THE IMAGING SCIENCE JOURNAL, 2017
https://doi.org/10.1080/13682199.2017.1358528

3D face detection based on salient features extraction and skin colour detection using data mining

Abdul Rahman El Sayed[a], Abdallah El Chakik[b], Hassan Alabboud[c] and Adnan Yassine[a,d]

[a]Mathematics Department, Normandie University, UNIHAVRE, LMAH, Le Havre, France; [b]Computer Science Department, Beirut Arab University, Tripoli, Lebanon; [c]Business and Administration Faculty, Lebanese University, Tripoli, Lebanon; [d]Institut Supérieur d'Etudes Logistiques, Université du Havre, Le Havre, France

ABSTRACT
Face detection has an essential role in many applications. In this paper, we propose an efficient and robust method for face detection on a 3D point cloud represented by a weighted graph. This method classifies graph vertices as skin and non-skin regions based on a data mining predictive model. Then, the saliency degree of vertices is computed to identify the possible candidate face features. Finally, the matching between non-skin regions representing eyes, mouth and eyebrows and salient regions is done by detecting collisions between polytopes, representing these two regions. This method extracts faces from situations where pose variation and change of expressions can be found. The robustness is showed through different experimental results. Moreover, we study the stability of our method according to noise. Furthermore, we show that our method deals with 2D images.

ARTICLE HISTORY
Received 10 April 2017
Accepted 7 July 2017

KEYWORDS
3D facial point clouds detection; 3D point clouds saliency; 3D point clouds segmentation; classification; 3D object processing; points clouds

Introduction

Face detection becomes an important research field for computer vision applications such as human recognition and verification, analysing emotions for multimedia tasks, and visual tracking. Nowadays, the development of biometric systems has become an important challenge for researchers. Fingerprints and iris techniques are the mostly used, but the face recognition seems to be the best approach especially in airports and critical zones.

Most of face detection approaches for 2D images present many drawbacks such as illumination variations, pose and facial expressions [1]. In this context, this paper presents an accurate method for face detection on 3D coloured point clouds due to their less sensitivity to lighting conditions and viewpoint. In this method, we convert the 3D coloured point cloud into a weighted graph, and then we detect faces using the skin and saliency features of vertices. This proposed method deals with any data that can be represented by a weighted graph such as 3D point clouds, 3D meshes and 2D images.

State of the art

In this subsection, we review existing approaches for face detection and recognition based on skin region detection or geometry features.

In [2], we presented a method based on data mining techniques for face detection in 2D images. We have demonstrated the robustness of the method by showing some experimental results and comparing them with other related works. In this work, we propose a generalized approach to deal with 3D coloured point clouds.

In [3], Ma, Xiao and Bin Ghazali proposed a method for face detection under varying illumination conditions. This method considers the colour image and the colour balance model to convert the RGB colour space to YCgCr colour space. The algorithm is realized by merging Gaussian skin colour model, template matching and face verification.

Regarding skin detection, in [4], Joenes and Rehg used a method for skin detection based on Statistical Colour Models in 2D images. Their method achieved a correct detection rate of 80%.

Unlike approaches proposed in [2–4] that deal only with 2D images, we present a generalized facial detection approach to deal any data that can be represented by weighted graph such as 3D coloured point clouds, 3D meshes and 2D images.

Colombo et al. [5] proposed a method for 3D face detection that identifies the subject's eyes and nose and classifies eye and nose regions. Nonetheless, the authors mentioned that this technique is highly sensitive to noises and to the presence of holes surrounding the regions of nose and eyes.

Many of available approaches [6,7] aim at localizing faces, rather than detecting them, given the presence of the face in the first place. In [6], the face localization is carried out by detecting the nose tip, and then the

CONTACT Abdallah El Chakik a.alshakik@bau.edu.lb

segmentation can be made by cropping the sphere where the nose tip is its centre. This approach is strongly limited to the dataset used, as every input 3D object is inferred to contain only one face. Furthermore, the radius of the cropping sphere has a fixed value over the entire dataset and therefore the segmentation cannot handle scale changes.

In [8,9], authors divided the view sphere into a number of sectors and collect templates for each view. Given a new candidate object, the face detection is obtained by template matching.

Kheirkhah et al. [10] presented a hybrid approach, based on the skin colour information and AdaBoost-based face detection. The main key points of the proposed method are background elimination and down-sizing using the adaptive skin colour classification and segmentation. Therefore, the skin colour segments candidate were searched instead of the whole image. Finally, they adopted the Viola-Jones AdaBoost-based face detector to detect the faces.

In [11], authors proposed a human face detection method based on skin colour and neural networks. The demarche consists of several steps. First, the system searches for the candidate regions and forms a so-called skin map. Next, the locations of pair eyes in the candidate regions after removing noise and applying some morphological operations are detected. Finally, a three-layer back-propagation neural network verifies each of the face candidate regions.

Recently in [12], authors proposed a method for 3D facial detection on point clouds using an SVM model. They used a k-NN approach and a threshold to determine if a given point belongs to the face surface.

In this work, we define a set of face candidate regions based on skin information, and then we confirm the presence of the face by analysing the geometry features. This method deals with outliers and the presence of small holes. Moreover, it produces good results with faces of different sizes, alignments and directions.

Contributions

Our main contribution in this work is to propose a new method to extract 3D face parts from a coloured point cloud represented by a weighted graph.

As mentioned above, most of the related works detect faces from 2D images or 3D mesh. These works rely on colour features, geometry features, or a mixture of colour and geometry features. Our method deals with any data that can be represented by a weighted graph such as 2D images and 3D meshes. In this method, we define the human face as a skin region containing gaps (non-skin regions) such as eyes, mouth and eyebrows. Furthermore, some parts of the human face can be seen as salient regions. Thus, we can define a human face as a skin region containing some salient regions.

Our method steps can be summarized as follows:

- Modelling the 3D coloured point clouds surface by an undirected weight graph.
- Estimating vertices normal.
- Identifying the skin vertices.
- Using a custom version of Hill Climbing algorithm to segment the graph vertices into skin, non-skin and salient regions.
- Detecting and locating gaps (non-skin regions) in skin regions.
- Defining the candidate human face region using skin and gap regions.
- Finding salient regions in each candidate skin region.
- Using the collision between polytopes representing gaps and salient regions, we confirm the face detection.

The following flowchart in Figure 1 illustrates our method.

Paper organization

The remainder of this paper is organized as follows. Section 'Surface modelling and normal estimation' presents the 3D point clouds surface modelling, Section 'Proposed method' details the approach used in our proposed method: Skin detection, Saliency detection and faces recognition based on collision detection. Section 'Experimental results' presents some experimental results. Section 'Comparison with the state of the art' presents a comparison with the state of the art. Finally, Section 'Conclusion' concludes this paper.

Surface modelling and normal estimation

Let us consider the general situation where a point cloud can be viewed as an undirected weighted graph $G_d(V, E, W_d)$ where V represents a finite set of vertices and $E \subset V \times V$ a finite set of edges. Each edge $(v_i, v_j) \in E$ that connects two vertices v_i and v_j has a set of weights W_d that contains W_c the weight associated with colour information and W_g the weight associated with compute saliency.

The set of neighbours $D_n(v_i)$ of a given vertex v_i is defined using the concept of Delaunay triangulation. Thus, we define circumscribed spheres from a given 3D point P_i such that no point is inside these spheres. Then, the set of points located on the circumscribed spheres borders P_j will be defined as the set of P_i neighbours. In other words, we define the list of P_i neighbours as the set of vertices located on the corners of all triangles (faces) sharing P_i's neighbours, as shown in Figure 2. Thus, the set of neighbours $D_n(v_i)$ of a graph vertex v_i

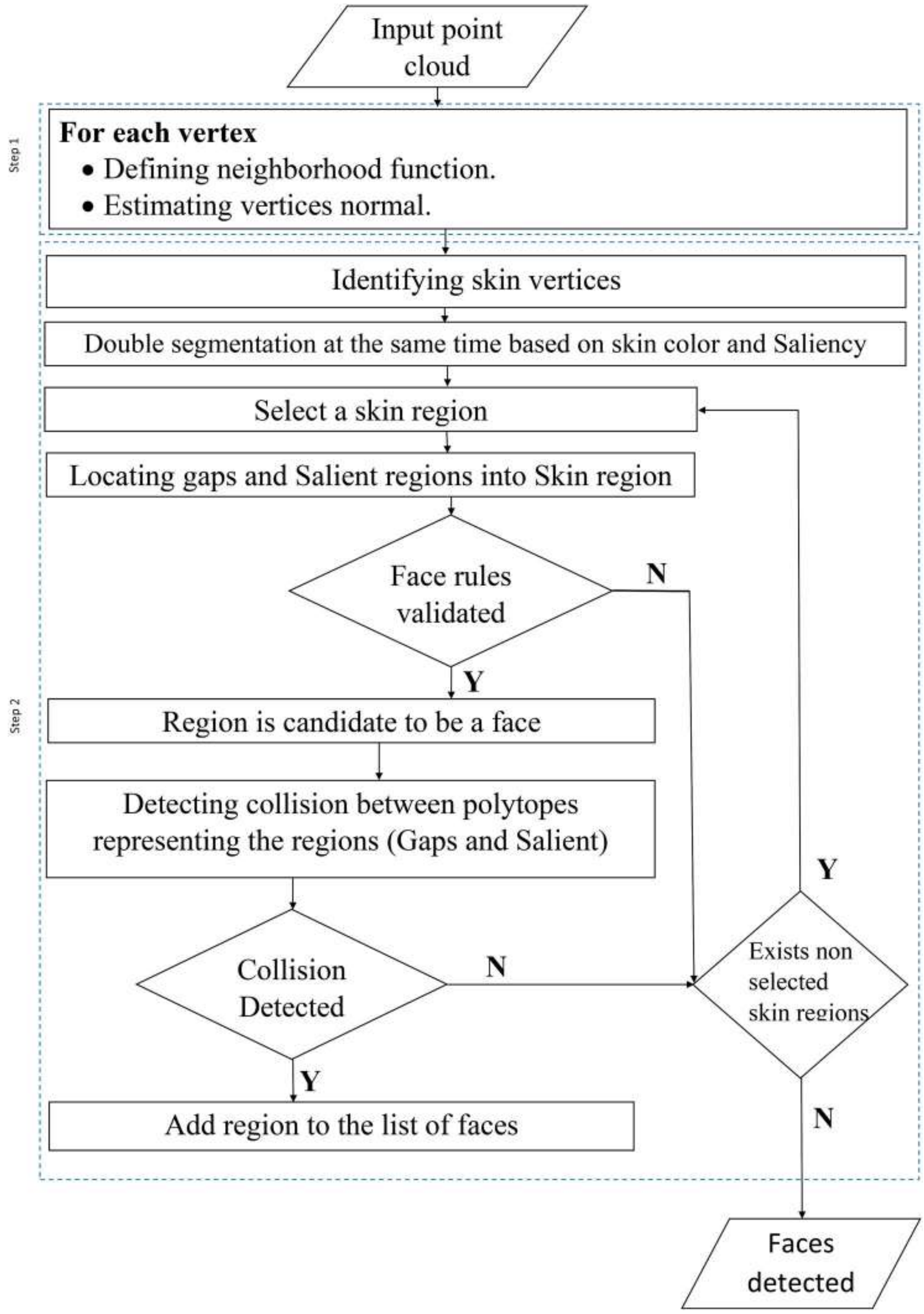


Figure 1. Flowchart of our method.

that represents a point P_i is defined as all vertices v_j representing P_j.

Then, we compute the normal vector $N(v_i)$ and the two-directional vectors for each vertex v_i following the *x*- and *y*-axes using the Eigen-values of the covariance matrix, as shown in Figure 3. Let $\alpha(v_i) = [N(v_i) \cdot t_1(v_i);\ N(v_i).t_2(v_i)]^T$ be the projection of the normal $N(v_i)$ on the tangent plane $(t_1(v_i);\ t_2(v_i))$ (see [13] for more details). Then, the covariance matrix is defined as follows:

$$\mathrm{Cov}(v_i) = \sum_{j \in D_n(v_i)} (v_j - \alpha'(v_i))(v_j - \alpha'(v_i))^T \in \mathbb{R}^{3\times 3} \quad (1)$$

where $\alpha'(v_i)$ represents the centroid of the projected normal.

Proposed method

Our method consists of detecting a human face using skin and geometry information. In this section, we present first some useful notations and the skin region detection method to generate face candidate regions. Then, we show the saliency degree calculation of vertices in the extracted candidate regions. Finally, we present the method of matching candidate regions based on colours and saliency measure.

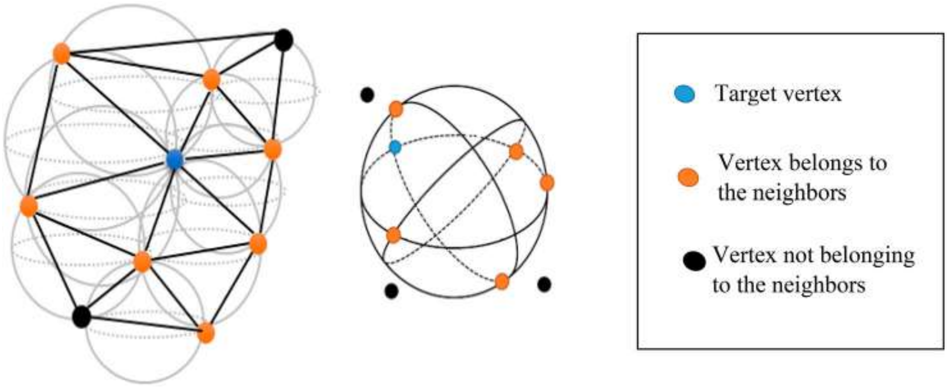


Figure 2. Illustration of the vertex neighbouring.

Notations and definitions

As we mentioned in the introduction, this work extends our work in [2] to deal with any data that can be represented by a weighted graph such as 2D images, 3D point clouds and 3D meshes. We begin this subsection by a brief summary of the method proposed in [2] followed by a review of some notations and definitions involved in this section.

To define a predictive rule to discriminate between skin and non-skin regions effectively, we have built a learning dataset and extraction decision rules. The demarche proposed consists of three steps:

- *Dataset construction*: we used a database composed of hundred images with different lighting conditions, races and sexes. This work led to a learning dataset consisting of 50,000 pixels of different classes (skin and non-skin).
- *Building learning file*: we created a learning data file (Table 1) to be used later on by some data mining tools to generate prediction rules.
- *Prediction rules generation*: to find the appropriate predictive model, we tried different techniques derived from inductive graphs J.48 and JRip rules using the tool WEKA 3.6.0. The resulting prediction rules based on RGB component of the learning data file lead to a definition of function called lambda $\lambda(v_r, v_g, v_b)$, which represents the components of vertices colour. See [2] for more details.

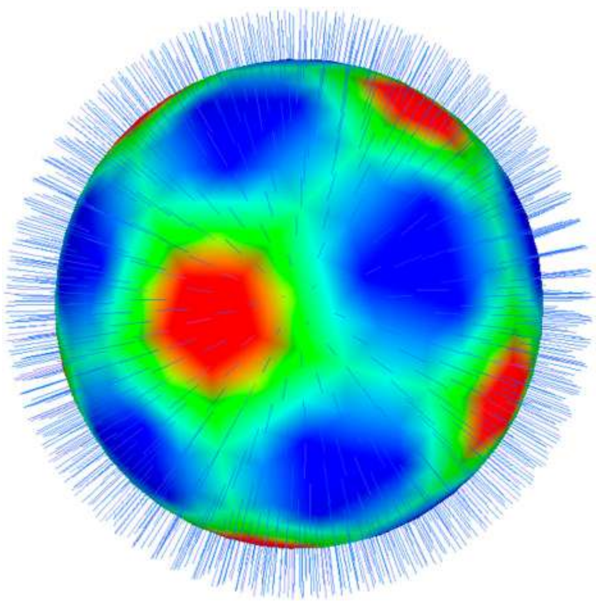


Figure 3. Vertices normal vectors and principal directions.

Let $\nabla(S)$ be a function defined as follows:

$$\nabla(S) = \mathrm{Card}\{v \in S \text{ such as } \lambda(v_r, v_g, v_b) = 1\} \quad (2)$$

where S is a set of vertices, $\nabla(S)$ the number of skin vertices in this set.

The associated weight function $\omega_c{:}V \times V \to [0, 1]$ that represents a homogeneity measure between a vertex and its neighbours based on skin colour is defined as follows:

$$\omega_c(v_i, v_j) = \begin{cases} \dfrac{\nabla(D_n(v_i))}{\mathrm{Card}\ \ (D_n(v_i))} & \text{If } \lambda(v_i^r, v_i^g, v_i^b) = 0 \\ 1 & \text{If } \lambda(v_i^r, v_i^g, v_i^b) = 1 \end{cases} \quad (3)$$

To classify graph vertices as skin and non-skin regions as shown in Figure 4, we define the two sets $F(G_d)$ (skin regions) and $B(G_d)$ (non-skin regions) as follows:

$$F(G_d) = \{v_i \in\ G_d \text{ such as } \lambda(v_i^r, v_i^g, v_i^b) = 1\} \quad (4)$$

$$B(G_d) = \{v_i \in\ G_d \text{ such as } \lambda(v_i^r, v_i^g, v_i^b) = 0\} \quad (5)$$

Table 1. Extract from learning data file: the rows in the table correspond to the representative pixels, while the columns correspond to the representation axes of the RGB space and the class (skin or no skin).

R	G	B	Class
X_1^r	X_1^g	X_1^b	X_1^C
…	…	…	…
X_K^r	X_K^g	X_K^b	X_K^C
…	…	…	…
…	…	…	…
X_N^r	X_N^g	X_N^b	X_N^C

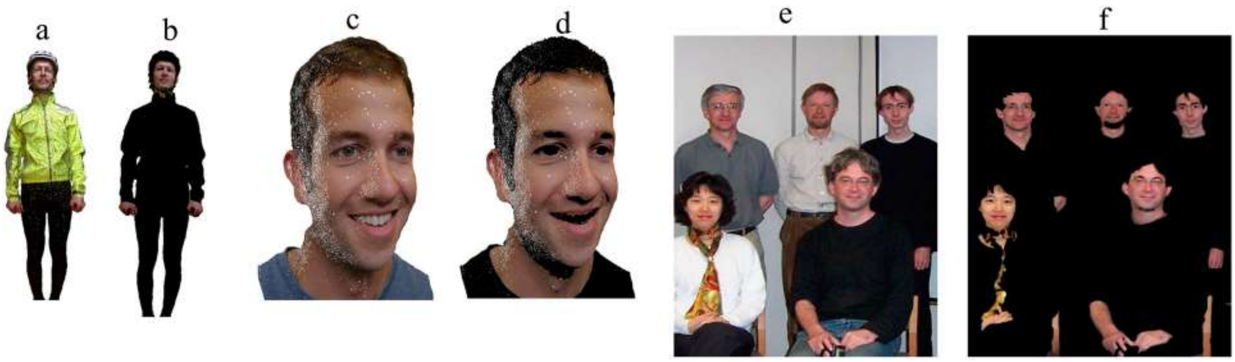


Figure 4. Skin detection with our approach images (a, c) shows the original 3D coloured point clouds, image (e) displays a 2D image, images (b, d, f) displays non-skin vertices in black.

The set of skin vertices $C(v_i)$ that represents the neighbours of a vertex v_i is defined as follows:

$$C(v_i) = \{v_j \in F(G_d) \text{ and } v_j \in D_n(v_i)\} \tag{6}$$

Let $E_q(v_i, v_j)$: $V \times V \rightarrow \{0, 1\}$ be a function that can be defined as follows:

$$E_q(v_i, v_j) = \begin{cases} 1 & 1 \text{ if } v_i \text{ and } v_j \text{ are considered skin} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

We define a function $H_m(v_i)$: $V \rightarrow \mathbb{N}$ that measures the homogeneity between v_i and its neighbours as follows:

$$H_m(v_i) = \sum_{vi \sim vj} E_q(v_i, v_j) \tag{8}$$

Candidate face region

The first phase in our method consists of defining a set of face region candidates based on skin information.

A skin region is considered as a face candidate if it belongs to the class $F(G_d)$ and contains at least three regions belonging to the class $B(G_d)$ (e.g. eyes, nose and mouth), as shown in Figure 5. The face candidate region must verify also the following conditions:

- $\exists 3$ regions non skin $B1$, $B2$ and $B3$ such as $B1 \sqsubset R$, $B2 \sqsubset R$, $B3 \sqsubset R$
- $\varphi(B1)/\varphi(R)$, $\varphi(B2)/\varphi(R)$ and $\varphi(B3)/\varphi(R) \in [0.05, \; 0.2]$
- $d(\theta(B1), \theta(B2)) \in [d0, d1]$
- $\Gamma(R, \theta(B1), \theta(B2), \theta(B3))$ is nearly an isosceles triangle.

where $\varphi(R)$ presents the area of the region R, $\theta(R)$ the gravity centre of the region R, and $\Gamma(R, p1, p2, p3)$ the triangle that connects $p1$, $p2$ and $p3$ in the region R.

Once we define the set of face candidate regions, we calculate the saliency degree of vertices in these regions to confirm the presence of a human face. To do so, we define a local matrix descriptor for vertices and we measure the deviation factor between tow vertices to compute finally the saliency degree at each graph vertex.

Deviation factor

To compute the saliency at a vertex v_i, we calculate the deviation factor that measures the fluctuation of the surface between two vertices v_i and v_j. Then, we construct patches by projecting the vertices onto a 2D plan and we fill the cells of patches associated matrix by the deviation factor values.

To measure the deviation factor $\mathrm{DF}(v_i, v_j)$, we compute the angle between the vertices of normal vectors using the following formula:

$$\boldsymbol{\theta}(v_i, v_j) = \arccos\left(\frac{(N(v_i) \cdot N(v_j))}{\sqrt{\left(\sum_{k=1}^{3} N(v_i)(k) * N(v_i)(k)\right) * \sum_{m=1}^{3} N(v_j)(m) * N(v_j)(m)}} \right) \tag{9}$$

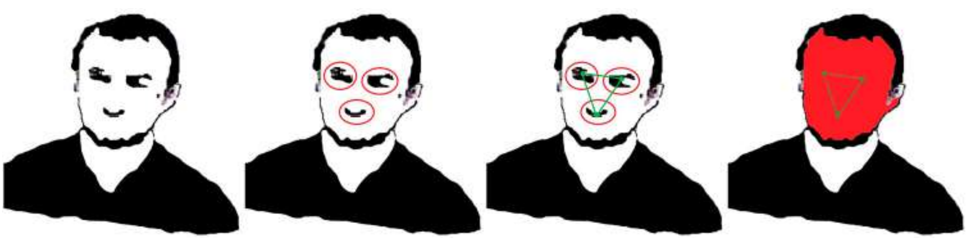


Figure 5. Skin region contains gap (skin in white and non-skin in black colour).

Then, we define the deviation factor between v_i and v_j as follows:

$$DF(v_i, v_j) = \frac{\theta(v_i, v_j)}{360} \times \lambda \tag{10}$$

where λ is a constant. If the deviation factor value of two vertices is almost zero, these two vertices are considered located in the same plane.

Local matrix descriptor

Once the face candidate regions are detected, we construct a patch to describe the local surface of each candidate region. To construct the local path descriptor of a vertex v_i, its neighbouring vertices (belonging to $D_n(v_i)$) are projected onto a 2D plan defined by the associated vectors (x and y), as shown in Figure 6.

As a result, we obtain a list of vertices v_j' for each projected vertex v_j and defined as follows:

$$v_j' = [(v_j - v_i) \cdot x(v_i), (v_j - v_i) \cdot y(v_i)]^T \tag{11}$$

To define a patch for a given vertex v_i, we construct a square grid centred at v_i according to its tangent plane. The patch length $L(v_i)$ is then defined as [14] follows:

$$L(v_i) = \max_{v_j \in D_n(v_i)} (||v_j - v_i||_2^2) \tag{12}$$

where $|| \cdot ||_2^2$ represents the Euclidean norm.

Thus, the patch at a vertex v_j' is represented by a rectangle of m cells where the side length of each cell is $L(v_i)/m$, where m is a constant that depends on applications.

Finally, we attach to the patch, a local matrix descriptor (LMD) of a size ($m \times m$). Then, we assign to each cell of the associated matrix LMD (v_i) the average of deviation factor of each vertex v_j according to v_i. To define the position of cells corresponding to the projected vertex v_j' of $v_j \in D_n(v_i)$ into the matrix *LMD* (v_i), we use the following equation:

$$\text{position}^d(v_j') = \left\lfloor \frac{v_j'^d}{L(v_i)/m} \right\rfloor \tag{13}$$

where $\lfloor\rfloor$ is the rounded integer.

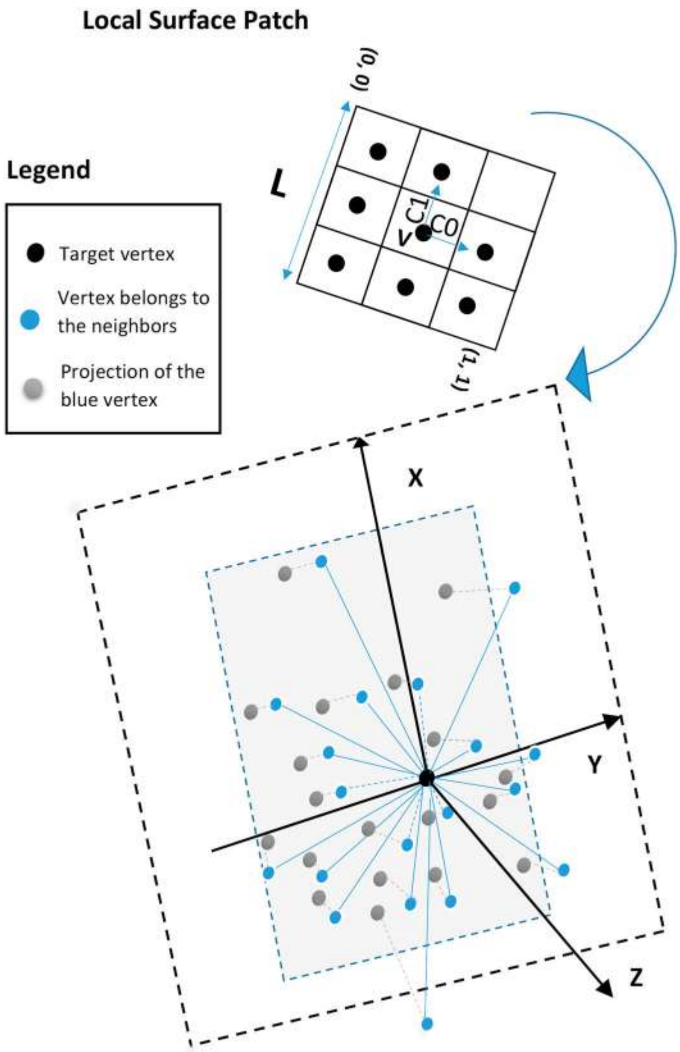


Figure 6. Illustration of the vertices projection.

Saliency computation

Once the patches are defined and the associated matrix is filled, we compute the saliency degree at a vertex v_i. To do so, we compute the patch saliency Patch_Saliency(v_i) using a similarity measure between the local matrix descriptor of $P(v_i)$ and the local matrix descriptor of patches associated with v_i's neighbours. Then, we calculate the saliency degree at v_i based on Patch_Saliency(v_i) and the weight functions of v_i and its neighbours.

Thus, the weight function $\omega_g{:}V \times V \rightarrow [0, 1]$ that represents a homogeneity measure between a vertex and its neighbours is defined as follows:

$$\omega_g(v_i,\ v_j) = \mathrm{e}^{-(K(v_j)*|A(v_j)-A(v_i)|)/\sigma(v_j)*\theta(v_i,v_j)} \tag{14}$$

where $k(v_j)$ represents the mean curvature [15] at v_j, $A(v_j)$ is the average of the local matrix descriptor LMD (v_j) elements, and $\sigma(v_i)$ is the scale parameter and can be calculated as follows:

$$\sigma(v_i) = \max_{v_k \in D_n(v_i)} (||v_k - v_i||^2) \tag{15}$$

To compute the local homogeneity for a given vertex, we compute the probability $P_r(v_i)$ as follows:

$$P_r(v_i) = \frac{C_x^i}{|V|} \tag{16}$$

where C_x^i is the number of vertices of $P(v_i)$ with a deviation factor (the value of its corresponding patch cell) greater than x (x is a constant that depends on applications) and $|V|$ is the number of vertices belonging to $P(v_i)$, as shown in Figure 7.

Then, we compute the entropy of v_i patch that measures the dissimilarity as follows:

$$\mathrm{entropy}(v_i) = -P_r(v_i) \times \log_2 P_r(v_i) \tag{17}$$

The saliency of v_i patch is computed as follows:

$$\mathrm{Patch\_Saliency}(v_i) = \frac{\sum_{j\in|V|} \mathrm{Abs}(\mathrm{entropy}(v_j) - \mathrm{entropy}(v_i))}{|\mathrm{Patches}|}, \tag{18}$$

where |Paches| is the number of distinct neighbour's patches.

Finally, the visual saliency of the vertex v_i on the 3D point clouds is defined as follows:

$$\mathrm{Saliency\text{-}degree}(v_i) = \frac{\sum_{j\in|V|} \mathrm{Patch-Saliency}(v_j) \times \ \omega_g(v_i, v_j)}{|V|} \tag{19}$$

The saliency degree is bounded between [0, 1] where the degree 1 means that v_i is very similar to its neighbourhood and 0 means the opposite.

Skin and saliency regions segmentation

During the segmentation process, a vertex will have two label holders, a label for the segmentation based on skin colours and a second label for the segmentation based on the saliency measure. The first step in our method consists of assigning the same label in the first and second label holder to each group of connected vertices that belongs to the class $F(G_d)$ and the similar vertices in terms of saliency. This step can be performed using a custom version of Hill Climbing algorithm. This algorithm uses a dilation morphological operator to propagate through weighted graph vertices and label homogenous regions in terms of skin and saliency similarities, respectively. Then, the algorithm traverses all graph vertices and creates a segment from each set of vertices having the same label. As a result, this algorithm produces two lists of segments (list of skin segments and list of salient segments) where a vertex can belong to skin and salient segment at the same time.

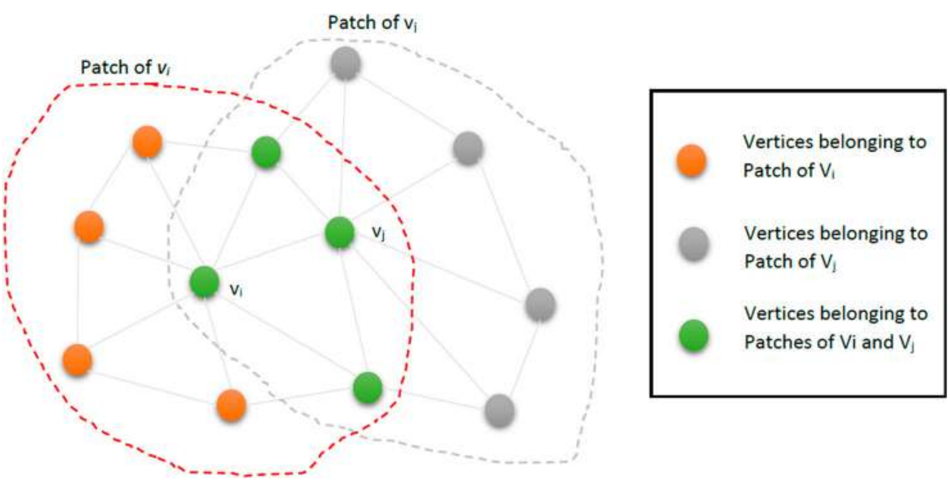


Figure 7. Vertices patches.

The main difference between current algorithm and the regular Hill Climbing algorithm is a multi-selection performed at each level. Furthermore, it avoids the local maxima solution because more than one solution can be found in the same iteration.

Multi-search hill climbing – algorithm

Let $C'(v_i)$ be a set of neighbouring vertices of a vertex v_i and similar to v_i in terms of saliency. We define a vertices labelling function $E'_q(v_i, v_j)$: $V \times V \rightarrow \{0, 1\}$ as follows:

$$E'_q(v_i, v_j) = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are similar in terms of saliency} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

We define a function $H'_m(v_i)$:V $\rightarrow \mathbb{N}$ that measures the homogeneity between v_i and its neighbours as follows:

$$H'_m(v_i) = \sum_{vi \sim vj} E'_q(v_i, v_j) \tag{21}$$

The custom version of Hill Climbing algorithm steps is listed as follows:

- Pick a random vertex v_i in the weighted graph.
- Consider all the neighbours $D_n(v_i)$ of the current vertex v_i.
  - o Choose the set of neighbours $C(v_i)$ from $D_n(v_i)$ that maximize the function $H_m(v_i)$.
  - o Apply the dilation operator from v to all vertices in $C(v_i)$.
  - o Choose the set of neighbours $C'(v_i)$ from $D_n(v_i)$ that maximize the function $H'_m(v_i)$.
  - o Apply the dilation operator from v_i to all vertices in $C'(v_i)$.
- Repeat 2 for each element in $C(v_i)$ and $C'(v)$ until $C(v_i)$ and $C'(v_i)$ are empty.

Figure 8 illustrates the behaviour of the precedent algorithm.

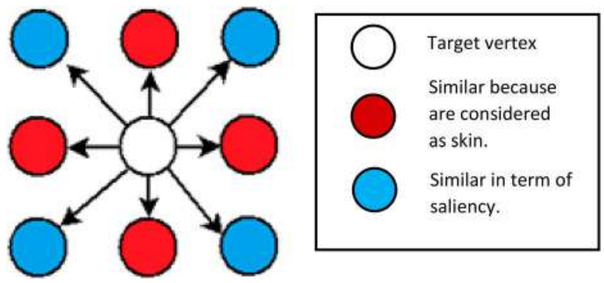


Figure 8. Illustration of Multi-Search Hill Climbing algorithm step.

Face detection method

In this subsection, we describe the process of matching candidate regions based on colours and saliency measure.

The definition of candidate face regions is based on gaps features (non-skin regions) located in skin regions. Therefore, each non-skin and salient region can be considered as a convex polytope, because it contains a finite number of vertices, it is bounded and closed [16]. Figure 9 presents some examples of polytopes.

The face detection process consists of checking whether there is a matching between the gap regions defining the face candidate regions and those of salient regions. The matching process is done by detecting collisions between polytopes representing these regions (gaps and salient regions).

Let R be a region in G_d (set of connected vertices), and $C_o(R) \subset \mathbb{R}^n$ where $n \in \mathbb{N}$ the set of polytope corners corresponding to R. According to Minkowski's Theorem of [17], a bounded and closed convex polytope $K \subset \mathbb{R}^n$ is the convex hull of its external points or corners.

Let S be the $C_o(R)$ of a region R. The convex hull of S is defined as

$$\mathrm{Conv}(S) = \left\{ \sum_{i=1}^{|S|} \alpha i \ xi | (\forall i: \alpha i \ \geq 0) \wedge \sum_{i=1}^{||S||} \alpha i \ = 1 \right\} \tag{22}$$

Let V and W denote $C_o(X)$ and $C_o(Y)$ of the two regions X and Y, respectively.

The Minkowski sum and the Minkowski difference of V and W are defined as follows:

$$V + W = \{v + w | v \in V, w \in W\} \tag{23}$$

$$V - W = \{v - w | v \in V, w \in W\} \tag{24}$$

Figure 10 shows a graphical example of the Minkowski sum of two sets in two-dimensional real space.

Let $D_s(S_1, S_2)$ be the function that represents the distance between two non-empty sets S_1 and S_2 which can be defined as follows:

$$D_s(S_1, S_2) = \inf\{d(x, y) \mid x \in S_1, y \in S_2\}, \tag{25}$$

where d is a metric defined on $\mathbb{R}^n$ and inf is the infimum.

To detect the collision between two polytopes U and V, we compute the distance between the convex hulls of both two polytopes. Thus, two polytopes U and V are collide if

$$D_s(\mathrm{Conv}(C_o(U)), \mathrm{Conv}(C_o(V))) = 0 \quad \text{where } 0 \in \mathbb{R}^n \tag{26}$$

As we mentioned before, both sets U and V can be represented by their convex hulls $\mathrm{Conv}(C_o(U))$ and $\mathrm{Conv}(C_o(V))$ of their corners, respectively. Therefore, formula (26) can be rewritten as

$$D_s(\mathrm{Conv}(C_o(U)), \mathrm{Conv}(C_o(V))) = 0$$
$$\Leftrightarrow 0 \in (\mathrm{Conv}(Co(U)) - \mathrm{Conv}(Co(V)))$$

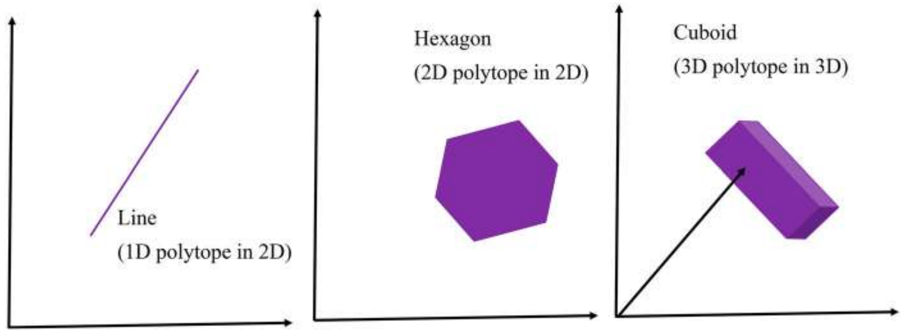


Figure 9. Examples of polytopes.

$$\Leftrightarrow 0 \in (\mathrm{Conv}(Co(U)) + \mathrm{Conv}(-Co(V)))$$

To find a representation of $0 \in \mathbb{R}^n$ created from the sum of two convex combinations of the same point where one combination is composed of points in $C_o(U)$ and the other one of points in $-C_o(V)$, we propose to use a linear system that can be written as follows:

$$\hat{A}X = 0$$

$$x \geq 0, \quad \sum_{i=1}^{|\mathrm{Co}(U)|} xi = 1, \quad \sum_{i=|C_0(U)|+1}^{|\mathrm{Co}(U)+|\mathrm{Co}(V)|} xi = 1$$

with two additional constraints:

$$a_{n+1}x = 1, \quad a_{n+2}x = 1$$

where $\hat{A} \in \mathbb{R}^{n\times(|C_0(U)|+|C_0(V)|)}$ that contains the elements of $C_0(U)$ and $-C_0(V)$, and

$$a_{n+1,i} = \begin{cases} 1 & \forall\, 1 \leq i \leq |S| \\ 0 & \forall (|S|+1) \leq i \leq (|S|+|T|) \end{cases}$$

$$a_{n+2,i} = \begin{cases} 0 & \forall\, 1 \leq i \leq |S| \\ 1 & \forall (|S|+1) \leq i \leq (|S|+|T|) \end{cases}$$

Finally, the linear system can be solved using the standard method of simplex and can be rewritten as

$$\begin{array}{ll} \text{Min} & f_A(x) = e^{T(b-Ax)} \\ \text{Subject to:} & Ax \leq b \\ & X \geq 0 \end{array}$$

where

$$A \in \mathbb{R}^{k\times m}, \; k = n+2, \; m = (|C_0(U))| + |C_0(V))|),$$

$$e = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ 1 \end{bmatrix} \in \mathbb{R}^k, \quad b = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \\ 1 \end{bmatrix} \in \mathbb{R}^k, \; k, m \in \mathbb{N}$$

Note that if the above linear program has a valid solution, then the two polytopes U and V are colliding.

Algorithm

In this subsection, we present an algorithm using the standard simplex method [18] to check the presence of collision between two polytopes U and $V \subset \mathbb{R}^n$.

Algorithm Check collision

- U, V: Polytopes
- Get the list of two polytopes corners.
- Check If Card $(Co(U)) = 0$ or Card $(Co(V)) = 0$, then there is no collision.
- Check if Card $(Co(U))$ = Card $(Co(V)) = 1$ and the two points are equals, then the two polytopes are colliding.
- Check if Card $(Co(U)) > 1$ or Card $(Co(V)) > 1$
- Solve the linear program come from formula (26).
- If the obtained minimal value is zero value, then the two polytopes are colliding.

End

Experimental results

3D point clouds and 3D meshes

In this section, we show the efficiency and robustness of our method by applying it on 3D meshes, and 3D point clouds. In our experiments, we used a sample of 140 coloured objects containing faces of different colour of skin, ages and positions. We will show and discuss the influence of different parameters on our approach such as graph Delaunay-level k, the variable x in Equation (16) that defines the deviation factor threshold in the calculation of patch entropy, and the number of patch cells m.

Figure 11 presents the results of our face detection method applied on a coloured point clouds. Images (a,e,k) show the original 3D point clouds, images (b,f, l) present the face regions candidates where the gaps defining the face are circled in red. Images (c,g,m) show the salient regions detected in the candidate face, using our method with the following parameters values ($k = 1$, $m = 9$ and $x = 12$). Images (d,h,n) display the detected faces by our method. The fuchsia areas on the 3D point cloud objects represent the detected faces. We can visually confirm the face detection in these point clouds.

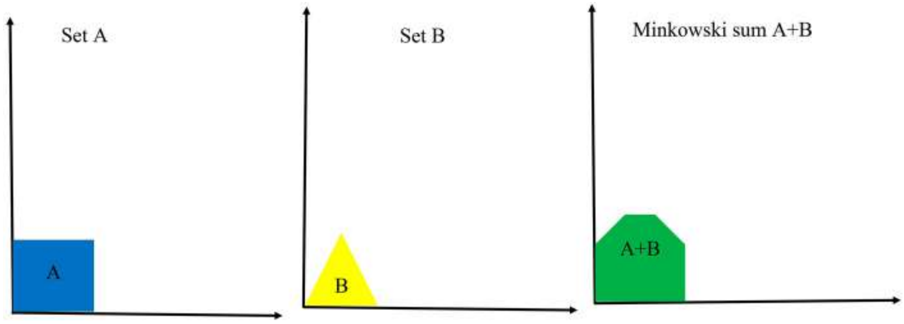


Figure 10. Examples of Minkowski sum of two sets in two-dimensional real space.

In Figure 12, images (a,c) present the original 3D coloured point cloud. Images (b,d) show the results obtained by applying first phase of our method (identifying the candidate regions). As shown below, two candidate regions are identified in the same object. The gaps defining the face are circled in red.

Figure 13 shows the result of applying the second phase of our method on the 3D point clouds presented in the previous paragraph (Figure 12). Image (a) displays the salient regions. The salient regions that collide with the gaps defining the candidates regions are circled in red. Note that there are salient regions located at the border of the second candidate region far than the gaps. Consequently, this candidate region is ignored.

Figure 14 presents the results of our method with a deviation factor $x = 9$.

As shown in Figure 14, we note that the number of salient regions increases when we decrease the deviation factor threshold. Practically, the similarity among the vertices relatively increases when the deviation factor threshold increases. This increase in the number of salient regions interprets the region candidate in the Figure 12(d) as human face where salient regions are detected near the gaps and collide with them.

Furthermore, we note that the number of salient regions with a local graph is greater than that of regions detected with a non-local graph due to the information of larger number of neighbours .

Figure 15 shows the result of applying our method on 3D coloured point clouds and meshes containing multi-faces. Images (a,c) show the original point clouds and images (b,d) show the results where the

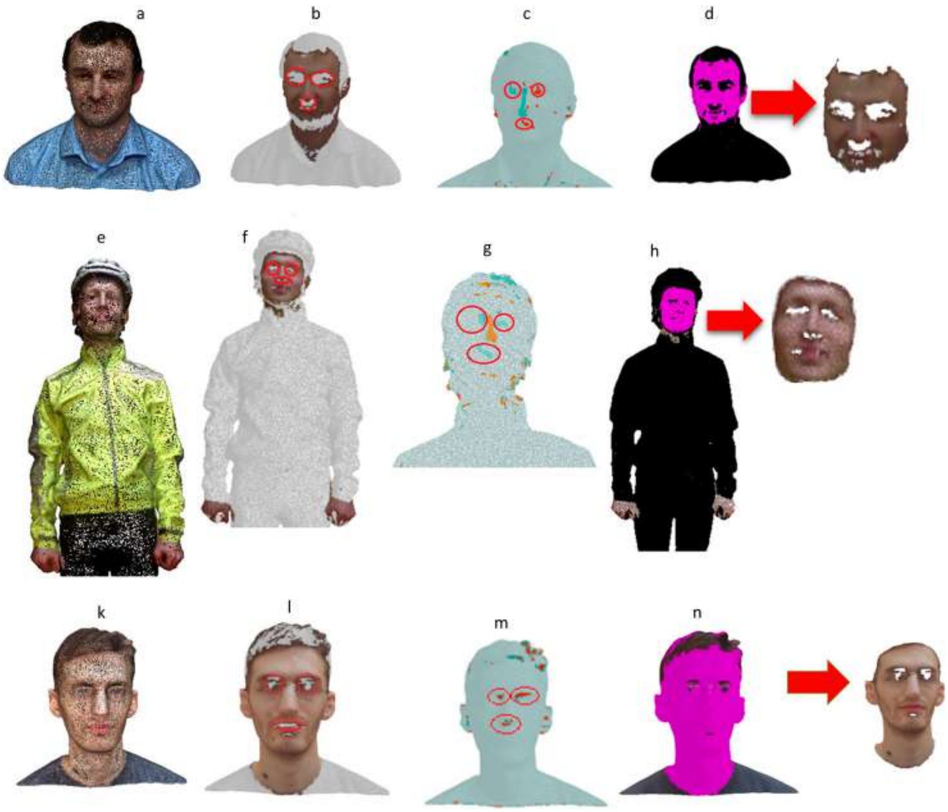


Figure 11. Results of applying our proposed model for face detection on some 3D coloured point clouds.

Figure 12. Two candidates regions are identified by applying the first phase of our proposed method.

region coloured with fuchsia are the detected faces. Image (e) shows the original 3D mesh and image (f) shows the detected faces.

Figure 16 presents the result of applying our method on the 3D mesh without colour. We can observe that the detected salient regions are located approximately in place of gaps that define the candidate face region. Thus, we can conclude that our method consists of two phases one to detect a face and the other to affirm the detection.

We tested our method on a sample of 140 3D coloured objects containing faces of different colour of skin, ages, and positions. These results are summarized in Table 2.

The accuracy of our proposed method is measured as follows:

$$\text{Accuracy} = 100 - (\text{FPR} + \text{FNR}) \tag{27}$$

where FPR (false positive rate) and FNR (false negative rate) can be computed as follows:

$$\text{FPR} = \frac{\text{number of false alarms}}{\text{number of dected faces}} \times 100 \tag{28}$$

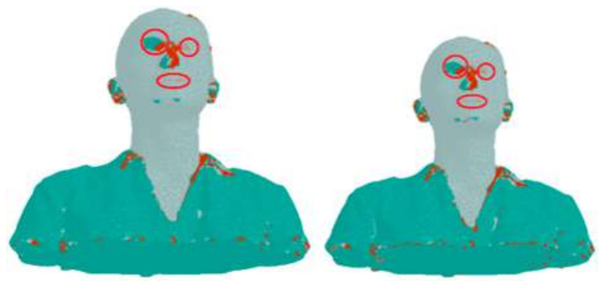


Figure 14. Influence of parameter deviation factor threshold x image a shows the salient regions detected with $x = 12$ and image b shows the salient regions detected with $x = 9$.

and

$$\text{FNR} = \frac{\text{number of missed faces}}{\text{number of all faces}} \times 100 \tag{29}$$

As mentioned in this section, the variation in the parameters x and k can affect the number of detected salient regions. Consequently, the error rate (FPR) is affected as shown in Figure 17.

2D images

As mentioned before, our method can be applied on any data that can be represented by a graph such as 2D coloured images and 3D point clouds. To deal with 2D images, we construct a *K-NN* graph, and then we construct the patch at a vertex v_i as a rectangle of m cells with a cell length $L(v_i)/m$. We assign to each patch at v_i the average of gradient of each vertex v_j according to v_i. The weight function ω_g is calculated as

$$\omega_g(v_i,\ v_j) \ = \ e^{-(|A(v_j)-A(v_i)|)/\ (\sigma(v_i)*\sigma(v_j))} \tag{30}$$

The figures below show some results of applying our method on some 2D coloured images. Note that a rectangle is drawn around the detected face region, as shown in Figure 18.

In Figure 19, all faces are detected using our method. Figure 20 presents a face detection from a picture with a complicated background.

Figure 21 demonstrates the efficiency of our method with images of different skin colours.

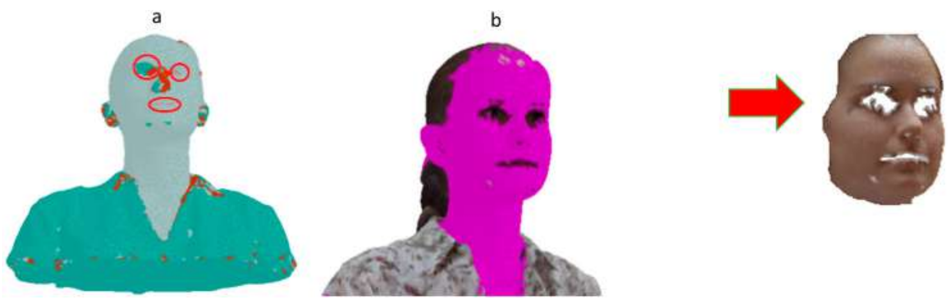


Figure 13. Two candidates regions detected by the first phase and a single face recognized after applying the second phase.

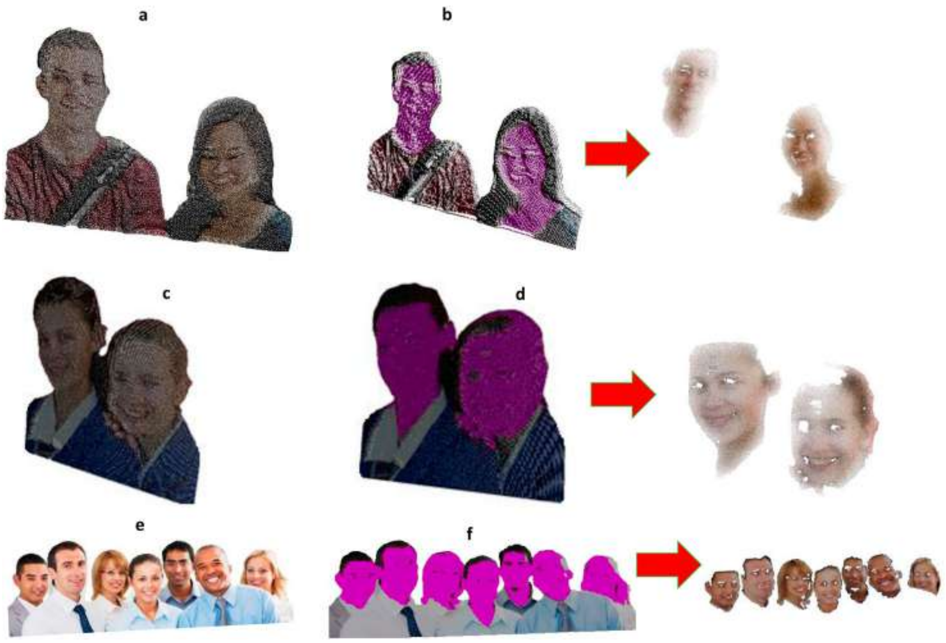


Figure 15. Using our method with 3D point clouds and meshes containing multi-faces.

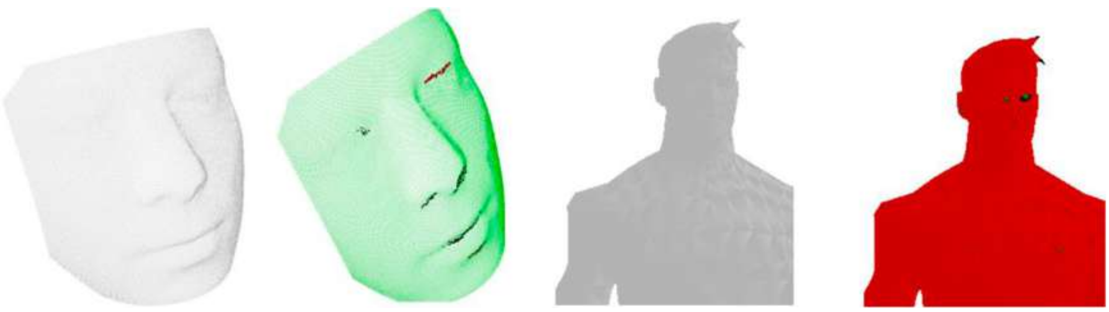


Figure 16. Using our method with 3D mesh not coloured.

Table 2. Experimental results.

Objects count	140
Number of all faces	380
Number of truly detected faces	361
False positive rate (FPR)	3.88%
False negative rate (FNR)	5%
Accuracy	91.12%

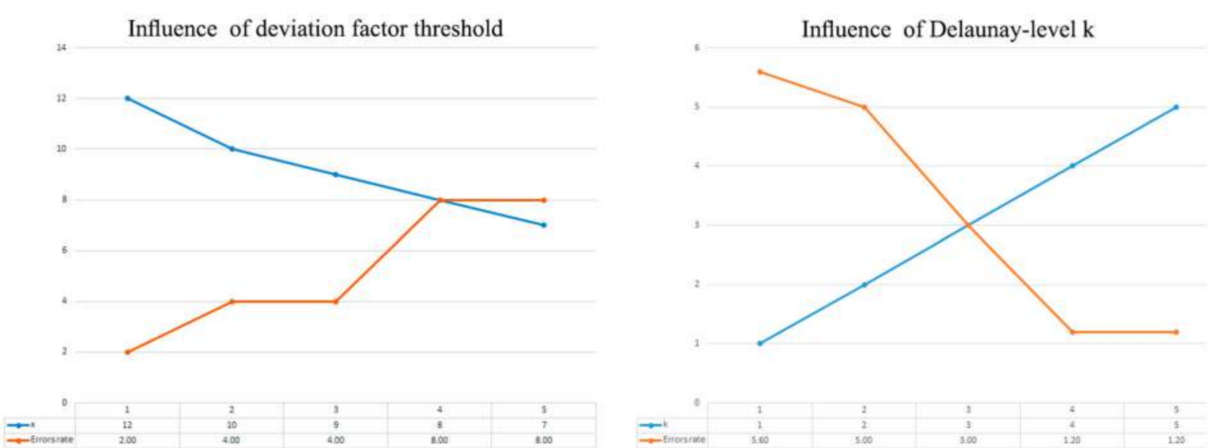


Figure 17. Influence of *x* and *k* on face detection process.

Figure 18. Using our method with 2D image.

Figure 19. Using our method with 2D coloured image containing multi-faces.

Figure 20. Using our method with 2D coloured image containing face and a complex background.

Figure 22 shows a failure case where an additional face is detected. This failure case is due to the colour of the Basketball that has been interpreted as a skin region. Then, the left hand with the Basketball has formed a region that verifies the face rules.

Finally, our experiments are made on a sample of 200 images with different lighting conditions, races and sexes chosen from 'Helen' dataset, 'Faces in the Wild' Dataset 'Caltech' image dataset and others according to the following parameter values ($k = 1$, $m = 9$ and $x = 12$) in addition to the following settings:

- [*NBFormsMin* = *3*, *NBFormsMax* = *20*]: Interval of number of non-skin regions in the face region.
- [*YShMin* = *0*, *YShMax* = *4*]: Differences between the ordinates of two eyes.
- [*MinDistance* = *5*]: Minimum distance between eyes.

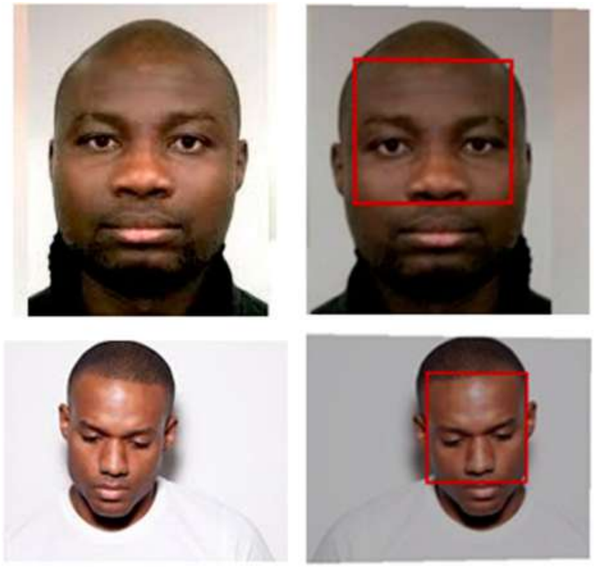


Figure 21. Using our method with 2D coloured image containing face of black skin.

Table 3 shows the statistical data.

Comparison with the state of the art

In this section, we compare our method with some related works. First, note that our method deals with 3D coloured meshes and 2D images. Most of the methods described in the literature deal with 2D images or 3D meshes that lie on geometry features. Table 4 shows the comparison of our approach with some other approaches in terms of detection accuracy. Authors in [3,11] proposed to use skin information to detect faces in 2D images, while the method proposed in [5] used the geometry information (curvature) in 3D meshes. In [10], authors used the skin information, then they applied a list of refinement steps such as the size of the candidate regions to detect faces in 2D images. In our method, we propose to combine the colour information (skin) and geometry information (saliency in our case) for a better face detection accuracy and to use graphs to deal with different types of data.

In addition to the statistical comparison, a visual example of a failure case presented by [3] and successfully detected by our method shown in Figure 23 [3].

The result obtained by [3] presents a failure case where a hand is interpreted as a face as shown in image (a). The author of [3] indicates in his experimental results that false alarms and misses still exist and this is an example. In contrast, by applying our method on the same image, all faces are detected correctly as shown in image (b). All other results presented in experimental result of [3] are also detected correctly by our method. Finally, [3] deals only with 2D coloured images.

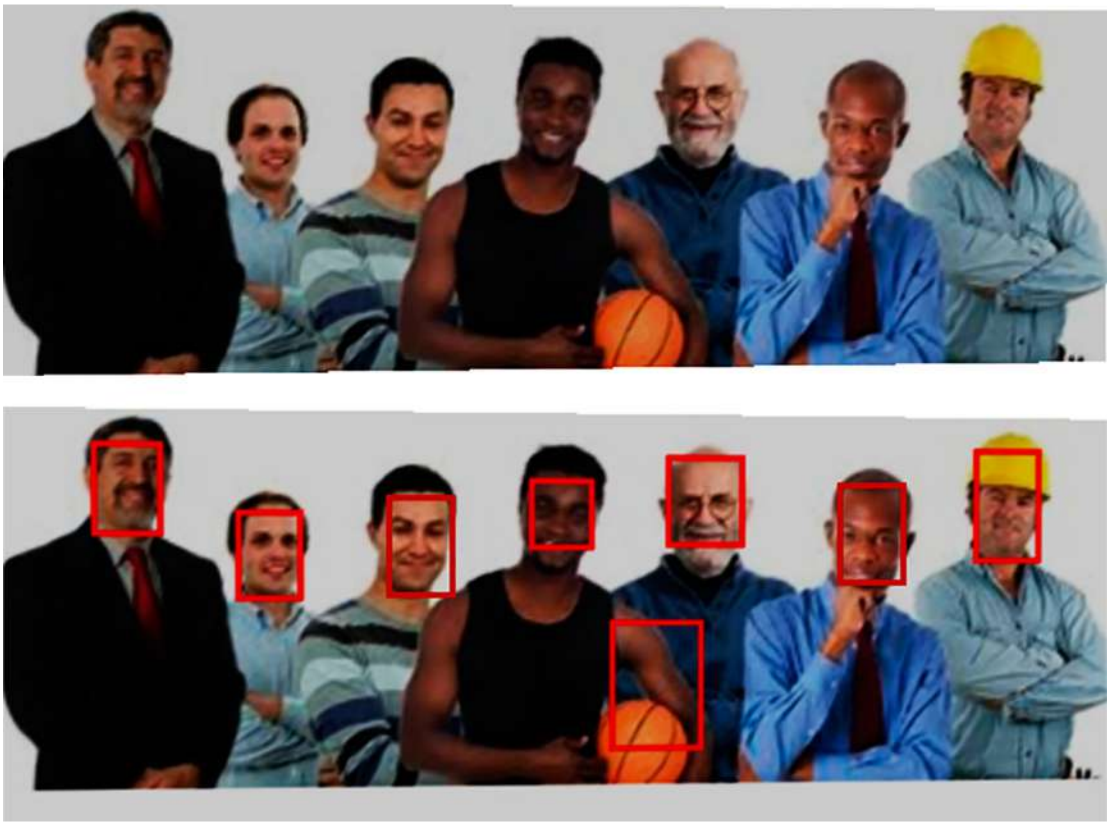
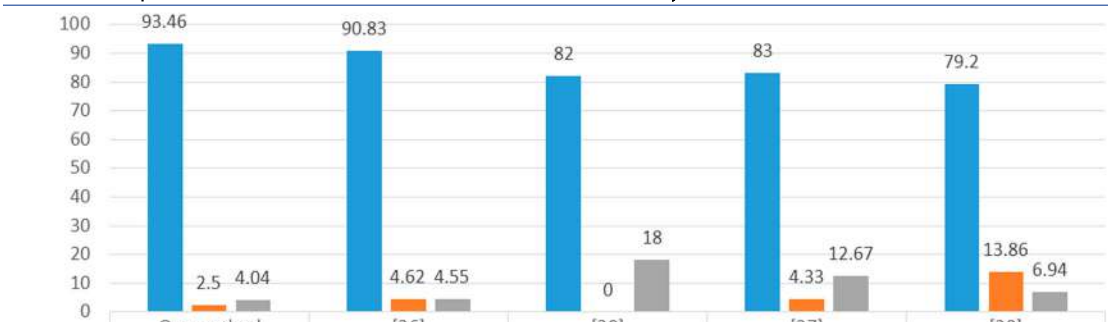


Figure 22. Result using our method applied on a coloured image containing multi-faces.

Table 3. Results obtained from 2D images.

Images count	200
Number of all faces	297
Number of really detected faces	279
False positive rate (FPR)	2.5%
False negative rate (FNR)	4.04%
Accuracy	93.46%

Table 4. Comparison of our method and others in terms of accuracy, FPR and FNR.



	Our method	[26]	[28]	[37]	[38]
Accuracy	93.46	90.83	82	83	79.2
FPR	2.5	4.62	0	4.33	13.86
FNR	4.04	4.55	18	12.67	6.94

Accuracy FPR FNR

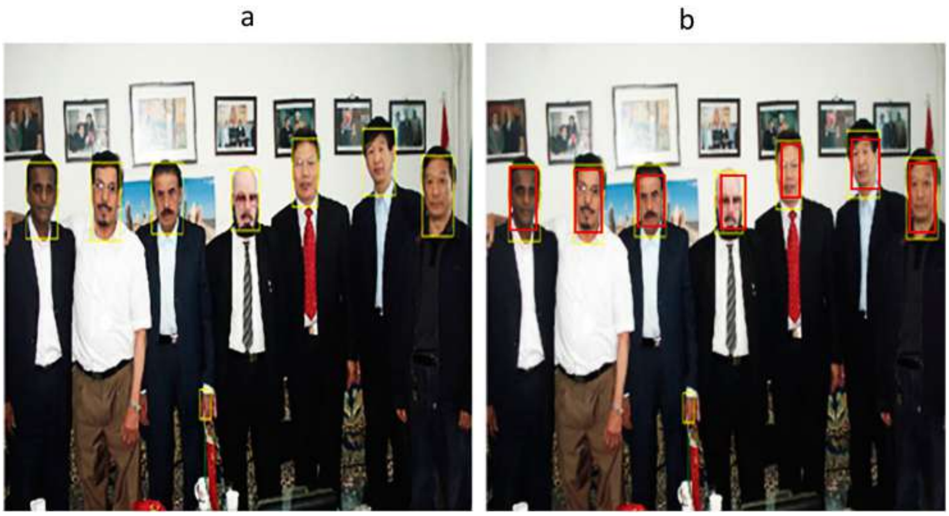


Figure 23. Image (a) shows a screen shot from experimental result of [3], image (b) showing the result of applying our method on the same screen shot.

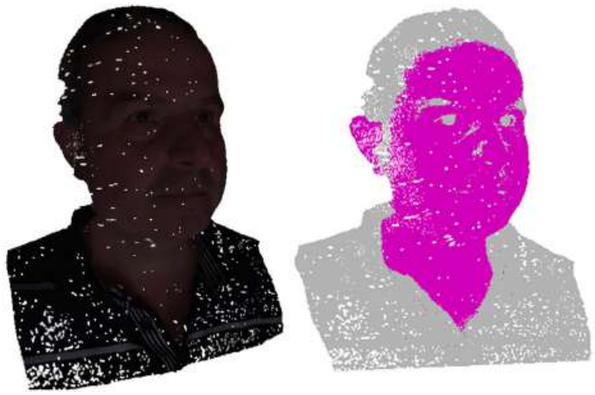


Figure 24. Face detection from a noisy 3D coloured point cloud.

The method presented in [5] deals with 3D faces, the authors mentioned that this technique is highly sensitive to noises and to the presence of holes surrounding the regions of nose and eyes. In contrast, our technique deals with noisy 3D coloured point clouds, as shown in Figure 24.

The approaches presented in [6,7] localize faces, rather than detecting them. In [6], the presented approach is strongly limited to the dataset used where every input 3D object must contain only one face. Otherwise, our method deals with 3D coloured point clouds, meshes and 2D images based on skin colour containing single or multiple faces of different sizes and locations.

Conclusion

In this paper, we have presented a novel method to detect 3D facial regions from coloured point clouds and 2D images. This method was divided into two phases. First, the candidates face regions based on skin features are detected, then the salient features are computed in each detected regions to affirm the presence of human face. We also presented a simple and efficient algorithm derived from the Hill Climbing algorithm using a morphological dilation operator to label skin and salient regions. We have showed the robustness of our proposed method through some experimental results. Then, we compared our method with various state-of-the-art methods.

Notes

Please see below for information relating to all images reproduced throughout this article:

Figures 4, 11 and 12. 3D Objects which can be obtained from https://sketchfab.com/models

Figure 15. 3D meshes created from 2D images using specific tools. Original images obtained from the following repositories:

1. http://www.german-pathway-colleges.org/wp-content/uploads/2015/11/International_students-300x266.jpg
2. https://www.ronitbaras.com/family-matters/parenting-family/three-is-a-crowd/
3. http://www.bytedigital.co.uk/who-we-are/our-company/

Figure 18. © The Authors. Reproduced with permission.

Figures 19 and 20. Used from dataset of Reference [2] (Moudani W, Sayed AR, 2010), also found at http://www.aiai.ed.ac.uk/project/plan/peapod/2002-05-13.html

Figure 21. Image obtained from https://ca.news.yahoo.com/roadside-body-could-missing-big-man-190107548.html

Figures 22 and 23. Used from experimental results of Reference [3] (Bin Ghazali KH, Ma J, Xiao R, 2011) in order to gain comparison.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Abdul Rahman El Sayed received the Master I degree in applied mathematics, from the Lebanese University, in 2002, the Master II degree (with honours) in informatics, from the Lebanese University, in 2008 and the M. Eng. in

computer engineering from the University of CNAM, Beirut, Lebanon, Center affiliated to CNAM Paris, in 2015. He has held lecturing positions at Tripoli Technical Institute and Tripoli Industrial Institute, Lebanon. In addition, he is an IT manager at Tripoli Governmental Hospital and the owner of a programming company in North of Lebanon. His research interests cover the image processing 2D and 3D with many publications in this scope.

Abdallah El Chakik received his Ph.D. and M.S. in computer science, both from the University of Caen, France. He received his B.S. in computer science from the University of Saint Etienne-France in 2006. He is an assistant professor in the Department of Computer Science at Beirut Arab University. He worked as a research engineer in Euridis Ingénierie, France.

Hassan Alabboud has been an assistant professor in the Lebanese University in Tripoli, Lebanon since 2008. He received his Ph.D. in 2007 at University of Le Havre in Le Havre, France. His Ph.D. thesis developed several methods and approaches in numerical optimization and operations research for binary and continuous Quadratic Programming Problems. He obtained the Master II degree in Pure Mathematics from the University of Sciences and Technology of Lille, France in 2003 and the Master I degree in Pure Mathematics from the Lebanese University in Tripoli, Lebanon in 2002. His research interests are combinatorial optimization, DC optimization, semidefinite programming, quadratic programming, crisis management and Image processing.

Adnan Yassine is a full professor in the University of Le Havre to Normandy in France and a member of the Laboratory of Applied Mathematics of Le Havre (LMAH). He teaches optimization and logistics in the Superior Institute of Logistic Studies (ISEL). He is author and co-author of numerous international publications in numerical and combinatorial optimization some of which concern the scheduling and logistics problems. He has supervised some thirty Ph.Ds. He obtained a Ph.D. in Applied Mathematics from the University of Grenoble (France) in 1989 and Habilitation to Supervise Research (HDR) from the University of Nancy (France) in 1998.

References

[1] Niese R, Al-Hamadi A, Michaelis B. A stereo and color based method for face pose estimation and facial feature extraction. ICPR; 2006. p. 299–302.
[2] Moudani W, Sayed AR. Efficient image classification using data mining. Int J Comb Optim Problem Inform. 2010;2(1):27–44.
[3] Bin Ghazali KH, Ma J, Xiao R. An innovative face detection based on skin color segmentation. Int J Comp Appl. 2011;34(2):6–10.
[4] Joenes M, Rehg J. Statistical color models with application to skin detection. IEEE conference computer vision and pattern recognition, CVPR'99; 1999. p. 274–280.
[5] Colombo A, Cusano C, Schettini R. 3D face detection using curvature analysis. Pattern Recognit. 2006;39(3):444–455.
[6] Mian A, Bennamoun M, Owens R. An efficient multimodal 2D-3D hybrid approach to automatic face recognition. IEEE Trans Pattern Anal Machine Intell. 2007;29(11):1927–1943.
[7] Niese R, Al-Hamadi A, Michaelis B. A novel method for 3D face detection and normalization. J Multimed. 2007;2(5):1–12.
[8] Payet N, Todorovic S. From contours to 3D object detection and pose estimation. ICCV; 2011. p. 983–990.
[9] Su H, Sun M, Fei-Fei L, et al. Learning a dense multi-view representation for detection viewpoint classification and synthesis of object categories. ICCV; 2009. p. 213–220.
[10] Kheirkhah E, Tabatabaie Z. A hybrid face detection approach in color images with complex background. Indian J Sci Technol. Jan 2015;8(1):49–60.
[11] Lin HJ, Wang SY, Yen SH. Face detection based on skin color segmentation and neural network. IEEE; 2015.
[12] Huang H, Mu Z, Zeng H, et al. 3D facial point cloud preprocessing based on skin color detection using SVM. 10th international conference on natural computation (ICNC); 2014.
[13] Hoppe H, DeRose T, Duchamp T, et al. Surface reconstruction from unorganized points. SIGGRAPH Comput Graph. 1992;26(2):71–78.
[14] Lozes F, Elmoataz A, Lézoray O. Nonlocal processing of 3D colored point clouds. 21st international conference on pattern recognition; 2012. p. 1968–1971.
[15] El Chakik A, Elmoataz A, Desquesnes X. Mean curvature flow on graphs for image and manifold restoration and enhancement. Signal Process. 2014;105:449–463.
[16] https://en.wikipedia.org/wiki/Polytope.
[17] https://en.wikipedia.org/wiki/Convex_hull#Convex_hull_of_a_finite_point_set
[18] István M, Gautam M. Simplex algorithms. In: Beasley JE, editor. Advances in linear and integer programming. New York (NY): Oxford University Press; 1996. p. 1–46.