (b)

Figure 8. Two different, user-robot configurations in a simulated environment, with a number of rooms. The robot location is depicted using a green rectangle. The user is depicted using a green circle with a line segment indicating the pointing direction. The visible POIs for each configuration are painted with green color.

To run the simulated experiments we randomly selected a large number of human-robot arrangements within each of these environments (1000). Each arrangement included a randomly selected POI that was visible by both the user and the robot and was assumed as pointed by the user. Fig. 8 depicts two such arrangements. In the first arrangement there are three visible POIs (P_3,P_4 and P_7, marked with green color) and the user points to P_4. The second arrangement involves two visible POIs (P_5 and P_6, marked with green color), with P_5 being pointed by the user. The user's perception regarding both the recognition of the hand pointing gesture as well as the face orientation was assumed imperfect, simulating the distributions shown in the confusion matrices depicted in Table 1 and Fig. 6, respectively.

The results obtained are shown in Table 2. The total number of arrangements (N) for each environment was 1000. In the vast majority of cases, the fusion algorithm returned a single result. In cases that there were ambiguities regarding the pointed POI (e.g., there were more than one POI outside the angular regions defined in Fig. 5), the algorithm returned more than one results with the same mass. The average number of returned results with the same, maximum, mass is indicted as n_{av}. The row labeled "Cor_m" provides the number of times that the correct result was within the returned results, even if the number of results with the maximum mass was more than one. The row labeled "Cor_s" provides the figures for the cases that the algorithm returned a single, correct result. Finally, the row labeled "$False$" corresponds to the cases that the algorithm failed to provide the correct POI within the returned results.

It is to be noted at this point that many of the above arrangements correspond to cases that the robot is not able to recognisee neither the hand pointing direction nor the face orientation (e.g., when the user points to an exhibit behind him/her). In these cases, the lack of input is a piece of evidence by itself, which is correctly utilized by our algorithm to limit the number of possible solutions. As can be easily seen from the results of Table 2, this is especially successful for environments like the ones depicted in Figures 7a and 8 where the spatial arrangements of POIs and the topology of the environment (walls, doors, etc.) help distinguish between different possible solutions.

Table 3 shows the results obtained for the same configurations when a single source of information was only available: either face orientation or hand pointing direction. V_p corresponds to the number of cases that the pointed POI was either within $[\phi_{h-min}, \phi_{h-max}]$ or within $[-\phi_{h-max}, -\phi_{h-min}]$, i.e., within one of the two ranges that the pointing gesture is recognizable (see Figure 5). C_p corresponds to the classification results that would have been obtained if the recognition of a pointing gesture in either of there ranges was the only information available to the classifier. The assumption used to derive the obtained results was that we had a correct classification if the gesture was correctly recognized (recognition rates were

assumed as shown in Table 1) and, additionally, a single POI existed within the pointed region. Similarly, V_f corresponds to the number of cases that the pointed POI was within $[-\phi_{f-max}, \phi_{f-max}]$ and C_f corresponds to the results that would have been obtained is we were using a classifier which could achieve the results depicted in Figure 6.

By comparing the results from Tables 2 and 3, one arrives at the conclusion that the proposed approach clearly outperforms both "single-evidence" classifiers describe above. In all three environments, the algorithms successfully combines evidence from both information streams, achieving recognition rates that could not have been obtained by any of the two information streams alone.

8. Conclusions

In this paper we have presented a novel method for estimating the pointing direction by fusing information regarding hand pointing gestures and the pose of the user's head. The proposed method is able to achieve surprisingly good performance by considering prior knowledge about the location of possible pointing targets which reduces the problem to deciding which is the pointed target rather than calculating the actual pointing direction.

Unlike other contemporary methods, our approach operates with a single camera and we have demonstrated its ability to achieve significant recognition rates even in cases that either of the two or both input streams are missing.

The method is readily applicable in a large variety of human-robot interaction scenarios. Future work will involve its enhancement by fusing additional sources of information, such as arm pose, body orientation and a-priori probabilities of POI selection.

9. Acknowledgements

This work was partially supported by the European Commission, under contract numbers FP6-045388 (INDIGO project), FP7-248258 (First-MM project) and FP7-270435 (JAMES project).

References

[1] H. Baltzakis and A. Argyros. Propagation of pixel hypotheses for multiple objects tracking. In *Proc. International Symposium on Visual Computating (ISVC)*, Las Vegas, Nevada, USA, Nov. 2009. 2

[2] H. Baltzakis, A. Argyros, M. Lourakis, and P. Trahanias. Tracking of human hands and faces through probabilistic fusion of multiple visual cues. In *Proc. International Conference on Computer Vision Systems (ICVS)*, pages 33–42, Santorini, Greece, May 2008. 2

[3] H. Baltzakis, M. Pateraki, and P. Trahanias. Visual tracking of hands, faces and facial features of multiple persons. *Machine Vision and Applications*, 2011. (under review). 2

[4] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(10):1025 –1039, Oct. 1998. 3

[5] E. Hosoya, H. Sato, M. Kitabata, I. Harada, H. Nojima, and A. Onozawa. Arm-Pointer: 3D Pointing Interface for Real-World Interaction. *Lecture Notes in Computer Science*, 3058:72–82, 2004. 1

[6] K. Hu, S. Canavan, and L. Yin. Hand pointing estimation for human computer interaction based on two orthogonal-views. In *Proc. Intl. Conf. on Pattern Recognition (ICPR)*, pages 3760–3763, 2010. 1

[7] N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang. Detection and estimation of pointing gestures in dense disparity maps. In *Proc. Intl. Conf. on Automatic Face and Gesture Recognition*, pages 1000–1007, Grenoble, France, March 2000. 1

[8] R. Kehl and L. van Gool. Real-time pointing gesture recognition for an immersive environment. In *Proc. Intl. Conf. on Automatic Face and Gesture Recognition*, pages 577–582, Seoul, Korea, May 2004. 1

[9] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, May 2007. 1

[10] K. Nickel and R. Stiefelhagen. Visual recognition of pointing gestures for human-robot interaction. *Image Vision Computing*, 25:1875–1884, December 2007. 1

[11] C.-B. Park and S.-W. Lee. Real-time 3d pointing gesture recognition for mobile robots with cascade hmm and particle filter. *Image and Vision Computing*, 29(1):51–63, 2011. 1

[12] M. Pateraki, H. Baltzakis, and P. Trahanias. Tracking of facial features to support human-robot interaction. In *Proc. IEEE ICRA*, pages 3755–3760, Kobe, Japan, May 2009. 3

[13] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for Human-Computer Interaction: A Review. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(7):677–695, 1997. 1, 2

[14] J. Richarz, A. Scheidig, C. Martin, S. Mller, and H.-M. Gross. A monocular pointing pose estimator for gestural instruction of a mobile robot. *International Journal of Advanced Robotic Systems*, 4(1):139–150, 2007. 1

[15] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976. 2

[16] Z. Černeková, C. Malerczyk, N. Nikolaidis, and I. Pitas. Single camera pointing gesture recognition for interaction in edutainment applications. In *Proc. of the 24th Spring Conference on Computer Graphics*, pages 121–125, 2008. 1

[17] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 3

[18] X. Zabulis, H. Baltzakis, and A. Argyros. Vision-based hand gesture recognition for human-computer interaction. In C. Stefanides, editor, *The Universal Access Handbook*, Human Factors and Ergonomics, pages 34.1 – 34.30. LEA Inc., June 2009. 2

[19] L. A. Zadeh. On the validity of dempster?s rule of combination of evidence. *Memo M*, 79:24, 1979. 5