Computación y Sistemas
ISSN: 1405-5546
computacion-y-sistemas@cic.ipn.mx
Instituto Politécnico Nacional
México

Fernández-Cervantes, Víctor; García, Arturo; Ramos, Marco Antonio; Méndez, Andrés
Facial Geometry Identification through Fuzzy Patterns with RGBD Sensor
Computación y Sistemas, vol. 19, núm. 3, 2015, pp. 529-546
Instituto Politécnico Nacional
Distrito Federal, México

ISSN 2007-9737

Facial Geometry Identification through Fuzzy Patterns with RGBD Sensor

Víctor Fernández-Cervantes[1], Arturo García[1], Marco Antonio Ramos[2], Andrés Méndez[1]

[1] Instituto Politécnico Nacional, CINVESTAV, Guadalajara, Mexico

[2] Universidad Autónoma del Estado de México, Mexico

{vfernand, aggarcia, amendez}@gdl.cinvestav.mx, mramos@univ-tlse1.fr

Abstract. Automatic human facial recognition is an important and complicated task; it is necessary to design algorithms capable of recognizing the constant patterns in the face and to use computing resources efficiently. In this paper we present a novel algorithm to recognize the human face in real time; the system's input is the depth and color data from the Microsoft KinectTM device. The algorithm recognizes patterns/shapes on the point cloud topography. The template of the face is based in facial geometry; the forensic theory classifies the human face with respect to constant patterns: cephalometric points, lines, and areas of the face. The topography, relative position, and symmetry are directly related to the craniometric points. The similarity between a point cloud cluster and a pattern description is measured by a fuzzy pattern theory algorithm. The face identification is composed by two phases: the first phase calculates the face pattern hypothesis of the facial points, configures each point shape, the related location in the areas, and lines of the face. Then, in the second phase, the algorithm performs a search on these face point configurations.

Keywords: Kinect, RGBD, fuzzy logic, face detection, face segmentation.

1 Introduction

Real life applications like authentication, robotics, and security can be substantially benefited through an automatic and robust system that handles facial detection with transparent interface [1, 2, 3, 4, 5]. Face recognition and tracking are two of the main problems which have received considerable attention recently [6, 7].

Automatic face recognition algorithms commonly assume that face images are well-aligning and have similar poses. These constraints are hardly to be found in the real world [8, 9]. The researchers of [8] tried to overcome this limitation. A common approach is to discretize the poses of the face and learn to detect each pose separately [8]. A number of studies rely on statistical models of the face shape and active appearance models [9]. Other methods based on facial characteristics need all facial characteristics to be fully visible on every pose or select the points manually and make them coincide with the generic model [9]. In general, these methods based on 2D images are sensitive to illumination changes, lack of features, and partial occlusions [10]. The work presented in [11] is a system based on a fuzzy neural network that combines histograms of color and depth information. Another method [12] detects the human face using a combination of the state of the art eye center locators and corner locations. This system estimates the visual gaze of a user in a controlled environment.

The main issues of this problem are face detection and gender classification. Here, several methods rely on the selection of the color space to detect skin region and face, and to perform gender classification. The relationship between face detection and gender classification is examined experimentally in [13]. Meanwhile, [14] reports a detailed analysis of how different normalizations can affect gender classification accuracy. The two different methods for alignment and gender classification studied in this work are support vector machines and two-layer Real Adaboost classifiers [15]. The most interesting findings, in the context of supervised learning for

face recognition [14], are that the support vector machine performs better than other classifiers and that the resolution of a face does not affect the classification rate [15]. Other works [16, 17, 18, 19, 20] use vector color models to separate RGB images in order to perform skin color detection. All these methods have been useful and they work very well under controlled environments.

Nevertheless, since devices like Kinect came out, we have a lot more information that we can use, such as depth data. The data obtained with Kinect normally cannot be directly fed into the designed computer vision algorithms. Most of the algorithms take advantage of rich information (RGB and depth) attached to a point.

In order to correctly combine an RGB image with the depth data, it is necessary to spatially align the RGB camera output and the depth camera output. In addition, raw depth data are very noisy and many pixels in the image may have no depth due to multiple reflections, transparent objects, or scattering in certain surfaces (such as human tissue and hair). Those inaccurate/missing depth data (holes) need to be recovered prior to being used.

Therefore, many systems based on Kinect start with a preprocessing module which conducts application-specific camera recalibration and/or depth data filtering. However, more information is not necessarily better. It is useful only if this information can be handled in an appropriate way. In order to overcome the overwhelming volume of information, we need new algorithms to process it, and face geometric feature standards have to be adapted to the face geometry feature standards like the cephalometric points to describe the human face [21].

Thus, as each face is unique, there are ridges and crevasses all wrapped around a spherical skull making the face an irregular terrain. Still, there are certain constant landmarks in the human face [21] that can be used. These points allow us to create the geography of the face, and ultimately to develop indices by which subtle relationships may be revealed. Facial cephalometric points correspond to underlying skeletal craniometrical points and certain knowledge of their correlations [21].

In our work, the proposed algorithm has two different phases. The first phase is calibration: we start from the key point in the head from Kinect Microsoft skeleton. Thus, the algorithm avoids searching in the entire image. Then, we divide the face in two parts (left and right) and create different hypothesis templates based on where the most probable location of each cephalometric point is. Then, depth information is used to compare subsets of points on each side of the face for similar shape topography, until the best match for the cephalometric point description is found or the time limit for searching this point is reached.

These subsets of points are translated into a point cloud structure. This gives way to starting the second phase of the algorithm: the point shape is modeled using the fuzzy pattern theory which finds the similarity between the shape descriptions of the cephalometric points and the point cloud deformation.

This representation is inspired from the Pattern Algebra Theory [22] and the description of proportions in topography to represent the relationship of each point with its neighbors; also rule-based fuzzy membership functions based on depth and color information are used. Afterwards, concentric square connections between points are used to locate facial information. The problem of occlusion using symmetric properties of the face is relaxed due to the search for similar individual cluster point shapes that match together with the face geometry characteristics [21]. Then, using the point cloud configuration, we can describe contours to obtain facial areas. This allows adjustment into the template structure to have a shape description of a particular face.

The present paper has the following structure: in Section 2, we review the Kinect device advantages for gathering visual information; in Section 3, we describe facial geometry theory on which this paper is based. Section 4 describes the proposed point cloud model and minimization of fuzzy rules. In Section 5, we present the idea behind contour points. In Section 6, we show the results of the cloud point model to detect facial geometry. Finally, in Section 7, conclusions are presented and possible future improvements are discussed.

2 Kinect

2.1 Hardware

The Kinect device consists of a pattern projector IR and an IR camera, which are used to triangulate points in Euclidian space. They work together like a depth and color camera (RGB).

1 RGB Camera delivers three basic color components of video. The camera operates at 30 Hz and can offer images at 640×480 pixels with 8-bit per channel. Kinect also has the option to produce higher resolution images, running at 10 f/s at the resolution of 1280 × 1024 pixels.
2 3-D Depth Sensor consists of an IR laser projector and an IR camera. Together, the projector and the camera create a depth map, which provides the distance information between an object and the camera. The sensor has a practical ranging limit of 0.8m – 3.5m of distance and outputs video at a frame rate of 30 frames/s with the resolution of 640 × 480 pixels. The angular field of view is 57◦ horizontally and 43◦ vertically.
3 The Motorized Tilt is a pivot for sensor adjustment. The sensor can be tilted up to 27◦ either up or down.

Therefore, the Kinect can be used to recognize the image content and the texture of 3D points. For measurements, the Kinect delivers three outputs: IR, RGB, and depth images [23, 24, 25, 26]. Actually, the Kinect has become the most practical way to capture RGB images with depth maps in real time [26]. It provides information of the face shape to different actions having similar 2D projections together with a single view projection on three orthogonal Cartesian planes [27, 28, 29, 30]. However, detection of the face with 2D/3D images is difficult due to the fact that the face is subjected to translation, scale, orientation, partial occlusion, rotation, and illumination changes [31]. Also high noise levels are still present in the data capture [31]. Since each local pattern of projected dots is unique, matching between the observed local dot patterns in an image with the calibrated projector dot patterns is feasible. The depth of a point can be deduced by the relative left-right translation of the dot pattern. This translation changes depending on the distance of the object to the camera- projector plane.

Table 1. OpenNi and Microsoft SDK comparison

	OpenNi	Microsoft SDK
Camera calibration	√	√
Body calibration	X	√
Standing skeleton	15 joints	20 joints
Seated skeleton	X	√
Gesture recognition	√	√
Hand gesture analysis	√	√
Facial tracking	√	√
Scene analyzer	√	√
3-D scanning	√	√
Motor control	√	√

2.2 Software

The Kinect software refers to the Kinect development library as well as the algorithms. Several software tools are available to develop products for various applications. These tools provide facilities to synchronize image signals, capture human 3D motion, recognize human voice, among others. The software components of the sensing package are the OpenNi [32], Microsoft Kinect SDK [33], and OpenKinect (LibFreeNect) [34]. OpenNi and NITE frameworks provide user identification, scene segmentation, and skeleton tracking. Working together with a compliant middleware called NITE, the version 2.0. Microsoft Kinect SDK is released by Microsoft, and its current version is 1.7. The Microsoft SDK (version 1.7) is available only for Windows, whereas OpenNi (version 2.0) is a multiplatform and open-source tool. Table 1 gives a comparison between these two tools in terms of their algorithmic components. In general, OpenKinect is a free, open source library maintained by an open community of Kinect people.

Generally, most corresponding components provided by these two libraries have functional

similarities. However, there are differences between them. The skeletal tracker in OpenNi is required to hold a predefined calibration pose by the user until the tracker identifies enough joints. The calibration time varies greatly depending on environment conditions and processing power. On the contrary, Microsoft SDK does not need a specific pose initialization. However, it is more prone to false positives than OpenNi, especially, when the initial pose of a human is too complicated. Moreover, the newest version of the Microsoft SDK is capable of tracking a user's upper body (ten joints) in case the lower body is not visible. This is particularly useful when analyzing human postures in a sitting position. Furthermore, OpenNi focuses on hand detection and hand-skeletal tracking, whereas Microsoft SDK realizes simple gesture recognition. It is worth highlighting that the new version of OpenNi (2.0) allows users to install Microsoft Kinect SDK on the same machine and run both packages using the Microsoft Kinect driver, which means that the OpenNi is now compatible with the Kinect driver. By doing so, switching between two drivers is not necessary anymore even when users want to benefit from both packages.

Kinect Performance Evaluation, from either the hardware or the software perspective, helps us to understand both the advantages and limitations of the Kinect sensor and thus to better design our own system for a given application. In [35], the authors present an experimental investigation of the depth measurement of Kinect in terms of its resolution and precision. Moreover, they make a quantitative comparison of the 3D measurement capability for three different cameras: the Kinect camera, a stereo camera, and a TOF camera. The experimental results reveal that Kinect is superior in accuracy to the TOF camera and close to a medium-resolution stereo camera. In [36], the researchers compare the Kinect sensor with two other TOF 3D ranging cameras. The ground truth data is produced by a laser range sensor with high accuracy, and the test is performed in an uncontrolled indoor environment. The experiments yield the following conclusions:

1 The performance of the Kinect sensor is very close to that of the laser for short range environments (distance < 3.5 meters);
2 Two TOF cameras have a slightly worse performance in the short range test; and
3 No sensor achieves performance comparable to that of the laser sensor at the full distance range.

This implicitly suggests that Kinect might be a better choice (over TOF cameras) if the application only needs to deal with short range environments, since TOF cameras are usually more expensive than the Kinect sensor.

Instead of comparing the Kinect with other available depth cameras, [37] provides an insight into the geometric quality of the Kinect depth data based on an analysis of the accuracy and resolution of the depth signal. The experimental results show that the random error of depth measurement increases when the distance between the scene and the sensor increases, ranging from a few millimeters at a close range to about 4 cm at the maximum range of the sensor.

The Kinect software performance is important when applying Kinect to human posture analysis. In [38], the 3D motion capturing capability offered by the Kinect is tested in order to know if the Kinect sensor has comparable accuracy to that of the existing marker-based motion acquiring systems. The result demonstrates that Kinect is able to capture relative 3D coordinates of markers with minor errors (< 1cm) in case the sensor is positioned in an ideal range (1m to 3m) and with an effective field of view.

In [39], the authors examine the accuracy of joint localization and the robustness of pose estimation with respect to more realistic setups. In their experiment, six exercises are conducted, in which the subject is either seated or positioned next to a chair. The exercise is generally challenging for human pose recognition since self-occlusion appears frequently and the capturing view angle is changed over time. The acquired 3D location of each joint is then compared to the data generated by a marker-based motion capture system, which can be considered as the ground truth data. According to the results, the Kinect has a significant potential as a low-cost alternative for real-time motion capturing and body tracking applications. The accuracy of the Kinect joint estimation is

Table 2. Cephalometric points

Code	Name	Points Set
v	Vertex	Cranial
sg	Supraglabella	Cranial
g	Glabella	Cranial
eu	Euryon	Cranial
at	Auriculotemporale	Cranial
zy	Zygion	Lateral
go	Gonion	Lateral
ec	Ectocanthion	Orbital
en	Endocanthion	Orbital
il	Iridion Laterale	Orbital
im	Iridion Mediale	Orbital
n	Nasion	Nasal
na	Nasale	Nasal
prn	Pronasale	Nasal
sn	Subnasale	Nasal
al	Alare	Nasal
sls	Superior Labial Sulcus	Labial
ls	Labiale Superius	Labial
sto	Stomion	Labial
li	Labiale Inferius	Labial
ch	Chelion	Labial
lm	Labiomentale	Mental
pog	Pogonion	Mental
gn	Gnathion	Mental
sa	Superaurale	Auricular
sba	Subaurale	Auricular
pra	Preaurale	Auricular
pa	Postaurale	Auricular
tr	Tragion	Auricular

comparable to that of the marker-based motion capture in a more controlled body pose.

However, in general poses, the typical error of Kinect skeletal tracking is about 10 cm. Moreover, the current Kinect algorithm frequently fails due to occlusions, not distinguishing depth (limbs close to the body) or clutter (other objects in the scene).

3 Facial Geometry

The human face is a marvel of contouring displaying hills and valleys, slopes, crests, ridges, and crevasses all wrapped around a spherical skull. Throughout this irregular terrain there are certain constant landmarks which are denoted by an array of precisely defined cephalometric points [21]. These cephalometric points are quite important because they are directly related to the underlying skeletal craniometric points. Although faces are not identical, they have the same pattern in distribution of the cephalometric points [21]. These points are shown in Table 2. They allow us to map the geography of the face, to zone its areas, and, ultimately, to develop indices by which subtle relationships may be revealed. Facial cephalometric points correspond to underlying skeletal craniometric points, and knowledge of their correlations forms the scientific basis of forensic facial approximation [21]. The points projected over the human face template are shown in Fig. 1. The code, name, and the membership set are shown in Table 2. The definition of the cephalometric points is given by Eq. 1:

$$F^{CP} = [v, sg, g, eu, at, zy, go, ec, en, il, im, n, na, prn, sn, al, sls, ls, sto, li, ch, lm, pog, gn, sba, pra, pa, tr], \quad (1)$$

where F^{CP} is the set of the cephalometric points in the face, each cephalometric point has a description of the geometrical shape and its position among the other cephalometric points.

This means that the areas of the face are contoured, and a true mathematical treatment of this surface would require solid geometry [21]. The face planes in Eq. 2 are defined as vectors that divide the face using a subset of cephalometric points where the n-tuple of F^{CP} is defined in Eq. 3. Then,

$$F^{L} = \left\{\overrightarrow{MSP} \bigcup \overrightarrow{MFP} \bigcup \overrightarrow{TNP} \bigcup \overrightarrow{TGP}\right\}. \quad (2)$$

The four most useful facial planes are defined as follows.

Midsagittal plane (MSP) divides the face into the bilaterally symmetrical right and left sides connecting all midline points from vertex to gnathion (v-gn):

ISSN 2007-9737

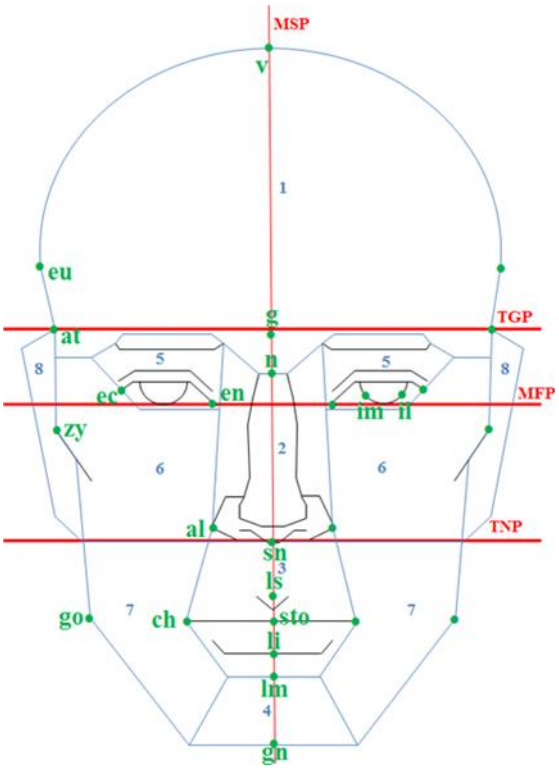

Fig. 1. Cephalometric points (green), planes (red), and areas (blue), frontal view

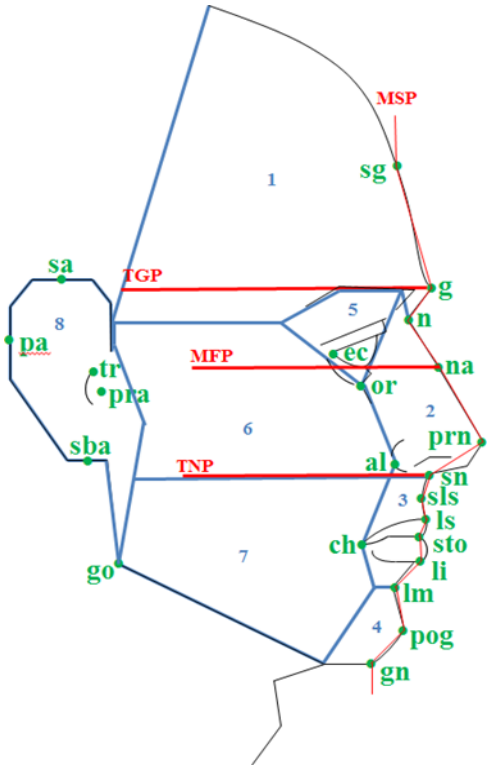

Fig. 2. Cephalometric points (green), planes (red), and areas (blue), lateral view

$$\overrightarrow{MSP} = [v,\ sg,\ g,\ n, na,\ prn,\ sn,\ sls,\ ls,\ sto,\ li,\ lm,\ pog,\ gn].$$

Midfacial plane (MFP) roughly divides the head into the upper and lower halves by a horizontal line tangent to the inferior poles of the irises:

$$\overrightarrow{MFP} = [pa, ec, im, il, en].$$

Transverse nasal plane (TNP) is a horizontal plane parallel to the MFP and passing through the subnasal:

$$\overrightarrow{TNP} = [sba, al, sn].$$

Transglabellar plane (TGP) is a horizontal plane passing through the glabella and marking the upper side of the facial square: $\overrightarrow{TGP} = [zy, n]$.

$$MSP, MFP, TNP, TGP \subseteq F^{CP}. \quad (3)$$

The eight areas of the face are contoured in Eq. 4, and a mathematical treatment of its surface would require solid geometry. One of the main objectives of the facial geometry, however, is to reveal facial relations via graphic facial analysis so we must be content with simpler Euclidian applications of geometry [21]. The eight areas of the face as seen in the front view and the 90 degree rotating face view are demarcated in Figures 1 and 2. This is preliminary concerned with the relationship between the eyes, nose, lips and ears:

1. Frontal area,
2. Nasal area,
3. Labial area,
4. Mental area,
5. Orbital area,
6. Zygomaxillary area,
7. Buccomandibular area,
8. Auricular area.

Then, the areas of the face are defined as a subset of the cephalometric points and the

geometrical shape inside the area. Thus, the face is described in Eq. 5.

$$F^A = [Frontal, Nasal, Labial, Mental, Orbital, Zygomaxillary, Buccomandibular, Auricular], \quad (4)$$

$$F = \{F^{CP} \cup F^L \cup F^A\}. \quad (5)$$

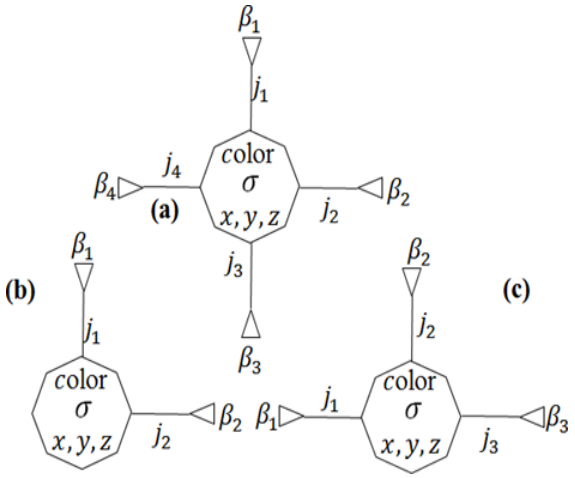

Fig. 3. Points: (a) interior, (b) corner, (c) edge

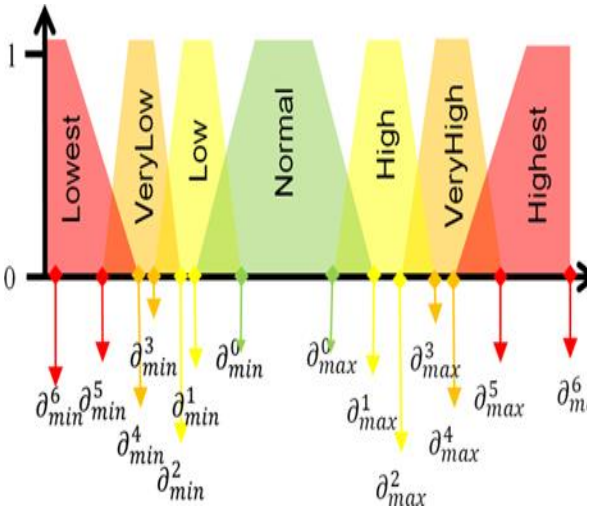

Fig. 4. Depth difference fuzzy membership function

4 Point Cloud Model

The point cloud model is a data structure that the system uses to recognize the human face from an RGBD image considering each point in the depth image. $P^d = [p_1, p_2, p_3, p_4, \ldots, p_n]$ has coordinates in the Cartesian plane$(\mathrm{x}, \mathrm{y}, \mathrm{z})$; $x, y, z \in \mathbb{Z}$, which is interpreted as a position in the plane. In order to insure that this only changes the information representation and not its values [22], it is necessary to generate a model representation that remains symmetric and invariant to rotation. For this, it is possible to use a transformation of the representation of each point in P into a generator representation from the Pattern Theory [22] which preserves distance information [24]. The number of points necessary to represent the information captured from the sensors RGB and IR is $\mathrm{n} = 640 \times 480 = 307200$. [24] However, only the hypothesis subsets of cephalometric points in the topography of the depth image are translated to the point cloud structure. Thus, each cephalometric point has three different concentric segments: the initial point, the interior points, and the border points. Finally, each point in the data structure is invariant. This means that each data point obtained with the Kinect and its data structure representation are the same [22]. In order to cluster all these point representations together into a point cloud, each point possesses a series of bonds $\mathrm{j}_1, \mathrm{j}_2, \ldots, \mathrm{j}_\mathrm{b}$.. [22] where $b = \mathrm{b(p)}$ means the bond-connectivity of p. The number of bonds can vary from point to point and each bond contains a Boolean value; this allows a bond to connect with other bonds. These connections are shown in Table 3 describing a connectivity matrix β. We have three different groups to describe the input RGBD image: edges points, corners points, and interior points.

A graphic representation of the points is shown in Fig. 3. Using these tree types of points, it is possible to describe the complete image information as a point cloud structure because corner points and edge points can rotate 90°,180°,270°, whereas the interior points are invariant under rotations. Also, the associated color RGB component $P^c = [p_1, p_2, p_3, p_4, \ldots, p_n]$ and the identification number of the σ configuration are invariant. The point cloud model makes use of depth information as follows. The ∂ value is defined as the depth difference between

Table 3. Label classification σ

Fuzzy Term	interval	Label σ
Lowest	∂^6_{min}-∂^5_{min}	G
Lowest-VeryLow	∂^5_{min}-∂^4_{min}	H
VeryLow	∂^4_{min}-∂^3_{min}	I
VeryLow-Low	∂^3_{min}-∂^2_{min}	J
Low	∂^2_{min}-∂^1_{min}	K
Low-Normal	∂^1_{min}-∂^0_{min}	L
Normal	∂^0_{min}-∂^0_{max}	M
Normal-High	∂^0_{max}-∂^1_{max}	N
High	∂^1_{max}-∂^2_{max}	O
High-VeryHigh	∂^2_{max}-∂^3_{max}	P
VeryHigh	∂^3_{max}-∂^4_{max}	Q
VeryHigh-Highest	∂^4_{max}-∂^5_{max}	R
Highest	∂^5_{max}-∂^6_{max}	S

two adjacent points linked by a bond, and it is calculated by Eq. 6. This term is calculated for every bond connection. This allows defining a threshold to characterize the shape of the point given the relation of the depth difference between the connected neighbors into the σ identification number:

$$\forall \beta_i : \exists\, \partial_{\mathrm{i}} \;\rightarrow \partial_{\mathrm{i}} \epsilon \beta_i = z\epsilon\rho_{(x,y)} - z\epsilon\rho_{(x,y)\beta_{nxn}}. \qquad (6)$$

Each connector is classified by a fuzzy function to describe the topographic shape of the point based on different ∂ depth values on each bond. These values are described by a fuzzy membership function and their own fuzzy rule set system. Since the points b and c from Fig. 3 are the corners and the edge, respectively, it is very unlikely to find any part of the face in those points. Then, the system only uses the interior points. The membership functions are used to define the linguistic variable depth difference between two connected points, see Fig. 4. The minimum $\partial_{\min}$ and the maximum $\partial_{\max}$ are obtained as integer values between 0 and 2047. The distance value ∂_{i} indicates the depth difference on the surface between two points. Next, the identifiable cephalometric points of the human face are described by fuzzy sets type-1.

These configurations are shown in Fig. 5. Finally, the set of fuzzy rules is used to describe the configuration in the topography point cloud. The topography is represented with an input fuzzy set corresponding to the depth change difference information: Lowest, VeryLow, Low, Normal, High, VeryHigh, and Highest. Thus, the p^{∂} set is defined in Equations 7 and 8 to describe fuzzy differences between point distances. Eq. 9 defines the fuzzy membership function between allowed distances.

$$\mu_{\mathrm{p}^{\partial}}.\, \partial \rightarrow [0,1], \qquad (7)$$

$$\mathrm{P}^{\partial} = [\mathrm{lowest}, \mathrm{verylow}, \mathrm{low}, \mathrm{normal}, \quad \mathrm{high}, \quad \mathrm{veryhigh}, \quad \mathrm{highest}], \qquad (8)$$

$$\sum_i \frac{\mu \mathrm{p}(\partial_i)}{\partial_i}. \qquad (9)$$

This fuzzy rule-based system is composed of if-then rules. Such set of rules is used to generate a topography description of the connected points. Fuzzy logic type-I is used because it is easy to describe the pattern on the face in terms of valid topography points where variations are present. The values of ∂ are used to classify each point change depending of the description of the cephalometric point; however, the relation to label each point is constant, and the relation between distances and labels is shown in Table 3. The label of each point is the code to represent the graphical shape on the topography of the depth image in order to label each bond over the dependency of the classification with the fuzzy rules set. Since we only classified the interior points, all the points (blue) are b(4) connected; the lines represent the bonds which connects the four neighbors around shown in figure 5, as follow: green is the adjacent point in j_1, orange is j_2 yellow j_3 and red j_4.

Using experimentation and available data, the following fuzzy rules r were designed:

$$\boldsymbol{r_1}: \partial_1 \text{ is } \mathrm{p}^{\partial}$$
$$\boldsymbol{r_2}: \partial_2 \text{ is } \mathrm{p}^{\partial}$$
$$\boldsymbol{r_3}: \partial_3 \text{ is } \mathrm{p}^{\partial}$$

$$\boldsymbol{r_4}: \partial_4 \text{ Is } \mathrm{p}^{\partial}$$
$$\boldsymbol{r}: p \text{ is } \sigma,$$

where ∂ is the depth information and p^{∂} is the membership function from Eq. 9. Below are examples of the rules for the configurations σ=MMMM from Fig. 5a, the cephalometric point prn from Figures 1 and 2, ∂^0_{min}=-2, and ∂^0_{max}=2:

– IF (∂_1 is 1) ∧ (∂_2 is 2) ∧ (∂_3 is 2) ∧ (∂_4 is -2) Then (p is MMMM),
– IF (∂_1 is -1) ∧ (∂_2 is 1) ∧ (∂_3 is 0) ∧ (∂_4 is -1) Then (p is MMMM).

Then, the configuration σ defined in Eq. 9 describes the bond connection in a set of points:

$$\boldsymbol{c} = \sigma(p_1, p_2, \ldots, p_n), \tag{9}$$

where $\boldsymbol{c}$ is the configuration of the set of points and σ is called a connector that connects some bonds $\rho_i = (j_1, \ldots, j_N)$ with others $\rho_i{}' = (j_1, \ldots, j_N)$. Here, $\rho = (j_1, \ldots, j_N)$ labels bonds in general. In addition, $\boldsymbol{c}$ describes two types of bonds: internal and external bonds in Eq. 9. This separates each point into the respective configuration. When bonds from internal points are connected to other bonds, these bonds will be called internal bonds. The remaining closed bonds are the external ones denoted by $ext(\boldsymbol{c})$ [22].

5 Face Configuration

The face configuration uses the Kinect skeleton SDK (S) to detect and initialize the face position and orientation to avoid searching into the entire image. Thus, we use a subset of joints S extracted from the Kinect skeleton:

$$S = \{Head, ShoulderCenter, Head, ShoulderLeft, ShoulderRight\}. \tag{10}$$

Each joint S_i in Eq. 10 has a position and orientation in the Euclidian coordinates. The system uses this information to calculate the position of the face and the orientation in the Cartesian plane F_p (facial position). The system searches a protuberance on the topography of the depth image looking for the initial point: the tip of

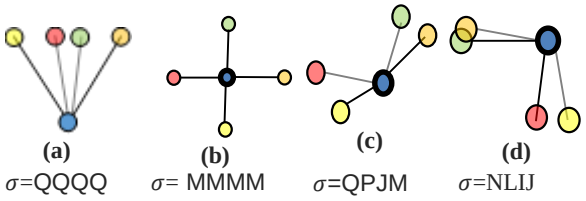

Fig. 5. The σ label configuration to points p (blue p, green j_1, orange j_2, yellow j_3 red j_4-)

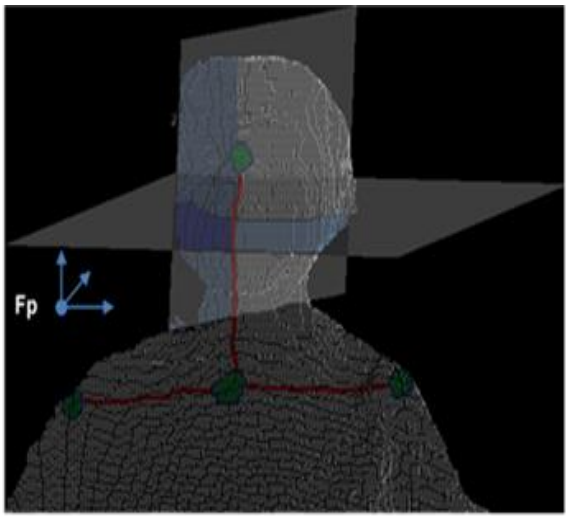

Fig. 6. Joints (green), calculation of the Cartesian plane

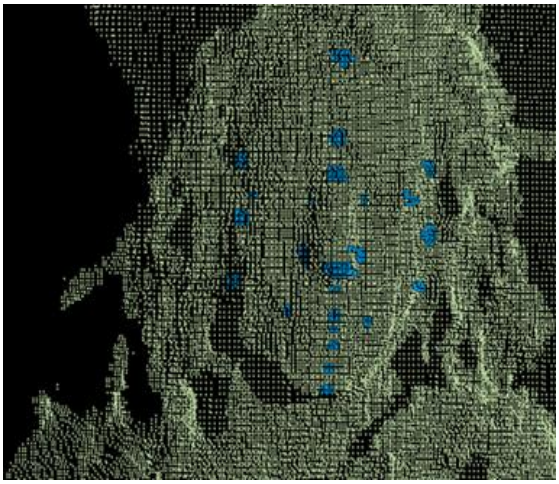

Fig. 7. Hypothesis of the cephalometric points (blue), over the point cloud

Algorithm 1: Inferring cephalometric point position

```
1://Collect absolute votes
2:Initialize Z_C=0 for all the configurations C_i
3:for all points P in the configuration C_i do
5:   lookup 3D Point position p_i=(x_p, y_p, z_p)^T
6:   for all tree in forest do
7:    descend tree to reach leaf node l
8:  for all configurations C_i do
9:    lookup weighted relative vote set V_lC
10:    for all (Δ_lck, w_lck) ϵ V_lC) do
11:      if ||Δ_lck||_2 ≤ distance threshold λ_{C_i} then
12:        compute absolute vote z = Δ_lck + p_i
13:         adapt confidence weight w = w_lck · Z_p
14:         Z_C := Z_C U{(z, w)}
15: // Aggregate weighted votes
16: sub-sample Z_C to contain N votes
17: aggregate Z_C using mean shift on Eq. 13
18: return weighted modes as final hypotheses
```

Algorithm 2: Learning relative votes

```
1://Collect relative Offsets
2:Initialize R_{lC_i} = 0 for all leaf nodes l and
  configurations C_i
3:for all points P in the template P^T do
4:   lookup ground truth joint position z_{C^T C_i}
5:   lookup 3D pixel position p_{C^T C_i}
6:   compute relative offset
          Δ_{P^T P→C_i} = z_{C^T C_i} − p_{C^T C_i}
7:   descend tree to reach leaf node l
8:    store Δ_{P^T P→C_i} in R_{lC_i} with reservoir sampling
9://Cluster
10: for all leaf nodes l and configurations C_i do
11:   cluster offsets R_{lC_i} using mean shift
12:   take top K weighted modes as V_lC
13: return relative votes V_lC for all l and C
```

the nose (prn). Fig. 6 shows the Kinect skeleton joints in green, the bones between these joints in red, and the Cartesian plane with the same correct orientation between these joints. The head is the start point to look for the tip of the nose (prn). From the initial point, we calculate the position of the cephalometric points based on the face size using the depth image. The configurations of the cephalometric points necessary to describe the face C are at most thirty under the point cloud model, see Fig. 7. Each configuration is used for describing the position of the cephalometric points and allows seeing the face as a union of all thirty configurations, Eq. 11:

$$C = [c_1 \cup c_2 \cup c_3 \cup c_4 \cup c_5 \cup c_6 \cup c_7 \cup c_8 \cup c_9 \cup c_{10} \cup c_{11} \cup c_{12} \cup c_{13} \cup c_{14} \cup c_{15} \cup c_{16} \cup c_{17} \cup c_{18} \cup c_{19} \cup c_{20} \cup c_{21} \cup c_{22} \cup c_{23} \cup c_{24} \cup c_{25} \cup c_{26} \cup c_{27} \cup c_{28} \cup c_{29} \cup c_{30}]. \quad (11)$$

The System infers the 3D position of several cephalometric points by aggregating votes cast in a regression forest. The regression forest is an ensemble of decision trees [3] that predicts continuous outputs [6]. A decision tree consists of split nodes and leaf nodes: the split nodes contain tests which evaluate image features to decide whether to branch to the left or right child; the leaf nodes contain some prediction (either categorical for classification, or continuous for regression). At the split nodes, the system uses the features from the cephalometric points which compare the depth at nearby pixels to a threshold. These features are fast to evaluate, and additionally they are depth-invariant shown to discriminate human face appearance in depth images well. At each leaf node l, we store a distribution over the relative 3D offset to each cephalometric point C_i of interest (potentially to all cephalometric points), a continuous regression output. The representation of the distribution is 3D relative vote vectors $\Delta_{lck} \epsilon \mathcal{R}$ obtained by the largest modes found by mean shift, and it assigns a confidence weight w_{lck} to each vote given by the size of the cluster, see Eq. 12 which describes the relative vote set for each cephalometric point C_i at node l:

$$V_{lc} = \{(\Delta_{lck}, w_{lck})\}_{k=1}^{K}. \quad (12)$$

The set Z_C of absolute votes cast by all points for each cephalometric configuration C_i is aggregated using mean shift. Only those relative votes that fulfill a per C_i distance threshold λ_{C_i} are used. This threshold prunes out long range predictions which are unlikely to be reliable and improves accuracy considerably. Then the system

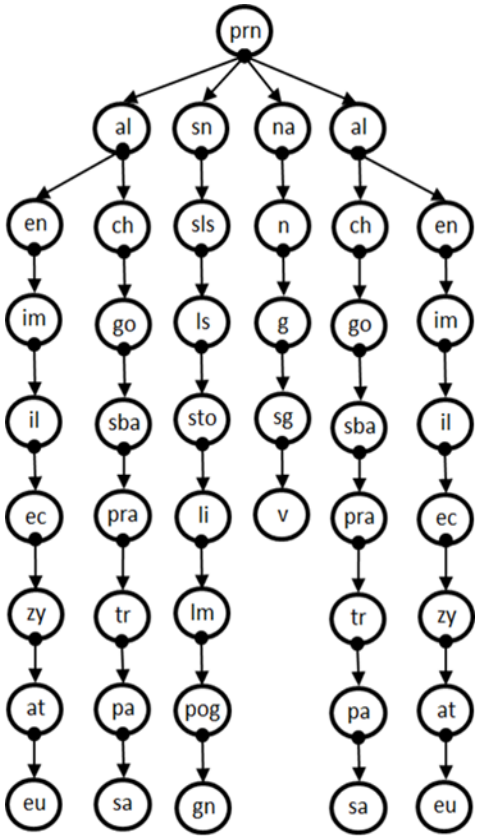

Fig. 8. The template tree structure of the facial cephalometric points

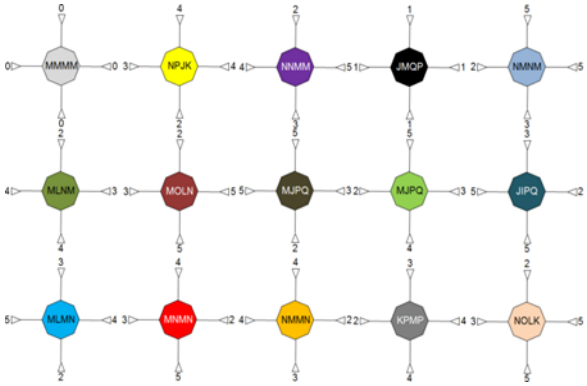

Fig. 9. Pattern points for the three level description of the prn point

re-weights the confidence of each relative vote to compensate for observing fewer points when the subject face is farther from the device. Optionally, the set Z_C can be subsampled to improve speed while maintaining high accuracy. The absolute votes Z_C are added to the kernel density estimation [40]. Given n points, $P = [p_1, p_2, \ldots, p_n]$ in the d-dimensional space $\mathcal{R}^3$, the multivariate kernel density estimator with kernel $K(P)$ and a symmetric bandwidth matrix computed in the point P is given by Eq. 13:

$$\hat{f}(P) = \frac{1}{n}\sum_{i=1}^{n} K_T(P_i - P_T), \qquad (13)$$

where P^T are the points in the template described in [21], $P^T = [p_1^T, p_2^T, \ldots, p_n^T]$ is the template configuration C^T.

The description of the template is presented in figure 8, as a graph, where each circle is compose by the characteristics of the corresponding relative votes V_{lc} in each cephalometric point C_i, starting at the nose point (prn). Algorithm 2 describes how the set of relative votes is learned, given the tree facial cephalometric point template structure. As shown in Fig. 8, each point indicates a relative offset to all ground true cephalometric points on the topography from the depth image. These are clustered using mean shift.

6 Pattern Connections

To recognize a specific cephalometric point from the topography requires finding a pattern relation between points like in chemistry: the atoms (points) are connected together into molecules (configurations), and the nature of the chemical bonds, ionic, covalent, etc. decides what combinations of atoms will be stable enough to form molecules. Using this idea, it is possible to get specific configurations from the set of points described in Section 4.

Now, it is necessary to split up the face space F into several level sets [22] F^{β}. Here, β is the variation over the same point space F_p. This allows having the partitions as in Eq. 14. Each point in F^{CP} is identified by a description denoted by a connectivity matrix β, this divides the pattern of each F_i^{CP} in three parts: the interior, boundary,

Algorithm 3: Pattern connection

1: Input: connectivity vector $\sigma[p_1, p_2, \ldots, p_n$
2: Output: configuration of the connection C_{NT}
3: Create $C_{NT} = 0$
4: **for** i = 1 unti li =|| σ || - 1
5: **for** j = i + 1 until j = || σ ||
6: **If** can establish the connection with C_{NT}, σ_j
7: $C_{NT} = C_{NT} \cup (\sigma_i, \sigma_j)$
8: end of loop
9:return C_{NT}

Algorithm 4: Calculation of the symmetry in the internal configurations of the points set

```
midpoints ⇐ new Vector[length(σ)]
for s ⇐ 0 to length(σ) - 1 do
  for p ⇐ 0 to length(σ[s]) - 1 do
    other ⇐ 0
    Find other point on σ[s] at same 'height' as p:
    for q ⇐ 0 to length(σ[s]) - 1 do
      next ⇐ q + 1
      if next >= length(σ[s]) then
        next ⇐ 0 {Wrap}
      end if
      if q = p _ next = p then
        continue
      end if
      if (q:y < p:y < next:y)_(q:y > p:y > next:y)
      then
        other ( interpolate(q; next)
        break
      end if
      end for
      if other ≠ 0 then
        match[s]:append((p + other)=2)
      end if
    end for
  end for
  return match
```

and exterior parts. To describe each one of these parts, we define three groups of point cloud structure: the 4-neighborhood, 3-neighborhood, and 2-neighnorhood sets. These sets are denoted $N_4(\rho)$, $N_3(\rho)$, and $N_2(\rho)$.

$$F^G = \bigcup_{\beta \epsilon F_p} F^{\beta}. \quad (14)$$

The connections to denote the prn point are shown in Table 3.

Table. 3. Connectivity matrix β

β	β_5	β_4	β_3	β_2	β_1	β_0
β_0	1	0	0	0	1	0
β_1	0	1	0	0	0	1
β_2	1	0	0	1	0	0
β_3	0	0	1	0	0	0
β_4	1	0	0	0	0	0
β_5	0	1	0	0	0	0

The connectivity matrix β_{6x6} is used to connect the different points following the pattern in a topography shape of F^{CP}. Each bond can be connected or disconnected. Thus, each bond j_i has a connector β_i which is the representation of the connection value with a neighbor. The value "1" indicates that a connection is complete, and "0" indicates that a connection is null. The configuration is restricted by local as well as global constraints on the product space BxB. This can be described by the true value function given in Eq. 15:

$$\rho: BxB \rightarrow \{TRUE, FALSE\}. \quad (15)$$

Therefore, detection of highly salient points depends on high values of β. Assuming the nose tip as high as a salient point, it is possible to use this feature to detect the face [24]. Cephalometric points are a region rather than one exact point.

Thus, different potential regions are treated as a nose tip configuration, see Fig. 10. The face can be represented as convex or concave in the point cloud describing the curvature of the face [41]. Inspired by this, a candidate region for the nose tip should maintain relative and symmetrical proportions as described in the facial geometry [21].

The matrix C_{NT} is the description between cephalometric point and the use of the face symmetry to find the configurations of F^{CP}.

$$C_{NT}=\begin{bmatrix} \sigma_{-2,2} & \sigma_{-1,2} & \sigma_{0,2} & \sigma_{1,2} & \sigma_{2,2} \\ \sigma_{-2,1} & \sigma_{-1,1} & \sigma_{0,1} & \sigma_{1,1} & \sigma_{2,1} \\ \sigma_{-2,0} & \sigma_{-1,0} & \sigma_{0,0} & \sigma_{1,0} & \sigma_{2,0} \\ \sigma_{-2,-1} & \sigma_{-1,-1} & \sigma_{0,-1} & \sigma_{1,-1} & \sigma_{2,-1} \\ \sigma_{-2,-2} & \sigma_{-1,-2} & \sigma_{0,-2} & \sigma_{1,-2} & \sigma_{2,-2} \end{bmatrix}. \quad (16)$$

Fig. 11 shows the distribution of the point configurations between the nose cephalometric points. C_{NT} is defined as the nose tip configuration. Each σ_i is described by the rule-based fuzzy system in Eq. 16. There, the configuration is compared with the point cloud. We treat first the points (0, 0, 0) in the F_p plane in Fig. 6, and Algorithm 3 compares the square symmetry configuration in the point cloud. The algorithm uses the concentric square shape in Fig. 11 because it is easy to measure the symmetry between the points [42].

7 Contour Connections

With the point cloud data structure, we can recalibrate the fuzzy function in Eq. 9, where the fuzzy rule-based system is composed of if-then rules. This set of rules is used to generate the contour description in a vector configuration c by Eq. 9 of each area of the face. A path between two points p and q in $\mathbb{R}^3$ is a sequence of distinct points $\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_n$, where $\Upsilon_0 = \text{initial point}$ and $\Upsilon_n = \text{end point}$, such that Υ_i is adjacent to $\Upsilon_{i-1}, 1 \leq i \leq n$, and n is the path length. A connected region $R \in b(x, y)$ is a set of points $p(x, y)$ such that there is a path starting in Υ_1 and ending in $\Upsilon_{|2}, \forall \Upsilon_1 \Upsilon_2 \in R$.

Each contour vector is the boundary of the points Υ_i such that it divides the points in two subsets: the interior subset and the exterior subset, Fig. 12. Thus, a point Υ_i is similar to a point P but with only one input bond and one output bond, Fig. 12. This restricts the configurations in local as well as global constraints on the cross space BxB of the bond value space B given a truth valued function shown in Eq. 15. Thus, for a pair $\{\beta', \beta''\}$ of bond values there is only one input bond and only one output bond. The pair is regular if $\rho\{\beta', \beta''\} = \text{TRUE}$, and all other bonds are irregular if $\rho\{\beta', \beta''\} = \text{FALSE}$. This partition divides BxB into two subsets. Thus, ρ is equivalent to a relation that separates interior points from the exterior ones.

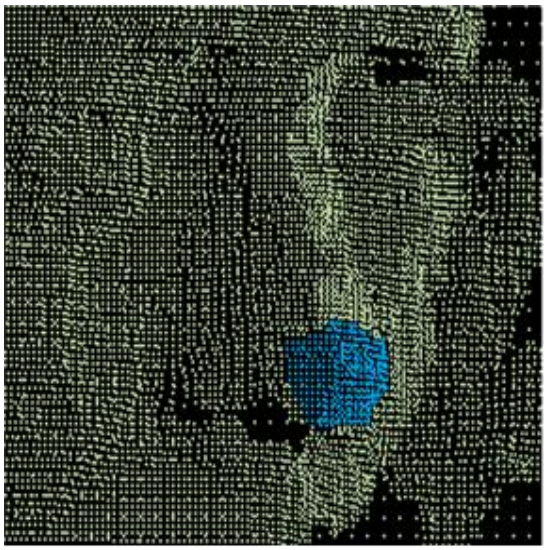

Fig. 10. Pattern points for the three levels projected in the points

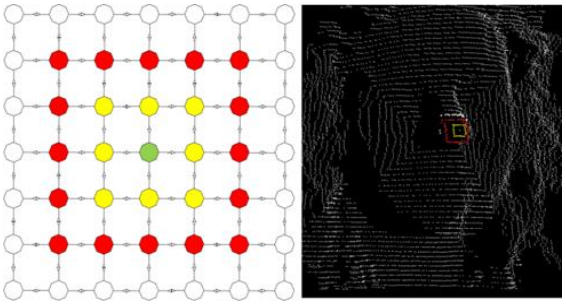

Fig. 11. Concentric square configuration

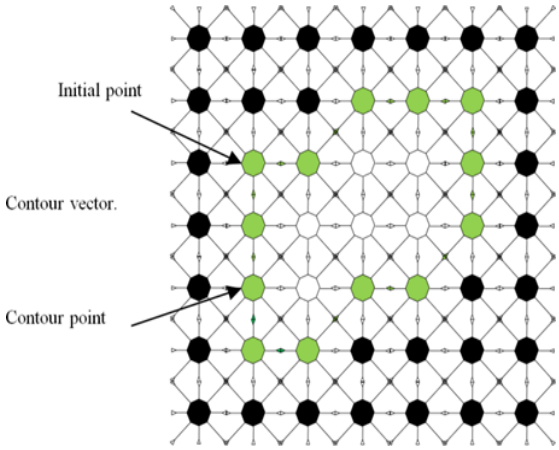

Fig. 12. Contour points

Fig. 13. Original color image

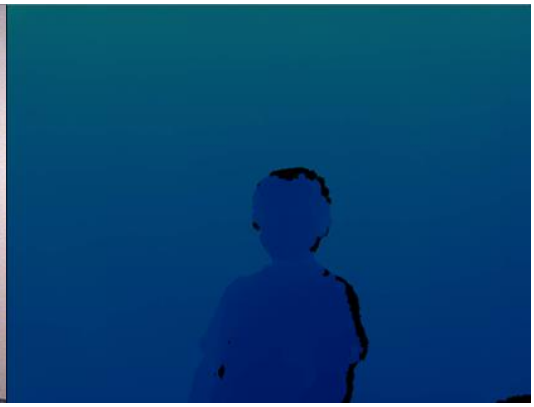

Fig. 14. Original depth image

Fig. 15. Cephalometric points

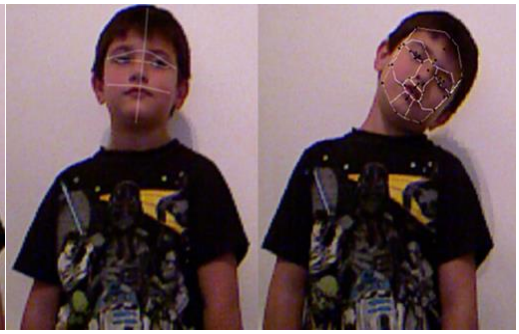

Fig. 16. Lines and sections

Now, we assume that ρ is invariant as well as the two internal bonds $\{i', j'\}$ of $p_{i'}$ and $\{i'', j''\}$ of $p_{i''}$.

This allows the definition of the contour vector Γ_i as a sequence of contour points Υ_i in Eq. 17. Then, the initial contour Υ_0 point is the same as the ending contour point Υ_k:

$$\Gamma = (\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_{k-1}). \tag{17}$$

The contour contains the necessary information in the face shape.

8 Experiments and Results

The proposed system performs real time face tracking. It was proven in several different faces of people ranging from children to seniors under different position of the face, fast movements, noisy background, illumination changes, and a combination of male and female. The system showed a very good performance even with small faces (of children). As shown in Fig. 13, it was necessary to impose distance constraints from the device to the subject. The distance used was between 0.4 meters to 1.8 meters, Fig. 13. The person is standing at 90 cm from the device to the prn cephalometric point. Under these constraints, the data produced by the device was adequate for the system.

Once calibration of the template was finished, the proposed system was able to correctly find the cephalometric points in the face. This can be seen in Fig. 15. Then, using these points, it was possible to follow the face contours, and divide the face in four contours, Fig. 16. Finally, the face

Fig. 17. Different captures from the system with free fast movements

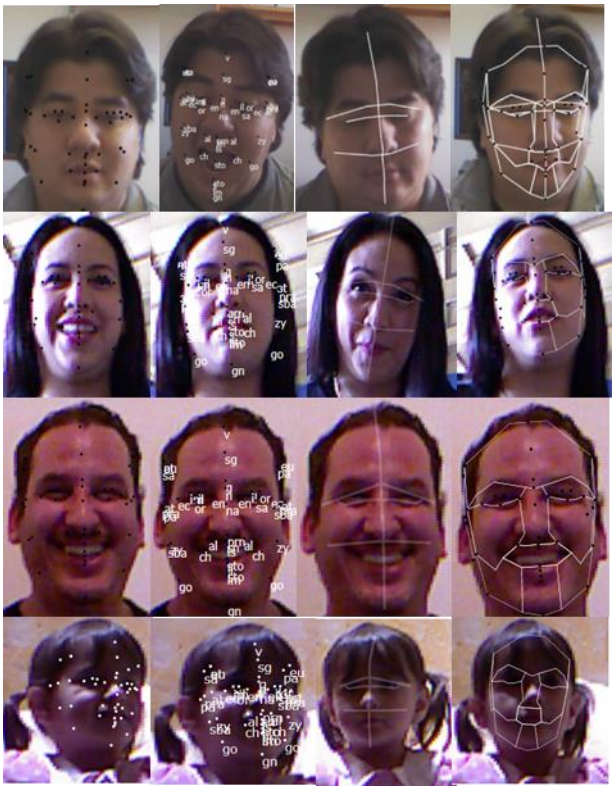

Fig. 18. Different captures from the system with adult people and free fast movements

was segmented into eight parts as shown in Fig. 16. Even though there is occlusion by hair, the proposed system was able to adapt to the face configuration and approximate the cephalometric points in the image. This allowed us to build a 3D mask that can be used to track the face of an individual or to obtain other information.

Fig. 17 shows different poses used in the system. Fig. 18 shows the points with which it was possible to follow the face contours and divide the face into four contours.

Finally, the topography of the face is segmented in eight areas. Although in several parts of the image images show partial occlusion, the proposed algorithm was capable to overcome

and adapt to the face configuration patterns and mark the cephalometric points in the image. The template of the face in the tridimensional space fits in to the user face. This will be useful to identify emotions expressions in the face, using the distance relation between facial points.

9 Conclusions

This work presents a novel system which deals with the RGBD information provided by the Kinect sensor. This novel algorithm is able to recognize the subject's face under behavior without movement restriction, Fig. 16. Under these lax conditions, the system was able to identify face biometry using the description of the facial geometry. This task is especially difficult because the human face topography changes all the time with different gestures in the facial expression. However, even using simple fuzzy type-I set configurations, the algorithm was able to adapt correctly the mask configuration. In addition, the proposed algorithm also has an acceptable accuracy to measure and track the face. Finally, the description of the configuration can be adapted to different granularities, but it is necessary to design a configuration template before executing the algorithm. In future, we will report on an implementation of a fuzzy type-II set configuration which is being designed now to increase the robustness of the proposed algorithm.

Acknowledgements

This work has been funded by CONACYT scholarship number 212753. The authors want to thank Victor Rodriguez, Isela Ayar, Juan Manuel Rodriguez, Emily "Godzilla" Marlen Rodriguez, Stuart and Maria Embleton who participated in the project.

References

1. **Inaba, R., Watanabe, E., & Kashiko, K. (2003).** *Security applications of optical face recognition system: access control in e-Learning*. Departlnent of Mathematical and Physical Sciences, Japan Woinen's University, pp. 255–261.
2. **Belhumeur, P.N., Hespanha, J.P., & Kriegman, D.J. (1997).** Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 19, pp. 711–720, doi: 10.1109/34.598228.
3. **Blanz, V. & Vetter,T. (2003).** Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 25, No. 9, pp. 1063–1074, doi: 10.1109/TPAMI.2003.1227983.
4. **Kare, S., Samal, A., & Marx, D.D. (2008).** Using bidimensional regression to assess face similarity. *Machine Vision and Applications,* Vol. 21, No. 3, pp. 261–274, doi: 10.1007/s00138-008-0158-7.
5. **Singh, R., Vatsa, M., & Noore, A. (2009).** Face recognition with disguise and single gallery images. *Image and Vision Computing,* Vol. 27, No. 3, pp. 245–257, doi: 10.1016/j.imavis.2007.06.010.
6. **Kresimir, D. & Mislav, G. (2007).** *Face Recognition*. I-Tech Education.
7. **Ponce, J. & Karahoca, A. (2009).** *State of the Art in Face Recognition*. I-Tech Education.
8. **Xie, S., Shan, S., Chen, X., & Gao, W. (2008).** V-LGBP: Volume based Local Gabor Binary Patterns for face representation and recognition. *19th International Conference on Pattern Recognition (ICPR), pp. 1–4, doi: 10.1109/ICPR.2008.4761374.*
9. **Zou, J., Ji, Q., & Nagy, G. (2007).** A comparative study of local matching approach for face recognition. *IEEE Trans. on Image Proc.,* pp. 2617–2628, doi: 10.1109/TIP.2007.904421.
10. **Breitenstein, M.D., Kuettel, D., Weise, T., Van Gool, L., & Pfister, H. (2008).** Real-time face pose estimation from single range images. *IEEE Conference on Computer Vision and Pattern Recognition,* doi: 10.1109/CVPR.2008.4587807.
11. **Seemann, E., Nickel, K., & Stiefelhagen, R. (2004).** Head pose estimation using stereo vision for human-robot interaction. *Proc. 6th Int. Conf. AFGR*, Seoul, Korea, doi: 10.1109/AFGR.2004.1301603.
12. **Valenti, R., Sebe, N., & Gevers, T. (2008).** Simple and efficient visual gaze estimation. *Workshop on Multimodal Interactions Analysis of Users in a Controlled Environment.*
13. **Makinen, E. & Raisamo R., (2008).** Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol. 30, No. 3, doi: 10.1109/ICSPC.2007.4728416.

14. Yang, Z., Li, M., & Ai, H. (2006). An Experimental Study on Automatic Face Gender Classification. *Proc. 18th IEEE Int'l Conf. Pattern Recognition,* Vol. 3, pp. 1099–1102, doi: 10.1109/ICPR.2006.247.

15. Ravi, S. & Wilson, S. (2010). Face Detection with Facial Features and Gender Classification Based On Support Vector Machine. *International Journal of Imaging Science and Engineering.*

16. Yang, M.H., Kriegman, J., & Ahuja, N. (2002). Detecting Faces in Images: A Survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 24, No. 1, doi: 10.1109/34.982883.

17. Zadeh, L.A. (2010). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-3, No. 1, pp. 28–44, doi: 10.1109/TSMC.1973.5408575.

18. Yang, M.H., Kriegman, D. & Ahuja, N. (2002). Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 24, No. 1, pp. 34–58, doi: 10.1109/34.982883.

19. Hjelmas, E. (2001). Face Detection: A Survey. *Computer Vision and Image Understanding,* Vol. 83, pp. 236–274, doi: 10.1006/cviu.2001.0921.

20. Nasrabadi, A. & Haddadnia, J. (2010). Face detection base on fuzzy skin region segmentation. *Education Technology and Computer (ICETC), 2nd International Conference,* Vol. 5, pp. 22–24, doi: 10.1109/ICETC.2010.5530048.

21. Robert, M. & George, P. (2007). *Facial Geometry, Graphic Facial Analisis for Forensic Artist.* Charles Thomas Publisher.

22. Grenander, U. (1996). *Elements of Pattern Theory.* Baltimore: Johns Hopkins University Press.

23. Smisek, J., Jancosek, M., & Pajdla, **T. (2011).** 3D with Kinect. *IEEE International Conference on Computer Vision Workshops (ICCV Workshops),* pp. 1154–1160, doi: 10.1007/978-1-4471-4640-7_1.

24. Burrus, N. (Online) *RGBDemo Ver. 0.5, Available Kinect calibration*. Available: http://burrus.name/index.php/Research/KinectCalibration

25. Freedman, B., Shpunt, A., Machline, M. & Arieli, Y. (2012*).* *Depth mapping using projected patterns*. US Patente 8,150,142,B2.

26. Hartley, R. & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision.* 2 ed.*,* Cambridge,

27. Bowyer, K., Chang, K., & Flynn, P. (2006). A survey of pproaches and challenges in 3D and multi-modal 3D+2D face recognition. *Computer Vision and Image Understanding,* Vol. 101, No. 1, pp. 1–15, doi: 10.1016/j.cviu.2005.05.005.

28. Samir, C., Srivastava, A., & Daoudi, M. (2006). Threedimensional face recognition using shapes of facial curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 28, No. 11, doi: 10.1109/TPAMI.2006.235.

29. Gokberk, B., Dutagaci, H., Ulas, A., Akarun, L., & Sankur, B. (2008). Representation plurality and fusion for 3D face recognition. *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 38, No, 1, pp. 155–173, doi: 10.1109/TSMCB.2007.908865.

30. Llonch, R.S., Kokiopoulou, E., Tosic, I., & Frossard, I. (2008). 3D face recognition using sparse spherical representations. *19th International Conference on Pattern Recognition (ICPR 2008),* pp. 1–4, doi: 10.1109/ICPR.2008.4761682.

31. Bevilacqua, V., Casorio, P., & Mastronardi, G. (2008). Extending hough transform to a points cloud for 3D-Face Nose-Tip detection. *4th International Conference on Intelligent Computing (ICIC 2008)*, Shanghai, China, pp. 15–18, doi: 10.1007/978-3-540-85984-0_144.

32. OpenNI (Online). *OpenNI*. Available: http://www.openni.org/.

33. Microsoft (Online). *Microsoft Kinect SDK.* Available: http://www.microsoft.com/en-us/kinectforwindows/.

34. OpenKinect (Online). *OpenKinect.* Available: https://github.com/OpenKinect/libfreenect/.

35. Smisek, J., Jancosek, M., & Pajdla, T. (2011). 3D with Kinect. *IEEE ICCV Workshops,* pp. 1154–1160.

36. Stoyanov, T., Louloudi, A., Andreasson, H., & Lilienthal, A. (2011). Comparative evaluation of range sensor accuracy in indoor environments*. Eur. Conf. Mobile Robots*, pp. 19–24,

37. Khoshelham, K. & Elberink, S. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors,* Vol. 12, No. 2, pp. 437–1454, doi: 10.3390/s120201437.

38. Dutta, T. (2012). Evaluation of the Kinect sensor for 3-D kinematic measurement in the workplace. *Appl. Ergonom,* Vol. 43, No. 4, pp. 645–649, doi: 10.1016/j.apergo.2011.09.011.

39. Obdrzalek, S., Kurillo, G., Ofli, F., Bajcsy, R., Seto, E., Jimison, H., & Pavel, M. (2012). Accuracy and robustness of Kinect pose estimation

in the context of coaching of elderly population. *IEEE EMBC,* pp. 1188–1193, doi: 10.1109/EMBC.2012.6346149.

40. **Comaniciu, D. & Meer, P. (2002).** Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 24, No. 5, pp. 603–619, doi: 10.1109/34.1000236.
41. **Jenke, P., Wand, M., Bokeloh, M., Schilling, A., & Straßer, W. (2006).** Bayesian Point Cloud Reconstruction. *Computer Graphics Forum,* Vol. 25, No. 3, pp. 379–388, doi: 10.1111/j.1467-8659.2006.00957.x.
42. **Meers, S. & Ward, K. (2009).** Face Recognition Using a Time-of-Flight Camera. *Computer Graphics, Imaging and Visualization,* pp. 377–382, doi: 10.1109/CGIV.2009.44.

Victor Fernández-Cervantes received his Ph.D. from the Center for Research and Advanced Studies of the IPN (CINVESTAV), Guadalajara campus, in 2014. His research interests are serious games, fuzzy logic, pattern recognition, and rehabilitation.

Arturo García is a Ph.D. student in Computer Science. He received his M.Sc. from the Center for Research and Advanced Studies of the IPN (CINVESTAV), Guadalajara campus, in 2012. His research interests are fuzzy logic, image recognition, and optimization.

Marco Antonio Ramos received his Ph.D. from the University of Toulouse, France. He is a research professor of Artificial Intelligence and Virtual Reality at the Autonomous University of Mexico State. His research interests include animation techniques, artificial life, artificial intelligence, distributed system, and intelligent agents.

Andrés Méndez received his Ph.D. in Computer Engineering at University of Florida, USA in 2008. He is a research professor at the Center for Research and Advanced Studies of the IPN (CINVESTAV), Guadalajara campus. His research interests include machine learning and data mining, artificial intelligence, computer vision, analysis of algorithms, and numerical optimization.

Article received on 24/11/2014; accepted on 17/04/2015. Corresponding author is Victor Fernández Cervantes.