A MARKERLESS APPROACH FOR CONSISTENT ACTION RECOGNITION IN
A MULTI-CAMERA SYSTEM

Simone Calderara\textsuperscript{1}, Andrea Prati\textsuperscript{2}, Rita Cucchiara\textsuperscript{1}

\textsuperscript{1}Dipartimento di Ingegneria dell’Informazione, University of Modena and Reggio Emilia, Italy
\textsuperscript{2}Dipartimento di Scienze e Metodi dell’Ingegneria, University of Modena and Reggio Emilia, Italy

ABSTRACT
This paper presents a method for recognizing human actions in a multi-camera setup. The proposed method automatically extracts significant points on the human body, without the need of artificial markers. A sophisticated appearance-based tracking able to cope with occlusions is exploited to extract a probability map for each moving object. A segmentation technique based on mixture of Gaussians is then employed to extract and track significant points on this map, corresponding to significant regions on the human silhouette. The point tracking produces a set of 3D trajectories that are compared with other trajectories by means of global alignment and dynamic programming techniques. Preliminary experiments showed the potentiality of the proposed approach.

Index Terms— Action recognition, mean tracking, mixture of Gaussians, dynamic programming.

1. INTRODUCTION AND RELATED WORKS
Labeling actions taking place in a given scene is a task of paramount importance for behavior analysis. The main challenge relies on developing a method able to cope with almost every type of action, even if they are very similar to other one and also in the case of cluttered and complex scenarios. In the recent past, many researchers have addressed action recognition in video sequences in different contexts and with different purposes, ranging from sports video analysis to video surveillance to human-centred computing. For several years, researchers have concentrated on ad-hoc solutions to identify, often with heuristic rules, specific actions, such as fighting, talking, etc. [1]. However, recent advances in computer vision and statistical pattern recognition offer an effective and often efficient help for the recognition of higher-level actions, such as abandoned luggage detection, repetitive and abnormal path detection, or people-to-people interactions.

Basic approaches for recognizing human actions are based on either the analysis of body shape (in 2D or 3D) or the analysis of the dynamics of prominent points or parts of the human body. More specifically, action recognition approaches can be divided into two main groups [2] depending on whether the analysis is performed directly in the image plane (\textit{2D approaches}) or using a three dimensional reconstruction of the action itself (\textit{3D approaches}). The latter ones have been widely adopted where building and fitting a 3D model of the body parts performing the action is relatively simple due to controlled environmental conditions and high-resolution view of the object. For instance, Regh and Kanade in [3] used a 27 degree-of-freedom (DOF) hand model to recognize poses and gestures, while Goncalves et al. in [4] addressed the problem of analyzing human arm positions against a simple uncluttered background.

These methods are sometimes unfeasible in many real-time surveillance applications. Gavrila and Davis in [5] adopted a 22-DOF human-body model to detect actions against complex background but their approach constrains the user to wear a tight-fitting body suit with contrasting limb colors to simplify the edge detection problem in case of self-occlusions. Despite the complexity of the approach used, these methods can be applied only if a more or less sophisticated model of the target exists.

On the contrary, 2D approaches analyze the action in the image plane relaxing all the environmental constraints of 3D approaches but lowering the discriminative power of the action-classification task. People action classification can be performed in the image plane by either observing and tracking explicitly feature points (\textit{local feature approaches} [6]), or considering the whole shape-motion as a feature itself (\textit{holistic approaches} [7, 8]).

Yilmaz and Shah in [9] exploited people contour-points tracking to build a 3D volume describing the action and their work represents an example of local feature approaches. A compact representation of this action-specific volume was presented and proved to be effective in distinguishing among several predefined actions. Although this proposal results effective in most situations, contour-points tracking is a difficult task to achieve in real-time systems leading to a NP-hard optimization problem when points are occluding each other and one-to-one matching is impossible.

Niebles et al. in [10] proposed a feature-based approach...
that searches for “spatio-temporal words” as a time-collection of points of interest and classify them into actions using a pLSA (probabilistic latent semantic) graphical model. Training a complex graphical model requires many examples of the desired action and the “bag of words” feature extractor can be imprecise when a close view of the subject is not available.

Holistic approaches, instead, directly map low-level image features to actions, preserving spatial and temporal relations. Feature choice is a crucial aspect to obtain a discriminative representation. An interesting holistic approach that detects human action in videos without performing motion segmentation was proposed by Shechtman and Irani in [11]. They analyzed spatio-temporal video patches to detect discontinuities in the motion-field directions. Despite the general applicability of this method, the high computational cost makes it unusable for real-time surveillance applications.

All these related works, however, are inherently addressing single camera scenarios. Unfortunately, several actions cannot be easily caught by a single view; additionally, occlusions in cluttered environments can seriously affect action recognition, by hiding all or part of the action and bringing to missed or erroneous classifications. The use of multiple cameras looking simultaneously at the scene from different viewpoints allows to consistently recognize the action. Mimicking the use of the term “consistent labeling” (to identify the process of assigning the same label to different instances of the same person from different views [12]), we call this process “consistent action recognition”.

Few previous works have addressed explicitly the “consistent action recognition” problem. Most of them tackle the problem by using a view-independent representation of the action [13, 14]. Li and Fukui in [13], for instance, considered the case of an action seen by a moving camera: by continuously changing the point of view, the action can be confused. The authors used non-rigid factorization [15] of non-rigid shapes (like the human body is) and HMMs to model the action as a dynamic linear combination of basis shapes, whose weights contain the crucial information to recognize (in a view-invariant way) the action. Gritai et al. in [14], instead, use anthropometric spatial constraints among body parts and proportions of them as cues for matching actions performed from different viewpoints and in different environments. Non-linear time warping is used to perform temporally invariant matching. Despite the generality of these two approaches, both of them exploit artificial markers or manual intervention to extract human body points to be analyzed.

Some other works explicitly use multiple sources of information and thus tend more to a “consistent” recognition of the actions. For instance, in [16] the action was modeled by the so-called “action cylinder”, i.e. a 3D object that models the evolution of the 2D object shape in time. View-independent action recognition is achieved by recovering the geometrical transformation between the action model and a given action cylinder. Cupillard et al. [1], instead, developed a combination mechanism to fuse recognition achieved in different views by combining the moving region graphs obtained by each view.

HMM-based techniques are commonly adopted to classify actions and learn motion semantics directly from data. Lee et al. in [17] proposed a method for recognizing actions from multiple views exploiting a combined local and global optical flow computed on segmented blobs. The view invariance was achieved using the normalize Zernike moments and recognition performed by a set of HMM-based classifiers, one for each known action, in a ML fashion. Although the method seems to perform correctly in many cases, optical flow approaches are unsuitable to recognize actions in low contrast environments and especially when single body parts are moving over the person silhouette, e.g. “drinking from a glass” action. In addition, HMM-based classifiers need an exhaustive training set to avoid model overfitting and poor classification rates. In [18] an HMM-based classifier that exploits state-level transition probabilities to model an action was proposed. One HMM was build for each action to classify and state sequence used to discriminate among them using a space time trajectory of marked point in the image plane as observed data. The authors described the classification stage avoiding the data extraction discussion. The proposed classifier could perform well if sufficient training data are available but the extraction of precise trajectories is the major weakness of trajectory-based classifiers because markers or body-part tracking is often unfeasible in real surveillance scenarios.

The system proposed in this paper is meant to solve the problem of using artificial markers by automatically segmenting the human silhouette into a certain number of relevant areas found in the image describing the motion evolution. The tracking of the areas’ centroids produces a set of 3D trajectories describing on a fine grain the action of the person. Therefore, in order to compare two actions, we define a novel approach for comparing two sets of trajectories based on sequence global alignment and dynamic programming. An important addition of this paper regards the fusion of information coming from different views of the same action. In this way, a consistent action recognition is exploited to solve errors due to occlusions, view-dependent action representations, etc. Preliminary results showed an excellent discriminative power of this approach.

The rest of the paper is structured as follows: Section 2 presents a system overview and introduces the main four steps of the proposed approach; a brief description of the object detection and tracking step is also reported; Section 3 explains in details how the action space-time trajectories are extracted from a single camera setup, whereas Section 4 describes the multi-camera extension; Section 5 explains how the STTs are compared both in a single and a multiple camera setup; eventually, Section 6 reports the experimental results.
Fig. 1. Scheme of the proposed system.

### 2. SYSTEM OVERVIEW

The proposed system is based on four main steps (Fig. 1):

- **object detection and tracking**: for each camera view $C_i$, the moving people are segmented and tracked from the image $I_i^t(x, y)$ at instant time $t$; by this step, a probability map $PM_i^t(x, y)$ is obtained for each moving person (section 2.1);

- **iterative space-time trajectory extraction**: $K$ main components of $PM_i^t$ corresponding to $K$ main body parts are automatically extracted and tracked; they are used to model the action; EM algorithm is used to infer the parameter set $\hat{A}_i^t$ of a 3-variate mixture of Gaussians (MoG) on $PM_i^t$; the means of the $K$ components are tracked frame-by-frame using a minimum distance approach in the pdf domain based on the Bhattacharyya distance; finally, the iterative tracking allows to obtain $K$ space-time trajectories $STT_i^t = \{T_i^1, \ldots, T_i^K\}$ for each view (section 3);

- **consistent action recognition**: the list of tracked people from different (overlapped) cameras can be used to consistently assign the same label to different instances of the same person in different views (problem of consistent labeling); this assignment is used in the consistent action recognition step to fuse together the STT of the same person coming from different views (section 4);

- **action recognition**: a new action modeled as the set $STT$ of consistent $STT_i^t$ from different views is compared using global alignment [19] to compute a measure of distance/similarity from all the existing actions; the classification is performed with a minimum distance classifier (section 5).

### 2.1. Object Detection and Tracking

For the sake of brevity, we report here only the key concepts of the object detection and tracking step in order to provide sufficient information for further steps. Complete description of the object detection step and tracking algorithm can be found in [20] and [21], respectively. These steps have a twofold scope: the first is to separate foreground/moving objects from the background; the second is to obtain a rich feature set characterizing the action.

The first scope is achieved with the approach called SAKBOT (Statistical And Knowledge-Based Object Tracker) [20] which is based on background suppression where the background model updating is performed with an adaptive model and temporal median filtering. The updating is empowered by selectivity, i.e. the background model should not include the interesting moving objects if their motion is low or zero for a short period.

Once that moving objects/people have been segmented, they need to be tracked along time. For our purposes, track-
ing also represents a way for incorporating action evolution in a single observation by integrating in time single pixel membership. In other words, we create a probability map \( PM(p) \) where each value defines the probability that the point \( p \) belongs to the object. The value of \( PM(p) \) is updated with the segmentation results of the last \( n \) frames: further details can be found in [21]. As shown in Fig. 2, \( PM \) represents a fine-grain description of the action, removing useless information such as the person’s appearance.

![Images of different actions: walking, abandoning pack, sitting, leaving off the jacket](Image)

**Fig. 2.** Examples of \( PM \) computed for different actions. Brighter pixels correspond to higher probability.

### 3. ACTION SPACE-TIME TRAJECTORIES

To model the complete evolution of the action we use a 3D observation \( O^t = (x, y, z) \), where \( z = PM^t(x, y) \). In other words, the \( PM \) image is treated as 3D data to be clustered jointly in space and probability domains. Data clustering has the objective to identify main areas in the person’s silhouette characterized by a 3D distribution similar to a given model. The 3D data are modeled with a 3-variate mixture of Gaussians:

\[
p(O^t|A^t) = \sum_{k=1}^{K} \pi_k N(O^t|\mu_k, \Sigma_k)
\]

where \( A^t = \{\mu^t, \Sigma^t, \pi^t\} \) is the set of parameters of the MoG, including the mean vector \( \mu = \{\mu_1, \cdots, \mu_K\} \) for each of the \( K \) components, the covariance matrix \( \Sigma \) and the weight vector \( \pi \). The single 3-variate Gaussian can be written as:

\[
N(O|\mu, \Sigma) = \frac{1}{\sqrt{2\pi} (\det(\Sigma))^{1/2}} \exp \left\{ -\frac{1}{2} (O - \mu)^T \Sigma^{-1} (O - \mu) \right\}
\]

By using the EM algorithm the set of estimated parameters \( \hat{A}^t = \{\hat{\mu}^t, \hat{\Sigma}^t, \hat{\pi}^t\} \) can be easily inferred. To initialize the EM on the first frame \( (t = 0) \) we use the k-means clustering on \( PM^0 \). Conversely, the initialization for the subsequent frames \( (t > 0) \) is based on the estimate on the previous step, i.e. \( A^t = \hat{A}^{t-1} \) (see also the feedback arrow in Fig. 1). This re-initialization process reinforces the next mean tracking step by increasing the probability that the means of the MoG components assume the same ordering in successive frames. Some examples of the segmentation achieved by this process with \( K = 3 \) components are reported in Fig. 3, where a person leaving a pack is shown.

In order to guarantee the association of the same component number to the same mean in successive frames, a specific mean tracking algorithm has been designed. The main idea is to compare the set of estimated parameters \( \hat{A}^t \) with the previous one \( \hat{A}^{t-1} \) to match corresponding MoG components. To do this we assign to each new component \( i \) in \( \hat{A}^t \) the component \( j \) of \( \hat{A}^{t-1} \) at the minimum distance. The distance between two components (i.e., two Gaussian distributions) is computed by using the closed-form expression of the Bhattacharyya distance provided by Kailath in [22]:

\[
\hat{A}^t_i \iff \hat{A}_j^{t-1} \quad \text{iff} \quad j = \arg \min_{q=1, \cdots, K} d(\hat{A}^t_i, \hat{A}_q^{t-1})
\]

\[
d(A_s, A_p) = d_{BHATT}(N(O|\mu_s, \Sigma_s), N(O|\mu_p, \Sigma_p))
\]

\[
= \frac{1}{8} (\mu_s - \mu_p)^T \Sigma^{-1} (\mu_s - \mu_p) +
\]

\[
+ \frac{1}{2} \ln \left( \frac{\det \Sigma}{\sqrt{\det \Sigma_s \det \Sigma_p}} \right)
\]

where \( 2 \cdot \Sigma = \Sigma_s + \Sigma_p \).

The mean tracking algorithm allows to correlate parameters’ estimation in successive frames. Consequently, the 3D means \( \mu_i = (x_i, y_i, z_i) \) of a component \( i \) and the associated covariance matrix \( \Sigma_i \) can be merged to form the sequence of the tracked Gaussian distribution \( T_i = \{N(O^t|\mu_i^t, \Sigma_i^t)\} \) for the point/component \( i \). The set \( \text{STT} = \{T_1, \cdots, T_K\} \) can be used to analyze the action.

### 4. CONSISTENT ACTION RECOGNITION

This problem of consistent labeling is solved adopting a geometric approach that exploits the relations and constraints among cameras’ fields of view (FoV) to impose identities’ consistency. This approach has been called HECL (Homography and Epipolar-based CONsistent Labeling) [23]. In detail, when cameras partially overlap, the shared portion of the scene is analyzed and people identities are matched geometrically. After an initial unsupervised and automatic training phase the overlapping regions among fields of view, groundplane homographies and the epipole location for pairwise overlapping cameras are estimated. The consistent labeling problem is then solved on-line whenever a new object \( \tau \) appears in the field of view of a given camera. The multi-camera system must check whether \( \tau \) corresponds to a completely new object or to one which is already present in the FoV of other cameras. Moreover, the system should deal with groups and identify the objects composing them.
To achieve this, HECOL exploits homographic mapping on the ground plane to warp the point corresponding to the person’s feet between the two cameras, and the epipolar geometry to estimate the vertical axis of the moving person. Let us suppose the new object \( \tau \) appears on camera \( C^1 \). The feet’s point of each of the \( K \) objects in another overlapped camera \( C^2 \) is warped to the image plane of \( C^1 \), forming a set of hypothesis for the matching. Likelihood is then computed by testing the fitness of each hypothesis against current evidence. The main goal is to distinguish between single hypothesis, group hypotheses and possible segmentation errors exploiting only geometrical properties in order to avoid the uncertainties due to color variation and adopting the vertical axis of the object as an invariant feature. It is worth emphasizing that the axis of the object can be warped correctly only with the homography matrix and the knowledge of epipolar constraints among cameras.

Based on geometrical constraints, the warped axis of \( \tau \) in the image plane of \( C^2 \) is univocally identified but its computation is not error free. In order to improve the robustness to computation errors, we account also for the dual process that can be performed for each of the \( K \) potential matching objects: the axis of each object in \( C^2 \) is warped on camera \( C^1 \). The measure of axis correspondence is not merely the distance between axes, but is defined as the number of matching pixels between the warped axis and the foreground blob of the target object. With this approach, two distinct contributions (one forward from \( C_1 \) to \( C_2 \) and one backward from \( C_2 \) to \( C_1 \)) can be computed. At the end, the likelihood is defined as the maximum value between forward and backward contribution and the assignment is performed using a MAP framework.

5. STT COMPARISON FOR ACTION RECOGNITION

The previous sections describe our methodology to model an action as a set of \( K \times n \) (with \( K \) MoG components and \( n \) camera views) trajectories \( T \), given as a sequence of Gaussian distributions. In order to cluster or classify similar actions, single trajectories \( T \) from different actions should be compared. To this aim, defined \( a \) and \( b \) the actions to be compared, a similarity measure \( \Omega (T_i, T_j) \) between every trajectory \( i \) of action \( a \) and every trajectory \( j \) of action \( b \) is needed. Due to segmentation inaccuracies and different velocities in performing actions, the Gaussian distributions composing \( T_i \) and \( T_j \) can not be directly compared point by point. We can borrow from bioinformatics a method for comparing sequences of data in order to find the best inexact matching between them, also accounting for gaps and statistical uncertainty. Among the many different techniques, we used the global alignment [19].

Global alignment of two sequences \( S \) and \( T \) is obtained by first inserting spaces, either into or at the ends of \( S \) and \( T \) so that the length of the sequences will be the same, and then placing the two resulting sequences one above the other so that every symbol or space in one of the sequences is matched to a unique symbol in the other. The algorithm is based on the concept of “modification” to the sequence (analogous to the mutation in a DNA sequence). The modifications to a sequence can be due to indel operations (insertion or deletion of a symbol) or to substitutions. By assigning different weights/costs to these operations it is possible to measure the degree of similarity of the two sequences.

Unfortunately, this algorithm results very onerous in terms of computational complexity if the sequences are long. For this reason, dynamic programming is used to reduce computational time to \( O(n_i \cdot n_j) \), where \( n_i \) and \( n_j \) are the lengths of the two sequences. Without going too much into details, dynamic programming overcomes to the problem of the recursive solution to global alignment by not comparing the same subsequences for more than one time, and exploiting tabular representation to efficiently compute the final similarity score (for further details refer to [19]).

Each element \((p,q)\) of the table contains the alignment score of the distribution \( \mathcal{N} (\mathbf{O}^p | \mu^p_j, \Sigma^p_j) \) of sequence \( T_i \) with the symbol \( \mathcal{N} (\mathbf{O}^q | \mu^q_j, \Sigma^q_j) \) of sequence \( T_j \). Thus, the score can be measured statistically as a function of the distance between the corresponding distributions. If the two distributions result sufficiently similar the score should be high and positive, while if they differ much the score (penalty) should be negative. Assigning zero to the gap penalty, the best alignment can be found by searching for the alignment that maximizes the global score.

Specifically, we measured the distance between distribu-
tions \( p \) and \( q \) using the Bhattacharyya distance:

\[
d_{BHATT}(p, q) = \sqrt{1 - \int_{-\infty}^{+\infty} p(\theta)q(\theta) d\theta}
\]  

(5)

If \( p \) and \( q \) are two Gaussians, the closed-form expression
reported in equation 4 can be used to compute the distance.
Since global alignment requires a score which must be posi-
tive and less than 1 (with 1 corresponding to identical se-
quences), we transform the distance \( d_{BHATT} \) in the coeffi-
cient \( c_{BHATT} = \exp\{-d_{BHATT}\} \).

Assuming that two distributions are sufficiently similar if
the coefficient is above 0.5 and that the score for perfect match
is +2, whereas the score (penalty) for the perfect mismatch is
-1 (that are the typical values used in DNA sequence align-
ments), we can write the general similarity score as follows:

\[
\sigma(p, q) = \begin{cases} 
2 \cdot c_{BHATT} & \text{if } c_{BHATT} \geq 0.5 \\
2 \cdot (c_{BHATT} - 0.5) & \text{if } c_{BHATT} < 0.5 \\
0 & \text{if } p \text{ or } q \text{ are gaps}
\end{cases}
\]  

(6)

After computing the similarity score, it must be normal-
ized to obtain the distance \( \Omega(T_i, T_j) \).

5.1. Single Camera STT Comparison

Since the STTs are composed by a set of trajectories a global
similarity measure that accounts for all the trajectories in
the sets should be adopted to compare the actions. A \( K \times K \)
distance matrix \( \Delta_{a,b} \) is build comparing all the trajectories
of the STT modeling action \( a \) with the STT modeling action \( b \)
using the alignment technique:

\[
\Delta_{a,b} = \begin{bmatrix}
\delta_{1,1} & \delta_{1,2} & \ldots & \delta_{1,K} \\
\delta_{2,1} & \delta_{2,2} & \ldots & \delta_{2,K} \\
\ldots & \ldots & \ldots & \ldots \\
\delta_{K,1} & \delta_{K,2} & \ldots & \delta_{K,K}
\end{bmatrix}
\]  

(7)

where \( \delta_{i,j} = \Omega(T_i, T_j) \) with \( T_i \in STT_a \) and \( T_j \in STT_b \).
The distance matrix is evaluated computing the maximum
eigenvalue. Since distance values are in the real domain,
matrices are guaranteed to have at least a singular positive
eigenvalue. The maximum eigenvalue \( \lambda_{a,b} \) of \( \Delta_{a,b} \) matrix
expresses the variance of distances along the main direction of
distances diffusion. This is equivalent to compare how much
different STT trajectories are in comparison with the ones be-
longing to another action performing the comparison globally
and without considering any association among trajectories
themselves.

5.2. Multi Camera STT extension

When multiple cameras are present, a set of STTs, one for
each camera observing the action, is build exploiting the con-
sistent labeling module previously described. The multi-camera
STT set embodies the information coming from multiple views
allowing to distinguish among actions that may appear simi-
lar from a specific point of view. This enriched discriminant
properties suggests to exploit a method that jointly compares
each STT coming from a specific camera. The \( \Delta \) matrix is
consequently extended to account for STTs coming from the
same camera views. More specifically if \( STT^a \) is the action
descriptor of action \( a \) observed on camera \( C^1 \) and \( STT^b \) ob-
erved on camera \( C^2 \), the matrix \( \Delta^{a,b}_{i,j} \) that compares two dif-
ferent actions observed from both \( C^1 \) and \( C^2 \) becomes:

\[
\Delta^{i,j}_{a,b} = \begin{bmatrix}
\Delta^i_{a,b} & 0 \\
0 & \Delta^j_{a,b}
\end{bmatrix}
\]  

(8)

As stated for the single camera case, the eigenvalues are com-
puted to obtain a scalar similarity measure from distance ma-
trices \( \Delta \). In particular, for every sub-matrix \( \Delta', \) its largest
eigenvalue \( \lambda' \) is computed and the sum used to identify whether
actions are jointly similar on all views.

6. EXPERIMENTAL RESULTS

The proposed approach is conceived for working on a multi-
camera setup. Nevertheless, we first evaluate its performance
on a single camera setup. The nine actions we considered are
summarized in Fig. 4 with an example frame for each action.
Videos are taken from static cameras with a side view, but the
action may also take place on a not-completely-lateral way,
as in the case of Fig. 4(a). The only strong assumption is
that the considered moving people are visible at a sufficient
resolution. In fact, if this assumption does not hold, the EM
algorithm on the MoG would have too few samples, making
it strictly dependent on the initialization seeds and may not
converge.

Several examples for each action have been collected to
form a set of 29 videos. The resulting confusion matrix with
respect to a manually labeled ground truth is reported in Table
1. In general, the system makes only 4 errors on the 29 videos,
resulting in an average accuracy of 86.21%.

In addition to the single camera experiments, we also con-
ducted a preliminary experimentation on a multi-camera setup.
Fig. 5 shows two examples of a “tying shoe laces” action
from two completely different viewpoints. We collected 10
videos of two actions (“walking” and “tying shoe laces”) and
compared them as reported in Section 5.2. Though very pre-
liminary, only two actions have been misclassified (resulting
in an average accuracy of 80%), basically due to the poor res-
olution of one of the cameras.

7. CONCLUSIONS

This paper proposes a markerless approach for body-part track-
ing based on a probability map (integrating pixel membership
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pack</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Walking</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Tying</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Sitting</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Jacket</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>50%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>50%</td>
</tr>
<tr>
<td>Drinking</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Jumping</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>33.3%</td>
<td>33.3%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Pointing</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Oscil.</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 1. Confusion matrix in the case of single camera setup.

Fig. 4. Examples of the actions considered in the single camera setup.

Fig. 5. An example from the multi-camera setup.

8. REFERENCES


