Web-Shaped Model for Head Pose Estimation: an Approach for Best Exemplar Selection

Paola Barra, Silvio Barra, *Member, IEEE,* Carmen Bisogni, Maria De Marsico, *Senior, IEEE,* and Michele Nappi, *Senior, IEEE,*

Published in: IEEE Transactions on Image Processing journal.
© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
The Version of Record is available online at: http://dx.doi.org/10.1109/TIP.2020.2984373

***Abstract*—Head pose estimation is a sensitive topic in video surveillance/smart ambient scenarios since head rotations can hide/distort discriminative features of the face. Face recognition would often tackle the problem of video frames where subjects appear in poses making it quite impossible. In this respect, the selection of the frames with the best face orientation can allow triggering recognition only on these, therefore decreasing the possibility of errors. This paper proposes a novel approach to head pose estimation for smart cities and video surveillance scenarios, aiming at this goal. The method relies on a cascade of two models: the first one predicts the positions of 68 well-known face landmarks; the second one applies a web-shaped model over the detected landmarks, to associate each of them to a specific face sector. The method can work on detected faces at a reasonable distance and with a resolution that is supported by several present devices. Results of experiments executed over some classical pose estimation benchmarks, namely Point '04, Biwi, and AFLW datasets show good performance in terms of both pose estimation and computing time. Further results refer to noisy images that are typical of the addressed settings. Finally, examples demonstrate the selection of the best frames from videos captured in video surveillance conditions.**

***Index Terms*—Head Pose Estimation, Smart Cities Applications, Web-Shaped Model, Head Pose Exemplar Selection**

I. Introduction

In the last decade the need for reliable tools and technologies for identity, resources and data protection has quickly increased. Along a different line, smart ambient proactivity calls for context awareness, which includes the identification of users, objects, activities, and conditions in the environment. Biometric recognition is one of the leading branches in both scenarios. It focuses on the recognition of subjects from either physical or behavioural characteristics. The first studies were carried out in controlled conditions, e.g., faces were captured with uniform illumination, frontal pose, and neutral expression. However, to target real-world applications, the research has to face ever more challenging issues, due to the increasingly demanding settings. In these challenging applications, the detection and recognition of a specific trait are usually affected by critical factors, like uneven illumination, natural and/or artificial occlusions or self occlusions, and subject pose or expression [1], [2]. These factors play an especially relevant role when dealing with unattended acquisition (no expert operator guides the subject).

Silvio Barra is with the Department of Mathematics and Computer Sciences, University of Cagliari, 09124, Cagliari, ITALY

Paola Barra, Carmen Bisogni and Michele Nappi are with the Department of Computer Sciences, University of Salerno, 84084 Fisciano (SA), ITALY

Maria De Marsico is with the Department of Computer Sciences, Sapienza University of Rome, 00198, Rome, ITALY

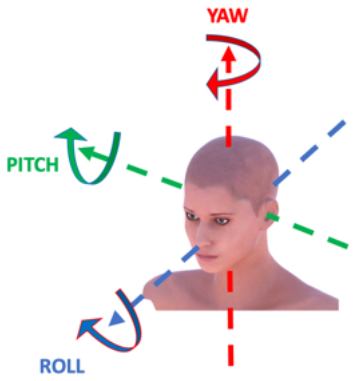

Fig. 1. The image shows the rotations of the head; the pitch, the yaw and the roll are related to the x, y and z axis respectively.

A relevant example is found in video surveillance [3], [4], involving either partially or even totally unaware subjects. More recently, smartphone/mobile applications [5], [6] are exploited by users that may have neither specific knowledge of the operations of a biometric recognition system nor technical competence to evaluate the quality of an acquired sample. These factors especially influence the processing and feature extraction steps for traits like face, ears, periocular region, and iris. In particular, the face is one of the traits which tend to be altered by illumination, pose, and expression (PIE). Low quality of images can also strongly affect the appearance of the trait. Occlusions by scarfs or sunglasses and self-occlusion of part of the face due to non-frontal head pose are a further source of problems [7]. These conditions can complicate the recognition right from the correct detection of the face, up to make it fail.

Amongst the mentioned factors, the pose is probably one of the most challenging to tackle, either regarding the entire body of a subject [8], [9] or the head alone [10]. Partial face self-occlusion due to the pose can complicate further processing operations [11]. Also, the possible rotations of the head around the x, y, and z-axis (pitch, yaw and roll, respectively), shown in Figure 1 [12], [13], cause intrinsic deformations in the geometrical relations among face elements which are quite hard to adjust [14]. Besides face recognition-based systems, also those exploiting the periocular region or the ear can suffer from changes in perspective.

The proposal presented herein entails a brand new method for a rapid estimation of head pose, expressed by the pitch, yaw, and roll rotation degrees. These are inferred by the positions of 68 relevant face landmarks. These are first de-

tected. In a second step, the method exploits the mapping of their positions in a kind of polar space. The model used for the facial landmark prediction is taken from [15]. It was chosen thanks to its performance and was used as-is. The main contributions of this work are:

- a fast method for quite highly accurate estimation of the pose of the head using the pitch, yaw and roll degrees;
- an approach for selecting the video frame/s in a sequence where the head is in its best pose for biometric trait detection (frontal images for face and periocular region, profile and semi-profile images for ear region).

The main advantages offered are:

- the method is robust against the dimension of the image since it exploits the face of the subject as a reference space for applying the model;
- the landmark mapping method does not need heavy and time-consuming training phases and relies on the results of the predictor that is already trained and does not need a novel training when changing the benchmark.

The rest of the paper is organised as follows: Section 2 summarises the state of the art in the head pose estimation; Section 3 presents the proposed approach; Section 4 shows the experimental results along with the performance obtained both in terms of computing time and in terms of accuracy; finally, Section 5 concludes the paper.

II. Related Work

It is quite hard trying to organise under a single taxonomy the myriad of approaches proposed for head pose estimation (hereafter HPE), especially due to hybrid approaches, which may lead to ambiguous classifications. The interested reader can refer to the survey in [16]. Some aspects that can characterise a single HPE method are listed below.

- **Basic reference features**: Tracking methods; Detector methods; Geometric methods; Appearance based methods; ...
- **Application Domain**: Videosurveillance; Advertisement; Best biometric sample selection; Medical applications;...
- **Kind of multimedia used for the pose estimation**: Still image; Consecutive video frames; Videos; Head sensors; ...
- **Use of camera system needed and FOV (field of view) exploited**: Near-FOV; Far-FOV; Master-Slave(s) System; Distributed System; ...

It is reasonable to assume that the kind of reference features is a main element to consider. In a nutshell, tracking methods (e.g. [17]) exploit the differences between consecutive frames (and therefore mostly exploit video) to infer the pose. Detector methods (e.g. [18]) rather train a set of classifiers to recognise relevant poses. Geometric methods (e.g. [19]) identify typical anatomical elements inside the face (generally eyes, nose, mouth) and then use their geometrical relations to determine the pose. Appearance-based methods (e.g. [20]) rely on a set of exemplars (prototypical poses) that are matched against the candidate image of a face, whose pose one wants to estimate, to find out the most similar one. The highest matching score identifies the pose related to the input sample. The proposal in this paper belongs to this last group. Figure 2 shows an example, in which the aim is to estimate the degrees of pitch, yaw, and roll of the Audrey Hepburn's pose. Once the information related to the pose is extracted, this is compared with each pose available in the exemplar set of images (the left part of Figure 2 shows a composition of the poses of the subject number 4 of the dataset Pointing'04 [21]). The label of the most similar pose is returned as output.

One of the most recent approaches in the group of appearance-based methods is described in [23]. The paper presents two different Convolutional Neural Networks (CNNs), for estimating the head pose and the full-body pose respectively. The proposal in [24] deals with a full-body estimation approach, consisting of three modules: 1) the first module uses the HOG method to extract the features related to the appearance of the person; 2) the second module updates a motion information classifier, according to the movement and the direction of the person; 3) the third module estimates the orientation of the body, by fusing the information gathered by the previous modules. In [25], the authors analyse the orientation of the nose to evaluate the orientation of the head, proving that this information has high discriminative power in pose classification. In [26] the authors use a SOM network to estimate the head pose jointly with the shape of the heat to build a new soft biometrics, suitable for being used *"in-the-wild"*. The head pose estimation approach in [27] uses video scene information to evaluate the orientation of the head using the direction of movement of the subject.

It is also possible to sketch a classification of appearance based-methods according to the possible use of CNNs. The methods listed below are also those that were considered for comparison in Section IV.

The use of CNNs has been widely investigated for head pose estimation, given that they are proven to be a perfect tool for processing multidimensional data, like images; in this direction, the works in [28] and [29] propose two interesting approaches, respectively named Multi-Loss ResNet50 and Hyperface. Specifically, the former uses ResNet50 for predicting the intrinsic Euler angles (Yaw, Pitch, and Roll) of faces, directly from the image, whereas the latter exploits a CNN for simultaneously detecting the face, localising the landmarks and estimating the pose. In [30], 3DDFA (3D Dense Face Alignment) is proposed, which fits and aligns a 3D face model to the 2D image through a CNN. Other NN based interesting approaches are KEPLER [31], which addresses the

Fig. 2. In the appearance-based methods, the pose of a face image is estimated by finding the most similar pose in a prototypical set of manually annotated images. The images in this figure are taken from the AFLW and Pointing'04 datasets, described in [22] and in [21] respectively.

face alignment problem by an H-CNN Regressor, and the tool presented in [32] which proposes a neural network to estimate the horizontal and vertical alignment of head orientation, starting from facial images. In [33], FAN is presented, which converts the 2D landmark annotation of many datasets to 3D, leading to the creation of LS3D-W, a dataset containing 3D information from about 230.000 images. In [34], the head pose is estimated by the use of a quad-tree based model; from this, a sparse vector is obtained, describing the pose feature vector. Also, hGLLiM [35] uses a mixture of linear regressions with a partially-latent output which map the feature vectors from many bounding boxes of faces to pitch, yaw, and roll to make them suitable for prediction in presence of unobservable phenomena. Similarly, Probabilistic HDR [36] maps the HOG descriptors from face bounding boxes to the corresponding head poses. The work by Gourier et al. [37] projects the normalized face region onto a set of small size images; an auto-associative memory that exploits the Widrow-Hoff correction rule is then trained. The classification of the head pose is obtained by comparing the small size images with the one reconstructed by the trained memory. A score based policy is then used to select the final head pose. The proposal in [38] exploits a multi-regression loss function, an L2 regression loss combined with an ordinal regression loss, to train a CNN able to estimate head poses from RGB images with no depth information. Among the most recent proposals using Neural Networks, [39] does not predict head poses through landmark or depth estimation but is rather based on regression and feature aggregation. Differently from other existing feature aggregation methods, it does not treat the input as a bag of features, therefore ignoring their spatial relationship in a feature map. The method rather learns a fine-grained structure mapping for spatially grouping features before aggregation. Another very recent work is presented in [40]. The deep neural network follows the Coarse-to-Fine strategy to estimate head poses. The scheme includes two subnetworks that are trained jointly: the Coarse classification phase classifies the input image into four categories and is created starting from the first 21 blocks of GoogleNet; afterwards a Fine Regression phase estimating the accurate pose parameters.

The method proposed and described in the remaining of the paper aims at quickly evaluating the orientation of the head, without using any training step. As a consequence, it does not need huge sets of images, like in [41], where a deep network is used for real-time estimation of the head pose. The challenge behind the deep learning-based approaches is that the images used for the training phase need to be properly annotated. This is a hard and time-consuming process when dealing with head pose estimation. Unless using accurate electronic sensors, it is quite hard to obtain a correct triple (*pitch, yaw, roll*) which reflects the actual pose of the subject. As a matter of fact, the accuracy of manual annotation cannot exceed a certain level, since a difference of 5° or less is hardly perceivable. On the contrary, in the proposed appearance-based method the training step is not necessary since the process is reduced to a $1:N$ matching, where N is the number of different poses contained in a specific prototypical set.

III. Spider-Web based Approach for Head Pose Estimation

The approach in this paper estimates the pose of a subject's head from a feature vector that is obtained by applying a cascade of two different models to the input face image:

- the first model is in charge of predicting the positions of 68 well-defined landmarks over the face, described by their coordinates within the image itself;
- the second model represents a novel proposal and is a spider-web shaped one; it is in charge of determining the specific model-relative locations of the identified landmarks on the face; these locations are established by a number of concentric circles and their sectors; the procedure assigns each landmark to a specific sector and uses the overall obtained information to build a feature vector to infer the head pose; the model is characterised by parameters whose values have been investigated during the experimental phase.

Once the pose feature vector has been obtained, it is compared with a set of prototypical vectors extracted in the same way from exemplars corresponding to known poses to determine the closets one.

The model used for the facial landmark prediction is described in [15]. We shortly summarise it for the sake of readers. The predictor takes as input a face image, and outputs the positions of 68 facial landmarks P_i expressed as pairs of Cartesian coordinates (x_i, y_i) with $i = 1, 2, ..., 68$. The coordinates correspond to the pixel locations of the landmarks. Point detection relies on an ensemble of regression trees. The model training exploits a set of faces that are manually annotated with the x and y coordinates of the landmarks, and with the probability of distances between single pairs of points. The obtained model predicts the locations of the points belonging to the following salient regions:

- the jawline of the face (the first 17 points)
- the eyebrows (from the 18th to the 27th point)
- the nose (from the 28th to the 36th point)
- the eyes (from the 37th to the 48th point)
- the mouth (from the 49th to the 68th point)

On the one end, the present proposal does not entail training the predictor again. The available model was chosen thanks to its robustness and works whatever the dataset used for the tests or the configuration of the model for the second step. On the other hand, the second step does not require any training at all. Therefore the overall approach can be used without any training/re-training when changing the benchmark data.

In the second step, the center and radius of the second model are determined according to the landmarks that have been identified over the face. The spider web-shaped model is centered on the nose tip (point number 33 from the previous model) and sized according to the measures of the face, therefore making the approach image size-independent. Being $O = (x_{33}, y_{33})$ the center of the model, and $P_j = (x_j, y_j)$, $j = 1, \ldots, 68$ except $j = 33$, one of the other landmarks, the radius r of the model is equal to the Euclidean distance d between O and the farthest landmark, i.e., $r = d(O, P_i)$ where $i = arg\max_j d(O, P_j)$. Figure 3 depicts a generic

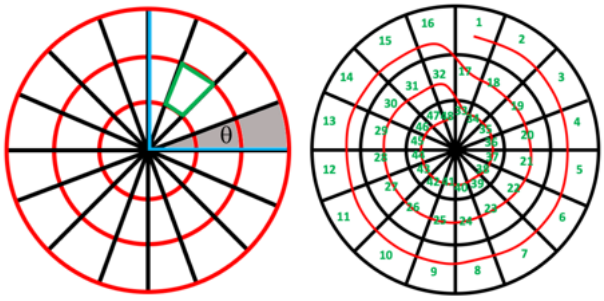

Fig. 3. Left: the web-shaped model applied on the face and centered on the landmark P_33; the radius of the web may change according to the farthest landmark from the center; red lines identify the *circles*, blue lines a *quarter*, the gray region is a *slice* of width , and green contour identifies a *sector*. Right: the order in which the sectors are analysed to build the pose feature vector, starting from the outer circle of the model.

model together with the variable parameters of the web, that are tested in the presented experiments: concentric *circles* appear in red, they delimit annuli(or *rings*), and are numbered starting from the outer one; a *quarter* of the model whole circle appears in blue in the figure, the number of quarters is fixed to 4 in the model, and they are numbered in clockwise order starting from the positive-positive one; *slices* divide quarters in a number of equal parts of width , their contours appear in black in the figure, where one slice is highlighted in grey, and they are numbered in clockwise order starting from the first slice in the first quadrant; *sectors* represent portions of slices comprised between two neighbouring concentric circles, i.e., the intersections of rings with slices, they are numbered starting from the intersection of the outer ring with the first slice and proceeding in clockwise order from outer to inner rings, and a sector appears with green contours in the figure. In the following, a ring will be referred to by the number of the outer circle limiting it. The configuration in Figure 3 is coded as $3C\_4S\_inv$, since it is formed by three circles and each *quarter* is divided into four *slices*. The suffix *inv* means that the model *circles* are equidistant from each other. In this configuration, their distance is determined by the radius and by their number. The number of elements in the feature vector returned by the second step is equal to the number of sectors, and the value of each vector element is the number of landmarks that fall in the corresponding sector, according to the order defined in the right part of Figure 3.

Figure 4 shows an example application of the cascade of models. The input image in the figure is taken from the AFLW (Annotated Facial Landmarks in-the-wild) dataset [22]. The middle image in Figure 4 shows the output of landmark prediction. The last image shows the superimposition of the web-shaped model with the same parameters as Figure 3.

Details on landmark/section association and on the construction/matching of the pose vectors are given in the following subsections.

A. Landmark/Sector association

To build a pose feature vector, each landmark needs to be associated with a specific *sector* of the model. This process is carried out by first detecting the *circle*, the *quarter* and then the *slice* to which a specific landmark belongs to.

Let the model be formed by n equidistant *circles* and m *slices* per *quarter* (therefore coded as nC_mS_inv). The radius of the outermost circle is always equal to the radius r of the web and, in this case, going inwards, the radius of each following circle is decreased by r/nC. The extension to circles with different distances is immediate.

Be O the origin of the axes (as determined by landmark P_33) and r_i the radius of the $i-th$ *circle*, for $i = 1, ..., n$, numbered from the outermost to the innermost. The point P belongs to the smallest *circle* C_i containing it (i.e. to the ring limited by C_i and C_{i+1}, if the latter exists). This is easily identified as the smallest circle whose radius is greater than or equal to the distance $d(O, P)$. This computation will appear in Algorithm 1 as `getBelongingCircle(P)`. In the running example of Figure 5, P belongs to C_1, i.e. the outermost *circle*. The *quarter* of the point is simply identified by considering the relative coordinates of the point in the Cartesian plane centered in O. As shown in Figure 5, since P coordinates are both positive it belongs to the first *quarter*. Finally, being α the angle between the Y axis and the $\overline{OP}$ segment

$$\alpha = \widehat{YOP}$$

and being θ the width of each of the m slices in a quarter

$$\theta = 90°/m$$

the slice s which the point P belongs to is computed as

$$s = \lceil \alpha/\theta \rceil$$

This procedure will be referred to in Algorithm 1 as `getBelongingSlice(P)`.

The *circle*, the *quarter* and the *slice* univocally identify the *sector* which each point belongs to. In the example in Figure 5, P belongs to the 1^{st} *circle*, 1^{st} *quarter*, 2^{nd} *slice*, therefore to 2^{nd} *sector* according to the order shown in the right part of Figure 3.

B. Pose Vector Construction

The application of the model allows identifying the sector in which each landmark is located. The feature vector related to the pose is built according to the model defined in Figure 3. The vector dimension is equal to the number of sectors. This is obtained as $m \times 4 \times n$, i.e. the number of *slices* multiplied by the number of quarters (always 4) by the number of *circles*

Fig. 4. An application of the cascade: the first model identifies the face landmarks; the elements of the second one are circles, slices, and their sectors where the procedure places the landmarks to build a feature vector.

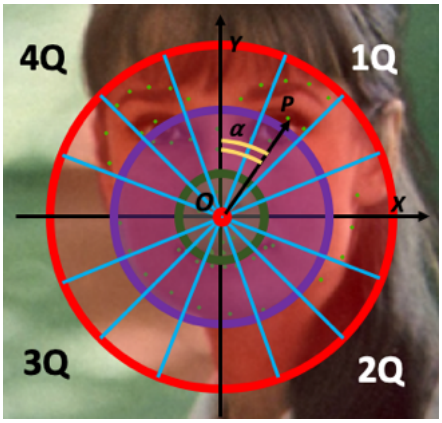

Fig. 5. The figure shows how each landmark is associated to a sector within a model like the one in Figure 3 with three *circles* (the innermost in green, the middle one in purple, and the outermost in red) and four *slices* per *quarter*.

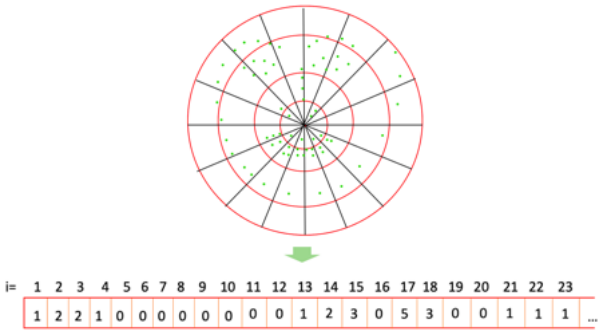

Fig. 6. The figure shows how each landmark is associated to a sector within a model with four *circles* and four *slices* per *quarter*, and the final construction of the pose vector with elements ordered according to the adopted convention.

(in the examples in the figures above, this leads to $4slices \times 4quarters \times 3circles = 48sectors$).

The vector is built according to the following algorithm:

Algorithm 1 Feature Vector Construction

$L \leftarrow$ number of *landmarks* = *68*
circles $\leftarrow$ number of *circles*
slices $\leftarrow$ number of *slices* for each quarter
$V \leftarrow$ empty array
for `j = 1 to` $circles \times 4 \times slices$ **do**
  `V[j] = 0` ▷ Array initialization
for `i = 1 to` L **do**
  ▷ Compute the *circle* which the *i-th* point belongs to
  `c = getBelongingCircle(i)`
  ▷ Compute the *slice* which the *i-th* point belongs to
  `s = getBelongingSlice(i)`
  `sector =` $circles \times slices \times (c-1) + s$
  `V[sector] = V[sector] + 1`
`return V` ▷ Return the feature vector

The obtained feature vector contains in its i-th position the number of landmarks located in the i-th sector (according to the ordering shown in the rightmost image in Figure 3). Figure

C. The pose estimation through the set of prototypes

The estimation of the pose of an input face image relies on the comparison of the pose feature vector extracted from this image using Algorithm 1 with those extracted from prototypical exemplars used as a reference. The returned result is the pose corresponding to the reference vector with the lowest Euclidean distance from the pose vector of the incoming image. For this work, a dataset of reference prototypes (exemplar poses) was built on purpose and called *Lara*. The 3D model of a standard synthetic head was automatically rotated along all the three axes as shown in Figure 7. Variable combinations of pitch, yaw and roll angles were used to obtain prototypical exemplars, from which pose reference vectors were extracted according to Algorithm 1. According to the proposed approach, these vectors were used in the experiments to classify the pose of an incoming image. Of course, it is possible to use a different set of reference poses/exemplars, given that they undergo the procedure in Algorithm 1 to extract the reference feature vectors.

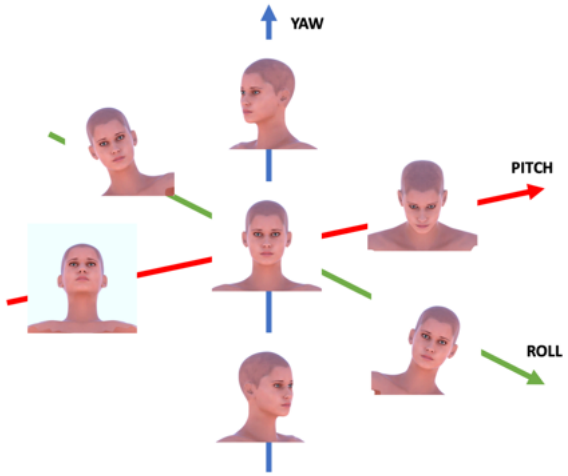

Fig. 7. Variations on pitch, yaw, and roll in Lara Dataset.

IV. Experimental Section

A. Datasets

To test the method, the exemplars of *Lara* were used as a reference to classify the pose of an input image.

The dataset used to create the input images for the experiments is the *Biwi Kinect Head Pose Database* [42] which contains over 15K images of 20 people (6 females and 14 males - 4 people were recorded twice). For each frame in the captured sequences, both the RGB-image and the depth image are provided. Besides this, *Biwi* also provides an *.obj* file per subject, which contains the 3D model of the head of the subject. Figure 8 shows some samples from the dataset. Using the provided data, the center of mass of the head was computed for each model and the head was adjusted manually so to place it in a frontal pose (0°of pitch, yaw, and roll). Then, the obtained head model was automatically modified to obtain the set of poses planned for the experiments. In particular,

for each 3D model the following pose rotations have been obtained:

- pitch: from -30°to +30 °
- yaw: from -45°to +45 °
- roll: from -20°to +20 °

A discretization step of 5°was adopted, so to obtain 13 variations in pitch, 19 variations in yaw and 9 variations in roll (41 kinds of variation in total per model). The total amount of head poses extracted from the *Biwi* dataset sums up to 2233 per model/subject, counting all the possible combinations of pitch, yaw and roll rotations, for a total of 44660. The maximum angular error for each of the rotations is equal to 5°since it is strictly related to the discretization step mentioned above. The head pose of each model was extracted in the described way and was mapped back onto a 2D synthetic face image, still presenting the texture features of the original subject. This procedure allowed creating a new benchmark with an even distribution of the desired head poses, where each image is annotated with the pitch, yaw, and roll parameters used to create it, to use them as clear ground truth for the experiments. The protocol used for experiments on different model configurations entails using these images as input to the pose recognition system, while the feature vectors extracted from the models in the Lara dataset are used as reference exemplars to match with.

To compare the proposed approach with the state of the art, further experiments were carried out with AFLW dataset [22] and Pointing'04 dataset [21].

Fig. 8. Samples of the Biwi Kinect Head Pose Database.

B. Results with different model configurations

To test different model configurations, a parametrization phase has been adopted before its application. The parameters involved in the definition of the model configuration determine the dimensions of the sectors the model is divided into along with their number. This number is also the size of the vector returned by the model application. The values of the elements of a pose vector are computed according to Algorithm 1 and are significantly affected by the sector size and position. In fact, a denser partition of the web space (greater number of circles and slices) produces larger vectors of generally lower values. Moreover, the change in sector density can also affect their positions beside their size. For instance, the second sector will start and end earlier in a denser partition than in a less dense one, besides being thinner. Therefore the landmark distribution in sector 2 in the two cases can differ not only for the number of landmarks but also for their labels (see Figure 9 and Figure 10 below). As a consequence, the same pair of poses can produce significantly different distances when using different model configurations. This is the rationale for testing different model configurations to identify the best performing one(s). The parameters are defined below:

- number of circles C;
- number of slices per quarters S;
- distance between two circles.

Table I reports the tested sets of parameters when considering equidistant circles and Figure 9 shows the behaviour of the different configurations.

TABLE I
THE MODEL CONFIGURATIONS IN WHICH THE CIRCLES ARE EQUIDISTANT FROM EACH OTHER. FOR EACH CONFIGURATION, THE PARAMETERS ARE LISTED TOGETHER WITH THE CORRESPONDING CODES, USED TO IDENTIFY A CONFIGURATION IN THE EXPERIMENTAL RESULTS, WITH #C THE NUMBER OF CIRCLES, #S THE NUMBER OF SLICES, AND VDIM THE OBTAINED VECTOR DIMENSION.

code	#c	#s	#radius of each circle (from the innermost to the outermost)	vdim
3C_4S_inv	3	4	R/3; 2/3*R; R;	48
4C_3S_inv	4	3	R/4; R/2; 3/4*R; R	48
4C_4S_inv	4	4	R/4; R/2; 3/4*R; R	64
5C_3S_inv	5	3	R/5; 2/5R; 3/5*R; 4/5*R; R	60
5C_4S_inv	5	4	R/5; 2/5R; 3/5*R; 4/5*R; R	80

The set of tests with fixed circle radii is aimed at assessing which of the parameter settings better fits with the discrimination of the pose, by keeping unaltered the distance between each pair of consecutive circles.

Fig. 9. The behaviour of configurations whose distance between each pair of consecutive circles is fixed. These configurations are built according to the parameters in Table I

The second set of model configurations has been built by slightly modifying the distances between the circles. The number of circles as well as of slices has been kept fixed (4 *circles* and 4 *slices* for each *quarter*) for all the configurations, but the distance between two consecutive circles changes along the radius. These configurations are detailed in Table II and their behaviour is shown in Figure 10.

The purpose of experiments with variable circle radii is to discover which landmarks are most significant for pose recognition and whether such significant landmarks are more densely contained in a specific sector.

TABLE II
THE MODEL CONFIGURATIONS IN WHICH THE CIRCLES HAVE VARYING DISTANCES FROM EACH OTHER. FOR EACH CONFIGURATION, THE PARAMETERS ARE LISTED TOGETHER WITH THE CORRESPONDING CODES, USED TO IDENTIFY THEM IN THE EXPERIMENTAL RESULTS, WITH #C THE NUMBER OF CIRCLES, #S THE NUMBER OF SLICES, AND VDIM THE OBTAINED VECTOR DIMENSION.

code	#c	#s	#radius of each circle (from the innermost to the outermost)	vdim
4C_4S_var1	4	4	1/15*R; 3/15*R; 7/15*R; R	64
4C_4S_var2	4	4	8/15*R; 12/15*R; 14/15*R; R	64
4C_4S_var3	4	4	1/10*R; 3/10*R; 6/10*R; R	64
4C_4S_var4	4	4	4/10*R; 7/10*R; 9/10*R; R	64

Fig. 10. The behaviour of model configurations with circles of variable radius. These configurations are built according to the parameters in Table II

Table III reports the average errors when the method is executed over the images synthesised from the entire set of 3D models in *Biwi* dataset. In all experiments, the face landmark predictor is applied first on the images, so to obtain once and for all the positions of the 68 landmarks over the face of the subject. Hence, each of the proposed spider-web model configurations is superimposed in turn over the cloud of points, and the pose is estimated from the feature vectors extracted according to Algorithm 1.

The results in Table III show that the highest estimation errors happen with the pitch rotations, whereas yaw and roll rotations mostly stay below the threshold of 5°, as given by the discretization step. In particular, it is immediate to deduce that the variations in pitch are harder to estimate than the variations in yaw. Taking a closer look at the reasons why this happens, this is because the landmark predictor outputs sets of points that are very similar to each other if applied on faces whose main variation is due to the pitch rotation, as shown in Figure 11. This does not happen in cases in which the main variations are either on yaw or roll. The obvious consequence is that there is a higher error in estimating the pitch rotation.

Table IV reports the percentage of tests, for each configuration, where the errors affecting pitch, yaw, and roll evaluation are all under a common threshold, starting from 5°. It is worth noticing that this means that all angles are identified with comparable accuracy and is therefore different from having the average error under the same threshold.

TABLE III
THE TABLE REPORTS, FOR EACH OF THE TESTED CONFIGURATIONS, THE MEAN ABSOLUTE ERROR (MAE) EXPRESSED IN DEGREES FOR PITCH, YAW, AND ROLL VARIATIONS.

configuration code	*Pitch MAE*	*Yaw MAE*	*Roll MAE*
3C_4S_inv	11,5525°	4,8488°	3,3405°
4C_3S_inv	11,5783°	**4,6517°**	**3,3080°**
4C_4S_inv	11,5683°	4,6871°	4,6871°
5C_3S_inv	11,5327°	4,8428°	3,5553°
5C_4S_inv	**11,4034°**	4,8405°	3,4001°
4C_4S_var1	12,0856°	5,1850°	3,3845°
4C_4S_var2	10,8884°	4,6533°	**3,1997°**
4C_4S_var3	11,8988°	5,0982°	3,3865°
4C_4S_var4	**10,7139°**	**4,6357°**	3,2309°

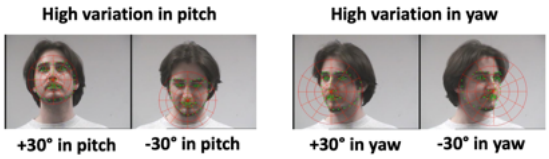

Fig. 11. The images on the left vary only in pitch (+30°on the left and -30°on the right), whereas those on the right vary on their yaw rotations (+30°on the left and -30°on the right)

Interestingly the best results are generally achieved by models with 4 circles, and with variable circle distance (last two rows). On the other hand, results obtained by models with 5 circles present a discontinuous trend, as well as the only reported one with 3 circles (first row). This seems to testify that the distribution of sectors obtained by 4 concentric circles/rings better captures the layout of relevant landmarks.

C. Comparison with the state of the art

The ranking of the web configurations observed over Biwi in the previous subsection is confirmed for the other datasets with no significant difference. Therefore, for sake of space, the following results only take into account the best one, namely *4C_4S_var4*. The experiments in this section compare the proposed approach against the main proposals in the state of the art, among those presented in Section II. How discussed in Section II, these approaches can be divided into two classes: 1) model-based methods and 2) neural network-based methods. There are pros and cons in adopting one strategy rather than the other. For instance, on one side model-based approaches do not require any training phase, resulting in a quick-to-test method building. On the other side, neural network-based methods result more reliable against the misdetection of face landmarks, which is an easy-to-fail task for model-based approaches. The experiments presented here aim at comparing the proposed method with works from both kinds of approaches. Moreover, we also report the results achieved

TABLE IV
PERCENTAGE OF TESTS ACHIEVING PITCH, YAW, AND ROLL ERROR UNDER A COMMON THRESHOLD, IN TERMS OF DEGREES

configuration	<5°	<10°	<15°	<20°	<25°	<30°	<35°
3C_4S_inv	22.5058	53.7807	79.7722	93.0287	97.9964	99.3276	99.6844
4C_3S_inv	21.408	53.52	78.0294	91.1075	96.5967	98.6689	99.4236
4C_4S_inv	21.5315	53.1769	78.3313	91.9857	97.4338	98.8747	99.5334
5C_3S_inv	19.5005	54.0003	80.4035	92.5347	96.9946	98.4905	99.2041
5C_4S_inv	21.1198	54.165	80.7191	93.2757	97.7494	98.9982	99.4511
4C_4S_var1	23.4527	52.6554	77.3981	91.1349	97.0495	99.0668	99.753
4C_4S_var2	24.084	57.1977	81.419	93.358	97.2005	98.614	99.259
4C_4S_var3	21.3463	52.1387	77.5353	91.8348	**97.6259**	**99.2178**	**99.7393**
4C_4S_var4	**24.976**	**59.0092**	**83.2716**	**94.1128**	97.4887	98.6963	99.3687

through the application of some general methods like Support Vector Regression (SVR) [43], Gaussian processes - GPR [44], and Partial Least Square Regression - PLS [45], and with an approach based on 3D face morphing with depth parameters proposed by Kong et al. [46].

Not all the compared methods exploit all the datasets taken into account here for performance evaluation Tables V, VI and VII respectively show the comparison of results obtained on AFLW dataset [22], Pointing'04 dataset [21] and BiWi dataset [42]. The comparative results are in terms of MAE (Mean Absolute Error) for yaw, pitch, and roll degrees and MAE for all the Euler angles. When not available in the referenced papers, the reported total MAE is the average of the separate mean absolute errors achieved for yaw, pitch, and roll.

Table V compares the results of the presented approach with other ones tested on the AFLW Dataset. The cited works in the table exploit neural networks. On this dataset, our approach, though not always overcoming the others' results, still provides an MAE value lower than the other methods except for QuatNet.

TABLE V
MEAN ABSOLUTE ERROR OF PITCH, YAW, AND ROLL ANGLES (EXPRESSED IN DEGREES) ACROSS DIFFERENT METHODS OVER THE AFLW DATASET [22].

Method	Yaw	Pitch	Roll	MAE
Multi-Loss ResNet50 [28]	6.470	6.559	5.436	6.155
Hyperface [29]	7.61	6.13	3.92	5.89
KEPLER [31]	6.45	5.85	8.75	7.01
3DDFA [30]	5.400	8.530	8.250	7.393
FAN [33]	6.358	12.277	8.714	9.116
QT_PYR [34]	7.6	7.6	7.17	7.45
QuatNet [38] *	3.933	**4.316**	2.590	3.613
Our Approach - 4C_4S_var4	**3.11**	4.82	**2.25**	**3.39**

Table VI shows the results obtained over Pointing'04 dataset [21]. In this case, the proposed approach actually competes with the best among the others. In fact, it is to further take into account that almost all the other approaches achieve the estimation of the head pose over a manually annotated bounding box, whereas our method also achieves automatic face detection, and automatically determines a bounding box before exploiting the prediction of the face landmark positions. Pointing'04 dataset does not contain Roll information, therefore none of the approaches has evaluated performance against it. We reported a partial MAE anyway, to highlight the good overall performance of the proposed approach.

TABLE VI
MEAN ABSOLUTE ERROR OF PITCH, YAW, AND ROLL ANGLES (EXPRESSED IN DEGREES) ACROSS DIFFERENT METHODS OVER THE POINTING'04 DATASET [21]. ALL THE APPROACHES MARKED WITH (*) USE A MANUAL ANNOTATED BOUNDING BOX FOR THE FACE DIFFERENTLY FROM OUR METHOD WHICH EXPLOITS AN AUTOMATIC DETECTION PROCEDURE.

Method	Yaw	Pitch	MAE
Stiefelhagen [32] *	9.7	9.5	9.6
Gourier et al. [37] *	12.1	7.3	9.7
SVR [43] *	12.82	11.25	12.035
hGLLiM [35] *	**7.93**	8.47	**8.2**
Probabilistic HDR [36]	8.70	8.85	8.775
Kong et al. [46]	10.98	9.71	10.345
Our Approach - 4C_4S_var4	10.63	**6.34**	8.485

It is possible to appreciate a similar situation for the results computed over the BiWi dataset [42] and reported in Table VII. Taking into account that many approaches use manually annotated bounding boxes, the results achieved by the proposed approach favourably compete with the others. Some exceptions are represented by approaches based on deep networks that sometimes obtain better results in some columns, and by QuatNet, that the proposed approach overcomes for pitch estimation. Also in this case, the overall MAE is similar to the best methods using CNNs.

TABLE VII
MEAN ABSOLUTE ERROR OF PITCH, YAW, AND ROLL ANGLES (EXPRESSED IN DEGREES) ACROSS DIFFERENT METHODS OVER THE BIWI DATASET [42]. ALL THE APPROACHES MARKED WITH (*) USE A MANUALLY ANNOTATED BOUNDING BOX FOR THE FACE DIFFERENTLY FROM OUR METHOD WHICH EXPLOITS AN AUTOMATIC DETECTION PROCEDURE.

Method	Yaw	Pitch	Roll	MAE
Multi-Loss ResNet50 [28]	5.17	6.97	3.39	5.177
GPR [44] *	7.72	9.64	6.01	7.79
PLS [45] *	7.35	7.87	6.11	7.11
SVR [43] *	6.98	7.77	5.14	6.63
hGLLiM [35] *	6.06	7.65	5.62	6.44
QT_PYR [34]	5.41	12.80	6.33	8.18
FSA-Net [39]	4.27	4.96	2.76	**3.996**
Coarse-to-Fine [40]	4.76	5.48	4.29	4.84
QuatNet [38]	**4.010**	5.492	**2.936**	4.146
Our Approach - 4C_4S_var4	6.21	**3.95**	4.16	4.77

D. Robustness against distortions, use in video surveillance, and best frame selection

A final set of experiments aimed at evaluating the performance of the proposed pose estimation approach when dealing with noisy images, with videos captured in a video surveillance setting, and as a support for choosing the frames in a video that are best suited for face recognition. The experiments with noisy images used some samples from AFLW dataset, that were modified adding some blur, motion, and noise. As for the use in video surveillance, the experiments relied on GOTCHA [47] dataset. GOTCHA collects images of faces taken from the GOTCHA-I dataset, including videos of subjects walking / following a path / climbing the stairs, in either cooperative or non-cooperative manner, either indoor or outdoor, indoor with light, and indoor with a flashlight. Finally, frames from two sample videos are processed to demonstrate the ability to extract those with approximately frontal faces, better suited for recognition. To the best of our knowledge, there is no public dataset available at present to compare the achieved performance with competing methods.

a) Noisy images from AFLW and GOTCHA videos: To just evaluate the effectiveness of the web model, we applied it to images where the landmark predictor provided acceptable results. The choices relating to the filters to be applied to the images were made by evaluating their visual result, to make an image very noisy / blurred / moved but still recognizable. A different set of images was used for each condition, that was obtained by filtering AFLW images.

- AFLW_blur was obtained from the application of a Gaussian filter with standard deviation equal to 7, which simulates the blur.
- AFLW_motion was obtained from the application of a filter that simulates a 9-pixel linear horizontal movement of the camera / video camera.
- AFLW_noise was obtained using a Gaussian filter with mean equal to 0 and variance equal to 0.15, which simulates noise in RGB.

Figure 12 shows some samples from GOTCHA dataset, while Figure 13 shows some distorted images from AFLW dataset.

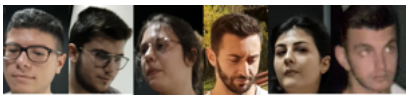

Fig. 12. Samples from GOTCHA dataset.

Without noise, the percentage of images of AFLW where the face was correctly detected and landmarks were identified in a satisfactorily way is about 93%. Of course, the percentages on images affected by noise, blur and motion are (not dramatically) different as each type of distortion affects the data in a different way. There are two things to underline: the first one is that these percentages are in line with the visual quality of the respective images. If for example one considers AFLW_noise, the images have really become very bad, hindering any possible recognition. The second one is that, in real settings, generally this type of ”problems” (blur, noise, motion) are not found on a single image, but on frames deriving from video sequences. As a consequence, following the strategy to attempt recognition only on ”reliable” images, the fact that the quantity of detected faces (and therefore whose pose is estimated) decreases is not a problem, especially aiming at using the best ones for face recognition. The latter does not require recognizing the subject in each and every frame.

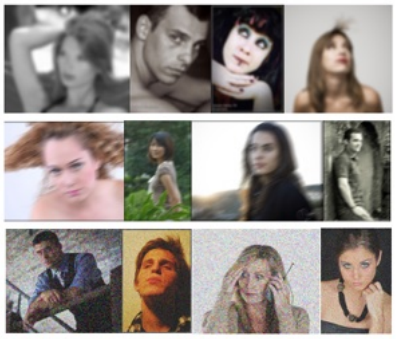

Fig. 13. Distorted samples from AFLW dataset: blur (top), motion (middle), and noise (bottom).

Table VIII shows the results obtained using the best configuration of the web model, namely 4C_4S_var4, that, compared with the other ones, provided consistently better results in almost all situations and on all datasets. Moreover, a further configuration was also tested, namely 4C_5S_var4, that divides each quarter into 5 sectors, using the same distances among circles as 4C_4S_var4 (4/10*R; 7/10; 9/10*R; R). This configuration was not previously compared with other approaches since it provided better results on roll than 4C_4S_var4 (4.0 vs. 5.16 on AFLW and 3.24 vs. 4.16 on Biwi) but an overall worse MAE due to higher mean absolute errors for yaw and pitch. The same happened with GOTCHA. However, it seems to work better for distorted images. Table IV-D0a reports the achieved results

TABLE VIII
MEAN ABSOLUTE ERROR OF PITCH, YAW AND ROLL ANGLES (EXPRESSED IN DEGREES) AND MAE OF TWO WEB CONFIGURATIONS ACROSS DIFFERENT SETS OF IMAGES. 4S STANDS FOR 4C_4S_VAR4, AND 5S STANDS FOR 4C_5S_VAR4

Conf.	Dataset	Yaw	Pitch	Roll	MAE
4S	GOTCHA	11.8	9	7	9.26
4S	AFLW_blur	9.05	8.33	3.81	7.06
4S	AFLW_motion	9.78	7.17	3.91	6.95
4S	AFLW_noise	18.75	5.62	6.87	10.41
5S	GOTCHA	12.17	10.65	6.52	9.78
5S	AFLW_blur	7.14	6.42	4.76	6.11
5S	AFLW_motion	8.26	8.48	3.91	6.88
5S	AFLW_noise	18.75	5.62	5	9.79

It is interesting to notice that the overall MAE with 4C_5S_var4 is generally better on distorted images, even though the performance of this configuration is not uniform across all kinds of rotations.

b) Best frame selection: The approach was tested on videos to assess its ability to suggest the best frame to process

for face recognition, i.e., the one where a subject's face appears in a pose closer to the frontal one than in the rest of the video. Figure 14 and Figure 15 show frames from two example videos, one outdoor and the other one indoor, that demonstrate how the distance and the face size do not influence the accuracy of results. In fact, while in the first case the best frame corresponds with the closest one, in the second case this is not true because the best pose was found for a face at a larger distance. As described in Section III, the web model is resized to fit the face so that the distribution of landmarks inside the sectors is not affected. The figures further allow to notice that the quality of the best face (with respect to pose) detected in both cases is also good enough to justify this choice, since it is the best suited for a recognition attempt. It is worth reminding that, in videosurveillance, the aim is not to determine the exact pose of faces in each frame, but rather to select those allowing a reliable face recognition.

Fig. 14. Frames from video 1. The last frame in the figure reproduces the one chosen by the procedure using the proposed pose estimation. The same frame (the 15-th one) is highlighted in the sequence by a light rectangle.

Fig. 15. Frames from video 2. The last frame in the figure reproduces the one chosen by the procedure using the proposed pose estimation. The same frame (the 9-th one) is highlighted in the sequence by a light rectangle.

E. Computing times

Since the proposed method implies the application of two different models, the computing time is reported in a way that allows appreciating the computational demand of the individual steps:

- the first important element is the total computing time, starting from the application of the face landmarks predictor to the phase in which the head pose is estimated;
- the second element is the partial computing time that only refers to the application of the spider-web model, which is the actual innovation provided by the present work.

Figure 16 reports the total and the partial computing times, defined as above and expressed in seconds, for each adopted configuration of the spider-web model. It is possible to appreciate that the face landmarks predictor consumes the most of the time; as a matter of fact, this time is on average twice the time than the spider-web method. The configuration with three *circles* and 4 *slices* per *quarter* is the fastest one, with only $0,108$ seconds ($0,36$ seconds also considering the application of the landmarks predictor). From the image, it can be noticed that the computing time tends to increase as the number of sectors increases.

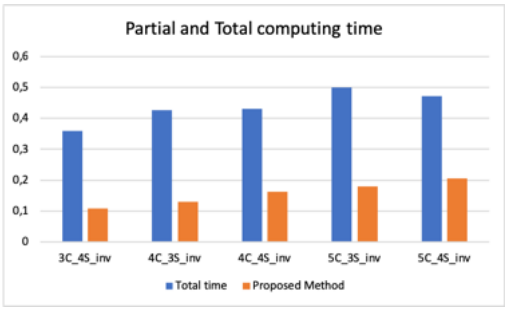

Fig. 16. For each of the configurations with fixed radius, the total (the blue bar) as well the partial (the orange bar) computing times are reported. The different configuration codes are reported on the x-axis, whereas the time (in seconds) is on the y-axis.

A similar trend regarding time/number of sectors cannot be appreciated for configurations with variable radius since in these cases the number of sectors was kept constant (see Table II). However Figure 17, notwithstanding the comparable computing times, highlights a lower demand for *4C_4S_var2* that has the radii of the circles set as 8/15*R, 12/15*R, 14/15*R, and R. This point will deserve further attention in future experiments. Overall, these configurations show slightly better results, if compared to the fixed radius ones, as shown in Tables IV and III.

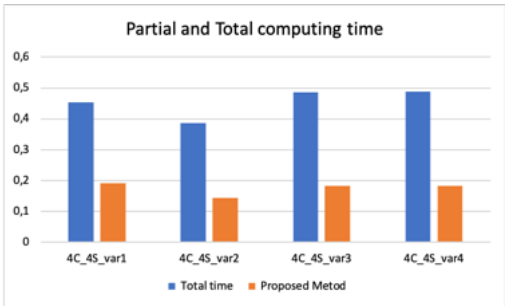

Fig. 17. For each of the configurations with variable radius, the total (the blue bar) as well the partial (the orange bar) computing time are reported. The different configuration codes are reported on the x-axis, whereas the time (in seconds) is on the y-axis.

Unfortunately, it is not possible to carry out a complete comparison of computing times with the methods taken into account in the previous subsection. Being the best-performing ones based on CNNs, a possible interesting parameter is the training time. QuatNet only mentions such time, which is equal to about 4.5 hours, but this also depends on the size of the

training set and on the length of the training in terms, e.g., of epochs. The paper presenting HyperFace-ResNet reports a time of 3 seconds per face, including face detection, landmark detection, pose estimation, but also recognition. 3DDFA is declared to take 63.9ms (15.65fps) for each sample. Some of these computing times are partially achieved using a GPU or extremely performing architectures. On the other hand, the tests presented here were carried out on a MacBook with a single Intel Core m3 processor 1.1 GHz, 8 GB RAM and an Intel HD Graphics 515. Such non-high performing configuration testifies the full viability of the proposed solution even with limited computational resources. Overall, the times achieved by the proposed approach seem to testify its suitability for real-time operation, even when compared with competing methods.

V. Conclusions

In this paper, a novel approach to head pose estimation is proposed. The complete application involves the use of a cascade of two models with different aims: the first one predicts the positions of 68 landmarks over a face image and is adopted from literature; the second one is a web-shaped model that is applied over an image in order to identify the model sector to which each landmark belongs. Different configurations for the spider-web model have been tested, with either concentric circles located at the same distance from each other, or with varying distances between consecutive circles. These tests highlighted that pitch rotation is the hardest angle to estimate, causing the overall results to significantly decrease. Therefore it can be considered as the most "difficult" problem to address for the proposed approach. This may be the reason why configurations with varying inter-circle distance presented slightly better results since they can better catch the variation in the vertical distribution of landmarks that is caused by pitch. This aspect surely deserves more future investigation. A comparison of results has been carried out against many state-of-the-art approaches. In many cases, the proposed approach performs better than many of those which exploit manually annotated bounding boxes or which are based on neural networks. However, the most recent proposals in the literature that exploit CNNs achieve generally better results. On the other hand, it is to consider that these require a huge training phase with a relevant amount of annotated training samples. Therefore, overall, the experimental results testify good performance, both in terms of accuracy of pose estimation and in terms of computing time. Regarding the latter, the obtained measures show really low computational demand, but, unfortunately, few works in literature report details on this aspect that may allow a thorough comparison. Further tests also demonstrate good performance with distorted images and the ability of the method to support the selection of video frames with the most favourable pose (close to the frontal one). Actually, this is its main intended use for real scenarios like video surveillance, smart cities, and mobile biometrics, where pose recognition is not used for image alignment but rather to select the best frames from a video to support an optimal recognition.

References

[1] M. De Marsico, M. Nappi, C. Riccio, and H. Wechsler, "Robust face recognition for uncontrolled pose and illumination changes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 1, pp. 149–163, Jan 2013.

[2] Y. Cho and K. Yoon, "Pamm: Pose-aware multi-shot matching for improving person re-identification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3739–3752, Aug 2018.

[3] J. C. Neves, G. Santos, S. Filipe, E. Grancho, S. Barra, F. Narducci, and H. Proença, "Quis-campi: Extending in the wild biometric recognition to surveillance environments," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Cham: Springer International Publishing, 2015, pp. 59–68.

[4] J. Neves, F. Narducci, S. Barra, and H. Proença, "Biometric recognition in surveillance scenarios: a survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 515–541, Dec 2016.

[5] S. Barra, A. Casanova, F. Narducci, and S. Ricciardi, "Ubiquitous iris recognition by means of mobile devices," *Pattern Recognition Letters*, vol. 57, pp. 66 – 73, 2015, mobile Iris CHallenge Evaluation part I (MICHE I). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865514003286

[6] S. Barra, M. De Marsico, C. Galdi, D. Riccio, and H. Wechsler, "Fame: Face authentication for mobile encounter," in *2013 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications*, Sep. 2013, pp. 1–7.

[7] M. De Marsico and M. Nappi, *Face recognition in adverse conditions: A look at achieved advancements*. IGI Global, 2018.

[8] M. Ding and G. Fan, "Articulated and generalized gaussian kernel correlation for human pose estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 776–789, Feb 2016.

[9] J. Chen, S. Nie, and Q. Ji, "Data-free prior model for upper body pose estimation and tracking," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4627–4639, Dec 2013.

[10] M. De Marsico, M. Nappi, and D. Riccio, "Face authentication with undercontrolled pose and illumination," *Signal, Image and Video Processing*, vol. 5, no. 4, p. 401, Aug 2011. [Online]. Available: https://doi.org/10.1007/s11760-011-0244-6

[11] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, Feb 2012.

[12] M. De Marsico, M. Nappi, and D. Riccio, "Measuring measures for face sample quality," in *Proceedings of the 3rd international ACM workshop on Multimedia in forensics and intelligence*. ACM, 2011, pp. 7–12.

[13] T. Jantunen, J. Mesch, A. Puupponen, and J. Laaksonen, "On the rhythm of head movements in finnish and swedish sign language sentences," vol. 2016-January. International Speech Communications Association, 2016, pp. 850–853.

[14] M. De Marsico, M. Nappi, and D. Riccio, "Measuring sample distortions in face recognition," in *Proceedings of the 2Nd ACM Workshop on Multimedia in Forensics, Security and Intelligence*, ser. MiFor '10. New York, NY, USA: ACM, 2010, pp. 83–88. [Online]. Available: http://doi.acm.org/10.1145/1877972.1877994

[15] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1867–1874.

[16] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, April 2009.

[17] M. Demirkus, D. Precup, J. J. Clark, and T. Arbel, "Probabilistic temporal head pose estimation using a hierarchical graphical model," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 328–344.

[18] D. Lee, M. Yang, and S. Oh, "Fast and accurate head pose estimation via random projection forests," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1958–1966.

[19] C. Papazov, T. K. Marks, and M. Jones, "Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4722–4730.

[20] B. M. Smith, J. Brandt, Z. Lin, and L. Zhang, "Nonparametric context modeling of local appearance for pose- and expression-robust facial landmark localization," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1741–1748.

[21] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial features," in *ICPR International Workshop on Visual Observation of Deictic Gestures*, 2004.

[22] M. Kstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov 2011, pp. 2144–2151.

[23] M. Raza, Z. Chen, S.-U. Rehman, P. Wang, and P. Bao, "Appearance based pedestrians: head pose and body orientation estimation using deep learning," *Neurocomputing*, vol. 272, pp. 647 – 659, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217312869

[24] H. Liu and L. Ma, "Online person orientation estimation based on classifier update," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 1568–1572.

[25] K. Pawelczyk and M. Kawulok, "Head pose estimation relying on appearance-based nose region analysis," in *Computer Vision and Graphics*, L. J. Chmielewski, R. Kozera, B.-S. Shin, and K. Wojciechowski, Eds. Cham: Springer International Publishing, 2014, pp. 510–517.

[26] H. Proena, J. C. Neves, S. Barra, T. Marques, and J. C. Moreno, "Joint head pose/soft label estimation for human recognitionin-the-wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2444–2456, Dec 2016.

[27] I. Chamveha, Y. Sugano, D. Sugimura, T. Siriteerakul, T. Okabe, Y. Sato, and A. Sugimoto, "Appearance-based head pose estimation with scene-specific adaptation," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov 2011, pp. 1713–1720.

[28] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[29] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, Jan 2019.

[30] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3d total solution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 78–92, Jan 2019.

[31] A. Kumar, A. Alavi, and R. Chellappa, "Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, May 2017, pp. 258–265.

[32] R. Stiefelhagen, "Estimating head pose with neural networks-results on the pointing04 icpr workshop evaluation data," in *Proc. Pointing 2004 Workshop: Visual Observation of Deictic Gestures*, vol. 1, no. 5, 2004.

[33] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks)," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[34] A. F. Abate, P. Barra, C. Bisogni, M. Nappi, and S. Ricciardi, "Near real-time three axis head pose estimation without training," *IEEE Access*, vol. 7, pp. 64 256–64 265, 2019.

[35] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1428–1440, March 2017.

[36] V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, and R. Horaud, "Head pose estimation via probabilistic high-dimensional regression," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 4624–4628.

[37] N. Gourier, J. Maisonnasse, D. Hall, and J. L. Crowley, "Head pose estimation on low resolution images," in *Multimodal Technologies for Perception of Humans*, R. Stiefelhagen and J. Garofolo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 270–280.

[38] H.-W. Hsu, T.-Y. Wu, S. Wan, W. H. Wong, and C.-Y. Lee, "Quatnet: Quaternion-based head pose estimation with multiregression loss," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1035–1046, 2018.

[39] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang, "Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1087–1096.

[40] Y. Wang, W. Liang, J. Shen, Y. Jia, and L.-F. Yu, "A deep coarse-to-fine network for head pose estimation from synthetic data," *Pattern Recognition*, vol. 94, pp. 196–206, 2019.

[41] B. Ahn, D.-G. Choi, J. Park, and I. S. Kweon, "Real-time head pose estimation using multi-task deep neural network," *Robotics and Autonomous Systems*, vol. 103, pp. 1 – 12, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889017303524

[42] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, Feb 2013. [Online]. Available: https://doi.org/10.1007/s11263-012-0549-0

[43] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug 2004. [Online]. Available: https://doi.org/10.1023/B:STCO.0000035301.49549.88

[44] C. E. Rasmussen, *Gaussian Processes in Machine Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_4

[45] H. Abdi, "Partial least square regression (pls regression)," *Encyclopedia for research methods for the social sciences*, vol. 6, no. 4, pp. 792–795, 2003.

[46] S. G. Kong and R. O. Mbouna, "Head pose estimation from a 2d face image using 3d face morphing with depth parameters," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1801–1808, June 2015.

[47] P. Barra, C. Bisogni, M. Nappi, D. Freire-Obregón, and M. Castrillón-Santana, "Gender classification on 2d human skeleton," in *2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART)*. IEEE, 2019, pp. 1–4.

Paola Barra Paola Barra received the B.S. degree in computer science from University of Salerno and the M.S. degree in Business Informatics from University of Pisa. She is currently pursuing the Ph.D. degree in computer science at Biometric and Image Processing Laboratory (BIPLAB) at University of Salerno, Italy. Her research interests include Machine Learning technics in facial and gait recognition, image processing and video games development. She is member of IEEE and GIRPR/IAPR.

Silvio Barra Silvio Barra was born in Battipaglia, Salerno, Italy, in 1985. He received the M.Sc. degree (cum laude) in Computer Science from the University of Salerno, Salerno, Italy, in 2012 and the Ph.D. in Computer Science from the University of Cagliari. He is currently an Assistant Professor at the Department of Computer Sciences of the University of Cagliari. His research interests include Biometrics, Pattern Recognition, Image Processing, Video Scene Understanding, Human-Computer Interaction, VR/AR. He co-authored over 30 papers in international conference, peer review journals and book chapters in these fields (see https://orcid.org/0000-0003-4042-3000 and https://dblp.uni-trier.de/pers/hd/b/Barra:Silvio). He is member of CVPL (ex GIRPR) and member of the Biometric and Image Processing Lab (BIPLAB), University of Salerno.

Carmen Bisogni Carmen Bisogni received the B.S. degree and M.S. degree (cum Laude) in Mathematics from University of Salerno in 2015 and 2017, respectively. She is currently pursuing the Ph.D. degree in Computer Science at Biometric and Image Processing Laboratory (BIPLAB) at University of Salerno, Italy. Her research interests include applied mathematics for Machine Learning, Biometrics, Image Processing and Statistical Analysis. She is member of IEEE and GIRPR/IAPR.

Maria De Marsico Maria De Marsico is Associate Professor (with qualification as Full Professor) at Computer Science Department of Sapienza University of Rome.Her research interest focus on image and signal processing, especially related to biometric techniques, and on human-computer interaction, especially related to multimodal interfaces. She has co-authored about 180 papers in international journals and conferences. She has co-edited books and journal special issues on topics related to biometrics. She is member of IEEE, ACM, CVPL (Italian association for research in Computer Vision, Pattern recognition and machine Learning (formerly GIRPR - national chapter of IAPR), EAB (European Association for Biometrics) and INSTICC (Institute for Systems and Technologies of Information, Control and Communication). She teaches courses about Biometric Systems and Multimodal Interaction in Master Degrees in Computer Science and in Cybersecurity at Sapienza University of Rome.

Michele Nappi Michele Nappi received the laurea degree (cum laude) in Computer Science from the University of Salerno, Italy, in 1991, the M.SC. degree in Information and Communication Technology from I.I.A.S.S. "E.R. Caianiello," in 1997, and the Ph.D. degree in Applied Mathematics and Computer Science from the University of Padova, Italy, in 1997. He is currently a full professor of Computer Science at the University of Salerno. Author of more than 160 papers in peer review international journals, international conferences and book chapters, He is co-editor of several international books. His research interests include pattern recognition, image processing, image compression and indexing, multimedia databases and biometrics, human computer interaction, VR. Dr. Nappi serves as associate editor and managing guest editor for several international journals. In particular he serves as Associate Editor for Pattern Recognition Letters and promoted many Special Issues for this journal. He is also member of TPC of international conferences. He is team leader of the Biometric and Image Processing Lab (BIPLAB) and received several international awards for scientific and research activities. IEEE Senior Member, GIRPR/IAPR Member, He has been the President of the Italian Chapter of the IEEE Biometrics Council. In 2014 He was one of the founders of the spin off BS3 (biometric system for security and safety).