Joint head pose and facial landmark regression from depth images

Jie Wang¹, Juyong Zhang¹ (✉), Changwei Luo¹, and Falai Chen¹

© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract This paper presents a joint head pose and facial landmark regression method with input from depth images for realtime application. Our main contributions are: firstly, a joint optimization method to estimate head pose and facial landmarks, i.e., the pose regression result provides supervised initialization for cascaded facial landmark regression, while the regression result for the facial landmarks can also help to further refine the head pose at each stage. Secondly, we classify the head pose space into 9 sub-spaces, and then use a cascaded random forest with a global shape constraint for training facial landmarks in each specific space. This classification-guided method can effectively handle the problem of large pose changes and occlusion. Lastly, we have built a 3D face database containing 73 subjects, each with 14 expressions in various head poses. Experiments on challenging databases show our method achieves state-of-the-art performance on both head pose estimation and facial landmark regression.

Keywords head pose; facial landmarks; depth images

1 Introduction

Estimation of human head pose and detection of facial landmarks such as eye corners, nose tip, mouth, and chin are of central importance to facial animation [1], expression analysis [2], and face recognition, etc. These two problems have been separately studied for many years [3–7], with significant progress for images [8–11]. However, image-based methods are always subject to illumination and pose angle variations [12], which lead to many limitations. In recent years, with

the development of RGBD technology like Microsoft Kinect and Intel Realsense, head pose estimation and facial landmark detection methods based on depth data have attracted more and more attention due to the rich geometric information in depth images.

In recent years, cascaded regression approaches have been successfully applied to pose estimation [13, 14] and facial landmark regression [15–17]. Initialization is a key factor in such regression methods. If the initial value lies far from the actual shape, the final regression result will be not satisfactory. In existing methods, mean shape is often used for initialization [13, 18], but it often fails in challenging cases with extreme head pose or partial occlusion. Tracking methods [7, 19, 20] can handle the problems caused by large pose variation and occlusion, and achieve fast speed at the same time, but leaving problems such as drift and initialization to be overcome. The main problem in depth images are noise and even missing data, so a good initial value is quite important for facial landmark detection. On the other hand, the structural information provided by facial landmarks [21] is a major clue for head pose estimation. Therefore, joint head pose and facial landmark regression can be helpful for mutual optimization and improving the accuracy of both tasks.

In this paper, we propose a joint optimization approach which regresses the head pose and facial landmarks together via the random forest framework. Figure 1 shows one result of joint 3D head pose and facial landmark regression. The main contributions of our approach are as follows:

- **Joint optimization.** The pose regression result provides supervised initialization for cascaded facial landmark regression, while the estimated

¹ University of Science and Technology of China, Hefei, Anhui, 230026, China. E-mail: juyong@ustc.edu.cn (✉).
Manuscript received: 2017-01-16; accepted: 2017-03-08

Fig. 1 Joint 3D head pose and facial landmark regression. Red points represent the ground truth shape while blue points represent the estimated shape. The orange axis indicates the head orientation.

facial landmarks can also help to refine the head pose in each stage. The head pose and facial landmarks are jointly optimized stage by stage, and iteratively converge to optimum results.

- **Classification-guided.** Unlike existing methods, which train one general model for a variety types of input, our proposed method divides the input data into 9 types according to head orientation, and trains a specific model for each type. In this way, our classification-guided facial landmark regression method achieves more accurate results than existing methods, and it can handle some challenging cases including large pose variations and occlusion.
- **A new database.** Since depth images of faces with well labelled head pose and facial landmarks are rarely available, we have constructed a database of such data with 73 persons, each with 14 different expressions in various head poses. It contains 130,000 well labelled depth images.

The remainder of this paper is structured as follows: Section 2 briefly introduces related work. In Section 3, we present our method of head detection, head pose estimation, and facial landmark localization. Experimental results are given in Section 4, and we conclude the paper in Section 5.

2 Related work

2.1 Head pose estimation

Fanelli et al. [22–24] adopted a voting method to directly determine head pose. However, their feature selection method for depth images degenerates into using 2D features, i.e., the RGB information used in 2D images was replaced by xyz -coordinate values in depth images. Tan et al. [20] pointed out that

when the head is translated along the camera's viewing direction, 2D features cannot determine the relationships between the object's projective view and the corresponding transformation parameters. Therefore, they captured subject-specific structures by combining a generalized model-based tracker with an online learning method. Papazov et al. [25] also used a random forest-based framework, in a similar way to the methods in Refs. [22–24]. They replaced depth features by more elaborate triangular surface patch (TSP) features to ensure view-invariance. Their result achieved better results, but with more computation because of the calculation of TSP features.

2.2 Facial landmark localization

Active appearance methods (AAMs) [26–30] fit the input image by minimizing the error between the reconstructed texture model and the input image. These methods always require global features, and thus lose details in some cases. Thus we mainly consider methods based on independent features, which lie in small patches around each landmark, such as active shape models (ASMs), constrained local models (CLMs), or models implicitly learned from the data by machine learning or deep learning. Ruiz and Illingworth [31] extended ASMs to 3D shape for automatic localization of landmarks. Their model was adapted to individual faces via a guided search, which guided the specific shape index models to match local surface patches. Gilani et al. [32] detected a large number of anthropometric landmarks in 3D faces, in common with Ruiz and Illingworth. Instead of using simple geometric matching, their algorithm builds dense correspondences. Baltrušaitis et al. [19] extended the CLMs to 3D (CLM-Z) for robust facial feature tracking under varying pose. Since methods based on direct shape-matching are often time-consuming, decisions from the local models are always fused with a global shape model. Jourabloo and Liu [33, 34] estimated both 2D and 3D landmarks and their 2D visibilities for a face image with an arbitrary pose.

3 Method

The pipeline and structure of our method are presented in Fig. 2. In order to facilitate our exposition, two concepts frequently used in this

Fig. 2 Pipeline of our method: (a) head detection, (b) head pose estimation, (c) supervised initialization for cascaded facial landmark localization, (d) pose-space chosen in cascaded regression phase, and (e) cascaded facial landmark localization via classification-guided approach.

paper are firstly explained.

Bounding box: The bounding box of a point cloud is defined as the smallest 3D cuboid which contains it, measured in terms of cuboid volume. For example, the bounding box in Fig. 3(c) is a blue cuboid. The bounding box of the reference face template shown in Fig. 6 is defined as R_b in our method.

3D patch: Just as a patch always represents a region in a 2D image, a 3D patch is a region that belongs to a 3D point cloud. See, e.g., Fig. 4: the red, green, and blue regions are all 3D patches.

3.1 Face detection

The discriminative random regression forests (DRRF) [23] method can be used to discriminate depth patches that belong to a head to ensure that only such patches are used to predict head localization and head orientation. We use the DRRF method only in order to estimate head localization. Remarkably, the head localization is defined as the barycenter of the head point cloud, which we call the head center in the following. We illustrate the head detection process in Fig. 3. We first estimated the head center by DRRF as the center of a sphere with an empirical fixed radius of 60 mm (small blue sphere). We then compute the mean of the points

Fig. 3 Head detection: (a) fixing the head center and generating the red sphere, (b) head point cloud, and (c) bounding box of the head point cloud.

Fig. 4 Feature selection. Top: feature used in head pose estimation. Middle, bottom: two kinds of features used in facial landmark localization framework. Note that the latter features are well aligned.

which lie within the blue sphere, setting this as the center (black point), and compute a further sphere of radius $d = 75$ mm (red sphere). All points which lie within the red sphere points are found; they constitute the head point cloud. Its bounding box is then determined.

3.2 Feature selection

Feature selection is a critical factor in the random forest framework [35–38]. A good feature selection method can help to appropriately classify the training samples in each non-leaf node, so we firstly consider the feature selection method before head pose regression and facial landmark localization.

3.2.1 Feature selection for pose regression

In head pose estimation phase, each tree in the forest is constructed from a set of 3D patches $\{L_i\}_{i=1}^N$ randomly sampled from the training samples. The scale of each L_i depends on the scale of the facial bounding box to which it belongs. Since each L_i is also a group of 3D points, we can get its bounding box B_i . At each non-leaf node, starting from the

root, suppose there are M 3D patches (e.g., the red 3D patches in the top row of Fig. 4), each with its corresponding bounding box $B_j, j = 1, \dots, M$. In order to divide these M 3D patches into two categories, and ensure that the difference between the 3D patches that fall into the same child node is as small as possible, we randomly choose K pairs of small 3D patches $\{l_k^1, l_k^2\}_{k=1}^K$ within R_b , and map them to each B_j with the following equation:

$$A(\text{Box}(l_{kj}^c)) = A(B_j) \frac{A(\text{Box}(l_k^c))}{A(R_b)}, \quad c = 1, 2 \quad (1)$$

where $\text{Box}(l)$ represents the bounding box of l , and A represents one attribute of the bounding box, i.e., its length, height, or width; in addition, when locating a small cube in a bounding box, A also represents the distance between each side of this cube and the corresponding side of the big bounding box it belongs to. Each pair of blue and green parts in Fig. 4 shows a pair of small 3D patches that belong to $\text{Box}(l_{kj}^c)$ for specific k, l , and c . It should be noted that R_b is only used as an arbitrary reference, within which we can choose 3D feature patches: we could use any other cube instead of R_b . Then, we can compute the feature difference $D(l_{kj}^1) - D(l_{kj}^2)$ for the j th training sample, where $D(l)$ is the mean depth of the point cloud of the 3D patch l .

3.2.2 Feature selection for facial landmark regression

In the facial landmark regression phase, each tree in the forest is constructed from training samples (a set of head point clouds). When using a cascaded random forest to locate facial landmarks, we hope that the selected features for each landmark are unaffected by the head pose. Consider Fig. 4, where each row corresponds to one feature. We can see that each pair of features has similar localization in different samples. Denote the transformation of the estimated head pose by R_t ; R_t^{-1} is its inverse. The influence of head pose on feature selection can be eliminated by transforming each head point cloud by R_t^{-1} . Having chosen features around each landmark in the template, these features are put into correspondence with each transformed head point cloud using the scale transformation between the bounding box of the template and that of the head point cloud. The feature difference is computed as $D(R_t^{-1}(l_i^1)) - D(R_t^{-1}(l_i^2))$, where $D(R_t^{-1}(l))$ is the mean depth of the point cloud in a 3D patches l .

3.3 Head pose estimation

Our supervised initialization for the cascaded random forest is based on an accurately estimated head pose, using a random forest framework. The training database are pre-processed, and annotated with ground truth poses. In this paper, we use axis-angle representation to indicate the head orientation, and the position of the head is represented by the tip of the nose. Compared to using Euler angles, axis-angle representation makes it easier to represent the direction vector of the head orientation, and we do not need to consider the problem of gimbal lock which arises in Euler representation. Head pose P can be denoted by a 6-dimensional vector $P = (\psi_x, \psi_y, \psi_z, t_x, t_y, t_z)$, where (t_x, t_y, t_z) is the location of the tip of the nose, while the head orientation is given by (ψ_x, ψ_y, ψ_z) . The orientation can be decomposed into the form: $(\psi_x, \psi_y, \psi_z) \Leftrightarrow ((e_x, e_y, e_z), \theta) \Leftrightarrow (\theta e_x, \theta e_y, \theta e_z)$, where (e_x, e_y, e_z) is a rotation axis encoded as a unit vector, and θ is the rotation angle.

3.3.1 Training

Beginning with a set of depth images, we can get the corresponding head point clouds using the method in Section 3.1. Each tree in the forest is constructed from a set of 3D patches $\{L_i, \Theta_i\}_{i=1}^N$ randomly sampled from these head point clouds, where $\Theta_i = \{\psi_{ix}, \psi_{iy}, \psi_{iz}, \Delta t_{ix}, \Delta t_{iy}, \Delta t_{iz}\}$ contains the head orientation $\{\psi_{ix}, \psi_{iy}, \psi_{iz}\}$, and $\{\Delta t_{ix}, \Delta t_{iy}, \Delta t_{iz}\}$ is the offset between the i th 3D patch's center and the ground truth localization of the tip of the nose.

We define the binary tests in each non-leaf node as $\beta_{L_i, \tau}$:

$$D(l_i^1) - D(l_i^2) > \tau \quad (2)$$

where $D(l)$ is the mean depth of the points in the 3D patch l , and l_i^1, l_i^2 belong to L_i , as described in Section 3.2. τ is a threshold. Using mean depth instead of a single point makes the feature less sensitive to noise.

Each non-leaf node chooses the best splitting function β^* out of $\{\beta^k\}$ by minimizing:

$$E = \frac{|L|}{|L| + |R|}(\sigma_L^\psi + \sigma_L^{\Delta t}) + \frac{|R|}{|L| + |R|}(\sigma_R^\psi + \sigma_R^{\Delta t}) \quad (3)$$

where $|L|$ and $|R|$ are the number of training samples that are routed to the left and right child nodes according to $\beta_{L_i, \tau}$. σ^ψ and $\sigma^{\Delta t}$ are the traces of covariance matrices of the rotation angles and

offsets. After choosing the best binary β^* in one non-leaf node, the training samples in this node can be split using Eq. (2). This process is iterated until a leaf is created when either the maximum tree depth is reached, or fewer than a certain number of 3D patches are left. When the training process is finished for one tree, each leaf node stores several training samples with their ground truth labels $\{\Theta_i\}$. We compute the mean vector and the corresponding determinant of the covariance matrix Σ of these ground truth labels and save them in each leaf node.

Equation (3) ensures that the differences of samples that fall into the same child node is as small as possible, while the difference of samples in different child nodes is as large as possible. We compare our entropy in Eq. (3) with the classical information gain function [23] in Eq. (12) in Section 4.

3.3.2 Testing

In the testing phase, the 3D patches $\{(L_k, \Theta_k^0)\}_{k=1}^K$ are densely extracted from one testing head point cloud, each with the label $\Theta_k^0 = \{0, 0, 0, c_{kx}, c_{ky}, c_{kz}\}$, $k = 1, \dots, K$. $\{c_{kx}, c_{ky}, c_{kz}\}$ is the center localization of the 3D patch L_k in the real world (camera coordinates). All of these $\{(L_k, \Theta_k^0)\}_{k=1}^K$ are sent into the forest, the binary tests stored in the non-leaf nodes guiding each 3D patch all the way to a leaf node of each tree in the forest. After all 3D patches have travelled over the forest, a set of leaves with their mean vectors and determinants of their covariance matrices are left as a basis for predicting the head pose. However, not all leaves can be considered for pose regression: only those whose covariance matrix determinant is less than the maximum allowed determinant should be taken into account. Suppose that $\bar{\Theta}^m = \{\bar{\psi}_x^m, \bar{\psi}_y^m, \bar{\psi}_z^m, \bar{\Delta t}_x^m, \bar{\Delta t}_y^m, \bar{\Delta t}_z^m\}$ is the mean vector for head pose stored in the m th leaf, and the 3D patch L_k lies in this leaf. Then, the head pose predicted by the chosen leaf and 3D patch L_k can be expressed as

$$P_k^m = \{\bar{\psi}_x^m, \bar{\psi}_y^m, \bar{\psi}_z^m, c_{kx} + \bar{\Delta t}_x^m, c_{ky} + \bar{\Delta t}_y^m, c_{kz} + \bar{\Delta t}_z^m\} \quad (4)$$

In order to further reduce the influence of outliers, we perform a k -means algorithm on the remaining leaves, and the cluster with the largest number of leaves is used for determining head pose.

3.4 Pose-based 3D facial landmark localization

3.4.1 Pose classification

Our classification-guided method divides the samples into 9 pose views (upper-left, upper-centre, upper-right, left, center, right, lower-left, lower-center, lower-right) and trains 9 models in each sub-space respectively. The training samples are split into these 9 sub-spaces before the phase of cascaded facial landmark regression, and the training phase is carried out in each sub-space in turn. When testing with a new sample, we must firstly determine which category it belongs to. The distance in each sub-space is measured with Gaussian probabilities: the orientation space is discretized into disjoint sets $\{P_i\}_{i=1}^9$, and for an estimated head pose $\mathcal{P}$, the Gaussian kernel between $\mathcal{P}$ and P_i is computed as

$$g(\mathcal{P}, P_i) = \frac{1}{2\pi\varsigma^2} \exp(-\|\mathcal{P} - w_i\|^2 / 2\varsigma^2) \quad (5)$$

where w_i is the centroid of P_i , and ς is a bandwidth parameter. We choose view space P_i which has the highest Gaussian probability when estimating the facial landmarks.

3.4.2 Supervised initialization via head pose

In Section 3.3, we presented our method of head pose estimation. Our estimated result is accurate, and as a result, provides important and helpful initialization information at the beginning of the cascaded shape regression phase.

Supervised initialization is based on estimated head pose and a mean template marked with 33 facial landmarks, as shown in Fig. 6. The estimated head orientation is firstly transformed by rotation matrix R_t , then the 33 facial landmarks in the mean template are rotated by R_t and translated to the estimated nose tip. Thus, a coarse initialization of facial landmarks is given by the transformed facial landmarks on the template. In order to refine the initial face shape, we further move the face shape along the direction of the head orientation using a rigid translation, so that the initial location of the nose tip is moved to one point belonging to the head point cloud.

3.4.3 Joint head pose and facial landmark regression

We now present our facial landmark localization method in one specific pose space. An n -point shape in 3D can be defined by concatenating all points'

coordinates into a $k = m \times 3$ vector:

$$S = [x_1, y_1, z_1, \dots, x_m, y_m, z_m]^T \quad (6)$$

where (x_j, y_j, z_j) represents the j th landmark. Beginning with an initial shape S_0 and the target shape $\tilde{S}$, the main idea of cascaded shape regression is to obtain a value R^t at stage t , so that by a certain stage T :

$$S_0 + \sum_{t=0}^T R^t \rightarrow \tilde{S} \quad (7)$$

In machine learning, given a training database with N samples $\{F_i, \tilde{S}_i\}_{i=1}^N$, where $\{F_i\}$ is a set of training samples and $\{\tilde{S}_i\}$ is the corresponding ground truth shape, Eq. (7) is typically satisfied by obtaining a set of regressors $(R^1, \dots, R^t, \dots, R^T)$ by solving:

$$R^t = \arg \min_R \sum_{i=1}^N \|\Delta \tilde{S}_i^t - R^t(F_i, S_i^{t-1}, P_i^{t-1})\|_2^2 \quad (8)$$

where $\Delta \tilde{S}_i^t = \tilde{S}_i - S_i^{t-1}$, and S_i^{t-1} and P_i^{t-1} are the estimated shape and pose of the i th sample at stage $t-1$. Ren et al. [18, 39] specified R^t as $R^t(F, S^{t-1}) = W^t \Phi^t(F, S^{t-1})$. But when placing a shape S in 3D, an additional global transformation is needed, and it can be represented by head pose at each stage. Thus, we define $R^t(F, S^{t-1}, P^{t-1}) = W^t \Phi^t(F, S^{t-1}, P^{t-1})$ in our method, where Φ^t is a feature mapping function which can be expressed by a set of leaves from the random forest, while W^t is a linear regression matrix that stores the displacement in each corresponding leaf node. The regression of Φ^t and W^t are solved separately.

Learning with Φ^t . When learning Φ^t using Eq. (8), we begin with N training samples $\{F_i, S_{i0}, P_{i0}, \tilde{S}_i\}_{i=1}^N$, where F_i is the i th head point cloud with its estimated head pose P_{i0} , while S_{i0} and $\tilde{S}_i$ are the initial shape (see Section 3.4.2) and the ground truth shape for the i th sample respectively. When training one tree using these samples, a group of 3D patches $\{L_k\}$ is chosen for each non-leaf node as described in Section 3.2, while the binary test in each non-leaf node is given by

$$D(R_t^{-1}(l_k^1)) - D(R_t^{-1}(l_k^2)) > \tau \quad (9)$$

The feature and binary test that give rise to minimum entropy 3 are chosen to split the samples in each node. The training process for one tree is iterated until a leaf is created when either the maximum tree depth is reached, or fewer than a certain number of training samples are left.

After training the random forest has finished after

the t th stage, each component of $\phi_j^t(F_i, S_i^t, P_i^t)$ for an arbitrary sample $\{F_i, \tilde{S}_i\}$ is determined to be either 0 or 1 by passing it through the random forest, where 1 indicates the leaves contain this sample while 0 indicates otherwise. All the ϕ_j^t are concatenated to form a high-dimensional binary feature Φ^t : $\Phi^t = [\phi_1^t, \dots, \phi_{33}^t]$, where ϕ_j ($j = 1, \dots, 33$) are the local binary features for each landmark. Let D_j represent the total number of leaf nodes in the random forest specified by the j th landmark. Then ϕ_j is a D_j -dimensional binary vector, and W^t that stores the displacement in each corresponding leaf node is a 3-by- $(\sum_{j=1}^{33} D_j)$ matrix in which each column is a 3D vector.

Learning with W^t . Since W^t does not contain global shape information, we re-computed W^t with the estimated Φ^t by performing the following minimisation:

$$\min_{W^t} \sum_{i=1}^N \|\Delta \tilde{S}_i^t - W^t \Phi^t(F_i, S_i^{t-1}, P_i^{t-1})\|_2^2 + \lambda \|W^t\|_2^2 \quad (10)$$

Variable update in each stage. After building the forest in stage t , we can get the local binary functions $\Phi^t = [\phi_1^t, \dots, \phi_{33}^t]$ for each training sample and the global linear regression W^t as described above. Then we update the face shapes of all training samples using:

$$\begin{cases} \Delta S_i^t = W^t \Phi^t(F_i, S_i^{t-1}, P_i^{t-1}) \\ S_i^t = S_i^{t-1} + \Delta S_i^t \\ P_{i(t)}^t = S_{i(t)}^t, P_{i(\psi)}^t = Q^t(S_i^{t-1}, S_i^t) \odot P_{i(\psi)}^{t-1} \end{cases} \quad (11)$$

where $P_{i(t)}^t$ and $P_{i(\psi)}^t$ represent the translation and orientation of head pose at stage t , and $Q^t(S_i^{t-1}, S_i^t)$ is a transformation for aligning the inner eye vector at stage $t-1$ to that in stage t . Thus, the pose orientation is updated along with the transformation for the inner eye vector in two adjacent stages, while $\odot$ represents the effect of $Q^t(S_i^{t-1}, S_i^t)$ on $P_{i(\psi)}^{t-1}$. Thus, we have built a random forest with global shape constraint in one step. Starting from the supervised initialization for facial shapes, Fig. 5 illustrates one result of facial landmark localization stage by stage.

4 Experiments

4.1 Experimental setup

We performed training and testing on a Windows 64-bit desktop with 4 GB RAM and a 3.3 GHz

Fig. 5 Cascaded facial landmark localization. Starting from the initial guess, the locations of the landmarks are gradually optimized.

Intel Core i5 CPU. We mainly used three databases. Two were from the ETHZ-Computer Vision Lab [22–24] and available online, the BIWI Kinect Head Pose database (BIWI) and the BIWI 3D Audiovisual Corpus of Affective Communication database (B3D(AC)²). In order to test our algorithm on different kinds of head pose and facial expression, we constructed our own 3D Faces with Expression and Pose database (3DFEP).

4.1.1 ETHZ databases

The BIWI database contains 24 sequences acquired with a Kinect sensor. The ground truth for the head pose indicates the head center in 3D space and the head orientation encoded as a 3×3 rotation matrix. In this paper, we acquire the ground truth localization of the nose tip with the aid of iterative closest point (ICP) algorithm, and the head orientation is transformed to axis-angle representation.

The B3D(AC)² database contains 14 persons, each captured in 40 poses using a 3D scanner. We manually marked 33 landmarks on the mean template, as shown in Fig. 6. We mapped these marked points to each sequence to obtain ground truth facial landmarks, while the ground truth head orientation in axis-angle representation and the position of nose tip were computed at the same time.

4.1.2 3DFEP database

In order to test our algorithm on different kinds of

Fig. 6 Ground truth facial landmarks for one example in the B3D(AC)² database.

head pose and facial expression, we built a database with 73 persons, each with 14 expressions in various head poses. We scanned the subjects using an Intel Realsense sensor. The format of the depth data and RGB images are the same as that used to encode the BIWI database. To provide ground truth head poses and facial landmarks, we used 3D face modeling and tracking techniques, and poor calibration results were eliminated by checking each one in turn. Again, 33 landmarks are used, and the ground truth of the head pose is also given by axis-angle representation and nose tip.

4.2 Head pose estimation

Our proposed head pose estimation method was evaluated on the above three databases. Various fixed parameters are given in Table 1. Further parameters were: maximum variance to eliminate

Table 1 Parameters used for training

Database	Training samples	Features in each non-leaf node	Binary tests	Maximum tree depth	Minimum samples in each leaf node	Trees
BIWI	8041	520	40	14	20	10
B3D(AC) ²	435832	1040	10	8	20	10
BIWI + B3D(AC) ²	23527	520	40	13	20	10
3DFEP	6743	1040	40	15	20	10

the stray leaf node: 300, ratio of large 3D patch to face bounding box: 0.7, ratio of sub-3D patch to face bounding box: 0.3, number of large 3D patches sampled from each head point cloud: 25.

Figure 7(left) shows the relationship between the average error in head pose and the depth of tree. Error reduces as tree depth increases. Figure 7(middle) shows the relationship between accuracy and depth of tree. Accuracy increases with depth of tree. Figure 7(right) shows that as the maximum depth of the tree increases, the number of samples in each leaf node reduces, while the consumption of time shows slow growth. All experiments in Fig. 7 were done with the BIWI database.

Table 2 is in three parts. The first four rows show head pose estimation results using our method with different databases. For each database, we use 70% of the samples for training and the remaining

30% for testing. Note that the head pose estimation result for the B3D(AC)² database is more accurate than for the other databases, because all samples in B3D(AC)² database belong to the same pose-space. We also trained a head pose estimation model by combining the BIWI and B3D(AC)² databases; the results were similar to those obtained by only using the BIWI database. The results in the next three rows of Table 2 show that by using axis-angle representation, the head pose estimation is more accurate than using Euler angle form directly. Moreover, using entropy (Eq. (3)) is more effective than using information gain [23]:

$$I = \log(|\Sigma^v| + |\Sigma^a|) - \sum_{i=\{L,R\}} w_i \log(|\Sigma_i^v| + |\Sigma_i^a|) \quad (12)$$

in the splitting phase; here Σ^v and Σ^a are the covariance matrices of the offset and rotation angles. The splitting process in each non-leaf node is more stable using entropy, preventing extreme divisions,

Table 2 Head pose estimation results. The first four rows give results using different databases and our method; the associated training parameters are given in Table 1. The next three rows show results for head pose by transforming the estimated pose from axis-angle representation to Euler angle form ($A \rightarrow E \& N$). We also compare results with different loss functions: entropy and information gain. The last four rows give head pose estimation results from other methods

Database	Method	ψ_x/yaw (1/(°))	ψ_y/pitch (1/(°))	ψ_z/roll (1/(°))	t_x (mm)	t_y (mm)	t_z (mm)
BIWI	Our method	0.0237	0.0283	0.0162	1.56	1.56	1.40
B3D(AC) ²	Our method	0.0116	0.0193	0.0025	0.66	1.00	0.51
BIWI + B3D(AC) ²	Our method	0.0241	0.0294	0.0092	1.35	2.02	1.32
3DFEP	Our method	0.0419	0.0465	0.0163	3.04	3.06	2.42
BIWI	E & C	1.18°	1.40°	1.61°	1.65	1.43	1.31
BIWI	A $\rightarrow$ E & N	1.16°	1.30°	1.51°	1.91	1.47	1.55
BIWI	E & N with InfoGain	9.23°	12.21°	5.49°	15.88	7.84	6.61
BIWI	[20] (generalized)	1.9°	1.3°	1.0°	1.3	3.1	1.8
BIWI	[7] (CT)	4.7°	7.7°	5.3°	—	—	—
BIWI	[24] (DRRF)	8.9°	8.5°	7.9°	—	14.6	—
BIWI	[25] (TSP)	2.5°	1.8°	2.9°	—	6.8	—

Fig. 7 Left: relationship between average error of each head pose component and depth of tree. The left-hand vertical axis measures the positional error of the nose (shown as solid lines) while the right-hand vertical axis represents the orientation error (dotted lines). Center: relationship between accuracy and depth of tree. Orange: ratio of samples whose position error satisfies $[t_x < 5, t_y < 4, t_z < 4]$. Blue: ratio of samples whose position error satisfies $[t_x < 10, t_y < 10, t_z < 10]$. Right: Relationship between depth of tree and time.

Table 3 Estimation errors for facial landmark localization for B3D(AC)² and 3DFEP databases. M-initial: cascaded facial landmark localization with mean shape as initialization. V-based: cascaded facial landmark localization with position aware shape as initialization (Unit: mm)

Landmark	B3D(AC) ²			3DFEP	Landmark	B3D(AC) ²			3DFEP
	Fanelli	M-initial	V-based			Fanelli	M-initial	V-based	
LefEyeU	—	3.73	1.10	5.69	RigNoseUC	—	3.87	1.15	5.17
LefEyeL	4.4	3.73	1.20	5.94	RigNoseU	—	3.71	1.07	5.25
LefEyeD	—	3.95	1.10	5.40	NoseTip	—	3.81	1.14	6.11
LefEyeR	3.4	3.67	1.09	6.44	OutMouthL	5.3	3.82	1.08	5.87
RigEyeU	—	3.75	1.35	5.31	OutMouthUL	—	3.75	1.09	4.91
RigEyeL	3.8	3.89	1.25	5.86	OutMouthU	3.5	3.74	1.21	5.50
RigEyeD	—	3.84	1.08	5.47	OutMouthUR	—	3.65	1.16	5.73
RigEyeR	4.8	3.62	1.30	4.96	OutMouthR	5.3	3.80	1.61	6.20
LefNoseU	—	3.69	1.19	5.40	OutMouthDR	—	3.69	1.36	5.52
LefNoseUC	—	3.79	1.08	5.93	OutMouthD	6.2	3.80	1.22	6.25
LefNoseDC	—	3.72	1.08	5.85	OutMouthDL	—	3.69	1.09	5.17
LefNoseD	3.3	3.57	1.56	4.79	InnMouthL	—	3.84	1.37	4.57
CenNoseL	—	3.69	1.38	6.02	InnMouthU	—	3.94	1.49	4.47
CenNoseC	—	3.79	1.08	4.95	InnMouthR	—	3.76	1.11	4.75
CenNoseR	—	3.72	1.18	6.04	InnMouthD	—	3.84	1.10	4.59
RigNoseD	2.9	3.60	1.48	6.04	Chin	—	3.58	1.22	6.62
RigNoseDC	—	3.66	1.54	5.52	—	—	—	—	—

e.g., in which almost all samples are divided into the same leaf node. The last four rows of Table 2 provides comparative publicly available results, all computed with a random forest framework using the BIWI database.

We also later show some head pose estimation results for the BIWI database in the case of many missing head data—see Fig. 10.

4.3 Facial landmark localization

We used the B3D(AC)² and 3DFEP databases to perform some facial landmark localization experiments.

Since all samples in the B3D(AC)² database belong to the same pose space, they need not be classified with respect to pose space. Errors of each facial landmark are computed in the Euclidean norm. Table 3 lists the errors for each landmark using three methods. V-based gives results of the pose-based facial landmark localization method. It shows that the maximum estimation error of facial landmarks is less than 1.6 mm. In order to show the importance of pose-based initialization in cascaded shape regression, in the M-initial method we only locate the mean shape in the bounding box as the initial shape of each training sample, while the other parameters in the training phase are held the same as for the pose-based method. The error is two to three

Fig. 8 Success with varying occlusion for the B3D(AC)² database. **Fig. 9** Success of facial landmark localization in different pose-spaces for the 3DFEP database.

times larger than the errors for the V-based method. We also compare our result with the results reported by Fanelli et al. [40] for the B3D(AC)² database.

Figure 8 shows the rate of success for different degrees of occlusion, using three thresholds for assessing success. We also show some facial landmark

Fig. 10 Robustness of the proposed method to occlusion. Top two rows: results of head pose estimation on the BIWI database. Next two rows: facial landmark results on the B3D(AC)² database. Red points: ground truth shape. Blue points: estimated shape. Blank areas in the depth images in the second and fourth rows represent parts removed from the input data. Last row: facial landmark localization results for large pose angles, using 3DFEP. To show face shapes clearly, we have adjusted the corresponding point cloud to a suitable view.

localization results for the B3D(AC)² database in the third and fourth rows of Fig. 10. It should be noted that the estimation results are accurate even with missing data samples.

We also tested our algorithm on the 3DFEP database built by ourselves, containing different kinds of head poses and face expressions. We choose 6867 training samples from 3DFEP and classify the head pose space into 9 categories. The facial landmark localization model was then trained in each sub-space. Figure 9 shows the success rate of facial landmark localization when the mean error threshold of facial landmarks is set to 7 mm. The dataset was divided based on the projection of the head unit direction vector along the x - and y -axes, in areas of

0.25×0.25 . The color-coding represents the amount of data contained in each angle range, as indicated by the color bar on the right. The performance decreases in the corner regions, where the rotations are very large or the number of images is low. The last row of Fig. 10 shows facial landmark localization results for the 3DFEP database. Our pose-based 3D facial landmark localization method achieves good results on this database. Table 3 shows the error of face shapes with 3DFEP in neutral-view-space. Since the ground truth shapes in our database are inherently noisy, the errors are larger than those for the B3D(AC)² database. When testing with the 3DFEP database and 9 training models, each training model contained 198 trees, and 15 layers for

each tree. The average time taken for each stage in the cascaded facial landmark regression phase was about 35 ms.

5 Conclusions

We have presented a joint 3D head pose and facial landmark regression method for depth images. Our head pose estimation result is accurate and robust even for low quality depth data; it also provides a *supervised initialization* for the cascaded facial landmark estimation. Moreover, the head pose can be further refined using the estimated facial landmarks at each stage with our *joint optimization* framework. Our cascaded regression is based on a *classification-guided* approach, which divides the original complex problem into several easier sub-problems. In this way, our facial landmark localization approach can handle challenging cases including large pose angles and occlusion. We have also built a 3D face *database* which contains 73 persons, each with 14 expressions in various head poses, which can be used to train models on different kinds of head poses and facial expressions. The experiments demonstrate that the proposed method achieves a significant improvement on head pose estimation and facial landmark localization compared to existing methods.

Acknowledgements

We thank Luo Jiang and Boyi Jiang for their help in constructing the 3DFEP database. We thank the ETHZ-Computer Vision Lab for permission to use the BIWI Kinect Head Pose database and BIWI 3D Audiovisual Corpus of Affective Communication database. This work was supported by the National Key Technologies R&D Program of China (No. 2016YFC0800501), and the National Natural Science Foundation of China (No. 61672481).

References

- [1] Cao, C.; Weng, Y.; Lin, S.; Zhou, K. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 41, 2013.
- [2] Cao, C.; Hou, Q.; Zhou, K. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 43, 2014.
- [3] Breitenstein, M. D.; Kuettel, D.; Weise, T.; van Gool, L.; Pfister, H. Real-time face pose estimation from single range images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8, 2008.
- [4] Meyer, G. P.; Gupta, S.; Frosio, I.; Reddy, D.; Kautz, J. Robust model-based 3D head pose estimation. In: *Proceedings of the IEEE International Conference on Computer Vision*, 3649–3657, 2015.
- [5] Paderleris, P.; Zabulis, X.; Argyros, A. A. Head pose estimation on depth based on particle swarm optimization. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 42–49, 2012.
- [6] Seeman, E.; Nickel, K.; Stiefelhagen, R. Head pose estimation using stereo vision for human–robot interaction. In: *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 626–631, 2004.
- [7] Tulyakov, S.; Vieri, R. L.; Semeniuta, S.; Sebe, N. Robust real-time extreme head pose estimation. In: *Proceedings of the 22nd International Conference on Pattern Recognition*, 2263–2268, 2014.
- [8] Burgos-Artizzu, X. P.; Perona, P.; Dollar, P. Robust face landmark estimation under occlusion. In: *Proceedings of the IEEE International Conference on Computer Vision*, 151–1520, 2013.
- [9] Cao, X.; Wei, Y.; Wei, F.; Sun, J. Face alignment by explicit shape regression. *International Journal of Computer Vision* Vol. 107, No. 2, 177–190, 2014.
- [10] Dantone, M.; Gall, J.; Fanelli, G.; van Gool, L. Real-time facial feature detection using conditional regression forests. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2578–2585, 2012.
- [11] Zhang, Z.; Zhang, W.; Liu, J.; Tang, X. Multiview facial landmark localization in RGB-D images via hierarchical regression with binary patterns. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 24, No. 9, 1475–1485, 2014.
- [12] Zhu, Z.; Martin, R. R.; Pepperell, R.; Burleigh, A. 3D modeling and motion parallax for improved videoconferencing. *Computational Visual Media* Vol. 2, No. 2, 131–142, 2016.
- [13] Dollár, P.; Welinder, P.; Perona, P. Cascaded pose regression. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1078–1085, 2010.
- [14] Sun, X.; Wei, Y.; Liang, S.; Tang, X.; Sun, J. Cascaded hand pose regression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 824–832, 2015.
- [15] Chen, D.; Ren, S.; Wei, Y.; Cao, X.; Sun, J. Joint cascade face detection and alignment. In: *Computer Vision—ECCV 2014*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer International Publishing Switzerland, 109–122, 2014.
- [16] Lee, D.; Park, H.; Yoo, C. D. Face alignment

- using cascade Gaussian process regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4204–4212, 2015.
- [17] Tzimiropoulos, G. Project-out cascaded regression with an application to face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3659–3667, 2015.
- [18] Ren, S.; Cao, X.; Wei, Y.; Sun, J. Face alignment at 3000 fps via regression local binary features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1685–1692, 2014.
- [19] Baltrušaitis, T.; Robinson, P.; Morency, L. P. 3D constrained local model for rigid and non-rigid facial tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2610–2617, 2012.
- [20] Tan, D. J.; Tombari, F.; Navab, N. A combined generalized and subject-specific 3D head pose estimation. In: Proceedings of the International Conference on 3D Vision, 500–508, 2015.
- [21] Borovikov, E. Human head pose estimation by facial features location. *arXiv preprint arXiv:1510.02774*, 2015.
- [22] Fanelli, G.; Dantone, M.; Gall, J.; Fossati, A.; van Gool, L. Random forests for real time 3D face analysis. *International Journal of Computer Vision* Vol. 101, No. 3, 437–458, 2013.
- [23] Fanelli, G.; Gall, J.; van Gool, L. Real time head pose estimation with random regression forests. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 617–624, 2011.
- [24] Fanelli, G.; Weise, T.; Gall, J.; van Gool, L. Real time head pose estimation from consumer depth cameras. In: *Pattern Recognition*. Mester, R.; Felsberg, M. Eds. Springer-Verlag Berlin Heidelberg, 101–110, 2011.
- [25] Papazov, C.; Marks, T. K.; Jones, M. Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4722–4730, 2015.
- [26] Cootes, T. F.; Edwards, G. J.; Taylor, C. J. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 23, No. 6, 681–685, 2001.
- [27] Cristinacce, D.; Cootes, T. Boost regression active shape models. In: Proceedings of the British Machine Conference, 79.1–79.10, 2007.
- [28] Sauer, P.; Cootes, T.; Taylor, C. Accurate regression procedures for active appearance models. In: Proceedings of the British Machine Vision Conference, 30.1–30.11, 2011.
- [29] Tzimiropoulos, G.; Pantic, M. Optimization problems for fast AAM fitting in-the-wild. In: Proceedings of the IEEE International Conference on Computer Vision, 593–600, 2013.
- [30] Xiao, J.; Baker, S.; Matthews, I.; Kanade, T. Real-time combined 2D+3D active appearance models. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 535–542, 2004.
- [31] Ruiz, M. C.; Illingworth, J. Automatic landmarking of faces in 3D-ALF^{3D}. In: Proceedings of the 5th International Conference on Visual Information Engineering, 41–46, 2008.
- [32] Gilani, S. Z.; Shafait, F.; Mian, A. Shape-based automatic detection of a large number of 3D facial landmarks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4639–4648, 2015.
- [33] Jourabloo, A.; Liu, X. Pose-invariant 3D face alignment. In: Proceedings of the IEEE International Conference on Computer Vision, 3694–3702, 2015.
- [34] Jourabloo, A.; Liu, X. Large-pose face alignment via CNN-based dense 3D model fitting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4188–4196, 2016.
- [35] Breiman, L. Random forests. *Machine Learning* Vol. 45, No. 1, 5–32, 2001.
- [36] Schuster, S.; Leistner, C.; Wohlhart, P.; Roth, P. M.; Bischof, H. Alternating regression forests for object detection and pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, 417–424, 2013.
- [37] Schuster, S.; Wohlhart, P.; Leistner, C.; Saffari, A.; Roth, P. M.; Bischof, H. Alternating decision forests. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 508–515, 2013.
- [38] Wan, C.; Yao, A.; van Gool, L. Direction matters: Hand pose estimation from local surface normals. *arXiv preprint arXiv:1604.02657*, 2016.
- [39] Ren, S.; Cao, X.; Wei, Y.; Sun, J. Global refinement of random forest. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 723–730, 2015.
- [40] Fanelli, G.; Dantone, M.; van Gool, L. Real time 3D face alignment with random forests-based active appearance models. In: Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 1–8, 2013.

Jie Wang is currently an M.S. student in the School of Computer Science at the University of Science and Technology of China. Her research interest is in computer vision and machine learning.

Juyong Zhang is an associate professor in the Department of Mathematics at the University of Science and Technology of China. He received his Ph.D. degree from the School of Computer Science and Engineering at Nanyang Technological University, Singapore. Before that, he received his B.S. degree in computer science and engineering from the

University of Science and Technology of China in 2006. His research interests fall into the areas of computer graphics, computer vision, and machine learning.

Changwei Luo is a research assistant in the Department of Automation at the University of Science and Technology of China. His research interests cover computer vision and human computer interaction.

Falai Chen is a professor in the Department of Mathematics at the University of Science and Technology of China. He received his B.S., M.S., and Ph.D. degrees from the University of Science and Technology of China in 1987, 1989, and 1994, respectively.

His current research interests are in computer aided geometric design and geometric modeling, specifically, in algebraic methods in geometric modeling,

splines over T-meshes and their applications to iso-geometric analysis.

Open Access The articles published in this journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.