Neurocomputing 266 (2017) 206–215

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-level structured hybrid forest for joint head detection and pose estimation☆

Yuanyuan Liu[a], Zhong Xie[a], Xiaohui Yuan[b,*], Jingying Chen[c], Wu Song[c]

[a] Faculty of information engineering, China University of Geosciences, Wuhan, China
[b] Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA
[c] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China

ARTICLE INFO

Article history:
Received 18 November 2016
Revised 28 February 2017
Accepted 15 May 2017
Available online 19 May 2017

Communicated by Wang Gang

Keywords:
Multi-level structured hybrid forest
Head pose estimation
Head detection
Joint detection-estimation
Multiple structured features

ABSTRACT

In real-world applications, factors such as illumination variation, occlusion, and poor image quality, etc. make head detection and pose estimation much more challenging. In this paper, we propose a multi-level structured hybrid forest (MSHF) for joint head detection and pose estimation. Our method extends the hybrid framework of classification and regression forests by introducing multi-level splitting functions and multi-structural features. Multi-level splitting functions are used to construct trees in different layers of MSHF. Multi-structured features are extracted from randomly selected image patches, which are either head region or the background. The head contour is derived from these patches using the signed distance of the patch center to the head contour by MSHF regression. The randomly selected sub-regions from the patches within the head contour are used to develop the MSHF for head pose estimation in a coarse-to-fine manner. The *weighted neighbor structured aggregation* integrates votes from trees to achieve an estimation of continuous pose angles. Experiments were conducted using public datasets and video streams. Compared to the state-of-the-art methods, MSHF achieved improved performance and great robustness with an average accuracy of 90% and the average angular error of 6.6°. The averaged time for performing a joint head detection and pose estimation is about 0.44 s.

1. Introduction

Head detection and pose estimation are the key steps in many computer vision applications, such as human computer interaction (HCI), intelligent robotics, face recognition, and recognition of visual focus of attention [1–3]. The former locates the position of a face, and the latter estimates the three dimensional rotation angles of the head according to the orientation of the face. The existing techniques achieve satisfactory results in well-designed environments. Head detection[1] and pose estimation have been approached as separate problems, and various techniques were developed, such as scanning window classifiers, view-based eigenspace methods, and elastic graph models, etc. [4]. Head detection has been dominated by scanning window classifiers, among which is Viola and Jones face detector [5]. Recently, Convolutional Neural Networks (CNNs) have been applied to profile head detection and achieved improved results [6,7]. In real-world applications, however, factors, such as illumination variation, occlusion, poor image quality, etc., make the detection and pose estimation much more challenging [8,9]. Yet, its performance decreases from distortions and side-views of head pose, such as bowing the head and looking to the side, because most methods assume accurate head detection that shows a front or near-front view. Errors in head detection negatively impact pose estimation [6]. Hence, we propose a multi-level structured hybrid forest (MSHF) with multi-structured features for joint head detection and pose estimation. The MSHF is an ensemble learning model that aggregates multi-structured features extracted from randomly selected image patches. The multi-structured features extracted for head detection ensure the relevance to human head and, hence, provide an accurate description of the head pose. The many patches extracted within a head region allow ensemble to construct a diverse set of classifiers for a more robust detection and estimation that circumvents the aforementioned distortions.

This paper presents a MSHF approach for joint head detection and pose estimation that extends a framework of classification and

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail addresses: liuyy@cug.edu.cn (Y. Liu), xiezhong@cug.edu.cn (Z. Xie), xiaohui.yuan@unt.edu, xyuan@cse.unt.edu (X. Yuan), chenjy@email.ccnu.edu.cn (J. Chen), songwu@mails.ccnu.edu.cn (W. Song).

[1] Head location, face detection, and head contour detection are terms frequently used in the applications that require to locate the human face. In this paper, we use these terms interchangeably.

http://dx.doi.org/10.1016/j.neucom.2017.05.033
0925-2312/

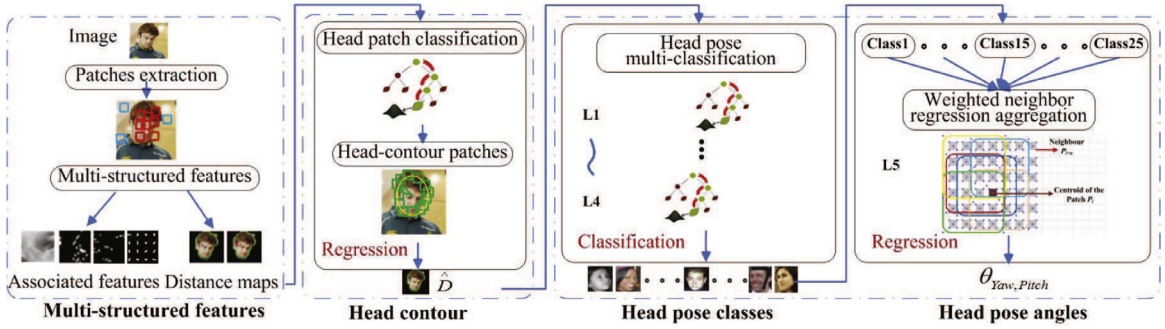


Fig. 1. The flowchart of our MSHF method. Multi-structured features are extracted from randomly selected image patches, which are either head region or the background, firstly. Then, the head contour $\hat{D}$ is derived from these patches using the signed distance of the patch center to the head contour by MSHF regression. The randomly selected sub-regions from the patches within the head contour are used to develop the MSHF for head pose estimation in a coarse-to-fine manner, as shown L1 to L4 layer in the figure. In the L5 layer, the *weighted neighbor structured aggregation* integrates votes from trees to achieve an estimation of continuous pose angles $\theta_{Yaw,\ Pitch}$.

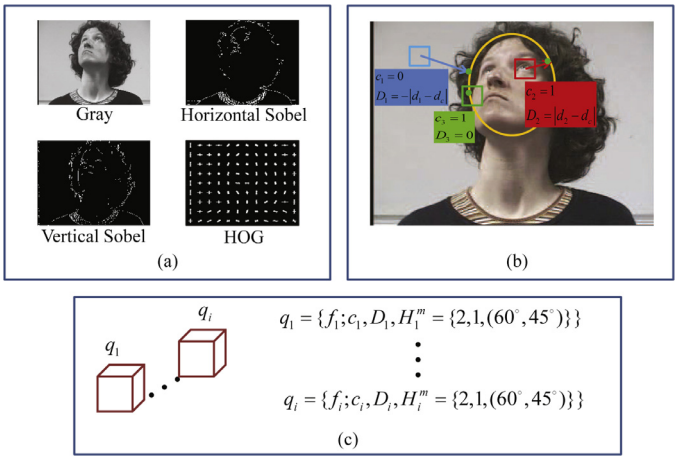


Fig. 2. Multi-structured features. (a) Intensity, Sobel edges and HOG features, (b) labels and each colored patch belong to a distinct class, (c) a set of multi-structured features $q_i = \{f_i; c_i, D_i, H_i^m\}$ that randomly sampled from the image, where $H_i^m = \{2, 1, (60^\circ, 45^\circ)\}$ represent that the head pose class labels (2, 1) in horizontal and vertical directions and the angles are (60°, 45°). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

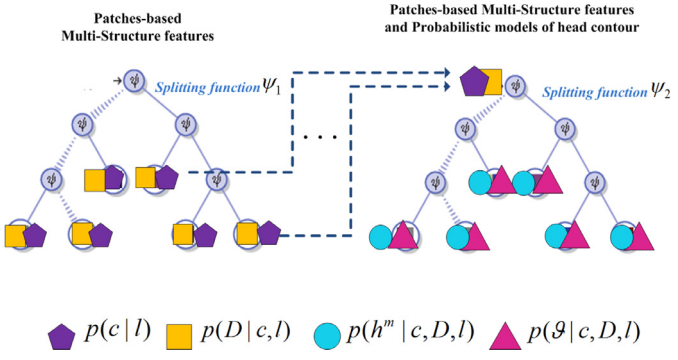


Fig. 3. The MSHF training model in head detection and pose estimation. The left image shows a tree constructed in the head detection layer, and the right image illustrates a tree constructed in detection-estimation layer.

regression forests by introducing multi-level splitting functions and multi-structural features to achieve a joint head detection and pose estimation. Multi-level splitting functions are used to construct trees in different layers of MSHF and the overall architecture is shown in Fig. 1. Multi-structured features are extracted from the randomly selected image patches, which ensure that these features are highly relevant to human head and, hence, provide an accurate description of the head pose. The identified head patches are aggregated into a head contour using sign distance with respect to the head contour. The randomly selected sub-regions within a head contour are used to develop a MSHF in a coarse-to-fine manner. The ensemble aggregates local features for head detection, which are used for pose estimation. The head pose is estimated by integrating weighted votes from the hybrid classification-regression trees, which achieves a continuous angle. The many patches extracted within a head region allow ensemble to construct a diverse set of classifiers for a more robust detection and estimation.

Our contributions include the following:

1. A multi-level structured hybrid forest method is proposed for joint head detection (including head location and contour detection) and continuous head pose estimation (including pose classification and regression) in unconstrained challenging environments.
2. Continuous head poses are decided using patches based on the detected head contour, where the *weighted neighbor structured aggregation* is used to achieve continuous pose angles in multi-probabilistic model regression.
3. The MSHF detects head region (a classification process) and obtains contour location (a regression process), which delivers more accurate head contour detection by suppressing errors in head pose estimation.

The rest of this paper is organized as follows: Section 2 reviews the related work on head detection and pose estimation. Section 3 presents our multi-level structured hybrid forest method. Section 4 discusses the experimental results using publicly available data sets and our data sets. Section 5 concludes this paper with a summary of our method.

2. Related work

Many methods have been proposed for head detection and pose estimation as separate problems. We refer the readers to the recent surveys [1,10] and the references therein. For head detection, a widely used method is Adaboost classifiers with Haar-like features [11], a popular method of which is the Viola Jones face detector [5]. Deformable Parts Model (DPM) [12] based face detection methods have also been proposed in the literature, where a face is defined as a collection of parts. It is shown that in unconstrained environments, partially visible face detection is still a challenging problem. Yet, far distance, various illuminations, occlusion, low image resolution, expression, and make-up degrade its performance. Recently, Deep Convolutional Neural Networks (DNNs) is applied for head detection [6,7,13], which achieved improved performance in cases such as multi-view occlusion and low image resolution.

The improvement, however, heavily relies on the large number of training sets and high performance computing power.

Head pose estimation is usually achieved using template matching, subspace embedding, and tracking methods [1]. Methods, such as neural networks (NN) [14], support vector machines (SVM) [8], nearest prototype matching [15], manifold embedding [16] and random forest [17–19], have gained popularity for head pose estimation in natural environment. Gourier et al. [14] applied an auto-associative network to learn the mapping for head pose estimation on low-resolution images. The method achieved a precision of 10.3° in the yaw angle and 15.9° in the pitch angle on the Pointing'04 database. Orozco et al. [8] trained a multi-class SVM for head pose classification in crowed scenes. The performance on videos acquired in crowd public spaces with low resolution reached 80% accuracy rate in a four-pose classification. Wu et al. [15] proposed a two-stage framework for head pose estimation based on a geometrical structure. Peng et al. [16] proposed a coarse-to-fine pose estimation framework, where the unit circle and 3-sphere are employed to model the manifold topology on the coarse and fine layers, respectively. The pose-related and unrelated factors can be decoupled in a latent instance parametric subspace. This method achieved much improved performance in estimation of yaw angle on in-the-wild datasets. Yet, its performance for a combination of yaw and pitch is unclear. This method can achieve superior performance for yaw estimation on in-the-wild datasets. Multi-class and regression random forest becomes a popular method for head pose estimation on low resolution images owing to their robustness. Liu et al. [18] extended random forest by introducing a Dirichlet-tree distribution.

Structured learning addresses the problem of learning a mapping where the input space may be arbitrarily complex [20,21]. Several structured learning approaches have been developed including Conditional Random Field [22], Structured Support Vector Machine [23], and Structured Random forest (SRF) [20,24,25]. Other improved structured learning methods such as Hough forest [17,26] have been proposed for pose estimation, which introduced Hough transform for voting. Zhang et al. [17] developed a head pose estimation based on Hough forests on low-resolution images.

One of the earlier approaches for joint addressing the tasks of face detection and pose estimation was proposed in [27], which employs a mixture of trees with a shared pool of parts. Multi-task learning using CNNs has also been developed [6], which learns five horizontal head poses to improve landmark localization.

When pose estimation and head detection are processed separately, the final pose estimation is inevitably affected by the errors induced from head detection process. And most of the aforementioned pose estimation methods rely on accurate head detection. The obstructed face and excessive background included in the head region make features extracted for pose estimation error-prone. Hence, the significant features used for head detection could be used for pose estimation, which ensure their relevance to human head and provide an accurate description of pose. In addition, a diverse set of features offers greater robustness for head detection and pose estimation when distortions exist, which necessitates the development of an ensemble architecture.

3. Multi-level structured hybrid forest

Fig. 1 illustratesan overview of our proposed MSHF approach for joint head detection and pose estimation. Randomly selected image patches are extracted with multi-structured features and classified into head patches and background patches, firstly. Then, the head patches are aggregated into the head contour using the signed distance of the patch center to the head contour. The randomly selected sub-regions from patches within the head contour are used to develop a multi-level structured hybrid forest for joint head pose estimation in a coarse-to-fine manner. Finally, the head poses are estimated by integrating votes from hybrid classification-regression trees to achieve a continuous angle.

3.1. Multi-structured features

Multi-structured features are extracted from randomly selected image patches as shown in Fig. 2. The features are used for head detection as well as for pose estimation, which ensure their relevance to human head and provide an accurate description of poses.

In each training image, we randomly select a set of patches Q, $Q = \{q_i\}$ and $q_i = \{f_i; c_i, D_i, H_i^m\}$, where f_i is the associated image features and c_i, D_i, H_i^m are the labels and annotations. $f_i = \{f_i^1, f_i^2, f_i^3\}$, f_i^1 contains the gray values of the neighbor patches, f_i^2 represents the Sobel edge descriptors in the horizontal and vertical directions, and f_i^3 represents the HOG descriptors extracted from the patches. $\{c_i, D_i, H_i^m\}$ are structural labels and features. c_i is the label to indicate if the patch is inside a head area. The distance maps assign a n-dimensional distance vector $D_i = \pm|d_i - d_c|$, where d_c is the distance from the center of a patch d_i to the closest boundary point d_c on the head contour. The negative distance represents that the patch is outside of the head, while the positive distance represents that the patch is within the head region. When a patch is on the head contour, the distance is zero. $H_i^m = \{h_i^m, \vartheta_{y,p}\}$ contains the head pose and angle. The annotation of h_i^m in different layers of MSHF follows the scheme in [18], $\vartheta_{y,p}$ represents the head pose angle in both the horizontal and vertical directions.

3.2. MSHF construction

The multi-structured features extracted within a head region allow an ensemble to construct a diverse set of classifiers for a more robust detection and estimation. In the training of a MSHF, each tree is constructed using a set of feature patches Q. Fig. 3 illustrates the elements in the training model. The left image of Fig. 3 shows a tree constructed in the head detection layer, which is built and selected randomly from a set of the image patch-based multi-structured features. The right image of Fig. 3 illustrates a tree constructed in detection-estimation layer, which is grown using multi-structured features and probabilistic models of head contour detection. The hybrid probabilistic models in leaves can be seen under the tree.

To construct a tree in MSHF, a node divides a set of training patches Q into two subsets Q_L and Q_R, i.e.,

$$Q_L = \{q_i|\varphi < \tilde{\varphi}\}, \quad \text{and} \quad Q_R = \{q_i|\varphi > \tilde{\varphi}\}, \tag{1}$$

where φ is the difference between two patches as follows:

$$\varphi = \frac{1}{|R_1|}\sum_{j\in R_1} f(j) - \frac{1}{|R_2|}\sum_{j\in R_2} f(j), \tag{2}$$

where R_1 and R_2 are arbitrary regions in a patch. $|\cdot|$ gives the size of a patch, $f(j)$ is the image feature, j denote a pixel. Fig. 4 shows an example of randomly selected regions in a patch. $\tilde{\varphi}$ is decided by maximizing the Information Gain (IG) as follows:

$$\tilde{\varphi} = \arg\max_{\varphi}\left[H(Q) - \sum_{s\in\{L,R\}}\frac{|Q_s|}{|Q|}H(Q_s)\right], \tag{3}$$

where $\frac{|Q_s|}{|Q|}, s \in \{L, R\}$ is the ratio between the number of samples in Q_L (arriving at the left subset), set Q_R (arriving at the right subset), and Q. $H(Q)$ is the entropy of Q.

In order to construct different tree in different layer of the MSHF, multi-level splitting functions have been used. For the layer of head detection, the trees classify a patch as part of a head and

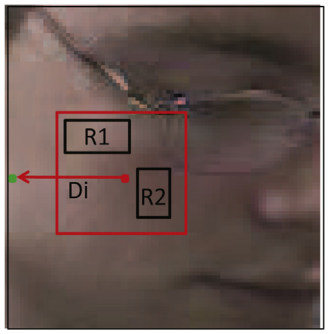


Fig. 4. A training patch with its distance vector (arrow) between the patch's center (red dot) and the closest boundary point on the head contour (green dot). R_1 and R_2 are randomly selected regions in a patch. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

cast votes into the spaces spanned by head contour locations. The integrated entropy H_d in head detection trees is computed as follows:

$$\begin{aligned} H_d(Q) &= -\sum_c \int_D p(c,D)\log p(c,D)dD \\ &= -\sum_c p(c)\log p(c) \\ &\quad +\sum_c p(c)(-\int_D p(D|c)\log p(D|c)dD) \end{aligned} \tag{4}$$

$$p(D|c) \propto p(c)\exp\left(-\frac{|D|}{\lambda}\right), \tag{5}$$

where $p(c)$ is the probability that the patch belongs to the head area, $p(D|c)$ represents the probability of the head contour. The factor λ controls the steepness of this function. c labels if the patch is inside a head area, D is the distance offset vector.

For the layer of pose estimation, our goal is to learn the head pose probability $p(H|c=1, D=0)$ given the head contour $p(c=1, D=0)$. A joint detection-estimation tree is trained with multi-structured features from the neighboring head contour-patches. We rewrite this multi-level probabilistic distribution as $p(H|c=1, D=0) = p(c=1, D=0)p(h^m|h^{m-1})p(\theta|h^m)$. So to learn the multi-level distribution within the MSHF, we define the entropy H_e in each sub-layer of pose estimation as follows:

$$\begin{aligned} H_e(Q) = & -\sum_{h^m}\int_\theta p(h^m,\theta|c=1,D=0) \\ & \log p(h^m,\theta|c=1,D=0)d\theta \\ = & -\sum_{h^m} p(h^m|c=1,D=0,h^{m-1}) \\ & \log p(h^m|c=1,D=0,h^{m-1}) \\ & +\sum_{h^m} p(h^m|c=1,D=0,h^{m-1}) \\ & (-\int_\theta p(\theta|c=1,D=0,h^m) \\ & \log p(\theta|c=1,D=0,h^m)d\theta) \end{aligned} \tag{6}$$

where $p(h^m|c=1, D=0, h^{m-1})$ is the head pose probability in the m-th sub-layer of an estimation forest and $p(\theta|c=1, D=0, h^m)$ is the probability of head rotation angle.

The training continues until the tree reaches the maximum depth or the number of samples in a node falls below a threshold and a leaf l is created. A leaf node stores the structured probability for the patch $p(c)$, the distance to the head contour $p(D|c)$, and the head pose $p(H|c, D)$.

For a leaf node in a head detection tree, we can simplify the distributions over multiple structured probabilities by adopting multivariate adaptive Gaussian mixture models (GMM) [18,28]: (1) patch class probabilistic distribution $p(c|l) = N(c; \bar{c}, \Sigma_l^c)$, (2) head contour's distance distribution $p(D|c) = N(D; \bar{D}, \Sigma_l^D)$. For a leaf node in a head pose tree, the distributions is modeled as multivariate GMM: (3) discrete head pose class distribution $p(h^m|c, D) = N(h^m; \overline{h^m}, \Sigma_l^{h^m})$, and (4) continuous head angle distribution $p(\theta|h^m) = N(\theta; \bar{\theta}, \Sigma_l^\theta)$. In these multivariate GMM, $\bar{c}, \bar{D}, \overline{h^m}, \bar{\theta}$ and $\Sigma_l^c, \Sigma_l^D, \Sigma_l^{h^m}, \Sigma_l^\theta$ are the mean and covariance of leaves' probabilities, respectively.

3.3. Head detection

For head detection in unconstrained environment, the MSHF classifies the random patches into inside (or on) a head or outside the head, which are integrated into a head contour. In the procedure of detection, the image patches pass away the trees in a head detection sub-forest of MSHF. All patches end in a set of leaves of the sub-forest. In these leaves, the task need to classify a patch as part of a head and to regress head contour location.

Given a forest $F = \{F_t\}_{t=1}^T$ and a set of patches Q, let p_F be the average joint probability of c and D:

$$p_F(c,D) = \frac{1}{T}\sum_{t=1}^{T} p(c,D|l_t(Q)), \tag{7}$$

where T is the number of trees in the forest. The most probability of head contour position $\hat{D}$ for a patch q_i is obtained as follows:

$$\begin{aligned} \hat{D} &= \arg\max_D p_F(c,D) \\ &= \arg\max_D p_F(c)p_F(D|c) \\ &\propto \arg\max_D p_F(c)g(q_i), \end{aligned} \tag{8}$$

where $p_F(c|q_i)$ is the head class probability of a patch q_i and it can be obtained by averaging the outputs of all trees of the sub-forest. $g(q_i)$ is the regressor for head contour position D at the patch location q_i:

$$g(q_i) \propto \sum_l w_s K\left(\frac{D-(q_i+\tilde{D}_l)}{h}\right)\sigma(l), \tag{9}$$

where

$$\sigma(l) = \begin{cases} 1 & exp\left(-\frac{|D|}{\lambda}\right) \geq \alpha \\ 0 & otherwise \end{cases},$$

and K is a Gaussian kernel and the bandwidth parameter h, w_s is the weight of leaf l, and the confident factor $\sigma(l)$ avoids a bias towards an average face configuration. To integrate the votes by different patches, we aggregate them into a Hough image $V(q_i)$ and the hypothesis head contour location:

$$V(q_i) = \sum_{q_i} p(D|c=1), \tag{10}$$

where $c = 1$ represents that patches belong to a head area. The location of head contour computes the Hough image V and identifies the most likely locations. Only the head patches around head contours have be used for head pose estimation.

3.4. Pose estimation

In pose estimation, yaw and pitch are estimated from random patches within the head region. Our idea is to integrate pose estimated from a number of randomly selected patches to reach a continue angle.

Fig. 5 shows an example of cascaded head pose estimation with the head contour-patches in the horizontal and vertical directions.

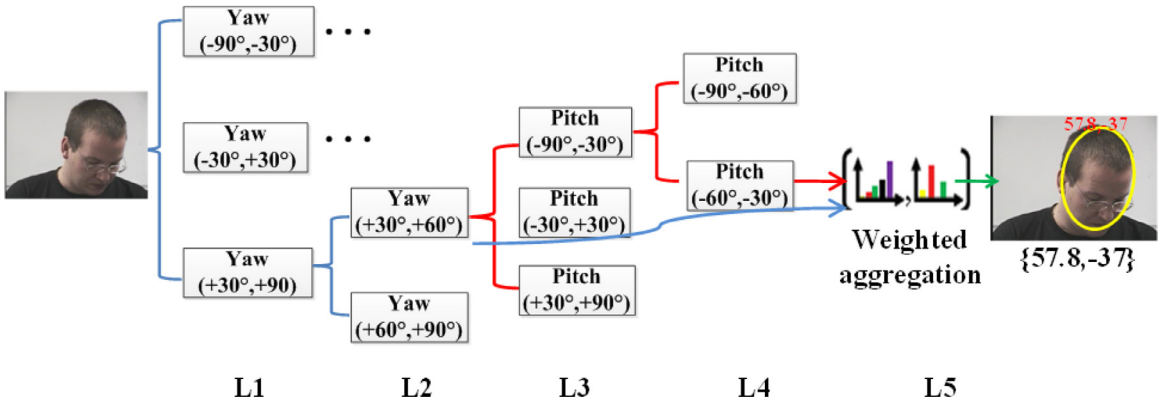


Fig. 5. Our hierarchical head pose estimation scheme.

It includes five cascaded estimations. In L1 sub-layer, three yaw angles are classified. For refined horizontal estimations in the L2 sub-layer, five yaw angles are classified based on the results of L1. The procedure of vertical estimations is similar to the horizontal estimations. In the L3 sub-layer, there are three pitch angles under five yaw angles. In the L4 sub-layer, five refined pitch angles are classified. Finally, in the L5 sub-layer, continuous head poses are estimated using patches based on the detected head contour, where *the weighted neighbor structured aggregation* is used to achieve continuous pose angles in multi-probabilistic model regression. Eqs. (11)–(13) describe the multi-probabilities of head poses computed by GMM and *weighted neighbor structured aggregation*. Note that this *weighted aggregation* is a weighted summation process as shown in Eq. (13).

When head contour patches reach the leaves of the MSHF, pose probability is computed as follows:

$$p(H|l) = p(h^m, \theta|l_m) = p(h^m|l_m)p(\theta|h^m, l_m), \tag{11}$$

where l_m is a leaf in the m-th sub-layer of the head pose forest.

In L1–L4 sub-layers, we simplify the distribution over the discrete head pose class by an adaptive multi-variance Gaussian Mixture Model (GMM):

$$p(h^m|l_m) = N\left(h^m; \overline{h^m}, \Sigma_{l_m}^{h^m}\right), \tag{12}$$

where $\bar{h}^m$ and $\Sigma_{l_m}^{h^m}$ are the mean and covariance matrix of the contextual head pose class.

In the L5 sub-layer, to estimate the head pose angle, a *weighted neighbor structured aggregation* method is used. Different from the conventional random forest, which assigns an class label to each patch given a test patch q_i, our *weighted aggregation* makes a prediction by taking into consideration of the neighboring ones. We randomly select M patches in the neighborhood u and, hence, get M predictions. The probability of a head pose θ in a patch is calculated by integrating the estimations in the neighborhood:

$$\begin{aligned} p(\theta|h^m, q_i) &= \frac{1}{|M|}\sum_u w_s p(\theta|h^m, q_{i+u}) \\ &= \frac{1}{|M|}\frac{1}{|T|}\sum_u\sum_t w_s N(\theta; \overline{\theta}, \Sigma_{l_{i+u}}^{\theta}), \end{aligned} \tag{13}$$

where T is the number of the trees t in the L5 sub-layer, $\overline{\theta}$ and $\Sigma_{l_{i+u}}^{\theta}$ are the mean and covariance matrix of pose angles. This process is illustrated in Fig. 6. To account for the imbalance of the training samples, we store the weight $w_s = P_s/P$ (the ratio of the number of samples in each subset P_s) and the number of samples P in each tree.

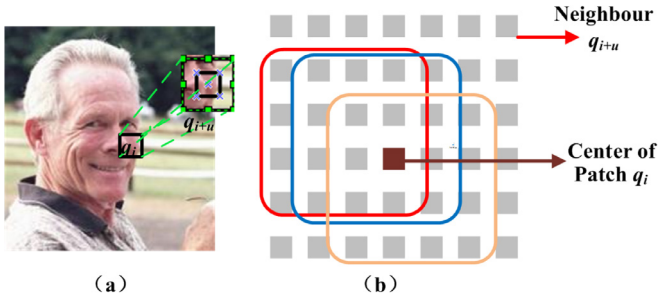


Fig. 6. Weighted aggregation for head pose. (a) The patch and neighbor pixels in the image. (b) Each patch collects class hypotheses from the structured labels predicted for itself and the neighboring patches. For clarity purpose, only three out of nine patches are shown.

4. Experimental results

4.1. Datasets and settings

To evaluate our approach, five challenging face datasets were used: Pointing'04 dataset [29], LFW dataset [30], AFW [27] and CCNU head pose dataset in the wide classroom [18]. These datasets were chosen since they contained unconstrained face images with poses ranging from −90° to +90°. The Pointing'04 head pose dataset is a benchmark of 2790 monocular face images of 15 people with variations of yaw and pitch angles from −90° to +90°. For every person, 2 series of 93 images (93 different poses) are available. The CCNU dataset was collected included an annotated set of 58 people with 75 different head poses from an overhead camera in the wide scene. The LFW dataset consists of 5749 individual facial images. The images were collected in the wild, and varied in poses, lighting conditions, resolutions, races, occlusions, make-ups, etc. In AFW dataset, images tend to contain cluttered backgrouds with large variations in both face viewpoint and appearance (aging, sunglasses, skin color, expression, make-ups etc.).

In these datasets, each face is labeled with a bounding ellipse based on 68 landmarks and a discretized viewpoint (−90° to +90° every 15°) along pitch and yaw directions. Our method was trained with 2000 images from Pointing'04, 5000 images from LFW dataset, and 4000 images from CCNU dataset. In evaluation, we used 500 images from Pointing'04 dataset, 2000 images from LFW dataset, 478 images from AFW dataset, 1500 images from CCNU dataset, and real life videos were used. The methods were implemented using C++, OpenCV library, and Boost library and the experiments were conducted in a PC with Intel(R) Core(TM) i5-2400 CPU@ 3.10 GHz, RAM 8 GB.

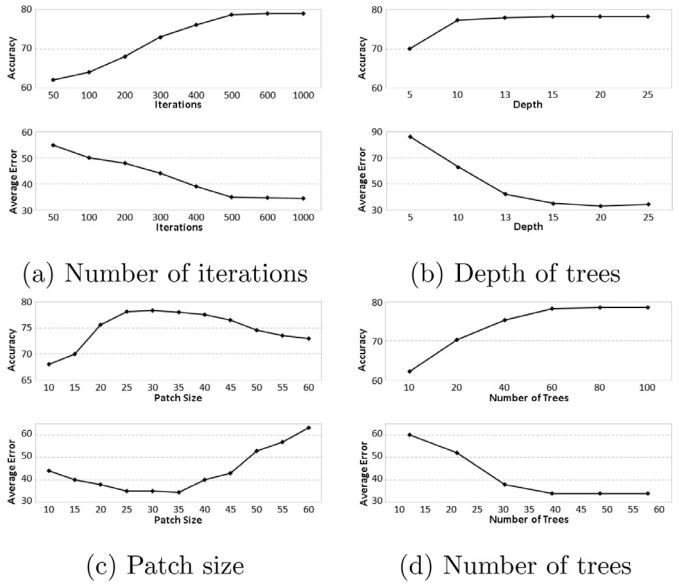


Fig. 7. Accuracy of pose estimation (%) and average error of head detection (in pixels) with different parameter settings in a MSHF.

4.2. Parameter selection

In training the trees, we adopted bagging for sample selection, and each tree was trained on a random subset that contains 10% of the examples. Fig. 7 presents the accuracy of pose estimation and average error of head detection with respect to four key parameters.

Fig. 7(a) depicts the accuracy and average error with respect to the number of splitting iterations. It is clear that as the iterations increased the performance of MSHF in term of both accuracy and error improved, and it reached a plateau at about 500. Allowing greater number of splitting iterations makes no further changes. Hence, we used 500 splitting iterations per node.

Fig. 7(b) depicts the accuracy and average error with respect to the number of tree depth. The trend was very similar to that of the number of iterations except that the accuracy of pose estimation reached a plateau quickly. However, considering the error, we adopted 15 as the maximum depth of trees in the random forests.

Fig. 7(c) depicts the accuracy of pose estimation and average error of head detection with respect to patch size. There existed a peak accuracy and a minimum error when the range of patch size varied from 25 to 30. Hence, it is plausible to take the patch size of 30×30 as a balance to the accuracy and error. Either a small or a large patch size produced sub-optimal outcomes. Indeed, a small patch fails to provide an indication for the expected estimation; whereas a large patch is prone to mistakes due to occlusion and background noise.

Fig. 7(d) depicts the accuracy of pose estimation and average error of head detection with respect to the number of trees. As shown in the plots, 60 trees for head detection and 80 trees for pose estimation yielded the greatest performance and hence were used in the rest of our experiments.

4.3. Head contour detection

Fig. 8 illustrates exemplar results of head detection using our proposed method on AFW, LFW, CCNU, and Pointing'04 datasets, which include various cases of occlusions, illuminations, resolutions, and make-ups in unconstrained challenging environments. The ellipses outline the detected heads in the images. Our MSHF method obtained qualitative results on these challenging datasets.

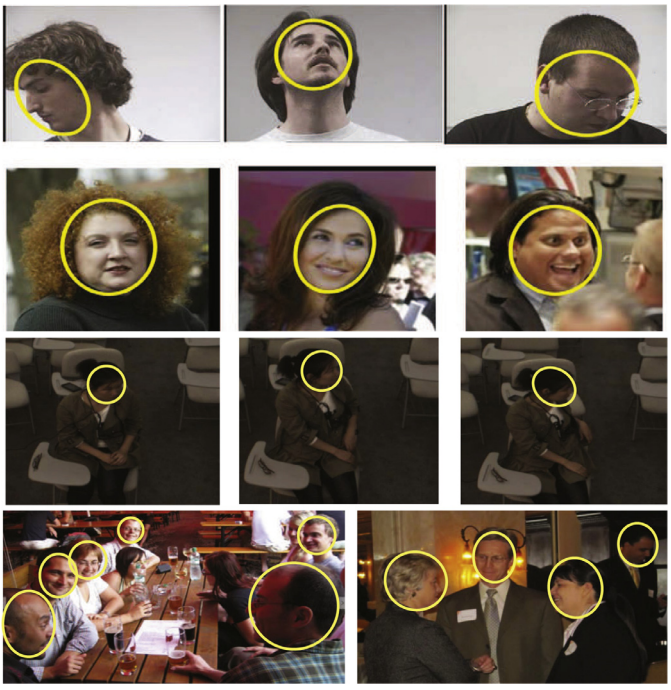


Fig. 8. Qualitative results of our detector on AFW, LFW, CCNU, and Pointing'04. The ellipses outline the detected head in the images, which are used for the pose estimation. One can see that our MSHF method can obtain qualitative results on these challenging datasets.

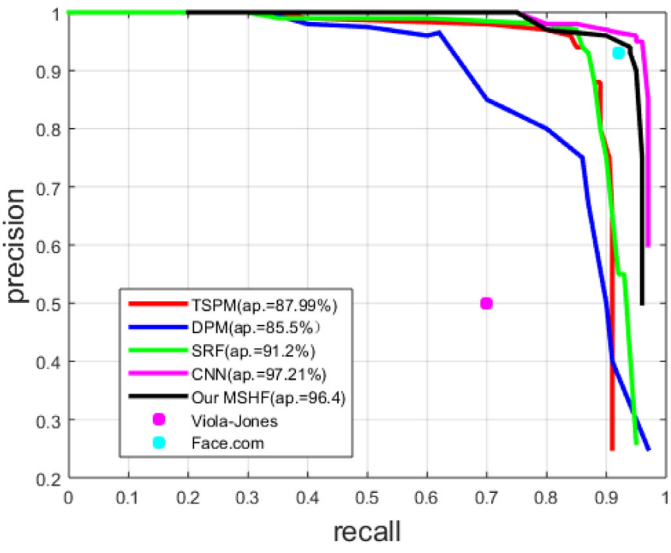


Fig. 9. On the AFW dataset we compare our performance with the state-of-the-art methods including OpenCV frontal+profile Viola–Jones detector [5], Tree-structured part models (TSPM) [27], structured random forests (SRF) [25], convolutional neural networks (CNN) [7], DMP [12] and face.com.

In order to evaluate head detection results of our methods, we compare the method with OpenCV frontal+profile Viola–Jones detector [5], Tree-structured part models (TSPM) [27], structured random forests (SRF) [25], CNN [7], DMP [12] and face.com detector on AFW dataset. We adopt the PASCAL VOC precision–recall protocol for face detection (requiring 50% overlap). Evaluation results on AFW dataset are summarized in Fig. 9. Our method outperform Viola–Jones, TSPM, DMP and SRF significantly and are only slightly below CNN. The CNN model needs more training images and GPU supporting, while our MSHF can achieve the similar performance in CPU instead of GPU. Noted that our performance is similar to

Table 1
Average MSD in head delineation using different methods.

Methods	LFW	Pointing'04	CCNU	AFW
Viola–Jones [5]	77.5 (6.2)	58.4 (5.6)	70.3 (7.1)	80.4 (6.4)
SRF [25]	61.4 (4.0)	47.2 (3.3)	54.5 (4.8)	72.3 (5.2)
MSHF	39.7 (2.3)	36.5 (1.5)	38.6 (2.9)	55.3 (4.9)

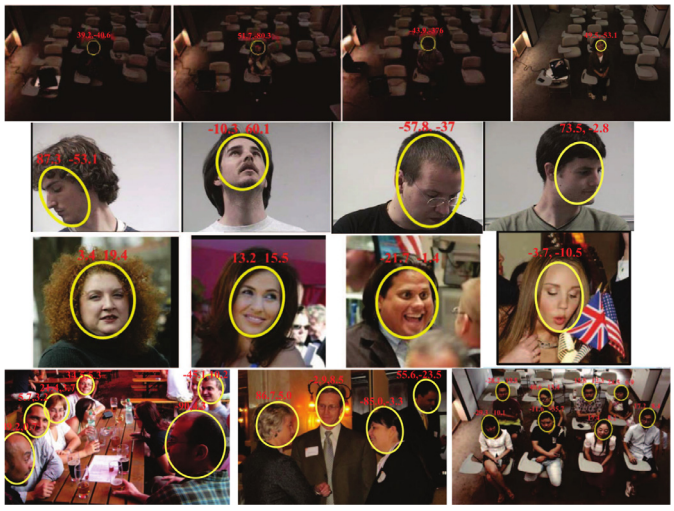


Fig. 10. Examples of pose estimation using the CCNU, LFW, Pointing'04, AFW datasets and a video (the right bottom one). Our MSHF method can obtain qualitative results on these challenging datasets with non-visible faces, illuminations, resolutions, poses, occlusions, expressions and make-ups, etc.

Table 2
Accuracy (%) and average errors (degrees) of MSHF method.

Datasets	Yaw	Pitch	Yaw+Pitch	Ave. error	STD
Pointing'04	92.3	90.7	84.0	6.6	3.5
LFW+AFW	85.6	80.3	70.4	11.6	5.9
CCNU	83.8	86.4	74.2	12.4	6.7

the performance of commercial face.com detector without needing hundred million of training images.

To gain a quantitative understanding of the delineation error of head contour regression, we adopted the mean surface distance (MSD) [24]:

$$\mathcal{D} = \sum_{x,y} \left\| d(x,y) - \dot{d}(x,y) \right\|_2, \tag{14}$$

where $d(x, y)$ is a point on the detected head boundary and $\dot{d}(x, y)$ is a nearest point on the reference head boundary. MSD computes the cumulative distance of the head contour to the reference. We also compared our method with among detected head contour methods. Table 1 presents the average MSD and STD. on the four datasets by using the three methods. The unit of error is pixel. With the four datasets, MSHF consistently outperforms both Viola–Jones detector [5] and SRF [25] in head delineation with much smaller MSD as well as a smaller standard deviation. MSRF reduced the MSD by approximately 30% with respect to SRF. It is evidential that MSRF yielded more accurate head detection results.

4.4. Accuracy of pose estimation

Fig. 10 depicts examples of head pose estimation results on CCNU, LFW, Pointing'04, AFW datasets and real-life videos. The ellipses enclose the detected head within the images. The estimated head pose angles are written above the ellipses, where the left is the yaw angle and the right is the pitch angle. Additionally, Fig. 10 shows some example of pose estimation in a real-life video of a crowd. One can see that our MSHF method can obtain qualitative results on these challenging datasets with various non-visible faces, poses, illuminations, resolutions, occlusions, expressions and make-ups, etc.

Table 3
Accuracy (%) and average error (in degrees) using different methods on Pointing'04 dataset.

Methods	Yaw	Pitch	Yaw+Pitch	Ave. error	STD
MLD [31]	84.30	86.24	72.3	7.19	4.9
D-RF [18]	83.52	86.94	71.83	13.4	5.5
HF [32]	82.3	84.86	70.54	13.7	5.2
M-SVM [8]	80.6	82.5	60.46	20.2	5.7
M-RF [19]	78.4	68.73	62.23	26.3	8.4
NN[14]	79.5	70.36	56.7	29	7.5
MSHF	92.3	90.7	84.0	6.6	3.5

Table 4
Accuracy (%) and average error (in degrees) using different methods on AFW and LFW datasets.

Methods	Yaw	Pitch	Yaw+Pitch	Ave. error	STD
AVM [33]	80.56	74.75	58.33	17.2	–
D-RF [18]	80.8	77.4	58.9	13.5	7.3
Embedding [34]	72.5	60.13	43.38	28.15	–
TSPM of [27]	81.0	–	–	15.3	–
MSHF	85.6	80.3	65.65	11.6	5.9

Table 5
Computation time comparison (per second).

Method	MSHF	D-RF [18]	HF [32]	M-RF [19]	Viola–Jones+SVM
Mean	0.4357	0.98995	1.06547	1.36859	1.0446
STD.	0.11	0.15	0.19	0.18	0.13

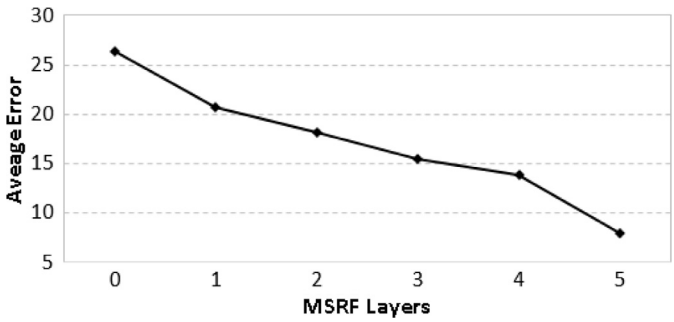


Fig. 11. Average errors at different sub-layers of the MSHF.

Table 2 lists the accuracy with respect to the yaw and pitch rotations of our MSHF method as well as the average error in terms of degrees. A 4-fold cross-validation was conducted. Among the four datasets, our method achieved the greatest performance with Pointing'04 dataset. The accuracy in yaw and pitch angles were in the range of 80% to 90%, respectively. Note that LFW, AFW and CCNU datasets consist of great variation of poses, lighting, occlusions, etc. For these three more challenging datasets, the accuracy was also above 80% for yaw rotation and above 70% for pitch rotation. The average error reached 6.6° for the Pointing'04 and those of the other two were close to 10%. In all cases, the standard deviation of the average error was fairly low.

In our method, a multi-level hybrid forests based on multi-structured features is used and refined estimations of head pose are produced within this hierarchical structure. Fig. 11 shows the average errors at different sub-layers of the MSHF for pose estimation. The starting point where the MSHF sub-layer is zero indicates that a conventional random forest was used to make an estimation of the head pose. The average error of this conventional RF was 26.3°. The MSHF sub-layers of 1 through 4 give the intermediate estimations and the MSHF sub-layer 5 gives the final pose estimation. As shown in Fig. 11, as the analysis traced through the random forests, the pose was refined to be much more

accurate. The final average estimation error was 7.9° as the result of the 5th sub-layer in the MSHF. This indicates that with multi-level hybrid forests improved the estimation accuracy. In theory, additional sub-layers of MSHF could further improve the accuracy; however, in practice the complexity in training such random forests and overfitting issue make more sub-layers a less favorable choice.

In comparison with the state-of-the-art head pose estimation methods, we conducted experiments using the Dirichilet-tree enhanced random forest (D-RF) [18], Multivariate label distribution (MLD) [31], Hough forest (HF) [32], multi-class random forest (M-RF) [19], multi-class SVM (M-SVM) [8], and neural networks (NN) [14] on Pointing'04 head pose dataset. The same training and testing datasets were used, and we employed a 4-fold cross-validation. Table 3 lists the average accuracy and error across using these methods. MLD [31], D-RF [18] and HF [32] yielded comparable results with an accuracy of approximately 70% in yaw and pitch rotations. MLD [31] proposed to associate a multivariate label distribution to each image for head pose estimation in yaw and pitch rotation. D-RF [18] proposed a dirichlet-tree distribution enhanced random forest to 25 class head pose estimation. HF [32] improved random forests with Hough voting for real-time head pose estimation. M-SVM [8], M-RF [19], and NN [14] produced similar accuracy in the range of 60%. MSHF exhibited the highest accuracy of 84% and the accuracy of the estimation of the yaw component reached 92.3%. The multi-structured features from head contour-patches and a *weighted neighbor aggregation* method removes the unwanted patches from face deformation and large rotation angle in unbalanced sample sets, which ensures improved accuracy in our proposed method. The improvement with respect to the second best (MLD [31]) is about 9%. We get the same observation from the average estimation error. The average error of MSHF method was 7.9°. In addition, the standard deviation of MSHF indicates that MSHF achieved the greatest consistency with a smallest STD. It is evidential that our MSHF improved the head pose estimation with great robustness.

Table 4 lists the average accuracy and error across on more challenging AFW and LFW datasets using AVM [33], D-RF [18], Feature-embedding [34], TSPM [27] and our MSHF. AVM [33] proposed features-based manifold embedding for head pose estimation in unconstrained environments. The average accuracy reached to 58.33% within 15° in horizontal and vertical direction estimations. TSPM of [27] presented a unified the mixture tree-structured part model for face detection, pose estimation, and landmark estimation in real-word and wild images. The TSPM method only estimated head poses in the horizontal direction with an average accuracy of 81%, while our proposed method can estimate head contour and head poses in horizontal and vertical directions. Feature-embedding [34] proposed a feature embedding based regression function method and achieve the average accuracy of 43.38% in the challenging datasets. The compared results can be shown in Table 4. Our MSHF method outperforms other methods with an average accuracy of 65.65% and average error of 11.6° in the horizontal and vertical estimation on the challenging datasets.

4.5. Analysis of different image features and head detectors with respect to neighborhood patches

In our method, various image features can be used as input for training of a MSHF. It is to our interest to study the impact of image features to the estimation accuracy. However, the coverage of image patches is an integral factor and cannot be separated from the employment of image features. To understand the effects of features and patch coverage, we conducted experiments with four features including multi-structured features, LBPH, Gabor filter bank with eight different rotations and five different phase shifts, and gray values of raw input image.

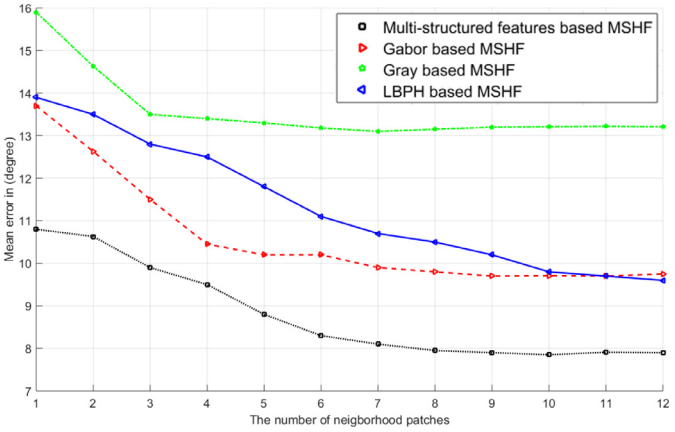


Fig. 12. Accuracy of MSHF using various image features with respect to the number of neighborhood patches.

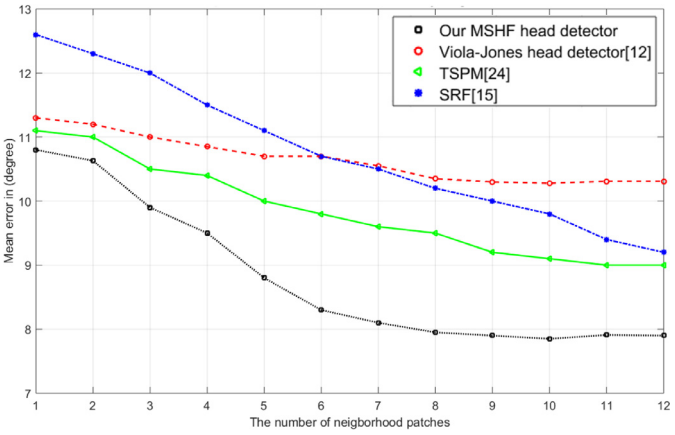


Fig. 13. Using different head detectors with respect to the number of neighborhood patches.

Fig. 12 illustrates the curve of average error of pose estimation using different image features with respect to the number of neighborhood patches. The horizontal axis is the number of neighborhood patches in the textbfweighted neighbor aggregation; the vertical axis is the average error of pose estimation. As shown in the plot, estimation error of head pose decreases with the increment of the number of neighborhood patches regardless of the image feature used. The decrement gradually reached a plateau and any further increment of number of patches had little impact to the error. Such elbow point is about 7 for all cases. Among the four features, our proposed multi-structured features extracted from head contour-patches consistently performed better than the others did. Gray value exhibited the highest error. Meanwhile, one can see the number of neighborhood patches 9 is a good choice in our experiments.

In order to evaluate our joint detection-estimation method, we compared the average head pose error with different head detectors with respect to the number of neighborhood patches. We compared our MSHF detector with the following: (1) OpenCV frontal+profile Viola–Jones detector [5], (2) Tree-structured part models (TSPM) of [27], (3) Structured random forests (SRF) of [25]. As shown in Fig. 13, our proposed MSHF demonstrated a significant advantage to other methods when different number of patches were used. Our MSHF method all outperformed other methods for joint head pose estimation due to declining the influence of head detection by the joint detection-estimation method. It is interesting to note that the error of using head-contour patches from

Viola–Jones head detector is less influenced by the number of patches. It is proved the benefit of our proposed joint detection-estimation method.

4.6. Analysis of time complexity

Table 5 reports the average computational time of five methods on AFW dataset. The methods were implemented using C++, OpenCV library, and Boost library and the experiments were conducted in a PC with Intel(R) Core(TM) i5-2400 CPU@ 3.10 GHz, RAM 8 GB. It can be seen that all methods are fairly efficient in processing the test images. MSHF yielded an average of 0.4357 s, whereas the others are about 1 s or above. The standard deviation of the computational time of MSHF is also the minimum among all. It is evident that MSHF is more efficient; the reduction with respect to D-RF (the second most efficient method) is about 56%.

5. Conclusion

This paper describes a multi-level structured hybrid forest (MSHF) for joint head detection and pose estimation. The MSHF extends random forest to integrate classification-regression forests by introducing multi-level splitting function and multi-structural features to achieve a joint head detection and pose estimation. Multi-level splitting functions are used to construct different tree in different layer of MSHF. Multi-structured features are extracted from randomly selected image patches and head contour is derived using the signed distance of the patch center to the head contour. The randomly selected sub-regions from these patches are used to construct a multi-level structured random forest. The *weighted neighbor structured aggregation* is introduced to the MSHF by integrating discrete votes from hybrid trees to achieve continuous pose angles in horizontal and vertical directions. Our proposed MSHF can do head region location, head contour detection, head pose classification, and continuous head angle estimation in a joint way.

Experiments were conducted using public challenging datasets and video streams. Our experimental results demonstrated that among the four image features adopted in our experiments, multi-structured features extracted from head contour-patches consistently outperformed the others. In comparison to the state-of-the-art methods, MSHF yielded more accurate head contour detection results. The averaged time for performing a joint head detection and pose estimation using neighborhood multi-structured feature around head contour is about 0.44 s. Our method achieved the greatest performance with an average accuracy of 90% and the average error of 6.6°. The standard deviation of the average error was fairly low. It is evidential that our MSHF improved the head pose estimation with great robustness. In future, we plan to investigate on-line learning methods to achieve real-time estimation by integrating head movement tracking.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61602429 and 61401188), China Postdoctoral Science Foundation (No. 2016M592406), and Research Funds of CUG from the Colleges Basic Research and Operation of MOE (No. 26420160055).

References

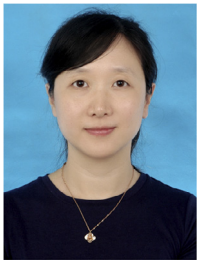
[1] E. Murphy-Chutorian, M. Trivedi, Head pose estimation in computer vision: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2009) 607–626.
[2] H. Kim, M. Sohn, D. Kim, S. Lee, Kernel locality-constrained sparse coding for head pose estimation, IET Comput. Vis. 10 (8) (2016) 828–835.
[3] S. Wu, M. Kan, Z. He, S. Shan, X. Chen, Funnel-structured cascade for multi-view face detection with alignment-awareness, Neurocomputing 221 (2017) 138–145.
[4] D. Zhu, X. Ramanan, Face detection, pose estimation and landmark localization in the wild, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2879–2886.
[5] M. Jones, P. Viola, Fast Multi-view Face Detection, Mitsubishi Electric Research Lab TR-20003-96 3 (2003) 14.
[6] Z. Zhang, P. Luo, C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: Proceedings of European Conference on Computer Vision, Zurich, 2014, pp. 94–108.
[7] H. Li, Z. Lin, X. Shen, J. Brandtz, G. Huay, A convolutional neural network cascade for face detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 5325–5334.
[8] J. Orozco, S. Gong, T. Xiang, Head pose classification in crowded scenes, in: Proceedings of British Machine Vision Conference, London, UK, 2009, pp. 1–3.
[9] B. Ma, A. Li, X. Chai, S. Shan, Covga: a novel descriptor based on symmetry of regions for head pose estimation, Neurocomputing 143 (2014) 97–108.
[10] R. Jafri, H. Arabnia, A survey of face recognition techniques, J. Inf. Process. Syst. 5 (2) (2009) 41–68.
[11] B. Heisele, T. Serre, T. Poggio, A component-based framework for face detection and identification, Int. J. Comput. Vis. 74 (2) (2007) 167–1811.
[12] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained partbased models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[13] S. Farfade, M. Saberian, L. Li, Multi-view face detection using deep convolutional neural networks, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ACM Shanghai, China, 2015, pp. 643–650.
[14] N. Gourier, J. Maisonnasse, D. Hall, Head pose estimation on low resolution images, Proceedings of International Evaluation Workshop on Classification of Events, Activities and Relationships (2007) 270–280.
[15] J. Wu, M. Trivedi, A two-stage head pose estimation framework and evaluation, Pattern Recognit. 41 (3) (2008) 1138–1158.
[16] X. Peng, J. Huang, Q. Hu, S. Zhang, A. Elgammal, D. Metaxas, From circle to 3-sphere: head pose estimation by instance parameterization, Comput. Vis. Image Underst. 136 (2015) 92–102.
[17] M. Zhang, K. Li, Y. Liu, Head pose estimation from low-resolution image with hough forest, in: Proceedings of IEEE Conference on Chinese Conference on Pattern Recognition, Chongqing, China, 2010, pp. 1–5.
[18] Y. Liu, J. Chen, Z. Shu, Z. Luo, L. Liu, K. Zhang, Robust head pose estimation using dirichlet-tree distribution enhanced random forests, Neurocomputing 173 (2016) 42–53.
[19] C. Huang, X. Ding, C. Fang, Head pose estimation based on random forests for multiclass classification, in: Proceedings of IEEE International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 934–937.
[20] P. Dollr, C. Zitnick, Structured forests for fast edge detection, in: Proceedings of IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 1841–1848.
[21] Y. Lu, X. Hou, X. Chen, A novel travel-time based similarity measure for hierarchical clustering, Neurocomputing 173 (2016) 3–8.
[22] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of International Conference on Machine Learning, San Francisco, CA, USA, 2001, pp. 282–289.
[23] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large margin methods for structured and interdependent output variables, J. Mach. Learn. Res. 6 (9) (2005) 1453–1484.
[24] B. Glocker, O. Pauly, E. Konukoglu, Joint classification-regression forests for spatially structured multi-object segmentation, in: Proceedings of Europe Conference on Computer Vision, Florence, Italy, 2012, pp. 870–881.
[25] P. Kontschieder, S.R. Bul, H. Bischof, Structured class-labels in random forests for semantic image labelling, in: Proceedings of IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2190–2197.
[26] A. Tejani, D. Tang, R. Kouskouridas, T. Kim, Latent-class hough forests for 3d object detection and pose estimation, in: Proceedings of European Conference on Computer Vision, Zurich, 2014, pp. 462–477.
[27] X. Zhu, D. Ramanan, Face detection,pose estimation and landmark localization in the wild, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2012) 2879–2886.
[28] J. Gall, V. Lempitsky, Class-specific hough forests for object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, FL, USA, 2009, pp. 143–157.
[29] N. Gourier, D. Hall, J. Crowley, Estimating face orientation from robust detection of salient facial features in pointing, Proceedings of International Conference on Pattern Recognition Workshop on Visual Observation of Deictic Gestures (2004) 1379–1382.
[30] G. Huang, M. Ramesh, T. Berg, E. Learned-Miller, in: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical report, University of Massachusetts Amherst, 2007, pp. 07–49.
[31] G. Xin, Y. Xia, Head pose estimation based on multivariate label distribution, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Ohio, USA., 2014, pp. 1837–1842.
[32] M. Garca-Montero, C. Redondo-Cabrera, R. Lpez-Sastre, T. Tuytelaars, Fast head pose estimation for human computer interaction, in: Proceedings of Iberian Conference on Pattern Recognition and Image Analysis, Springer Santiago de Compostela, Spain, 2015, pp. 101–110.
[33] K. Sundararajan, D. Woodard, Head pose estimation in the wild using approximate view manifolds, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 2015, pp. 50–58.

[34] M. Torki, A. Elgammal, Regression from local features for viewpoint and pose estimation, in: Proceedings of IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2603–2610.

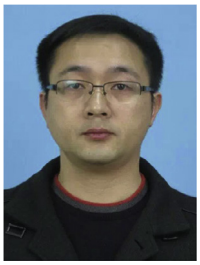
Yuanyuan Liu received B.E. degree from NanChang University, NanChang, China, in 2005, M.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2007, and Ph.D. degree from Central China Normal University. She is currently a lecturer in China University of Geosciences. Her research interests include image processing, computer vision and pattern recognition.

Zhong Xie received B.E, M.E and Ph.D. degree from Xincai Wu in the China University of Geosciences, Wuhan, China, in, respectively, 1990, 1998 and 2002. Now, he is a professor in the Faculty of Information Engineering, China University of Geosciences. His research interests are 3D rebuilding and spatial analysis, image processing.

Xiaohui Yuan received the B.S. degree in electrical engineering from the Hefei University of Technology, Hefei, China in 1996 and the Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA in 2004. He is currently an Associate Professor at the Department of Computer Science and Engineering in the University of North Texas. His research interests include computer vision, data mining, machine learning, and artificial intelligence. His research findings have been reported in over one hundred peer-reviewed papers. Dr. Yuan is a recipient of Ralph E. Powe Junior Faculty Enhancement award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013.



Jingying Chen received B.E and M.E degree from Huazhong University of science and technology, Wuhan, China, in, respectively, 1996 and 1998, and Ph.D. degree in Nanyang Technological University, Singapore. Now, she is a professor in the National Engineering Research Center for E-Learning, Central China Normal University. Her research interests are Computer vision and pattern recognition, image processing.



Wu Song is a Ph.D.student at the National Engineering Research Center for E-Learning (NERCEL), Central China Normal University. He received his B.S. degree in Technology of Computer Application from National University of Defense Technology, Changsha, China in 1998 and his M.S. degree in Technology of Computer Application from Yunnan University, Kunming, China in 2004. His research interests include video processing, data mining, and e-learning.