Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

HuPBA8k+: Dataset and ECOC-Graph-Cut based segmentation of human limbs

Daniel Sánchez [a,*], Miguel Ángel Bautista [a,b], Sergio Escalera [a,b]

[a] Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007 Barcelona, Spain
[b] Centre de Visió per Computador, Campus UAB, Edifici O, 08193 Barcelona, Spain

ARTICLE INFO

Article history:
Received 22 January 2014
Received in revised form
5 July 2014
Accepted 23 July 2014

Keywords:
Human limb segmentation
ECOC
Graph-Cuts

ABSTRACT

Human multi-limb segmentation in RGB images has attracted a lot of interest in the research community because of the huge amount of possible applications in fields like Human–Computer Interaction, Surveillance, eHealth, or Gaming. Nevertheless, human multi-limb segmentation is a very hard task because of the changes in appearance produced by different points of view, clothing, lighting conditions, occlusions, and number of articulations of the human body. Furthermore, this huge pose variability makes the availability of large annotated datasets difficult. In this paper, we introduce the *HuPBA8k*+ dataset. The dataset contains more than 8000 labeled frames at pixel precision, including more than 120 000 manually labeled samples of 14 different limbs. For completeness, the dataset is also labeled at frame-level with action annotations drawn from an 11 action dictionary which includes both single person actions and person–person interactive actions. Furthermore, we also propose a two-stage approach for the segmentation of human limbs. In the first stage, human limbs are trained using cascades of classifiers to be split in a tree-structure way, which is included in an Error-Correcting Output Codes (ECOC) framework to define a body-like probability map. This map is used to obtain a binary mask of the subject by means of GMM color modelling and Graph-Cuts theory. In the second stage, we embed a similar tree-structure in an ECOC framework to build a more accurate set of limb-like probability maps within the segmented user mask that are fed to a multi-label Graph-Cut procedure to obtain final multi-limb segmentation. The methodology is tested on the novel *HuPBA8k*+ dataset, showing performance improvements in comparison to state-of-the-art approaches. In addition, a baseline of standard action recognition methods for the 11 actions categories of the novel dataset is also provided.

1. Introduction

Human analysis in RGB images is a challenging task because of the high variability of the human body, including the wide range of human poses, lighting conditions, cluttering, clothes, appearance, background, point of view, and number of human body limbs. Even so, human analysis in visual data has become one of the more interesting areas of research in Computer Vision and Pattern Recognition because of its capabilities in final applications (i.e. human–computer interaction, surveillance, gaming, eHealth, and interactive virtual reality systems). In this sense, the common pipeline for human body analysis in visual data uses to be defined in a bottom-up fashion. First, the human body limbs are segmented and the body pose is estimated (often with a prior person/background segmentation or person detection step). Then, once the body pose is estimated higher abstraction analysis can be performed. Usually, the following step in the pipeline is action/gesture recognition, since actions can be seen as a set of estimated body poses varying over time.

The first step of the pipeline, which concerns human limb segmentation or pose estimation in RGB images, has been a core problem in the Computer Vision field since its early beginnings. In this particular problem the goal is to provide a complete segmentation of each of the defined human body parts appearing in an image, discriminating human limbs from each other and from the rest of the image. Usually, human body segmentation is treated in a two-stage fashion. First, a human body part detection step is performed, and then, these human part detections are used as prior knowledge to be optimized by segmentation/inference strategies in order to obtain the final human-limb segmentation. In the literature one can find many works that follow this two-stage scheme. Bourdev and Malik [1] used body part detections in an AND–OR graph to obtain the pose estimation. Vinet et al. [2]

[*] Corresponding author.
E-mail addresses: gammarl@gmail.com (D. Sánchez), mbautista@ub.edu (M. Ángel Bautista), sergio@maia.ub.es (S. Escalera).

http://dx.doi.org/10.1016/j.neucom.2014.07.069
0925-2312/© 2014 Elsevier B.V. All rights reserved.

proposed to use Conditional Random Fields based on body part detectors to obtain a complete person/background segmentation. Nevertheless, one of the methods that have generated more attraction is the well known pictorial structure for object recognition introduced by Felzenszwalb and Huttenlocher [3]. Some works have applied an adaptation of pictorial structures using a set of joint limb marks to infer spatial probabilities [4–7]. Later on, an extension was presented by Yang and Ramanan [8,9] which proposed a discriminatively trained pictorial structure that models the body joints instead of limbs. In contrast, there is also current tendency to use Graph-Cuts optimization to segment the human limbs [10] or full person segmentation [11].

The common step after estimating the pose of a subject within the pipeline of human body analysis is analyzing non-verbal communication in terms of actions and/or gestures, which can be interpreted as a set of poses varying over time. In this sense, in order to deal with action/gesture recognition there exist a wide number of methods based on dynamic programming algorithms for alignment and clustering of temporal series [12,13]. One of the most common methods for Human Gesture Recognition based on dynamic programming is Dynamic Time Warping (DTW) [14,15,13], since it offers a simple yet effective temporal alignment between sequences of different lengths. Other probabilistic methods such as Hidden Markov Models (HMM) or Conditional Random Fields (CRF) have been commonly used in the literature [16]. Some other methods also considered for action/gesture recognition include Neural Networks approaches, Boosting variants, and Random Forest [17,18].

The Computer Vision community has been lately focusing their efforts on developing methods for both pose estimation and action/gesture recognition. However, one of the main problems is the necessity of public available datasets containing annotations of all the variabilities the methods have to deal with. Substantial effort has been put on designing datasets with different scenarios, people and illumination characteristics. Datasets such as Parse [7], Buffy [19], UIUC People [20], and Pascal VOC [21] are widely used to evaluate different pose estimation and action/gesture recognition methods. However, these public available datasets fail to provide a sound framework to validate pose recovery systems (i.e. the number of samples per limb is small, the labeling is not accurate, and there are no interactions of actors). Given this lack of sound and refined public datasets for human multi-limb segmentation and/or action/gesture recognition, we introduce the *HuPBA8k+* dataset, which to the best of our knowledge is the biggest RGB human-limb labeled dataset. The dataset contains more than 8000 labeled frames at pixel precision and more than 120 000 manually labeled samples of 14 different limbs. In addition, the *HuPBA8k+* dataset is also labeled with action annotations drawn from an 11 action dictionary which includes both single person actions and interactive actions (actions performed by more than one person).

Furthermore, we also extend our work of [22] by proposing a two-stage approach for the segmentation of human limbs. In a first stage, a set of human limbs is normalized by main orientation to be rotation invariant, described using Haar-like features, and trained using cascades of Adaboost classifiers to be split in a tree-structure way. Once the tree-structure is trained, it is included in a ternary Error-Correcting Output Codes (ECOC) framework. This first classification step is applied in a windowing way on a new test image, defining a body-like probability map, which is used as an initialization of a binary GRAB Cuts optimization procedure. In the second stage, we embed a similar tree-structured partition of limbs in a ternary ECOC framework and we use Support Vector Machines (SVMs) with HOG descriptors to build a more accurate set of limb-like probability maps within the segmented user binary mask that are fed to a multi-label Graph-Cut optimization procedure to obtain the final human multi-limb segmentation. We tested our ECOC-Graph-Cut based approach in the novel *HuPBA8k+* dataset and compared with state-of-the-art pose recovery approaches, obtaining performance improvements in both person/background and multi-limb segmentation steps. For completeness, we also provide action recognition results as a baseline for the *HuPBA8k+* dataset. Summarizing, our key contributions are the following:

- We introduce the *HuPBA8k+* dataset, the largest RGB labeled dataset of human limbs, with more than 120 000 manually annotated limbs. The dataset also includes frame-level annotation for 11 action/gesture categories.
- We propose a two stage approach based on ECOC and Graph-Cuts for the segmentation of human limbs in RGB images.
- We provide a baseline for Action Recognition in the novel dataset.

The rest of the paper is organized as follows: Section 2 introduces the novel dataset. Section 3 introduces the proposed method. Section 4 presents the experimental results, and finally, Section 5 concludes the paper.

2. *HuPBA8K+* dataset

Automatic human limb detection and segmentation, human pose recovery and human behavior analysis are challenging problems in computer vision, not only for the intrinsic complexity of the tasks, but also for the lack of large public and annotated datasets. Usually, public available datasets lack refined labeling or contain a very reduced number of samples per limb (e.g. *Buffy Stickmen V3.01*, *Leeds Sports* and *Hollywood Human Actions* [19,23,24]). In addition, large datasets often use synthetic samples or capture human limbs with sensor technologies such as *MoCap* in very controlled environments [25].

Being aware of this lack of public available datasets for multi-limb human pose detection, segmentation and action/gesture recognition, we present a novel fully limb labeled dataset, the *HuPBA8k+* dataset. This dataset is formed by more than 8000 frames where 14 limbs are labeled at pixel precision.[1] Furthermore, the *HuPBA8k+* dataset also contains gesture/action annotations for 11 isolated and collaborative action categories. The main characteristics of the dataset are the following:

1. The images are obtained from 9 videos (RGB sequences) and a total of 14 different actors appear in those 9 sequences. In concrete, each sequence has a main actor (9 in total) which during the video interacts with secondary actors performing a set of different actions.
2. Each video (RGB sequence) was recorded with a mean 15 fps rate.
3. RGB images were stored with resolution 480 × 360 in BMP file format.
4. For each actor present in an image 14 limbs (if not occluded) were manually tagged: Head, Torso, R–L Upper-arm, R–L Lower-arm, R–L Hand, R–L Upper-leg, R–L Lower-leg, and R–L Foot.
5. Limbs are manually labeled using binary masks and the minimum bounding box containing each subject is defined.
6. The actors appear in a wide range of different poses and performing different actions/gestures.
7. For each video we manually labeled a set of 11 gesture/action categories: Wave, Point, Clap, Crouch, Jump, Walk, Run, Shake Hands, Hug, Kiss, and Fight.

[1] The whole number of manual labeled limbs exceeds 120 000.

Table 1
Easy and challenging aspects of the *HuPBA8k+* dataset.

Easy
Fixed Camera
Frontal point of view
Full body capture
The main actor is kept within a sequence
Several instances of each gesture/action
Gestures/actions differentiated by an idle pose in most cases
Fixed background across all video sequences
Challenging
Within each sequence
Gestures/actions execution involve most limbs
Large variability of poses
Some gestures/actions imply the interaction of various actors
Some parts of the body may be occluded
Between sequences
Variations in clothing, skin color, gender, height and corporal conditions
Some parts of the body may be occluded

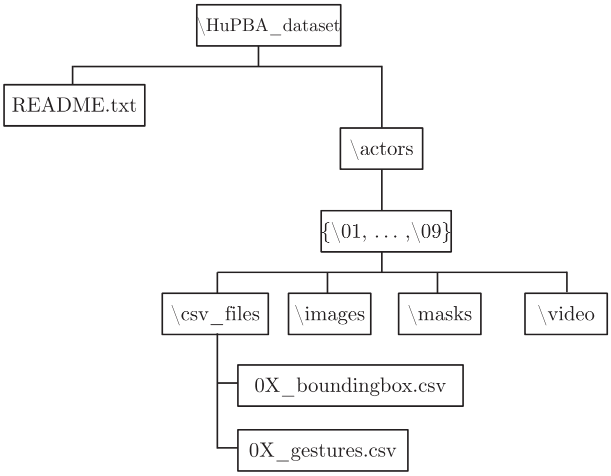


Fig. 1. Folders structure.

Finally, the easy and challenging aspects of the *HuPBA8k+* dataset are listed in Table 1.

2.1. *Data format and structure*

The dataset we introduce is composed of RGB images, labeled limbs (binary masks) and additional information that has a specific structure to distinguish the location of limbs and gestures/actions for each actor. Additionally, for each actor, a pair of structured files is created to store the location of the bounding-boxes for each RGB image and the start-end frames associated with the gestures/actions executed. The folder structure that contains the *HuPBA8k+* dataset is shown in Fig. 1.[2]

2.1.1. Folder \images

In this folder, we store the set of frames for a given video sequence. The folder *images* contains the sequence of RGB images (480 × 360 pixels). Each image name has the structure *idActor_numberFrame.bmp*, where

- **idActor**: Numerical identifier of the actor $\{01, 02, \ldots, 09\}$.
- **numberFrame**: Numerical identifier of the image in the sequence.

2.1.2. Folder \masks

This folder contains the binary masks for each one of the 14 limbs appearing on each frame. In the case of two actors appearing in a frame, there will be an *id* for each one in order to distinguish limbs. Each binary mask name has the structure *idActor_numberFrame_idUser_idLimb.bmp*, where

- **idActor**: Numerical identifier of the actor $\{01, 02, \ldots, 09\}$.
- **numberFrame**: Numerical identifier of the image in the sequence.
- **idUser**: Numerical identifier for the actor that appears in the image. Values $\{1, 2, \ldots, n\}$. In case of appearing two actors: The main actor and another, the main actor is 1, the second is 2, and so on.
- **idLimb**: Numerical identifier of the limb, which is described in Fig. 2.

2.1.3. Bounding-boxes

In addition, for each sequence there is a file *0X_boundingbox.csv* located in the directory *csv_files* that contains the bounding-boxes of all actors that appear in that sequence. That is, for each actor that appears in an image, its bounding-box is given. In the case of two actors appearing in an image, two bounding-boxes will be described, one for each actor, as shown in Fig. 3. The *csv* file contains the following structure:

- **id_user**: Numerical identifier for the actor that appears in the image. Values $\{1, 2, \ldots, n\}$. In case of appearing two actors: The main actor and another, the main actor is 1 and the second is 2. Thus, there will be two bounding-boxes, one for 1, another for 2, and so on.

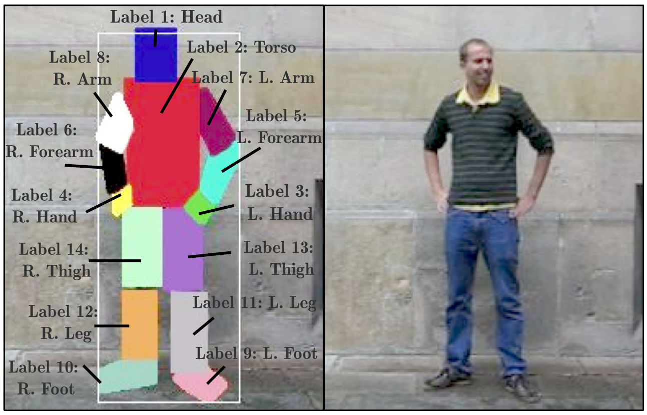


Fig. 2. Human-limb labelling on the *HuPBA8k+* dataset.

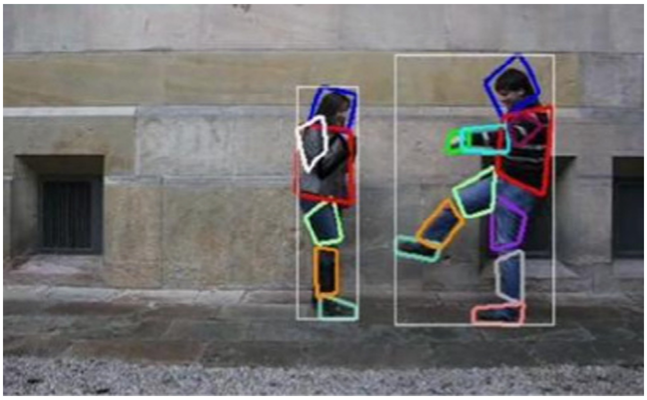


Fig. 3. Sample of two bounding-boxes in a frame.

[2] Web page of the dataset will be public after acceptance of the paper.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

- **number_frame**: Numerical identifier of the image in the sequence.
- **x**: Minimum position of X. That is, the leftmost.
- **y**: Minimum position of Y. That is, the uppermost.
- **width**: Width of the bounding-box.
- **height**: Height of the bounding-box.

2.1.4. Gestures/Actions

Besides the human-limb labeling provided on the dataset, we also annotated gestures/actions performed by the actors. The 11 gesture/action categories labeled are the following: Wave, Point, Clap, Crouch, Jump, Walk, Run, Shake Hands, Hug, Kiss and Fight. An example of key frames for the different gesture/action categories is shown in Fig. 4. Each set of gestures/actions performed by an actor is associated with a file *./csv_files/0X_gestures.csv* that contains the following structure:

- **id_user**: Numerical identifier for the actor that appears in the image. Values $\{1, 2, \ldots, n\}$.
- **label_gesture**: Numerical identifier related to the gesture/action performed. There are gestures/actions that involve just one actor (i.e. walk or run), and others more than one actor (i.e. fight or kiss).

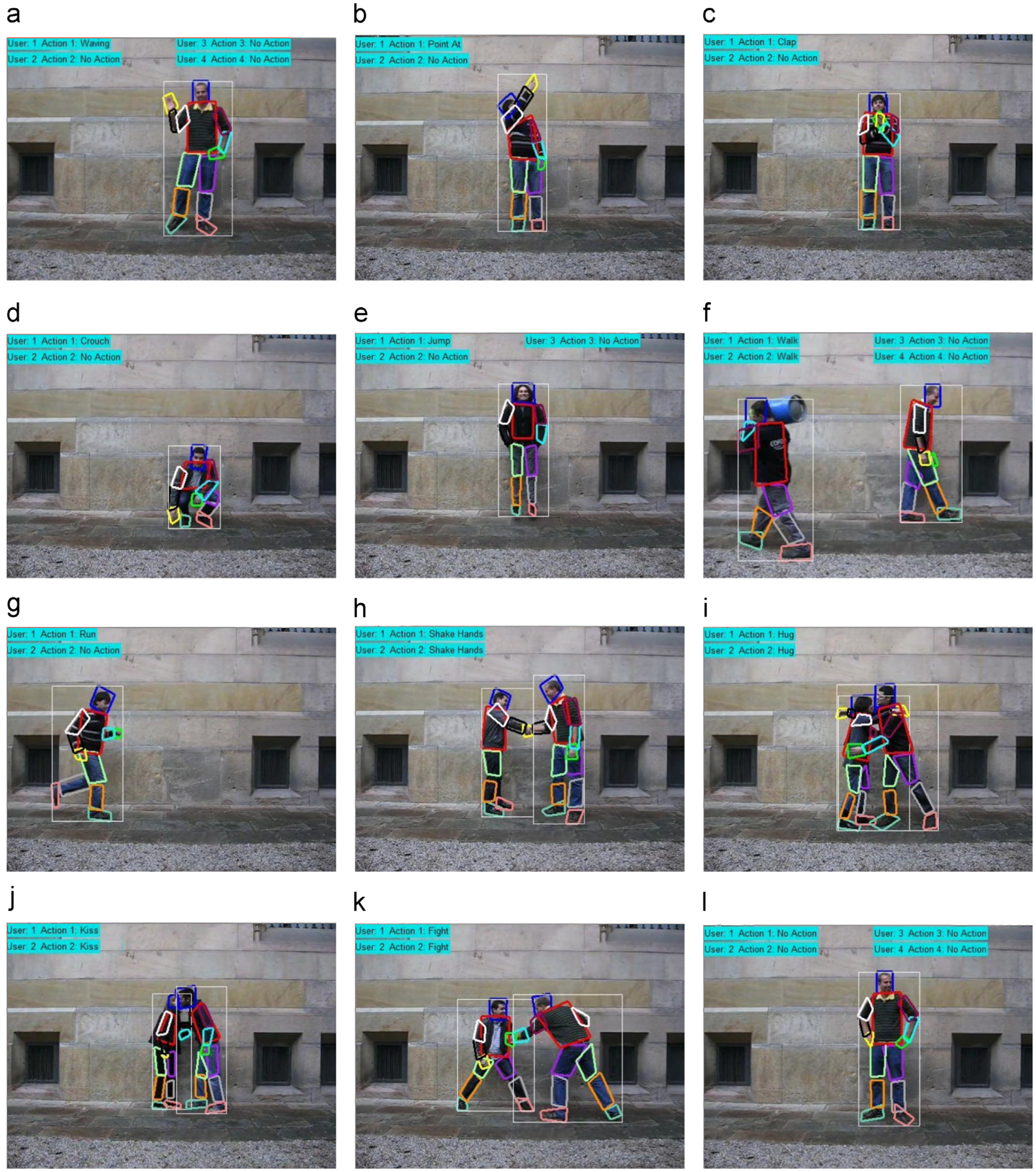

a
User: 1 Action 1: Waving
User: 2 Action 2: No Action
User: 3 Action 3: No Action
User: 4 Action 4: No Action
b
User: 1 Action 1: Point At
User: 2 Action 2: No Action
c
User: 1 Action 1: Clap
User: 2 Action 2: No Action
d
User: 1 Action 1: Crouch
User: 2 Action 2: No Action
e
User: 1 Action 1: Jump
User: 2 Action 2: No Action
User: 3 Action 3: No Action
f
User: 1 Action 1: Walk
User: 2 Action 2: Walk
User: 3 Action 3: No Action
User: 4 Action 4: No Action
g
User: 1 Action 1: Run
User: 2 Action 2: No Action
h
User: 1 Action 1: Shake Hands
User: 2 Action 2: Shake Hands
i
User: 1 Action 1: Hug
User: 2 Action 2: Hug
j
User: 1 Action 1: Kiss
User: 2 Action 2: Kiss
k
User: 1 Action 1: Fight
User: 2 Action 2: Fight
l
User: 1 Action 1: No Action
User: 2 Action 2: No Action
User: 3 Action 3: No Action
User: 4 Action 4: No Action

Fig. 4. Different gesture categories labeled on the *HuPBA8k+* dataset. Images from (a) to (g) illustrate single actor gestures/actions, and images from (h) to (k) show gestures/actions that required interacting with a secondary actor. Additionally, (l) shows an example of an existing idle gesture/action.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

- **start_frame**: The number of image where the gesture/action starts.
- **end_frame**: The number of the image where the gesture/action ends.

Finally, in Table 2 we compare the *HuPBA8k+* dataset characteristics with some publicly available datasets. These public datasets are chosen taking into account the variability of limbs and gestures/actions. One can see that the novel dataset offers higher number of annotated limbs at pixel precision in comparison with state-of-the-art public available datasets. In case of gestures/actions, there is more equality in the number of gestures/actions set with the other datasets (i.e. HOLLYWOOD (HW), MMGR13, Human Actions). In contrast, MMGR13 presents much more variety of gestures/actions and samples than the proposed dataset.

3. ECOC and Graph-Cut based multi-limb segmentation

In the following subsections we describe the proposed system for automatic segmentation of human limbs. The main goal of this approach is to facilitate the multi-limb segmentation as a collection of softer sub-stages for the current database. Our main contributions are (i) The *HuPBA8k+* dataset, the largest RGB labeled dataset of human limbs, with more than 120 000 manually annotated limbs. (ii) A two stage approach based on ECOC and Graph-Cuts for the

Table 2
Comparison of public dataset characteristics.

Dataset	Labeling at pixel precision	# limbs	# labeled limbs	# frames	Full body	Bounding-box on limbs	Gesture annotation	# gestures	# gesture samples
HuPBA	Yes	14	124 761	8234	Yes	Yes	Yes	11	235
PARSE[7]	No	10	3050	305	Yes	Yes	No	–	–
BUFFY[19]	No	6	4488	748	No	Yes	No	–	–
UIUC people[20]	No	14	18 186	1299	Yes	Yes	No	–	–
LEEDS SPORTS[23]	No	14	28 000	2000	Yes	Yes	No	–	–
HW[24]	–	–	–	–	–	No	Yes	8	430
MMGR13[17]	No	16	27 532 800	1 720 800	Yes	Yes	Yes	20	13 858
H.Actions[26]	No	–	–	–	Yes	No	Yes	6	600
Pascal VOC[21]	Yes	5	8500	1218	Yes	Yes	No	–	–
MPII Human Pose [27]	Yes	14	567 308	40 522	Yes	Yes	Yes	20	491
FLIC[28]	No	29	145 087	5003	No	Yes	No	–	–
H3D[29]	No	19	38 000	2000	No	Yes	No	–	–

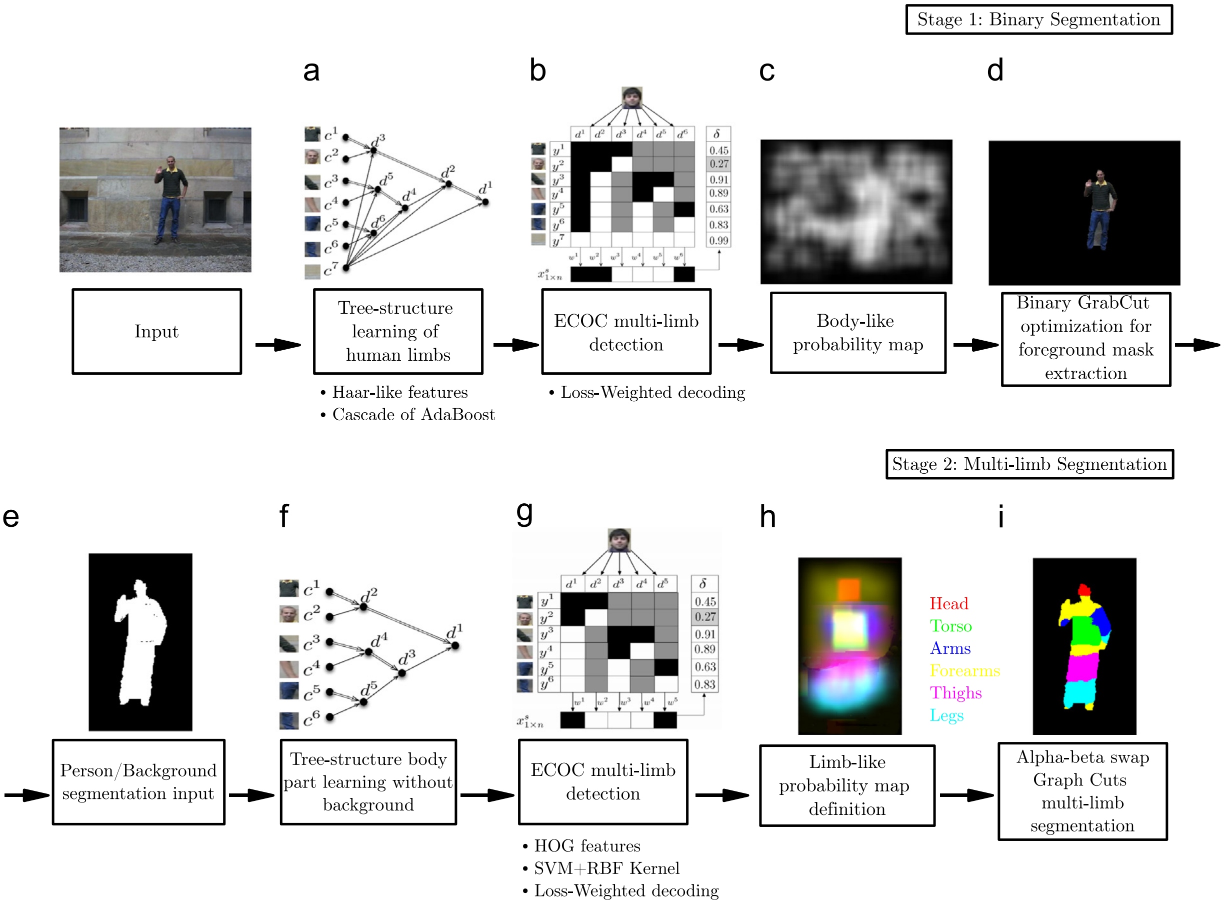

Stage 1: Binary Segmentation
a
b
c
d
Input
Tree-structure learning of human limbs
• Haar-like features
• Cascade of AdaBoost
ECOC multi-limb detection
• Loss-Weighted decoding
Body-like probability map
Binary GrabCut optimization for foreground mask extraction
Stage 2: Multi-limb Segmentation
e
f
g
h
i
Person/Background segmentation input
Tree-structure body part learning without background
ECOC multi-limb detection
• HOG features
• SVM+RBF Kernel
• Loss-Weighted decoding
Limb-like probability map definition
Head
Torso
Arms
Forearms
Thighs
Legs
Alpha-beta swap Graph Cuts multi-limb segmentation

Fig. 5. Scheme of the proposed human-limb segmentation method.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

segmentation of human limbs in RGB image. (iii) A baseline for action recognition in the dataset. The first stage, which focuses on binary person/background segmentation, consists of four main steps: (a) *Body part learning using cascade of classifiers with Haar-like features*. The selection of this model is based on its simplicity and fast computation, at the same time that allows training an unbalanced binary problem while reducing the false positive detection rate. (b) *Tree-structure learning of human limbs*. A tree structure approach allows defining the groups of body parts by taking into account their visual appearance and kinematic constraints. (c) *ECOC multi-limb detection*. The ECOC framework is used as a refinement process that permits us to enhance the soft cascade detections by learning a tree human limbs embedded in it. (d) *Binary Grab-Cut optimization for foreground extraction* is used to obtain a pixel-wise segmentation of the human body. In the second stage, we segment the person/background binary mask into different limb regions. This stage consists of four steps: (e) *Tree-structure body part learning without background*. A second tree structure is created with the same grouping than (b) but without taking into account the background removed by the cascade detections. (f) *ECOC multi-limb detection*. The soft cascade detections are enhanced with a second model composed of HOG descriptors and SVM classifiers as both jointly combined obtain a more robust performance. (g) *Limb-like probability map definition* and (h) *Alpha-beta swap Graph-Cuts multi-limb segmentation* as an extension of the binary GRAB cuts for multi-label segmentation that provides very accurate results by taking into account contextual information. The scheme of the proposed system is illustrated in Fig. 5.

3.1. Body part learning using cascade of classifiers

The core of most human body segmentation methods in the literature relies on body part detectors. In this sense, most part detectors in the literature follow a cascade of classifiers architecture [30–34]. Cascades of classifiers are based on the idea of learning an unbalanced binary problem by using the positive outputs of a classifier d^i as an input for the following classifier d^{i+1}. Particularly, this cascade structure allows any classifier to refine the prediction by reducing the false positive rate at every stage of the cascade. In this sense, we use AdaBoost as the base classifier in our cascade architecture. In addition, in order to make the body part detection rotation invariant, all body parts are rotated to the dominant gradient region orientation. Then, Haar-like features are used to describe the body parts.

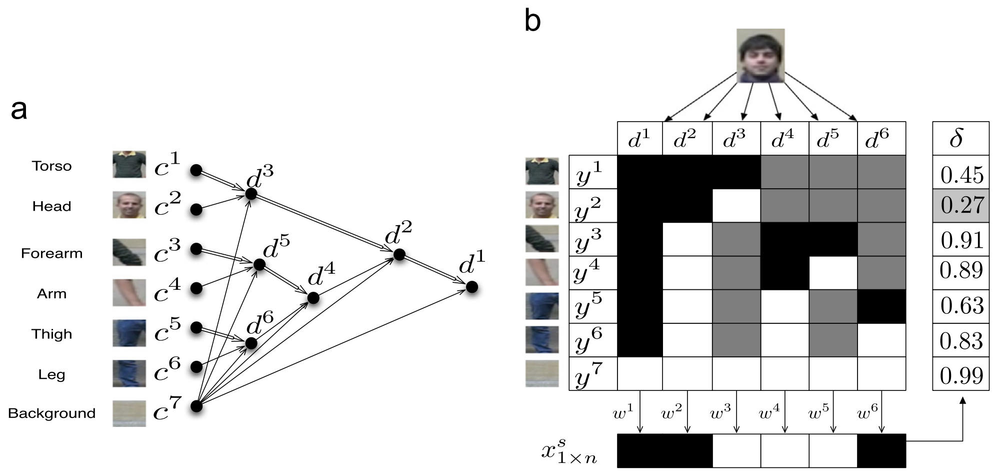
Because of its properties, cascade of classifiers are usually trained to split one visual object from the rest of the possible objects of an image. This means that the cascade of classifiers learns to detect a certain object (body part in our case), ignoring all other objects (all other body parts). However, if we define our problem as a multi-limb detection procedure, some body parts are similar in appearance, and thus, it makes sense to group them in the same visual category. Because of this reason, we propose to learn a set of cascade of classifiers where a subset of limbs is included in the positive set of a cascade, and the remaining limbs are included as negative instances together with background images in the negative set of the cascade. Applying this grouping for different cascades of classifiers in a tree-structure way and combining them in an Error-Correcting Output Codes (ECOC) framework enable the system to perform multi-limb detection [35].

3.2. Tree-structure learning of human limbs

The first issue to take into account when defining a set of cascades of classifiers is how to define the groups of limbs to be learnt by each individual cascade. For this task, we propose to train a tree-structure cascade of classifiers. This tree-structure defines the set of meta-classes for each dichotomy (cascade of classifiers) taking into account the visual appearance of body parts, which has two purposes. On one hand, we aim to avoid dichotomies in which body parts with different visual appearances belong to the same meta-class. On the other hand, the dichotomies that deal with classes that are difficult to learn (body parts with similar visual appearance) are defined taking into account few classes. An example of the body part tree-structure defined taking into account these issues for a set of 7 body limbs is shown in Fig. 6(a). Notice that classes with similar visual appearance (e.g. upper-arm and lower-arm) are grouped in the same meta-class in most dichotomies. In addition, dichotomies that deal with difficult problems (e.g. d^5) are focused only in the difficult classes, without taking into account all other body parts. In this case, class c^7 denotes the background.

3.3. ECOC multi-limb detection

In the ECOC framework, given a set of N classes (body parts) to be learnt, n different bi-partitions (groups of classes or dichotomies) are formed, and n binary problems over the partitions are trained [36]. As a result, a codeword of length n is obtained for each class, where each position (bit) of the code corresponds to a response of a given classifier d (coded by $+1$ or -1 according to their class set membership, or 0 if



	d^1	d^2	d^3	d^4	d^5	d^6	δ
y^1	■	■	■	▨	▨	▨	0.45
y^2	■	■	□	▨	▨	▨	0.27
y^3	■	□	▨	■	■	▨	0.91
y^4	■	□	▨	■	□	▨	0.89
y^5	■	□	▨	□	▨	■	0.63
y^6	■	□	▨	□	▨	□	0.83
y^7	□	□	□	□	□	□	0.99
$x^s_{1\times n}$	■	■	□	□	□	■	

Fig. 6. (a) Tree-structure classifier of body parts, where nodes represent the defined dichotomies. Notice that the single or double lines indicate the meta-class defined. (b) ECOC decoding step, in which a head sample is classified. The coding matrix codifies the tree-structure of (a), where black and white positions are codified as $+1$ and -1, respectively. c, d, y, w, X, and δ correspond to a class category, a dichotomy, a class codeword, a dichotomy weight, a test codeword, and a decoding function, respectively.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

a particular class is not considered for a given classifier). Arranging the codewords as rows of a matrix, we define a *coding matrix* M, where $M \in \{-1, 0, +1\}^{N \times n}$. During the *decoding* (or testing) process, applying the n binary classifiers, a code x is obtained for each data sample ρ in the test set. This code is compared to the base codewords (y^i, $i \in [1, \ldots, N]$) of each class defined in the matrix M, and the data sample is assigned to the class with the *closest* codeword [35].

The ECOC coding step has been widely tackled in the literature either by predefined or problem-dependent strategies. However, recent works showed that problem-dependent strategies can obtain high performance by focusing on the idiosyncrasies of the problem [37]. Following this fashion, we define a problem dependent coding matrix in order to allow the inclusion of cascade of classifiers and learn the body parts. In particular, we propose to use a predefined *coding* matrix in which each dichotomy is obtained from the body part tree-structure described in the previous section. Fig. 6(b) shows the coding matrix codification of the tree-structure in Fig. 6(a).

3.3.1. *Loss-weighted decoding using cascade of classifier weights*

In the ECOC *decoding* step an image is processed using a windowing method, and then, each image patch, that is, a sample ρ, is described and tested. In this sense, each classifier d outputs a prediction whether ρ belongs to one of the two previously learnt meta-classes. Once the set of predictions $x^{\rho}_{1\times n}$ is obtained, it is compared to the set of codewords of M, using a decoding function $\delta(x^{\rho}, M)$. Thus, the final prediction is the class with the codeword that minimizes $\delta(x^{\rho}, M)$. In [35] the authors proposed a problem-dependent decoding function (distance function that takes into account classifier performances) obtaining very satisfying results. Following this core idea, we use the Loss-Weighted decoding of Eq. (1), where M_W is a matrix of weights and L is a loss function ($L(\theta) = \exp^{-\theta}$).

$$\delta_{LW}(x^s, i) = \sum_{j=1}^{n} M_W(i,j) L(y^i_j \cdot d^j(x^s)) \tag{1}$$

In Eq. (1), M_W (weight matrix) corresponds to the product of cascade accuracies at each stage. Thus, each column i of M_W is assigned a weight w^i as

$$w^i = \prod_{j=1}^{k} \frac{TP(d^i_j) + TN(d^i_j)}{TP(d^i_j) + FN(d^i_j) + FP(d^i_j) + TN(d^i_j)}, \tag{2}$$

for a cascade of classifiers of k stages, where d^i_j stands for the i-th cascade and stage j, $j \in [1, \ldots, k]$, and TP, TN, FN, and FP compute the number of true positives, true negatives, false negatives and false positives, respectively. Finally, a body-like probability map $P^{bl} \in [0,1]^{l \times w}$, where l and w are the length and the width of I respectively, is built. This map contains, at each position P^{bl}_{ij}, the proportion of body part detections for each pixel over the total number of detections for the whole image. In other words, pixels belonging to the human body will show a higher body-like probability than the pixels belonging to the background. Examples of probability maps obtained from ECOC outputs are shown in Fig. 9(e) and (g) (see also step (c) in Fig. 5).

3.4. *Binary Grab-Cut optimization for foreground mask extraction*

Grab-Cut [10] has been widely used for interactive background/foreground extraction (binary segmentation). Formally, given a color image I, let us consider the array $z = (z_1, \ldots, z_q, \ldots, z_Q)$ of Q pixels where $z_i = (R_i, G_i, B_i)$, $i \in [1, \ldots, Q]$ in RGB space. The segmentation is defined as an array $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_Q)$, $\alpha_i \in \{0, 1\}$, assigning a label to each pixel of the image indicating if it belongs to background or foreground. A trimap T is defined consisting of three regions: T_B, T_F and T_U, each one containing initial background, foreground, and uncertain pixels, respectively. Pixels belonging to T_B and T_F are clamped as background and foreground respectively—which means Grab-Cut will not be able to modify these labels, whereas those belonging to T_U are actually the ones the algorithm will be able to label. Color information is introduced by GMMs. A full covariance GMM of U components is defined for background pixels ($\alpha_i = 0$), and another one for foreground pixels ($\alpha_j = 1$), parameterized as follows:

$$\boldsymbol{\theta} = \{\pi(\alpha, u), \mu(\alpha, u), \Sigma(\alpha, u), \alpha \in \{0, 1\}, u = 1 \ldots U\}, \tag{3}$$

with π being the weights, μ the means and Σ the covariance matrices of the model. We also consider the array $\mathbf{u} = \{u_1, \ldots, u_i, \ldots u_Q\}$, $u_i \in \{1, \ldots U\}$, $i \in [1, \ldots, Q]$ indicating the component of the background or foreground GMM (according to α_i) the pixel z_i belongs to. The energy function for segmentation E is then

$$\mathbf{E}(\boldsymbol{\alpha}, u, \boldsymbol{\theta}, z) = U(\boldsymbol{\alpha}, u, \boldsymbol{\theta}, z) + \lambda V(\boldsymbol{\alpha}, z), \tag{4}$$

where $\mathbf{U}$ is the likelihood potential based on the probabilities $p(\cdot)$ of the GMM:

$$\mathbf{U}(\boldsymbol{\alpha}, \mathbf{u}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i -\log\ p(z_i | \alpha_i, u_i, \boldsymbol{\theta}) - \log\ \pi(\alpha_i, u_i), \tag{5}$$

and $\mathbf{V}$ is a regularizing prior assuming that segmented regions should be coherent in terms of color, taking into account a neighborhood $\mathcal{N}$

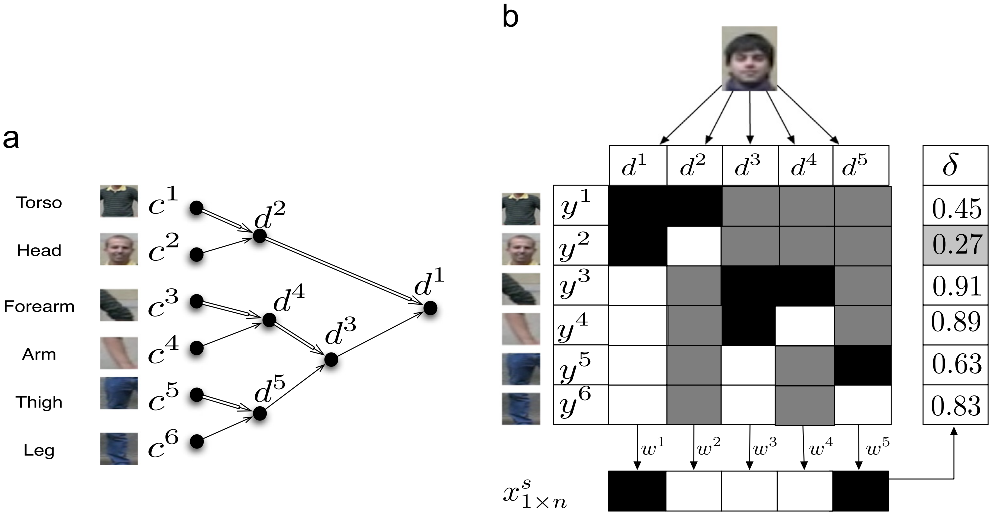


Fig. 7. (a) Tree-structure classifier of 6 body parts, (b) ECOC decoding step.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

around each pixel:

$$\mathbf{V}(\boldsymbol{\alpha},\mathbf{z}) = \gamma \sum_{\{m,q\} \in \mathcal{N}} [\alpha_q \neq \alpha_m] \exp(-\beta \|z_m - z_q\|^2), \tag{6}$$

where weight $\lambda \in \mathbb{R}^+$ specifies the relative importance of the boundary term against the unary term U.

With this energy minimization scheme and given the initial trimap T, the final segmentation is performed using a minimum cut algorithm. However, we propose to omit the classical semi-automatic trimap initialization by an automatic trimap assignment based on the human body probability map $P^{bl} \in [0,1]^{l \times w}$. In this sense, depending on the probability of each pixel it will be assigned to a certain tag T_B, T_F and T_U.

3.5. Tree-structure body part learning without background

Once the binary person/background segmentation is performed by means of Grab-Cut (mask shown in Fig. 5(e)), we apply a second procedure in order to split the person mask into a set of human limbs.

For this step, we define a new tree-structure classifier similar to the one described in Section 3.2 without including the background class c^7 shown in Fig. 6(a). An example of the tree-structure body part taking into account the set of 6 body limbs is shown in Fig. 7(a).

3.6. ECOC multi-limb detection

In order to obtain an accurate detection of human limbs within the segmented user mask, we base on HOG descriptor [38] and SVM classifier which have shown to obtain robust results in human estimation scenarios [38,10,30]. We extract HOG features for the different body parts (previously normalized to dominant region orientation), and then, SVM classifiers are trained on that feature space, using a Generalized Gaussian RBF Kernel based on Chi-squared distance [39].

Table 3
Prior cost between each pair of labels.

	Head	Torso	Arms	Forearms	Thighs	Legs	Background
Head	0	20	35	50	70	90	1
Torso	20	0	15	25	40	70	1
Arms	35	15	0	10	60	80	1
Forearms	50	25	10	0	30	60	1
Thighs	70	40	60	30	0	10	1
Legs	90	70	80	60	10	0	1
Background	1	1	1	1	1	1	1

This stage follows a similar pipeline as the one described in Section 3.3. In this sense, each SVM classifier learns a binary partition of human limbs but without taking into account the background class. As shown in Fig. 6(b), we train $n=6$ SVMs with different binary human-limb partitions.

At the ECOC decoding step, we also use the Loss-Weighted decoding [35] function shown in Eq. (1) (an example is shown in Fig. 7(b)). In this sense, for each RGB test image corresponding to the binary mask shown in Fig. 5(e), we adopt a sliding window approach and test each patch on our ECOC multi limb recognition system. Then, based on the ECOC output we construct a set of limb-like probability maps. Each map P^c contains, at each position P^c_{ij}, the probability of pixel at the entry (i,j) of belonging to the body part class c, where $c \in \{1,2,\ldots,6\}$. This probability is computed as the proportion of detections at point (i,j) over all detections for class c. Examples of probability maps obtained from ECOC outputs are shown in Fig. 5(h). While Haar-like based on AdaBoost gave us a very accurate and fast initialization of human regions for binary user segmentation, in this second step, HOG-SVM is applied in a reduced region of the image, providing better estimates of human limb locations.

3.7. Alpha-beta swap Graph-Cuts multi-limb segmentation

We base our proposal on Graph-Cuts theory to tackle our human-limb segmentation problem [10,11,40–42]. In [42], Boykov et al. developed an algorithm, named as α–β swap graph-cut, which is able to cope with the multi-label segmentation problem. The α–β swap graph-cut is an extension of binary GRAB cuts that performs an iterative procedure where each pair of labels (α_q, α_m), $\{m,q\} \in \{1,2,\ldots,6\}$, is segmented using Graph-Cuts. This procedure segments all α pixels from β pixels with Graph-Cuts and the algorithm will update the α–β combination at each iteration until convergence. However, to cope with the multi-label case, an extension of the binary GRAB Cuts optimization framework described in Section 3.4 is needed.

In this sense, $\alpha_i \in \{1,\ldots,c\}$ and an initial labeling $T \in \{T_1,\ldots,T_c\}$ are defined by an automatic trimap assignment based on the set of limb-like probability maps $P^c \in [0,1]^{l \times w}$ defined in the previous section. In addition, the coefficient that multiplies the exponential term in Eq. (6), $[\alpha_q \neq \alpha_m]$, is changed to $\Omega(c_q, c_m)$, which penalizes relations between pixels z_q and z_m depending on their label assignations and a user-predefined pair-wise cost to each possible

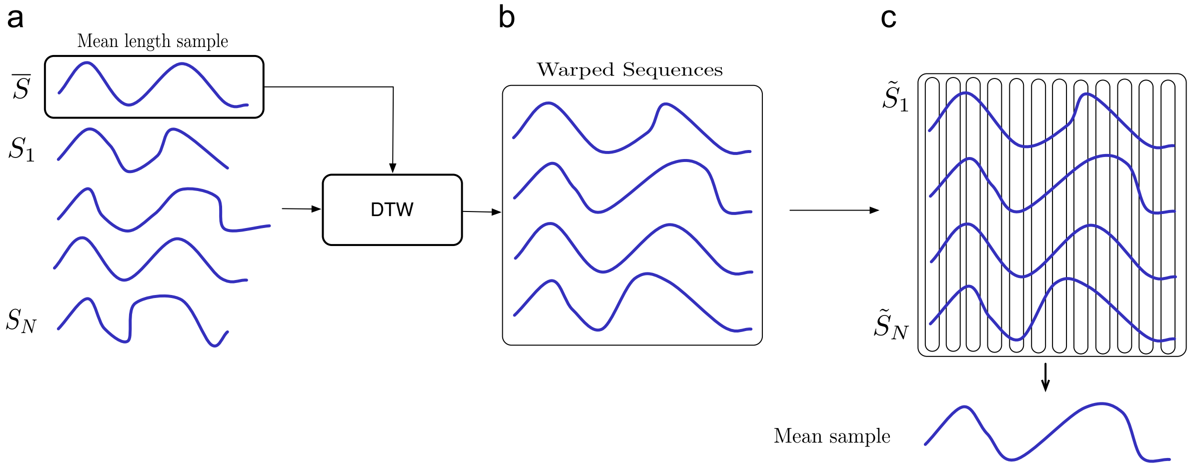


Fig. 8. (a) Action samples and selected median length sample. (b) Aligned samples with same length . (c) Computation of the mean sample.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

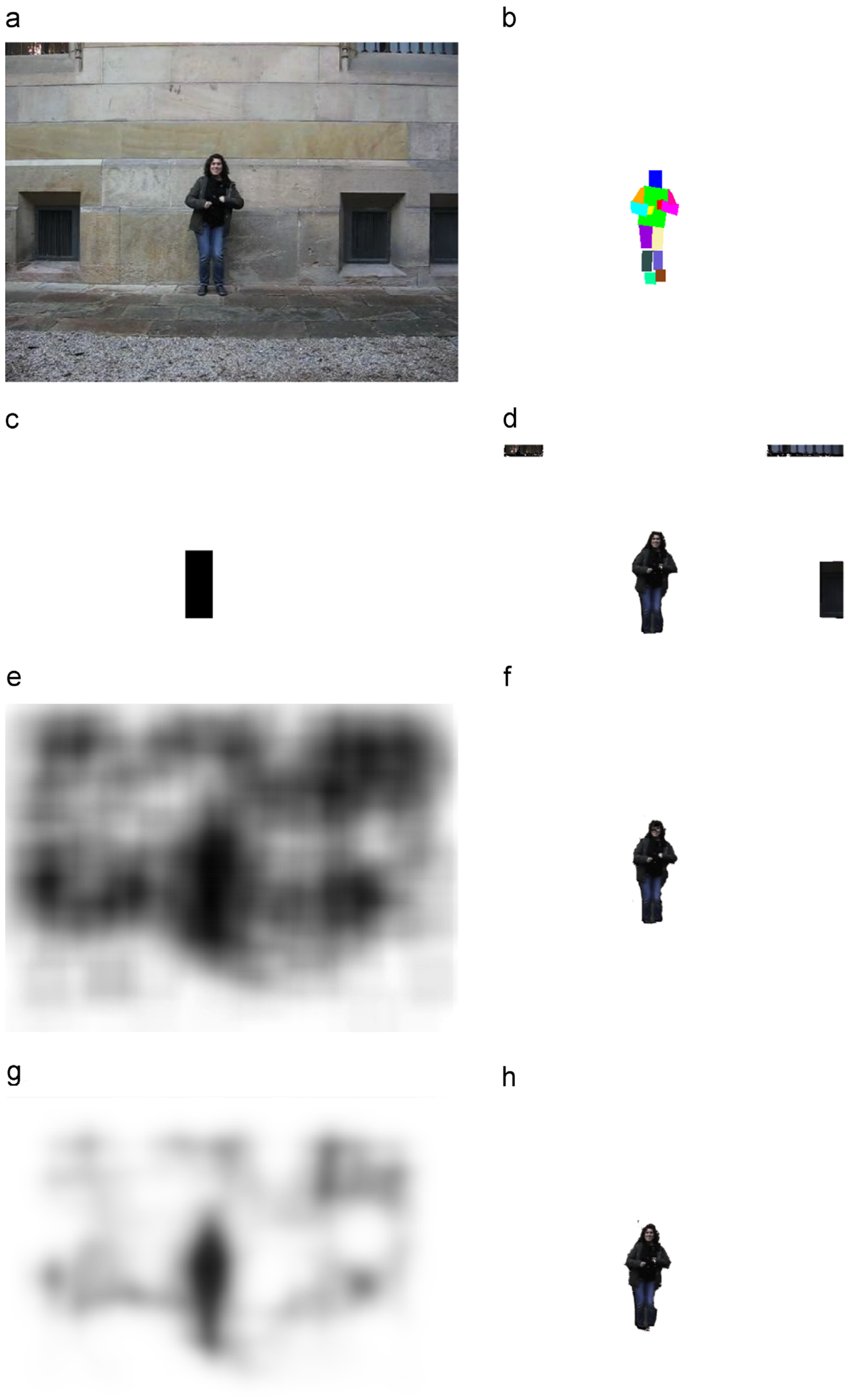


Fig. 9. (a) Original RGB image. (b) Multi-limb ground truth. (c) Probability map obtained by the Person Detector method. (d) Person/background segmentation of the Person Detector+GbCut approach. (e) Probability map yielded by the cascade class method. (f) Person/background segmentation of the cascade class method. (g) Probability map obtained from the ECOC method. (h) RGB segmentation obtained by the ECOC+GbCut approach.

combination of labels:

$$\mathbf{V}(\mathbf{c},\mathbf{z}) = \gamma \sum_{\{m,q\} \in \mathcal{N}} \Omega(c_q, c_m) \exp(-\beta \| z_m - z_q \|^2). \tag{7}$$

In concrete, in order to introduce prior costs between different labels, $\Omega(c_q, c_m)$ must fulfill some constraints related to spatial coherence between the different labels, taking into account the natural constraints of the human limbs (i.e. head must be closer to torso than legs and arms are nearer to forearms than head). In particular, we experimentally fixed the penalization function Ω as defined in Table 3.

4. Experimental results

In order to present the experimental results, we first discuss the data, experimental settings, methods and validation protocol.

4.1. Data

We use the proposed *HuPBA8k+* dataset described in Section 2. We reduced the number of limbs from the 14 available in the dataset to 6, grouping those that are similar by symmetry (right–left) such as arms, forearms, thighs and legs. Thus, the set of limbs of our problem is composed of *head*, *torso*, *forearms*, *arms*, *thighs* and *legs*. Although labeled within the dataset, we did not include hands and feet in our multi-limb segmentation scheme. Finally, in order to train the limb classifiers, ground truth masks are used to normalize all limb regions per dominant orientation, and both Haar-like features and HOG descriptors are computed based on the aspect ratio of each region, making the descriptions scale invariant.

4.2. Methods and experimental settings

In this section we introduce the different methods compared for *binary segmentation*, *multi-limb segmentation* and *action/gesture recognition* tasks. In addition, the experimental settings for these methods are explained.

4.2.1. Binary segmentation methods

As the first stage of our approach computes a binary person/background segmentation, we compare in this step the following methods:

- *P.Detector+GbCut*: The well-known Person Detector of [38] followed by Grab-Cut segmentation.
- *C.Class+GbCut*: The cascade of classifiers proposed by Viola and Jones [43], training one cascade of classifiers per limb and Grab-Cut segmentation.
- *ECOC+GbCut*: The proposed ECOC tree-structure body part classifier and automatic Grab-Cut segmentation for person/background segmentation.

4.2.2. Multi-limb segmentation methods

To evaluate the performance of our proposal for multi-limb segmentation, we compare our strategy with two state-of-the-art methods for multi-limb segmentation:

- **FMP:** This method was proposed by Yang and Ramanan [8,9] and it is based on Flexible Mixtures-of-Parts (FMP). We compute the average of each set of mixtures for each limb and for each pyramid level in order to obtain the probability maps for each limb category. In order to compute the probability map of the background category, we subtract 1 with the maximum probability $\in [0,1]$ of the set of limbs detection at pixel location.
- **IPP:** This method is proposed by Ramanan [7] and it is based on an Iterative Parsing Process (IPP). We use it to extract the limb-like probability maps followed by α-β swap graph-cut multi-limb segmentation. The background category is computed as shown in FMP method.
- **ECOC+Graph-Cut:** Our proposed human limb segmentation scheme shown in Fig. 5.

4.2.3. Action/gesture recognition methods

In the case of the action recognition task our goal is to provide a firm baseline of the recognition of the 11 actions categories labeled within the *HuPBA8K+* dataset. In order to do it, we compare performance of the following standard methodologies:

Table 4
Mean overlapping and standard deviation.

P.Detector+GbCut	C.Class+GbCut	ECOC+GbCut
49.60 ± 20.45	58.26 ± 17.31	**61.79 ± 14.02**

- *Dynamic time warping using a random sample*: We use the standard DTW algorithm to recognize the different actions categories in the dataset [13]. In order to compute the cost matrix for each of the gesture/action classes we choose a sample of that category at random.
- *Dynamic time warping using the mean sample*: Following the trend in [15], in order to compute the cost matrix we form a mean sample of each one of the action classes. That is, we choose the sample of each category and align all samples with it. Then, once all samples from the same class are aligned (they have the same length) we compute the mean, an example is shown in Fig. 8.
- *Hidden Markov model*: We use the standard discrete HMM framework [16]. Each HMM, was trained using the Baum–Welch algorithm, and 3 states were experimentally set for the every action category, using a vocabulary of 50 symbols computed using K-means over the training data features. Final recognition is performed with temporal sliding windows of different wide sizes, based on the training samples length variability.

The computation of the feature vector for training and testing the action recognition approaches is based on the segmentation results of our approach. Given the multi-segmentation of limbs, we computed the feature vector of a frame as the concatenation of the 6 limb-like probability maps, resizing each one of them to a 40 × 20 pixels region and vectorizing that region. We obtain a final vector of $d=40\cdot 20\cdot 6=4800$ dimensions, which is then reduced to $d=150$ dimensions using the Random Projection algorithm [44].

4.2.4. Experimental settings

In a preprocessing step, we resized all limb sample to a 32 × 32 pixels region for computational purposes. Then, we used the standard Cascade of Classifiers based on AdaBoost and Haar-like features [43], and we forced a 0.99 false positive rate and a maximum of 0.4 false alarm rate during 8 stages. We used the OpenCV implementation[3] to learn the cascading classifiers. On the detection step, we use a sliding window approach on a RGB test image I, each patch is decoded in the ECOC framework. Then, we consider all body-parts as a single body category. Thus, a body-like probability map $P\in[0,1]^{l\times w}$, where l and w are the length and the width of I respectively, is built. This map contains, at each position P_{ij}, the proportion of body part detections for each pixel over the total number of detections for the whole image. The sliding window has an initial patch size of 32 × 32 pixels up to 60 × 60 pixels and a geometric factor of 1.1 and 2 pixels separation between patches. Additionally, the ECOC framework for the two stages is used with the ECOClib.[4]. As a final part of the first stage, binary GRAB Cuts[3] were applied to obtain the binary segmentation, where the initialization values of foreground and background must be chosen. Thus, a set of value tags is assigned for background and foreground pixels: obvious background (GC_BGD), possible background (GC_PR_BGD) and possible foreground (GC_PR_FGD). These value tags

[3] OpenCV library: http://opencv.willowgarage.com

[4] ECOClib: http://sourceforge.net/p/ecoclib/code/HEAD/tree/

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

are tuned via cross-validation for different ranges. Empirically, we perform a grid-search for the following intervals: GC_BGD=[0, 0.5], GC_PR_BGD=[0.5, 0.8] and GC_PR_FGD=[0.8, 1.0] with a stepping of 0.07. For the second stage, we set the following parameters for the HOG descriptor[3]: 32 × 32 window size, 16 × 16 block size, 8 × 8 block stride, 8 × 8 cell size and 8 for number of bins. Then, we trained SVMs[5] with a Generalized Gaussian RBF kernel based on Chi-squared distance. The parameters C and γ were tuned via cross-validation in a grid-search where C values were obtained in a linear sampling in the range [0, 10^4] and γ values where obtained in a logarithm sampling in the range (0, 1]. Finally, the model selection step was done via a leave-one-sequence-out cross-validation. We obtain limb-like probability maps as in first stage but particularly considering the different established body parts instead of a map that includes all of them. Each map P^c contains, at each position P^c_{ij}, the probability of pixel at the entry (i,j) of belonging to the body part class c, where $c \in \{1, 2, \ldots, 6\}$. This probability is computed as the proportion of detections at point (i,j) over all detection for class c. For multi-limb segmentation we used the alpha-beta Graph-Cut implementation,[6] where we set a 8 × 8 neighboring grid, 7 labels (6 body-parts + background) and tuned the λ parameter in range [1, 10^3] with a stepping of 10.

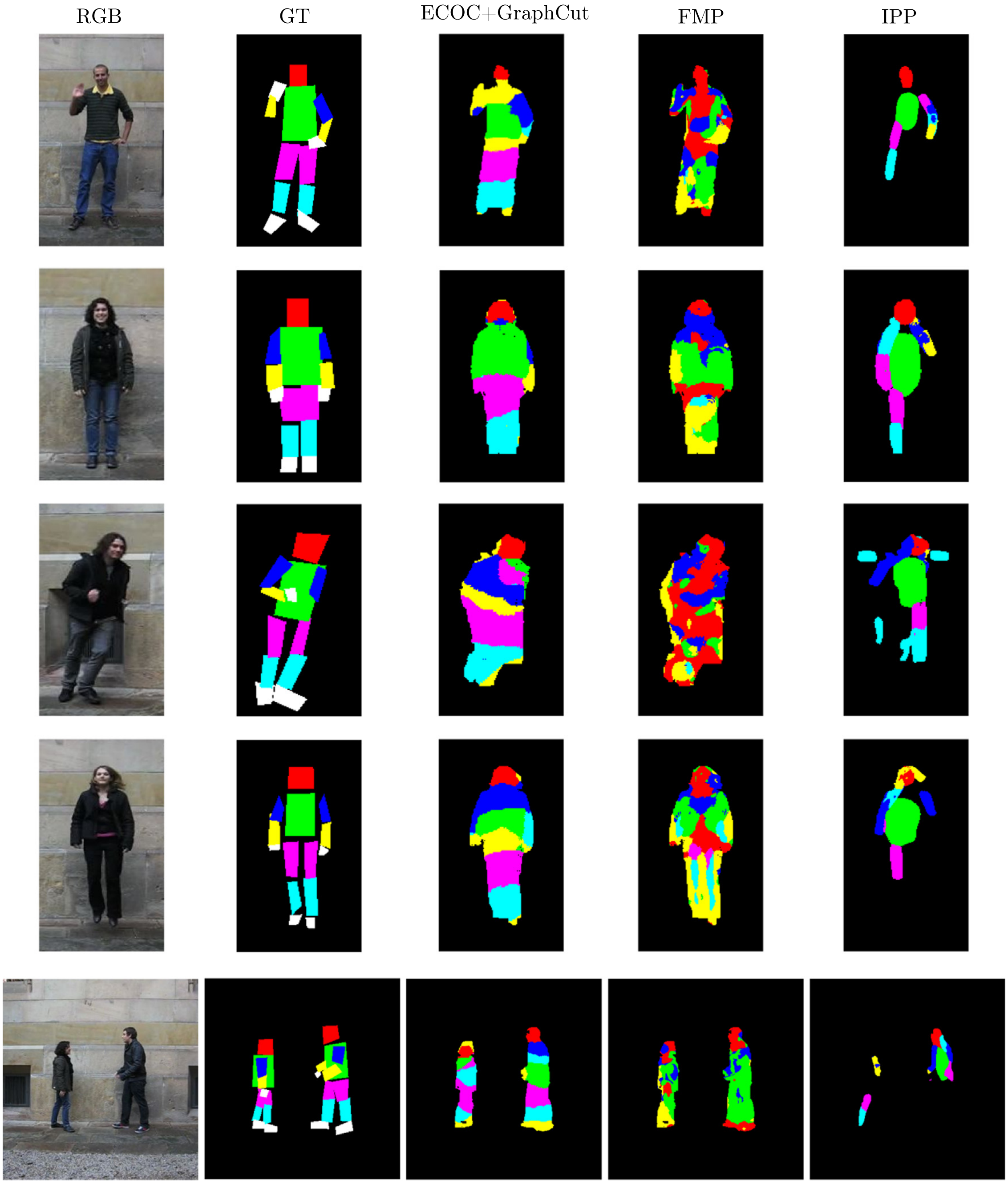


Fig. 10. Multi-limb segmentation results for the three methods, for each sample, we also show the RGB image and the ground-truth (GT).

[5] LIBSVM: http://www.csie.ntu.edu.tw/ cjlin/libsvm/.

[6] gco-3.0: http://vision.csd.uwo.ca/code/.

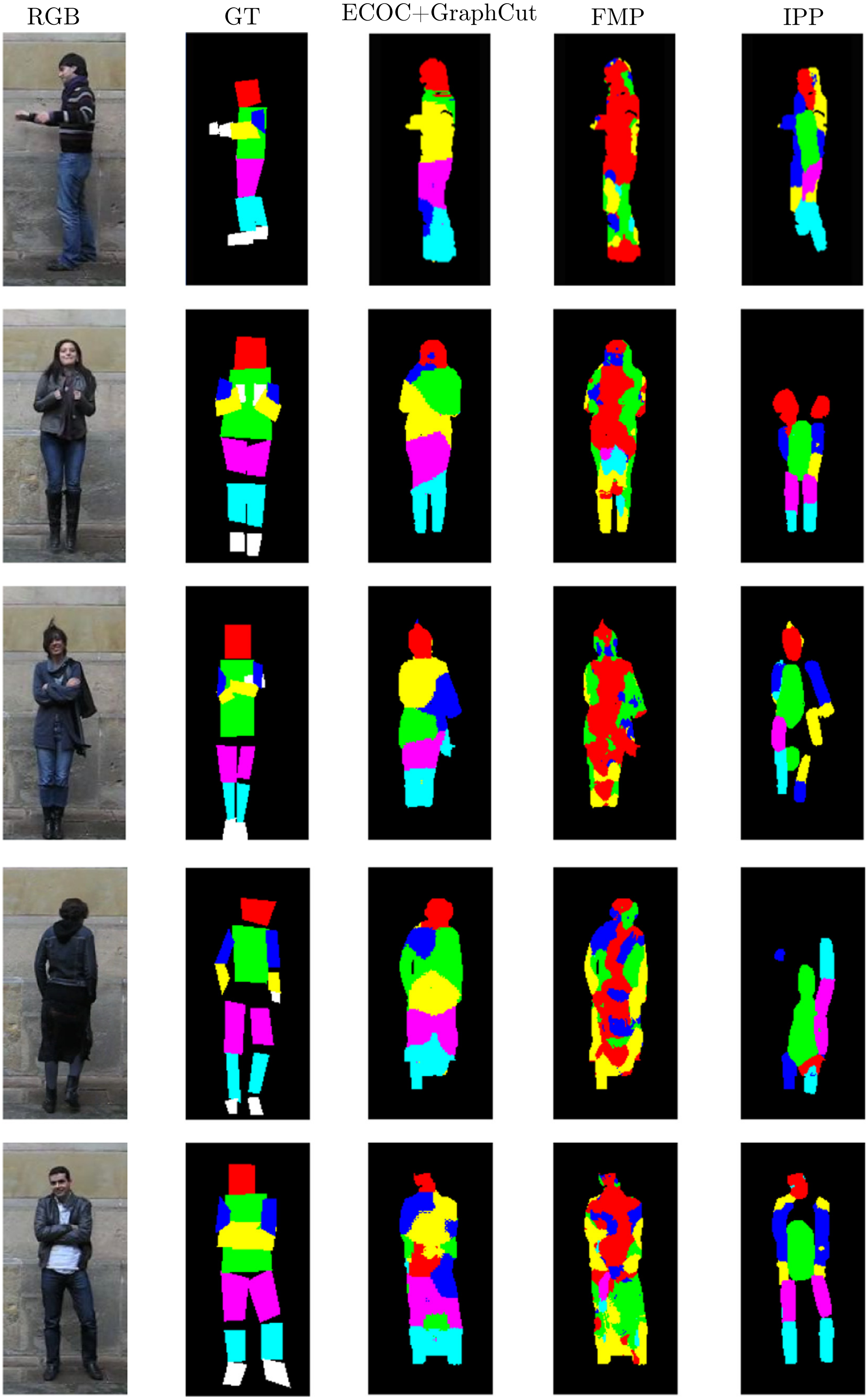


Fig. 11. Multi-limb segmentation results for the three methods, for each sample, we also show the RGB image and the ground-truth (GT).

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

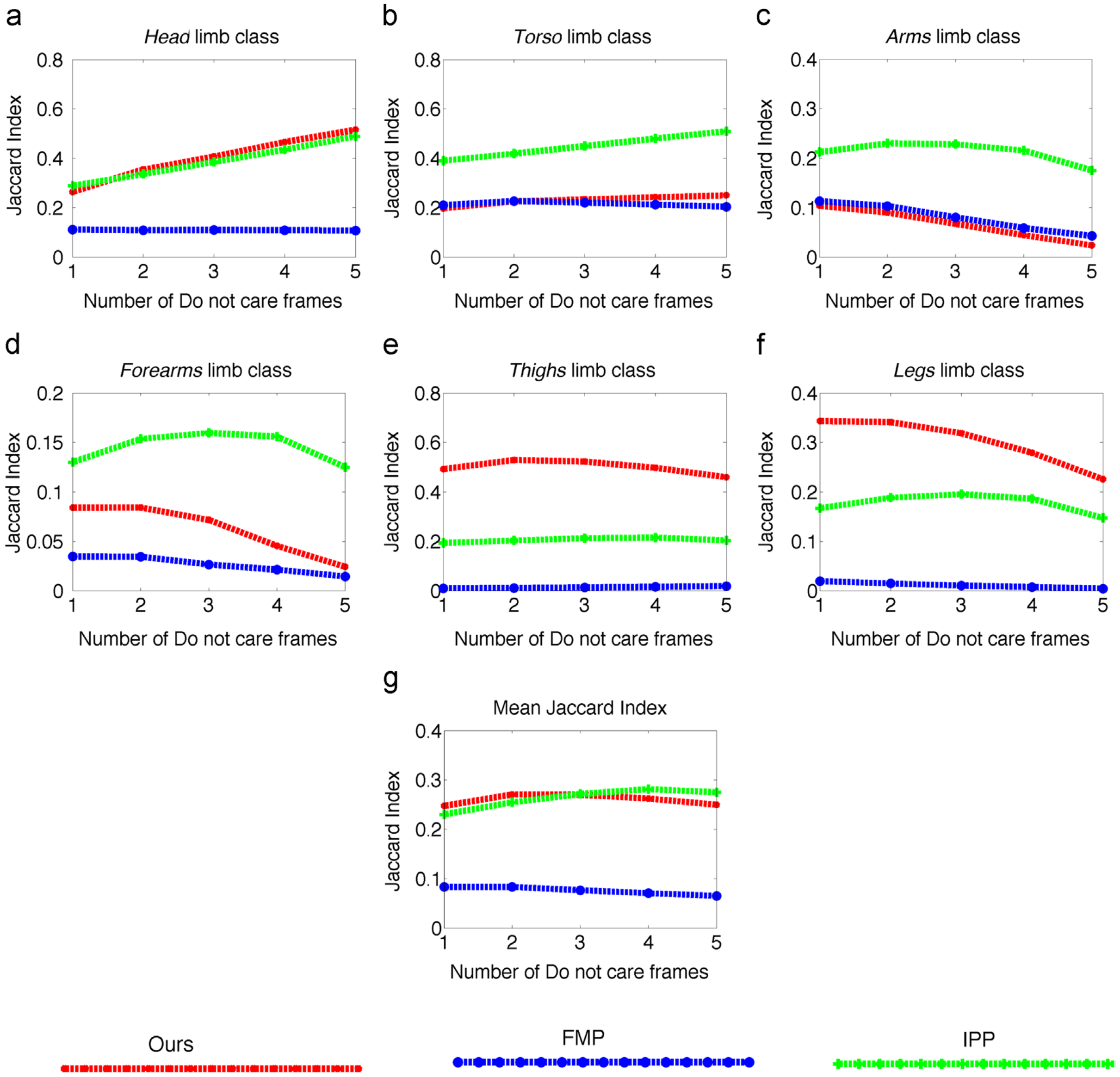


Fig. 12. Jaccard Indexes for the different limb categories from (a) to (f). (g) Mean Jaccard Index among all limb categories.

For the action recognition experiments the cost-threshold and the action/gesture model for both DTW experiments was obtained by cross-validation on training data, using a leave-one-sequence-out procedure. For HMM method, each HMM and its corresponding probability-threshold were obtained by cross-validation on training data, using a leave-one-sequence-out procedure.

4.3. Validation measurement

In order to evaluate the results for the three different tasks: binary segmentation, multi-label segmentation and gesture/action recognition, we use the Jaccard Index of overlapping ($J = A \bigcap B / A \bigcup B$) where A is the ground-truth and B is the corresponding prediction.

4.4. Experimental results

In this section we show results for the three different tasks: *binary segmentation*, *multi-label segmentation* and *action/gesture recognition*.

4.4.1. Binary segmentation results

In Fig. 9 we can see an example of the person/background segmentation obtained by the compared methodologies. In particular, we can see in Fig. 9(d) how the segmentation obtained by the Person Detector+GbCut method yields a poor result, segmenting dark regions of the image. Furthermore, when comparing Fig. 9(e) and (f), the improvement in the body-like probability map obtained by the ECOC+GbCut approach over the cascade class+GbCut method is clearly significant.

In order to evaluate the performance of the compared methodologies, Table 4 shows the mean overlapping obtained on the whole dataset together with the standard deviation. From the results one can see that the ECOC+GbCut method outperforms the compared methodologies at least by 5%. This improvement is the effect of two causes. The former is the Error-Correcting capabilities of the ECOC framework. The latter is the tree-structure definition of the coding matrix, which allows base classifiers to obtain accurate results.

4.4.2. Multi-limb segmentation results

For the Multi-limb segmentation task, we show in Figs. 10 and 11 qualitative results for some samples of the *HuPBA8k+* dataset. When comparing the qualitative results we can see how the FMP method [8,9] performs worse than its counter parts. In addition, one can see how IPP and our method obtain similar results in most cases. However, the IPP lacks a good person/background segmentation.

Furthermore, we provide quantitative results in terms of the Jaccard Index. In Fig. 12 we show the overlapping performance obtained by the different methods, where each plot shows the overlapping for a certain limb. In addition, we use a 'Do not care' value which provides a more flexible interpretation of the results. Consider the ground truth of a certain limb category in an image as a binary image, which pixels take value 1 when those pixels are labeled to belong to such limb. Then, the 'Do not care' value is defined as the number of pixels which are ignored at the limits of each one of the ground truth instances. Thus, by using this approach we can compensate the pessimistic overlap metric in situations when the detection is shifted some pixels. In this sense, we analyze the overlapping performance as a function of a 'Do not care' value that ranges from 0 to 4.

When analyzing quantitative results, we see how our method outperforms the compared methodologies for some limb categories. In particular, for the *head* region both methods obtain

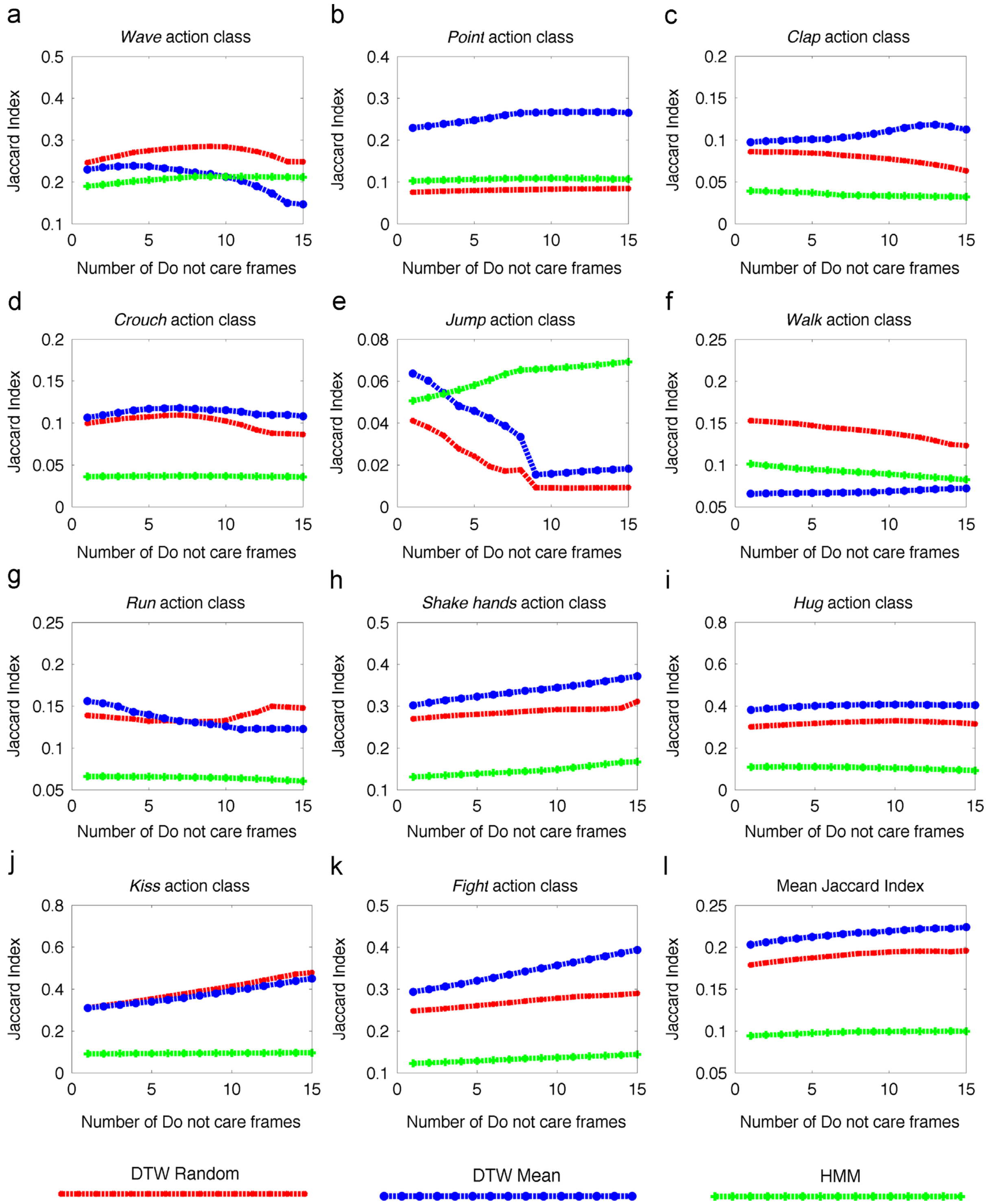


Fig. 13. Jaccard indexes for the different action categories from (a) to (k). (l) Mean Jaccard Index among all action categories.

similar results, which is intuitive since the method used to detect the head is the well known face detector. Finally, we see how FMP method is in almost all cases obtaining the worst performance. As shown in Fig. 12(g), for the mean overlapping considering all the segmented limbs our method outperforms the rest of the approaches up to 3 pixels of “Do not care” evaluation.

4.4.3. Action recognition results

In this section we show the quantitative results obtained by the different gesture recognition methods in terms of the Jaccard Index. Furthermore, to allow a deeper analysis of the proposed methodologies, in our evaluations we use a ‘Do not care’ value which provides a more flexible interpretation of the results. Consider the ground truth of a certain action category in a video sequence as a binary vector, which activates when a sample of such a category is observed in the sequence. Then, the ‘Do not care’ value is defined as the number of bits (frames) which are ignored at the limits of each one of the ground truth instances. Thus, by using this approach we can compensate the pessimistic overlap metric in situations when the detection is shifted some frames. The Jaccard Index as a function of the ‘Do not care’ value for the 11 action categories and the mean Jaccard Index among action categories are shown in Fig. 13.

When analyzing quantitative results we see how the DTW Mean methods outperforms for most action categories the standard DTW Random and HMM methods. In addition, when computing the mean Jaccard Index among all gesture categories the DTW Mean approach also ranks first, obtaining a mean Jaccard Index of 0.20. This good result is due to the use of information from all action samples which encodes the intra-class variability of the gesture categories. Finally, we can see how in all cases Hidden Markov Model achieves the lowest performance.

5. Conclusions

In this work, we introduced the *HuPBA8K+* dataset, which represents the largest available multi-limb dataset on RGB data up to date, with more than 120 000 manually labeled limb regions. In addition, we proposed a novel two-stage method for human multi-limb segmentation in RGB images. In the first stage, we perform a person/background segmentation by training a set of body parts using cascades of classifiers embedded in an ECOC framework. In the second stage, to obtain a multi-limb segmentation we applied multi-label Graph-Cuts to a set of limb-like probability maps obtained from a problem-dependent ECOC scheme.

We compared our proposal with state-of-the-art pose-recovery approaches on the novel dataset, obtaining very satisfying results in terms of both person/background and multi-limb segmentation steps. For completeness, the novel dataset was also labeled with different human actions drawn from an 11 gesture/action dictionary, including isolate and collaborative behaviors. In this sense, we also provided action recognition baseline results on the novel dataset considering DTW and HMM strategies.

Acknowledgments

The work of Miguel Ángel Bautista is supported by SUR-DEC of the Generalitat de Catalunya and FSE under the FI scholarship and research project TIN2013-43478-P.

References

[1] L. Bourdev, J. Malik, Poselets: Body part detectors trained using 3d human pose annotations, in: ICCV, IEEE, US, pp. 1365–1372, 2009.

[2] V. Vineet, J. Warrell, L. Ladicky, P. Torr, Human instance segmentation from video using detector-based conditional random fields, in: BMVC, 2011.

[3] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vis. 61 (2005) 55–79.

[4] M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: People detection and articulated pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, US, pp. 1014–1021.

[5] B. Sapp, C. Jordan, B. Taskar, Adaptive pose priors for pictorial structures, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, US, pp. 422–429.

[6] D. Ramanan, D. Forsyth, A. Zisserman, Tracking people by learning their appearance, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 65–81.

[7] D. Ramanan, Learning to parse images of articulated bodies, in: Advances in neural information processing systems, pp. 1129–1136.

[8] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, US, pp. 1385–1392.

[9] Yi Yang, Deva Ramanan, Articulated human detection with flexible mixtures of parts, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2878–2890.

[10] A. Hernández-Vela, N. Zlateva, A. Marinov, M. Reyes, P. Radeva, D. Dimov, S. Escalera, Graph cuts optimization for multi-limb human segmentation in depth maps, in: CVPR, pp. 726–732.

[11] C. Rother, V. Kolmogorov, A. Blake, “grabcut”: interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (2004) 309–314.

[12] F. Zhou, F. De la Torre, J.K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 35 (3) (2013) 582–596.

[13] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, in: Readings in Speech Recognition, 1990, pp. 159–165, ISBN:1-55860-124-4.

[14] M. Reyes, G. Dominguez, S. Escalera, Featureweighting in dynamic timewarping for gesture recognition in depth data, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, pp. 1182–1188.

[15] A. Hernández-Vela, M. Á. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol, C. Angulo, Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d, Pattern Recognit. Lett. (2013), http://dx.doi.org/10.1016/j.patrec.2013.09.009.

[16] T. Starner, A. Pentland, Real-time american sign language recognition from video using hidden Markov models, in: Motion-Based Recognition, Springer, US, 1997, pp. 227–243.

[17] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, H. J. Escalante, Multi-modal gesture recognition challenge 2013: Dataset and results, ChaLearn multi-modal gesture recognition grand challenge and workshop, in: 15th ACM International Conference on Multimodal Interaction, 2013, pp. 445–452.

[18] S. Escalera, J. Gonzalez, X. Baro, M. Reyes, I. Guyon, V. Athitsos, H. J. Escalante, L. Sigal, A. Argyros, C. Sminchisescu, R. Bowden, S. Sclaroff, ChaLearn multi-modal gesture recognition 2013: grand challenge and workshop summary, ChaLearn multi-modal gesture recognition grand challenge and workshop, in: 15th ACM International Conference on Multimodal Interaction, 2013b, pp. 365–368.

[19] V. Ferrari, M. Marin-Jimenez, A. Zisserman, Progressive search space reduction for human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, US, 2008.

[20] D. Tran, D. Forsyth, Improved human parsing with a full relational model, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 227–240.

[21] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2010) 303–338.

[22] M.A.B.S.E. Daniel Sanchez, Juan Carlos Ortega, Human body segmentation with multi-limb error-correcting output codes detection and graph cuts optimization, in: Proceedings of InPRIA, 2013, pp. 50–58.

[23] S. Johnson, M. Everingham, Clustered pose and nonlinear appearance models for human pose estimation, in: Proceedings of the British Machine Vision Conference http://dx.doi.org/10.5244/C.24.12, 2010.

[24] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, IEEE, US, pp. 1–8.

[25] F. De la Torre, J. K. Hodgins, J. Montano, S. Valcarcel, Detailed Human Data Acquisition of Kitchen Activities: the CMU-Multimodal Activity Database (CMU-MMAC), Technical Report, RI-TR-08-22h, CMU, 2008.

[26] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 3, IEEE, US, pp. 32–36.

[27] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, Human pose estimation: New benchmark and state of the art analysis, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, 2014.

[28] B. Sapp, B. Taskar, Modec: Multimodal decomposable models for human pose estimation, in: Proceedings CVPR, 2013.

[29] L. Bourdev, J. Malik, Poselets: Body part detectors trained using 3d human pose annotations, in: International Conference on Computer Vision, 2009.

[30] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: EuroCOLT, 1995, pp. 23–37.

[31] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, in: Computer Vision-ECCV 2004, Springer, US, 2004, pp. 69–82.

Please cite this article as: D. Sánchez, et al., *HuPBA8k+*: Dataset and ECOC-Graph-Cut based segmentation of human limbs, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.07.069

[32] Q. Zhu, M.-C. Yeh, K.-T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, vol. 2, IEEE, pp. 1491–1498.
[33] M. Enzweiler, D.M. Gavrila, Monocular pedestrian detection: Survey and experiments, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 2179–2195.
[34] Y.-T. Chen, C.-S. Chen, Fast human detection using a novel boosted cascading structure with meta stages, IEEE Trans. Image Process. 17 (2008) 1452–1464.
[35] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 120–134.
[36] M.A. Bautista, S. Escalera, X. Baró, P. Radeva, J. Vitriá, O. Pujol, Minimal design of error-correcting output codes, Pattern Recognit. Lett. 33 (2012) 693–702.
[37] M.Á. Bautista, S. Escalera, X. Baró, O. Pujol, On the design of an ECOC-compliant genetic algorithm, Pattern Recognit. 47 (2014) 865–884.
[38] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, vol. 1, 2005, pp. 886 –893.
[39] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, US, pp. 1794–1801.
[40] Y. Boykov, G. Funka-Lea, Graph cuts and efficient nd image segmentation, Int. J. Comput. Vis. 70 (2006) 109–131.
[41] Y. Boykov, V. Kolmogorov, Computing geodesics and minimal surfaces via graph cuts, in: CVPR, 2003, pp. 26–33.
[42] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 1222–1239.
[43] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: CVPR, 2001, vol. 1.
[44] E. Bingham, H. Mannila, Random projection in dimensionality reduction: applications to image and text data, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, ACM, New York, NY, USA, 2001, pp. 245–250.

Daniel Sánchez received his Bachelor degree in Computer Science at Universitat de Barcelona (UB), in 2012. He obtained his master degree in Artificial Intelligence at UPC (Universitat Politécnica de Catalunya), UB and URV, in 2014. He is currently doing his Ph.D. at University of Barcelona in computer vision approaches applied to human pose recovery and behavior analysis.

Miguel Ángel Bautista received both his B.Sc and M.Sc in 2010 at University of Barcelona and Polytechnic University of Barcelona, respectively. He is mainly interested in Machine Learning, Pattern Recognition and Learning Theory, as a tool to process huge quantities of data. Currently, his Ph.D. thesis is oriented to construct a sound theoretical framework to treat multi-class and multi-label problems in a efficient manner taking into account the idiosyncrasies of such problems.

Sergio Escalera obtained the P.h.D. degree on Multi-class visual categorization systems at Computer Vision Center, UAB. He obtained the 2008 best Thesis award on Computer Science at Universitat Autónoma de Barcelona. He leads the Human Pose Recovery and Behavior Analysis Group at UAB, CVC, and the Barcelona Graduate School of Mathematics. He is a lecturer of the Department of Applied Mathematics and Analysis, Universitat de Barcelona. He is a partial time professor at Universitat Oberta de Catalunya. He is a member of the Perceptual Computing Group and a consolidated research group of Catalonia. He is also a member of the Computer Vision Center at Campus UAB and International Foundation of Research & Analysis. He is an Editor-in-Chief of American Journal of Intelligent Systems and editorial board member of more than 5 international journals. He is an advisor and director of ChaLearn Challenges in Machine Learning. He is the co-founder of PhysicalTech company. He is an active member of the Cluster de Salud Mental de Cataluña. He is also a member of the AERFAI Spanish Association on Pattern Recognition and ACIA Catalan Association of Artificial Intelligence. He has different patents and registered models. He has published more than 150 research papers and organized scientific events, including CCIA2004, CCIA2014, and workshops at ICCV2011, ICMI2013, ECCV2014. His research interests include, between others, statistical pattern recognition, visual object recognition, and HCI systems, with special interest in human pose recovery and behavior analysis.