Fuzzy model based on RGBD images to identify biometrical facial geometry.

MC. Victor Fernández
Cinvestav Gdl
Zapopan Jalisco México
vfermamd@gdl.cinvestav.mx

Dr. Andres Mendez
Cinvestav Gdl
Zapopan Jalisco México
amendez@gdl.cinvestav.mx

Dr. Marco Antonio Ramos
UAEM
Toluca estado de México
mramos@univ-tlse1.fr

Psic. Monica Gudiño
MG Capital Humano SC
Guadalajara Jalisco México
direccion@mgcapitalhumano.com.mx

Abstract— **Automatic face recognition is an important and complex task. Therefore, it is necessary to design efficient algorithms able to cope with all the different problems of face recognition. Thus, a great progress has been done using the actual vision technology. For example, a lot of topography description of the facial geometry information has not been used for the lack adequate technology. In this paper, we present an algorithm to recognize the human face, and it uses inputs from RGBD images provided by the camera and depth sensor, Kinect. The images are translated to a point cloud structure to recognize facial geometry patterns on the topography with fuzzy rules set describing the pattern algebra of each point shape identifying the connectivity pattern with their neighbors. First, it finds the nose tip which is segmented with concentric squares to evaluate symmetry. Then the face is segmented in eight different areas. The algorithm was tested in an interview scenario with a person with free range of movements.**

Keywords—kinect, RGBD, Fuuzzy logic, face detection, face segmentation.

I. INTRODUCTION.

While face recognition algorithms commonly assume that face images are well align, and have similar poses. It is almost impossible in the real world to have this kind of constraints. [1,2,3]. Certain work has been done trying to overcome this limitation [2]. A common approach is to discretize the poses of the face, and learn to detect each pose separately [1]. Several studies rely on statistical models of face shape and active appearance models [3]. 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 [3]. In general these methods, based on 2D images, are sensitive to changing illumination, lack of features and partial occlusions [4]. [5] Presents a system based on fuzzy neural network that combines the histograms of color and depth information. Another method [6] detects human face using a combination of state of the art eye center locators and corner locations. This system estimates the visual gaze of a user in a controlled environment.

One of the challenging topics is the identification of the human face in an automatic way. This has many important applications in authentication, domotics, robotic, and security [7,8,9,10,11]. Main issues of this problem are face detection and gender classification. Here, several methods rely in the selection of the color space to detect skin region, face detection and gender classification. The connection between face detection and gender classification are examined experimentally in [12]. [13] Reports a detailed analysis of how different normalizations can affect gender classification accuracy. They had two different methods for alignment and gender classifiers: support vector machines and two-layer Real Adaboost classifiers [14].The most interesting findings, in the context of supervised learning for face recognition [13], are that the support vector machine which performed better than other classifiers. In addition, resolution of the face did not affect the classification rate [14]. Other works [15,16,17,18,19] use vector color models to separate the RGB images in order to do 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, as depth. For this, we need new algorithms to handle this new information. For example, we can adopt face geometry features standards, as cephalometric points, to describe the human face [20]. Thus each face is unique, there are ridges and crevasses are all wrapped around a spherical skull making the face an irregular terrain. Still, there are certain constant landmarks in the human face [20] than can be used. These points allow us to create 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 [20]. In our work, we described this information in terms of fuzzy patterns. This means that the input RGBD image from a Kinect device is turn into a cloud of point. This will be explained later in section IV. This representation draws its inspiration form Pattern Algebra Theory [21]. To describe the relation of each point with their neighbors, a rule-based fuzzy membership functions, based on the depth and color information, is used. Afterwards, concentric squares are created using facial information to help in the problem of occlusion using symmetric properties of the face [20]. Then, using the point cloud configuration, we can describe contours to obtain facial areas.

The present paper has the following structure: in section II, we review the Kinect device advantages for gathering visual information. In the section III, we describe facial geometry theory on which this paper is based. Section IV describes the proposed point cloud model. In section V, we present the idea behind contour points. In section VI, we show results of the cloud point model to detect facial geometry. Finally, in section VII conclusions are presented and possible future improvements.

II. KINECT.[1]

In order to obtain RGBD images, with 3D depth information, we used the Kinect device. It consists of a pattern projector IR and an IR camera, which are used to triangulate points in the space. They work together like a depth and color camera (RGB). Therefore, it can be used to recognize the image content and the texture of the 3D points. For measurements, the Kinect delivers three outputs: an IR image, RGB and depth image [22,23,24,25]. Actually, the Kinect has become the most practical way to capture RGB images with depth maps in real time [25]. It provides information of the face shape to different actions having similar 2D projections together with a single view projection on tree orthogonal Cartesian planes [26,27,28,29]. However, the detection of a 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 [30]. Also high noise levels are still present in the data capture [30].

III. FACIAL GEOMETRY.

The cephalometric points are quite important because are directly related to the underlying skeletal craniometric points. However, even the faces are not identical or even similar. Nevertheless, they have the same pattern in the distribution of the cephalometric points. These points are divided in cranial points Fig. 1 (denoted by facial geometry by v, sg, g, eu, at), lateral points (zy, go), orbital points(ec, en, il, im, or), nasal points(n, na, prn, sn, al), labial points(sls, ls, stm, li, ch), mental points(lm. pog, gn), auricular points (sa, sba, para, pa, tr) [20].

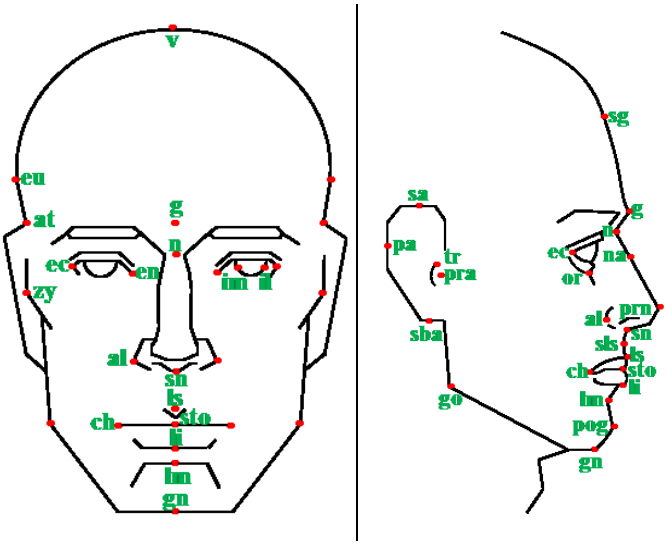


Figure 1. Cephalometric points directly related to the underlying skeletal craniometric points.

This means that areas of the face are contoured, and a true mathematical treatment of this surface would require solid geometry. The four most useful facial planes are midsagittal plane (MSP), midfacial plane (MFP), transverse nasal plane (TNP) and transglabellar plane (TGP). The eight areas of the face are 1) frontal area, 2) nasal area, 3) labial area, 4) mental area, 5) orbital area, 6) zygomaxillary area, 7) buccomandibular area, 8) auricular area. Fig. 2.

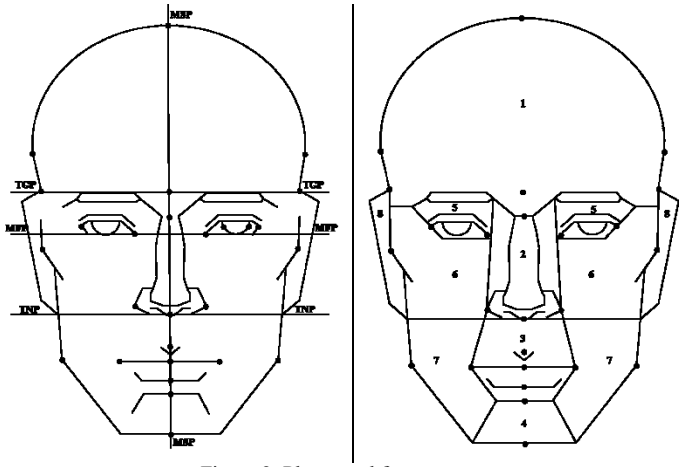


Figure 2. Planes and face areas.

IV. POINT CLOUD MODEL.

The point cloud model is the data structure that our algorithm uses to recognize human face from an RGBD image. Each point in the set $P = [p_1, p_2, p_3, p_4, \dots, p_n]$ has been associated with three integers (x, y, z); $x, y, z \in \mathbb{Z}$, which will later be interpreted as coordinates in the plane. The number of points necessary to represent the information captured from the sensors RGB and IR is $n = 640 \times 480 = 307200$ [23]. In order to insure that this only change the information representation and not its values [21], 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 Pattern Theory [21], which preserves distance and color information [23]. In addition, each point is associated to an index α. This α represents the cluster of points to which the point belongs.

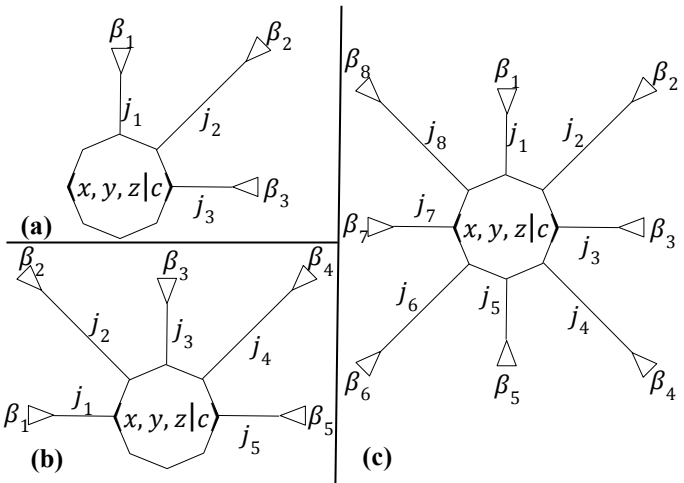


Figure 3. (a) Corner point; (b) edge; (c) interior points.

[1] Kinect consists of Infra-red (IR) projector, IR camera and RGB camera

For example, the nose and iris clusters. 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 [21]. In order to cluster all these point representations together into a point cloud, each point possess a series of bonds $j_1, j_2, \ldots, j_\omega$. [21]. Where $\omega = \omega(p)$ means the bond-connectivity of p. This means that the number of bonds can vary from point to point. We have three different groups: edges points, corners points and interior points. The graphic representation of the points is shown in the 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, edge points and interior points are invariant under rotations. Associated to each point ρ, there is a Cartesian coordinate (x, y) and a 8-neighborhood, 5-neighborhood or 3-neighnorhood set. These sets are denoted $N_8(\rho)$, $N_5(\rho)$ and $N_3(\rho)$. Their connections are denoted by the connectivity matrix β_{8x8} shown in Tab. 1. This connectivity matrix is used to connect the different points in the complete image. This connectivity matrix shows how they can be connected or disconnected. Furthermore, each bond j_i has a conector β_i, which is the representation of the connection value with a neighbor. The value "1" indicates the connection complete, and "0" indicates the connection is null. The connection validity is represented by the connectivity matrix $\boldsymbol{\beta}$.

TABLE 1. Connectivity matrix β.

β	β_8	β_7	β_6	β_5	β_4	β_3	β_2	β_1
β_1	0	0	0	1	0	1	0	0
β_2	1	0	1	0	1	0	1	1
β_3	0	1	0	1	0	1	0	1
β_4	1	0	0	0	0	0	1	0
β_5	0	0	0	0	0	1	0	1
β_6	0	0	0	0	0	0	1	0
β_7	0	0	0	0	0	1	0	0
β_8	0	0	0	0	1	0	1	0

Once this idea of connectivity is defined, the cloud of point model makes use of the depth information as follows. An integer ∂ is defined as the depth difference between two neighboring points, and it is calculated by the Eq. 1. This term is calculated for every bond connection generated by β.

$$\forall \beta_i : \exists\, \partial_i \rightarrow \partial_i \epsilon \beta_i = z \epsilon \rho_{(x,y)} - z \epsilon \rho_{(x,y)\beta_{8x8}}. \quad (1)$$

This allows defining a threshold to decide if there is a connection between points. The configurations define a way to characterize how points should be grouped to create different regions. Like in chemistry, the atoms (points) are connected together into molecules (configurations), and the nature of the chemical bonds, ionic, covalent, and so on, decides what combinations of atoms will be stable enough to form molecules. Using this idea, it is possible to get specific configurations from a set of points $p_1, p_2, \ldots, p_i, \ldots, p_n$. Here, the subscript i will be used to identify a seed point called a coordinator point p_i. Their bonds are denoted by $j_1, j_2, \ldots, j_\omega$, $\omega = \omega(p_i)$. Then, the configuration uses a connector function σ in Eq. 2 to describe the bond connection in the set of points.

$$\boldsymbol{c} = \sigma(p_1, p_2, \ldots, p_n). \quad (2)$$

Where $\boldsymbol{c}$ is the configuration of a 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 Eq. 2. Separating each point by the respectably 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 $ext(\boldsymbol{c})$ [21]. 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. 3.

$$\rho: BxB \rightarrow \{TRUE, FALSE\} \quad (3)$$

Each connector has two fuzzy functions. The first one describes the topographic shape of the point based on different ∂ depth values. These values are described by a fuzzy membership function, and their own fuzzy rule set system. An example of the membership functions used to define the linguistic variable the depth difference between two neighboring points Fig. 4. The minimum ∂_{min} and the maximum ∂_{max} are obtained by the 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 show in Fig. 5. Finally, the set of fuzzy rules are 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, highest*. For this, the p^∂ set is defined Eq. 4 to describe the fuzzy differences between point distances. Eq. 5 shows the fuzzy membership function between allowed distances. $\mu_{p^\partial}: \partial \rightarrow [0,1]$,

$$p^\partial = [\text{lowest, verylow, low, normal, high, veryhigh, highest}] \quad (4)$$

$$\sum_i \frac{\mu p(\partial_i)}{\partial_i} \quad (5)$$

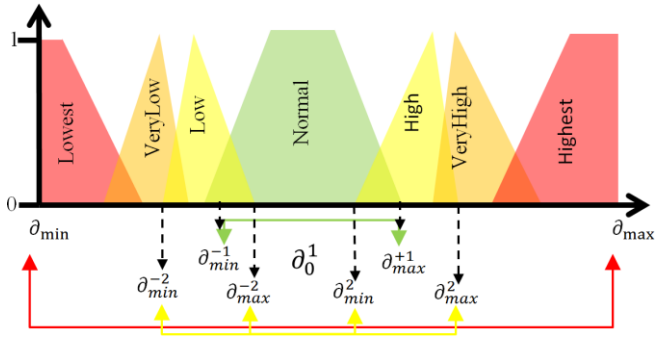
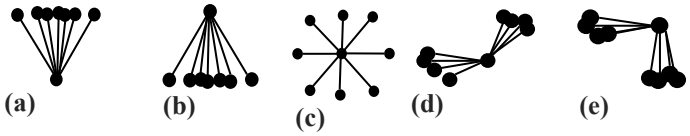


Figure 4. Depth difference fuzzy membership function.

This fuzzy rule-based system is composed of if-then rules. This set of rules is used to generate the topography description of the connected points. Fuzzy logic type-I is used because is easy to describe the pattern on the face in terms of valid topography points where variations are present. Some of these possible variations in configuration are show in the Fig. 5.



σ=sinking. σ=valley. σ=flat. σ=elevation. σ=slope.

Figure 5. some of the σ configuration to each point p

Using experimentation and the available data, the following fuzzy rules r were designed. $\boldsymbol{r_1}$: ∂_1 is p$^{\partial}$, $\boldsymbol{r_2}$: ∂_2 is p$^{\partial}$, $\boldsymbol{r_3}$: ∂_3 is p$^{\partial}$, $\boldsymbol{r_4}$: ∂_4 is p$^{\partial}$, $\boldsymbol{r_5}$: ∂_5 is p$^{\partial}$, $\boldsymbol{r_6}$: ∂_6 is p$^{\partial}$, $\boldsymbol{r_7}$: ∂_7 is p$^{\partial}$, $\boldsymbol{r_8}$: ∂_8 is p$^{\partial}$, $\boldsymbol{r}: p$ is σ. Where ∂ is the depth information and p$^{\partial}$ is the membership function Eq. 5. Examples of the rules for the configurations σ=flat, σ=slope Fig. 5c, 5e are:

1. If (∂1 is normal) ∧ (∂2 is normal) ∧ (∂3 is normal) ∧ (∂4 is normal) ∧ (∂5 is normal) ∧ (∂6 is normal) ∧ (∂7 is normal) ∧ (∂8 is normal) then (p is flat).
2. If (∂1 is Lowest) ∧ (∂2 is Lowest) ∧ (∂3 is Lowest) ∧ (∂4 is Lowest) ∧ (∂5 is normal) ∧ (∂6 is normal) ∧ (∂7 is normal) ∧ (∂8 is normal) then (p is slope).

For example, rule 1 describes the flat configuration where there is no great difference between the bonds have. An example of this is the area around the tip of the nose. The flat connector is described in the Fig. 5c. To recognize the color of the face is necessary to use the skin, hair and eyes color. Skin color has been proven to be an effective feature for the face detection introduced by [19]. It is more, each persons have different skin, hair and eyes color, and many studies have demonstrated that the major difference lies largely between their intensity rather than the chrominance [15,16,17,18]. Given RGB color space $p_c = \{r_p, g_p, b_p\}$, where $r_p, g_p, b_p \in [0..255]$. Then, given the color of a set points belonging to the skin $P_c = \{p_{1c}, p_{2c}, \dots, p_{nc}\}$, the histogram of $P_c, H(P_c)$, represent the probability distribution of each p_{ic}, for $i = 1,2,\dots,n$. $H(P_c)$ is normalized by P, then $H(P_c)$ take a color space P_c into the interval [0,1]; the fuzzy set can be defined as the pair (P_c, H), $H: P_c \rightarrow [0,1]$, is the normalized histogram. For each $p_c \in P_c$, $H(P_c)$ is the grade of membership of p_c so that $p_c \in (P_c, H) \leftrightarrow p_c \in P_c \; AND \; H(p_c) \neq 0$. Thus, the skin and not-skin classes membership function are mapped into a space color red, green and blue. The training has been performed using 806 images captured with Kinect. The membership function is modeling as a Gaussian function:

$$\mu_{SKIN_i}(P_c) = \beta_i e^{-\frac{(p_c-\alpha_i)^2}{2\sigma_i^2}} \tag{6}$$

Where $\{p_{c_R}\, p_{c_G}\, p_{c_B}\} \in [0..255]$; $i = \{R, G, B\}$; $\beta_i = \max\left(H(P_{c_i})\right)$, $\alpha = argmax\, H(P_{c_i})$ and σ_i^2 is the variance of each fuzzy set p_{c_i}. For the non-skin color a variation of model introduced by [31]. The membership value of a point to the object is determined by applying an S-function and Z-function to each color point:

$$\mu^{Z}_{NON-SKIN_i}(p_{c_i}) = \begin{cases} 0 & for\ p_{c_i} \le a_{si} \\ 2\left(\frac{p_{c_i}-a_{si}}{\gamma_{si}-a_{si}}\right)^2 & for\ a_{si} \le p_{c_i} \le b_{si} \\ 1-2\left(\frac{p_{c_i}-\gamma_{si}}{\gamma_{si}-a_{si}}\right)^2 & for\ a_{si} \le p_{c_i} \le b_{si} \\ 1 & for\ \gamma_{si} \le p_{c_i} \end{cases} \tag{7}$$

$$\mu^{Z}_{NON-SKIN_i}(p_{c_i}) = \begin{cases} 1 & for\ p_{c_i} \le a_{zi} \\ 1-2\left(\frac{p_{c_i}-a_{zi}}{\gamma_{zi}-a_{zi}}\right)^2 & for\ a_{zi} \le p_{c_i} \le b_{zi} \\ 2\left(\frac{p_{c_i}-\gamma_{zi}}{\gamma_{zi}-a_{zi}}\right)^2 & for\ a_{zi} \le p_{c_i} \le b_{zi} \\ 0 & for\ \gamma_{zi} \le p_{c_i} \end{cases} \tag{8}$$

The values b_{si}, b_{zi} are the cross-over points of the fuzzy sets defined by $\mu^{Z}_{NON-SKIN_i}(p_{c_i}), \mu^{Z}_{NON-SKIN_i}(p_{c_i})$, respectively, for $i = \{R, G, B\}$; the membership value of $p_{c_i} = b_{si}$ or b_{zi} that is equal to 0.5; $a_{si} = \gamma_{zi} = argmax\, H(p_{c_i})$; a_{zi}=0.5▪γ_{zi}; and $\gamma_{zi} = 1.5$▪a_{si}. $r\_p, g\_p, b\_p \in [0..255]$;

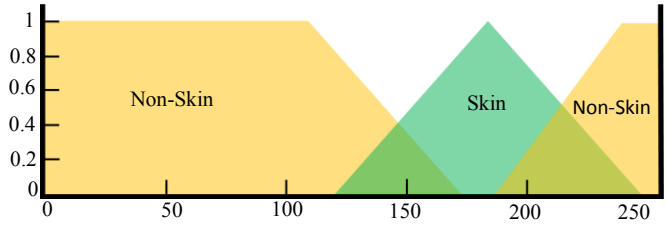


Figure 6. Membership function of the space color red.

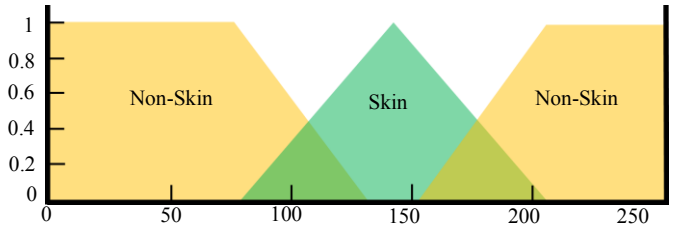


Figure 7. Membership function of the space color green.

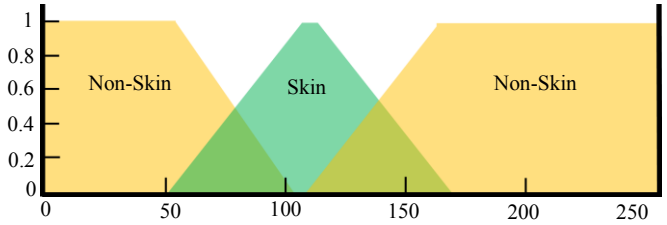


Figure 8. Membership function of the space color blue.

The results for the models of the fuzzy skin and non-skin classes are show in Fig. 6, 7 and 8, we have three classes for each color fuzzy set. The components $R, G, B.$ are fuzzified, according to the parameters defined in Eq. 6, 7 and 8. Then the inference system processes each point, and using available knowledge in the form of IF-THEN rules, it identifies and classifies skin color point in the point cloud.

V. AUTHOMATIC DETECTION OF THE FACE.

Trough experimentation, the present work has been able to discover that the necessary configurations necessary to describe the face F are at most 30 under the point cloud model Fig. 10. Each configuration is used for describing the position of cephalometric points in Fig. 9. This allows seeing the face as the union of al 30 configurations Eq. 9.

$$F = \{C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5 \cup C_6 \cup C_7 \cup C_8 \cup C_9 \cup C_{10} \cup C_{11} \cup C_{12} \cup C_{13} \cup C_{14} \cup C_{15} \cup C_{16} \cup C_{17} \cup C_{18} \cup C_{19} \cup C_{20} \cup C_{21} \cup C_{22} \cup C_{23} \cup C_{24} \cup C \cup C_{26} \cup C_{27} \cup C_{27} \cup C_{28} \cup C_{29} \cup C_{30}\} \quad (9)$$

Now, it is necessary to split up the face space F into several level sets, F^{β} [21]. Here, β is the variation over the same point space A. This allows having the partitions Eq. 10.

$$F = \bigcup_{\beta \epsilon A} F^{\beta} \quad (10)$$

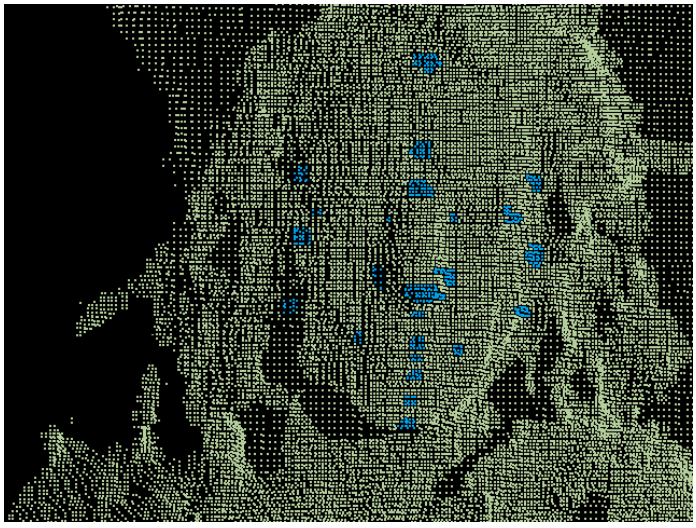


Figure 9. Cephalometric points in the point cloud.

Therefore, detection of high salient points depends on high values of β. Assuming the nose tip as high salient point, it is possible to use this feature to detect the face [23]. Using this idea, it possible to consider the nose tip as a region of more than one exact point. Thus, the entire point cloud is treated as a potential region of the nose tip Fig. 10. From the geometric point of view, the face may be represented as convex, concave, or a point cloud describing the curvature of the face [32]. Inspired by this, a candidate region to be the nose tip should maintain relative and symmetrical proportions as described in the facial geometry [20]. In order to use this fact, it is necessary to have the density of the point cloud in length units. Thus, it has been necessary to define a mean distance m_i of a point p_i within a set $\omega(p_i)$ Eq. 11 for the neighbors p_i' of a point.

$$m_i: \frac{1}{|\omega(p_j)|} \sum_{p_j' \epsilon \omega(p_j)_i} |p_j - p_j'| \quad (11)$$

With the mean m_i we know the number of points needed to find a configuration for the nose tip.

$$C_{NT} = \begin{bmatrix} \sigma_{-2,2} & \sigma_{-1,2} & \sigma_{0,2} & \sigma_{1,2} & \sigma_{2,2} \\ \sigma_{-2,1} & \sigma_{-1,1} & \sigma_{0,1} & \sigma_{1,1} & \sigma_{2,1} \\ \sigma_{-2,0} & \sigma_{-1,0} & \sigma_{0,0} & \sigma_{1,0} & \sigma_{2,0} \\ \sigma_{-2,-1} & \sigma_{-1,-1} & \sigma_{0,-1} & \sigma_{1,-1} & \sigma_{2,-1} \\ \sigma_{-2,-2} & \sigma_{-1,-2} & \sigma_{0,-2} & \sigma_{1,-2} & \sigma_{2,-2} \end{bmatrix} \quad (12)$$

Where C_{NT} is the nose tip configuration. Each σ_i is described by rule-based fuzzy system Eq. 12. There, the configuration is compared with the point cloud. We treat each point as a potential configuration with the shape and skin membership class function. Some of this are:

if ($\sigma_{0,0}$ is plane) ∧ (p_c is Skin) then (p_i is NosePoint).
if ($\sigma_{0,1}$ is plane) ∧ (p_c is Skin) then (p_i is NosePoint).
if ($\sigma_{1,1}$ is plane)∧(p_c is Skin) then (p_i is NosePoint).
if ($\sigma_{0,2}$ is elevation)∧(p_c is Skin)then(p_i is NosePoint).
if ($\sigma_{1,2}$ is elevation)∧(p_c is Skin) then (p_i is NosePoint).

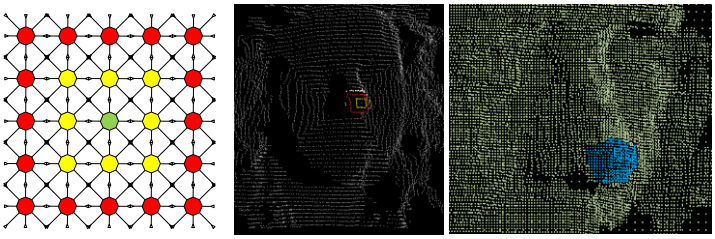


Figure 10. Concentric square configuration.

The final algorithm (1) compares the square symmetry configuration in the point cloud. The algorithm uses the concentric squares shape Fig. 10, because is easy to measure the symmetry between the points [33].

Algorithm 1: 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
```

VI. CONTOUR POINTS.

With the point cloud data structure, we can recalibrate the fuzzy function in Eq. 5, 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 Eq. 2. of each area of the face. A path between two points p and q in $\mathbb{R}^3$ is a sequence of distinct points $\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_n$ such that $\Upsilon_0 = \text{initial point}$, 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 called a contour point is a border point Υ_i such that divide the points in two subsets, interior and exterior Fig. 11. Thus, a point Υ_i, is similar to a point P, but with only one input bond, and one output bond Fig. 11. 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 show in Eq. 3. Thus, for a pair $\{\beta', \beta''\}$ of bond values the only one Input bond, and only one output bond. The pair is either regular, if $\rho\{\beta', \beta''\} = \text{TRUE}$, and all other bonds are irregular if $\rho\{\beta', \beta''\} = \text{FALSE}$. This partition divided BxB into two subsets. Thus, ρ is equivalent to a relation that separate interior points from the exterior points. Now, assuming that ρ is invariant, in the sense that if for two internal bonds $\{i', j'\}$ of $p_{i'}$ and $\{i'', j''\}$ of $p_{i''}$.

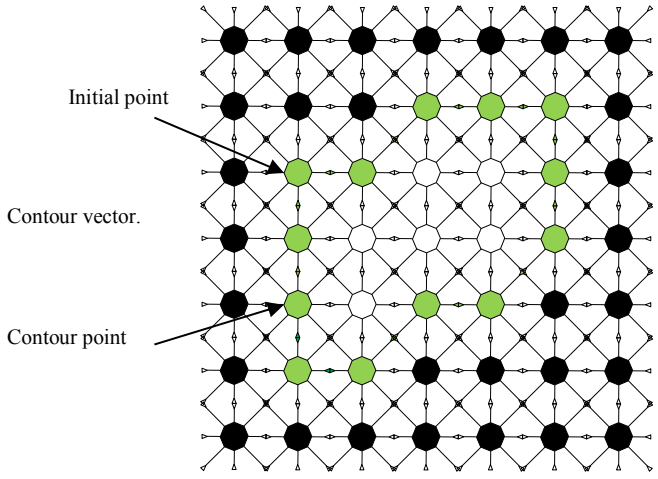


Figure 11. Contour points.

This allows the definition of the contour vector Γ_i as a sequence of contour points Υ_i Eq. 13. Then, the initial contour Υ_0 point is the same than the ending contour point Υ_k.

$$\Gamma = (\Upsilon_0, \Upsilon_1, \ldots, \Upsilon_{k-1}). \quad (13)$$

The contour contains the necessary information in the face shape. The proposed algorithm disregard interior areas, and it helps to solve the main problem of the face recognition (transposition, rotation) because it is invariant to these transformations. The contour contains the necessary information in the face shape.

Algorithm 2: Finding match for the configuration description.
Entrance: connectivity vector F = [F_1,F_2,…F_n].
Exit: configuration of the connection σ

```
    Create σ = 0
  for i = 1 until i =||F|| - 1
  for j = i + 1 until j = ||F||
  If can establish the connection with Fi, Fj
      σ = σ U (Fi,Fj)
  end of loop z
  return σ
```

VII. EXPERIMENTS AND RESULTS.

The algorithm was proven in a single face under possible different position of the face. These different poses can be seen in Fig. 16. In addition, it was necessary to impose distance constraints from the device to the subject. The distance used was between 0.8 meters to 1.40 meters Fig. 12. Under these constraints, the data produced by device was adequate for the algorithm. Once calibration was done, the proposed algorithm was able to find correctly the cephalometric points in the face. This can be seen in Fig. 13. Then, using these points, it was possible to follow the face contours, and divide the face in four contours Fig. 14. Finally the face is segmented in eight parts. This is shown in Fig. 15. Even though, there is occlusion by hair, the proposed algorithm was capable 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 the individual or to obtain other information. An example of this can be information about feelings.

Figure 12. Original image RGBD.

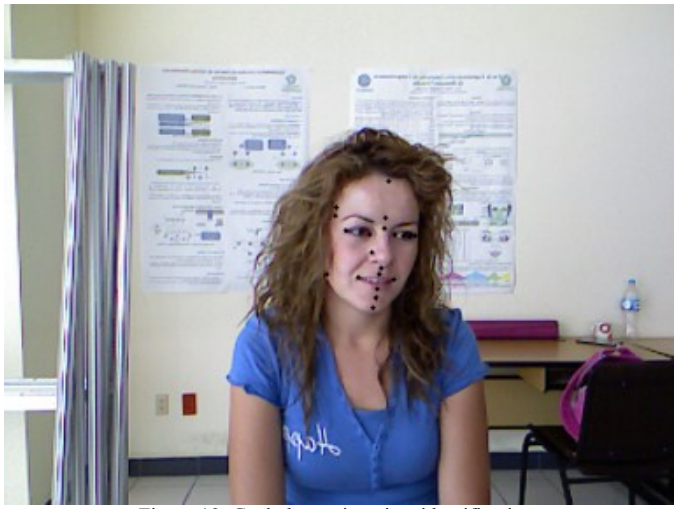


Figure 13. Cephalometric points identification.

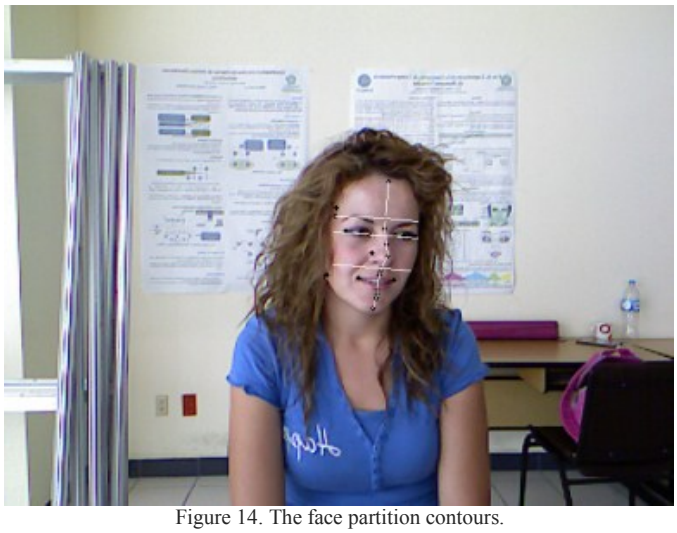


Figure 14. The face partition contours.

Figure 15. Segmentation on eight areas of the face.

VIII. CONCLUSIONS.

This work presents a novel algorithm to deal with the RGBD images provided by the Kinect sensor system. This novel algorithm was tested against a subject under behavior without movement restriction Fig. 16. Under these lax conditions, the novel algorithm 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 the different gestures in the facial expression. Therefore, 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 been adapted to different granularities, but it is necessary to design the configuration template before executing the algorithm. For future work, a fuzzy type-II is being designed to increase the robustness of the proposed algorithm.

Figure 16. Different configuration with the face tracking

IX. REFERENCES.

[1] S. Xie, S. Shan, X. Chen, and W. Gao, "V-LGBP: volume based local gabor binary patterns for face representation and recognition," in ICPR, 2008.

[2] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition," in *Proc. Int. Conf. Comput.Vis.*, pp. 786–791, Oct. 2005.

[3] J. Zou, Q. Ji, and G. Nagy, "A comparative study of local matching approach for face recognition," *IEEE Trans. on Image Processing*, 16(10), pp. 2617-2628, 2007.

[4] Michael D. Breitenstein and Daniel Kuettel and Thibaut Weise and Luc van Gool and Hanspeter Pfister : "Real-time face pose estimation from single range images, " *IEEE Conference on Computer Vision and Pattern Recognition 2008.*

[5] E. Seemann, K. Nickel, and R. Stiefelhagen, “Head pose estimation using stereo vision for human-robot interaction,” in *Proc. 6th Int. Conf. AFGR*, pp. 626–631, Seoul, Korea, May 17–19, 2004.

[6] Valenti, R. and Sebe, N. and Gevers, T., "Simple and efficient visual gaze estimation", *Workshop on Multimodal Interactions Analysis of Users in a Controlled Environment*, 2008

[7] Rieko Inaba, Eriko Watanabe and Kashiko Kodate. "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, 2003.

[8] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, pp. 711-720, 1997.

[9] Blanz, V., Vetter, T. “Face recognition based on fitting a 3d morphable model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9), pp. 1063–1074, 2003.

[10] Kare, S., Samal, A., Marx, D. “Using bidimensional regression to assess face similarity,” *Machine Vision and Applications,* 21(3), pp. 261–274, 2008.

[11] Singh, R., Vatsa, M., Noore, A. “Face recognition with disguise and single gallery images.” *Image and Vision Computing*. 27(3), pp. 245–257 2009;

[12] Erno Makinen and Roope Raisamo, “Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces”, *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 30, N0. 3, March 2008.

[13] Z. Yang, M. Li, and H. Ai, “An Experimental Study on Automatic Face Gender Classification,” *Proc. 18th IEEE Int’l Conf. Pattern Recognition*, vol. 3, pp. 1099-1102, Aug. 2006.

[14] S.Ravi S.Wilson. “Face Detection with Facial Features and Gender Classification Based On Support Vector Machine,” *Special Issue - International Journal of Imaging Science and Engineering*, 2010.

[15] M.H.Yang, J. Kriegman and N. Ahuja,"Detecting Faces in Images:A Survey,"*IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, 2002.

[16] Zadeh, Lotfi A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *Systems, Man and Cybernetics, IEEE Transactions on* , vol.SMC-3, no.1, pp.28,44, Jan. 1973.

[17] Ming-Hsuan Yang; Kriegman, D.; Ahuja, N., "Detecting faces in images: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.24, no.1, pp.34,58, Jan 2002.

[18] E. Hjelmas, "Face Detection: A Survey", *Computer Vision and Image Understanding*, vol. 83 , pp. 236-274, 2001.

[19] Nasrabadi, A.; Haddadnia, J., "Face detection base on fuzzy skin region segmentation," *Education Technology and Computer (ICETC), 2010 2nd International Conference on* , vol.5, no., pp.V5-379,V5-382, 22-24 June 2010.

[20] Robert M. George, Ph.D. “Facial Geometry, Graphic Facial Analisis for Forensic Artist,” Charles Thomas Publisher, LTD. 2007.

[21] Ulf Grenander. “Elements of Pattern Theory,” Baltimore : Jhons Hopkins University Press, 1996.

[22] Smisek, J.; Jancosek, M.; Pajdla, T., "3D with Kinect," *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, vol., no., pp.1154,1160, 6-13 Nov. 2011.

[23] N. Burrus. (2011, Nov.) RGBDemo Ver. 0.5, [Online], Available Kinect calibration. http://burrus.name/index.php/Research/KinectCalibration.

[24] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. “Depth mapping using projected patterns,” U.S. Patenet 8,150,142,B2, Apr. 3, 2012.

[25] R. Hartley and A. Zisserman. “Multiple View Geometry in Computer Vision,” Cambridge, 2nd edition, 2003.

[26] K.W. Bowyer, K. Chang, and P. Flynn. “A survey of pproaches and challenges in 3D and multi-modal 3D+2D face recognition”. *Computer Vision and Image Understanding*, vol. 101(1), pp. 1-15. Jan. 2006.

[27] C. Samir, A. Srivastava and M. Daoudi. “Threedimensional face recognition using shapes of facial curves” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, Nov. 2006.

[28] B. Gokberk, H. Dutagaci, A. Ulas, L. Akarun, B. Sankur. “ Representation plurality and fusion for 3D face recognition.” *IEEE Trans. on Systems, Man, and Cybernetics*, part B. vol. 38, no. 1, pp. 155-173. Feb. 2008.

[29] Llonch, R.S.; Kokiopoulou, E.; Tosic, I.; Frossard, P., "3D face recognition using sparse spherical representations," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* , vol., no., pp.1,4, 8-11 Dec. 2008.

[30] Vitoantonio Bevilacqua, Pasquale Casorio, and Giuseppe Mastronardi. “Extending hough transform to a points cloud for 3D-Face Nose-Tip detection.” 4th International Conference on Intelligent Computing, ICIC 2008 Shanghai, China, September 15-18, 2008.

[31] Murthy, C.A., Pal, S.K. “Fuzzy thresholding: Mathematical framework, Bound functions and weighted moving average technique,” *Pattern Recognition Letters* 11, 197–206 (1990).

[32] P. Jenke and M. Wand and M. Bokeloh and A. Schilling and W. Straßer, "Bayesian Point Cloud Reconstruction", *Computer Graphics Forum vol. 25, Issue 3,* pages 379–388, September 2006.

[33] Meers, S.; Ward, K., "Face Recognition Using a Time-of-Flight Camera," *Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference on* , vol., no., pp.377,382, 11-14 Aug. 2009.

[34] John Yen, Reza Langar. “Fuzzy Logic, Intelligence, Control and Information,” Prentice-Hall, Inc. 1999.