



Efficient 3D point clouds classification for face detection using linear programming and data mining

Abdul Rahman El Sayed^a, Abdallah El Chakik^b, Hassan Alabboud^c and Adnan Yassine^{a,d}

^aNormandie Univ, UNIHAVRE, LMAH, FR CNRS 3335, ISCN, Le Havre, France; ^bBeirut Arab university, Tripoli, Lebanon; ^cBusiness and Administration Faculty, Lebanese university, Tripoli, Lebanon; ^dNormandie Université, UNIHAVRE, ISEL, Le Havre, France

ABSTRACT

Most of the applications related to security and biometric rely on skin region detection such as face detection, adult 3D objects filtering, and gesture recognition. In this paper, we propose a robust method for skin detection on 3D coloured point clouds. Then, we extend this method to solve the problem of 3D face detection. To do so, we construct a weighted graph from initial coloured 3D point clouds. Then, we present a linear programming algorithm using a predictive model based on a data mining approach to classify and label graph vertices as skin and non-skin regions. Moreover, we apply some refinement rules on skin regions to confirm the presence of a face. Furthermore, we demonstrate the robustness of our method by showing and analysing some experimental results. Finally, we show that our method deals with many data that can be represented by a weighted graph such as 2D images and 3D models.

ARTICLE HISTORY

Received 30 April 2017

Accepted 30 August 2017

KEYWORDS

3D point clouds; face detection; face model; skin detection; data mining; linear programming

Introduction

The main goal of this paper is to present an accurate method for skin region and face detection on 3D coloured point clouds using weighted graphs representation. This model aims to extend skin models proposed for 2D images to any data that can be represented by a graph and accordingly with different types of visual data such as video and 3D objects.

Context and motivation

Nowadays, owing to the fast growing of digital field and informational data, 3D object processing, including skin and face detection, became an essential pre-processing phase for many applications such as human-machine interaction, gesture recognition, human identification, and robotics.

In this work, we propose to extend the skin detection method from 2D images to 3D coloured point clouds using weighted graphs representation. In addition, defining operators on graphs leads to generalize the face detection method to any data that can be represented by a weighted graph. The first step in our approach consists of modelling the 3D surface using a weighted graph, and a predictive model based on a data mining approach is constructed. This model leads to define a statistical function that classifies graph vertices into skin and non-skin vertices. Then, the graph is segmented into regions using linear programming algorithm where each region is defined as a set of vertices of similar class. All skin regions are

considered as candidate face regions and the presence of a face is confirmed by applying refinement steps such as face size. Our method is illustrated in the following flowchart [Figure 1](#).

Paper organization

The remainder of this paper is organized as follows. The section ‘State of the art’ presents a review of the state-of-the-art work. The section ‘Skin detection method’ provides a general definition and some algorithms for skin detection on the weighted graph. The section ‘Face detection’ presents our new definition for the human face and some algorithms. The section ‘Experimental results’ presents the numerical experiments that prove and strengthen our theoretical results. The section ‘Comparison with the state of the art’ presents a comparison with the state of the art. Finally, the section ‘Conclusion’ is a general conclusion of the results of this article.

State of the art

In this subsection, we discuss some available approaches for skin region segmentation and face detection.

Moudani et al. [1] presented a method based on data mining techniques for 2D face detection. They have demonstrated the robustness of the method by showing some experimental results and comparing them to other related works. In this work, they

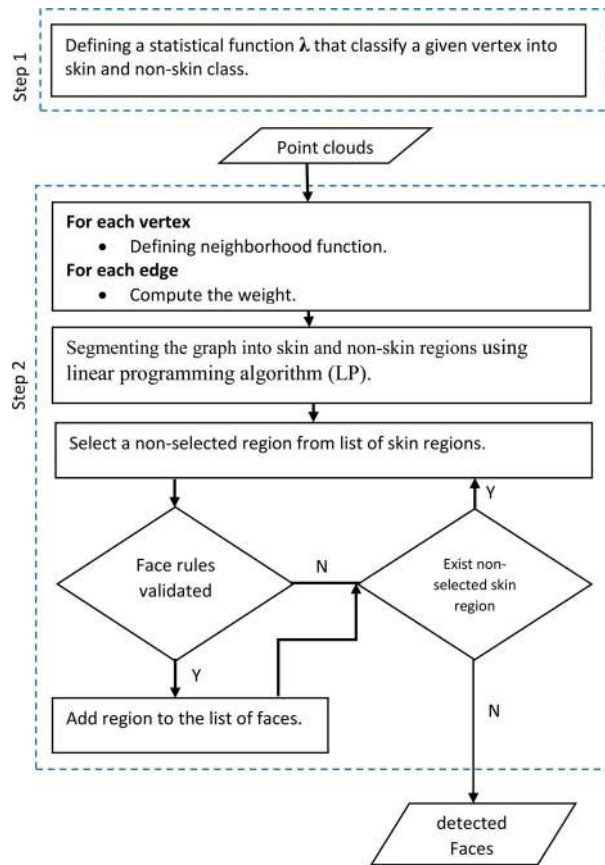


Figure 1. Flowchart of our method.

propose a generalized approach to deal with 3D coloured point clouds.

Bin Ghazali et al. [2] proposed a method for 2D images face detection under varying illumination conditions. This method considers the colour image and the colour balance model to convert the RGB colour space into the YCbCr colour space. Then, they proposed to detect faces by merging the Gaussian skin-colour model, template-matching, and face verification technique.

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

Mian et al. [4] proposed to localize the 3D face by detecting the nose tip, and then the segmentation can be made by cropping the sphere centred by the detected nose tip. They mentioned that their 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. Niese et al. [5] proposed a method for 3D face detection based on colour-driven clustering of the 3D points derived from stereo. They used a variant of the Iterative Closest Point algorithm to match the 3D mesh with

the post-processed face cluster. Then, they used the correspondence to derive the face pose. Finally, pose and model information are used for the synthesis of the face normalization. Some methods [6,7] divided the view sphere into a number of sectors and collected templates for each view. Given a new candidate object, the face detection is obtained using the template-matching approach.

Kheirkhah et al. [8] presented an approach using Ada-boost-based face detection based on skin-colour information. They proposed to extract the regions first, which have more potential to be faces in the image using image illumination correction, skin tone extraction, skin region segmentation, and connected component separation. Then they apply some rules to eliminate non-face regions. Finally, they used the Adaboost-based face detection system as an improved appearance-based approach. Lin et al. [9] proposed a human face detection method based on skin colour and neural networks technique. First, they proposed to construct a so-called skin map by searching for the candidates' regions. Then, they applied some morphological operations to detect the locations of pair eyes in the candidates' regions. Finally, they affirm a presence of a face using a 3-layer back-propagation neural network to verify the face candidates' regions. Wang et al. [10] proposed a hierarchical face detection method using the template-matching and 2DPCA algorithms. Their method consists in using two different classifiers. The first classifier filtrates the non-face regions using the template-matching algorithm, then the second one uses the 2DPCA algorithm to detect faces.

Recently, Huang et al. [11] proposed to use an support vector machine model with skin and non-skin colours to detect skin points in 3D facial point clouds. Then, they have used an approach based on the k-nearest neighbour (k-NN) method to fill holes on the obtained skin point clouds. Finally, they proposed to use a threshold to determine whether a point is on the face surface.

In this work, we define a set of possible faces from candidates' regions based on skin information, and then we confirm the presence of the face by verifying some proposed face rules. The experimental results show that our method deals with faces of different sizes, alignments, and directions.

Skin detection method

The first step in our approach consists in classifying the graph vertices as skin and non-skin based on data mining prediction rules. In this section, first we present the process of the prediction rules generation. Then we present a linear programme to segment the graph into segments based on vertices classes. We will extend then this method to solve the face detection problem in the next section.

Notations and definitions

As we mentioned above, our method deals 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 summary of the data mining rules generation that allow us to classify graph vertices as skin and non-skin. Then, we present a review of some basic definitions and operators defined on weighted graphs.

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

- *Dataset construction*: we used the existing dataset 'Helen dataset' [12] that contains 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 data mining techniques derived from inductive graphs J48 and JRip [13] rules using the statistical tool WEKA 3.6.0 [14]. The resulting prediction rules based on RGB component of the learning data file lead to a definition of function called lambda $\lambda: [0 \dots 255] \times [0 \dots 255] \times [0 \dots 255] \rightarrow \{0, 1\}$ as follows:

$$\lambda(v_r, v_g, v_b) = \begin{cases} 1 & \text{if the vertex } v(v_r, v_g, v_b) \\ & \text{has a skin color,} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where v_r, v_g, v_b are the red, green, and blue colour components of a vertex v . This function takes as input the colour components of a vertex and verifies whether this vertex has a skin or non-skin colour based on the generated prediction rules. The following flowchart illustrates the prediction rules generation Figure 2.

Once the prediction rules are defined, we model the surface to treat it. Consider the general situation where a point cloud can be viewed as a weighted graph. An undirected weighted graph $G = (V, E, W)$ consists of a

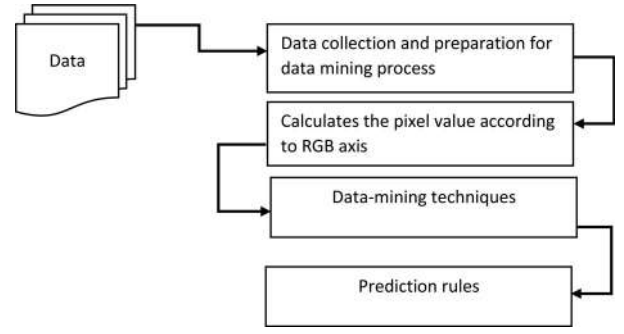


Figure 2. Flowchart of prediction rules generation.

finite set of vertices v , a finite set of edges $E \subset V \times V$, and a set of weight functions W .

Let (u, v) be the edge that connects two vertices u and v . The set of adjacent vertices $N(u)$ of a vertex u shown in (Figure 3) is constructed using the k-NN algorithm and defined as $N(u) = \{v / d(u, v) \leq k\}$, where $d(u, v)$ represents the three-dimensional Euclidean space distance between two vertices (u, v) and defined as

$$d(u, v) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + (u_3 - v_3)^2}, \quad (2)$$

The weight function $\omega: V \times V \rightarrow [0, 1]$ that represents a homogeneity measure between a vertex and its neighbours is defined as

$$\omega(u, v) = \begin{cases} \frac{\nabla(N(u))}{\text{Card}(N(u))} & \text{If } \lambda(u_r, u_g, u_b) = 0 \vee \lambda(v_r, v_g, v_b) = 0 \\ 1 & \text{If } \lambda(u_r, u_g, u_b) = 1 \wedge \lambda(v_r, v_g, v_b) = 1 \end{cases}, \quad (3)$$

where $\nabla(S)$ represents the cardinality of a skin vertices from a set S and defined as

$$\nabla(S) = \text{Card}\{u \in S \text{ such as } \lambda(u_r, u_g, u_b) = 1\}. \quad (4)$$

Skin region segmentation

Our method consists in classifying graph vertices into skin regions $(F(G))$ and non-skin regions $(B(G))$ using the generated predictive rules where

$$F(G) = \{v \in G \text{ such as } \lambda(v_r, v_g, v_b) = 1\} \quad (5)$$

$$B(G) = \{v \in G \text{ such as } \lambda(v_r, v_g, v_b) = 0\} \quad (6)$$

Figure 4 presents the graph vertices classification: images (a,c) show the original 3D point clouds,

Table 1. Information extraction 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

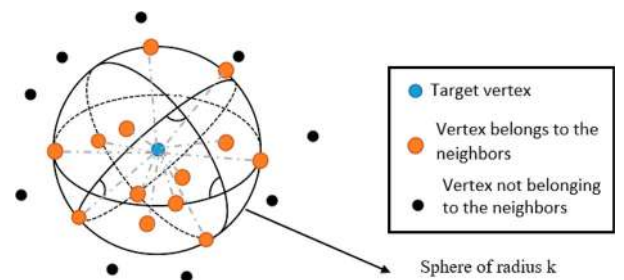


Figure 3. Vertices neighbours construction.

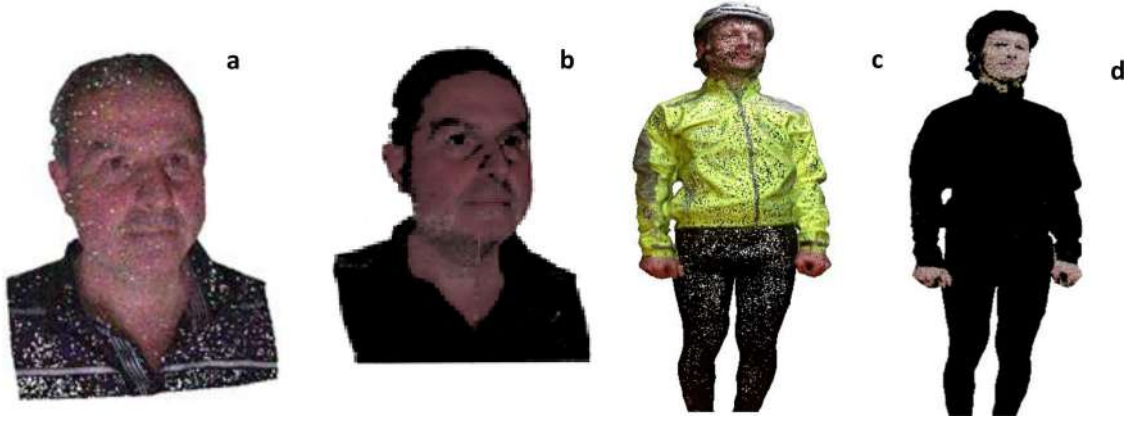


Figure 4. Vertices classification. Please see the image notes section for further details.

images (b, d) present the detected skin regions where the black regions represent non-skin vertices.

Vertices labelling once, we classify the graph vertices as skin and non-skin, we define a set of regions by adopting a linear programming model that groups the set of vertices of the same class in regions by minimizing a cost function. Let us start with the combinatorial problem of labelling the graph vertices where we search to assign the same label to adjacent vertices belonging to the same class using the minimal number of labels.

We propose to formulate the *vertex-labelling* process as an integer linear problem (ILP). To do so, we consider a set of vertices $V = \{v_1 \dots v_n\}$ and a set of n labels where the initial state of the problem considers that each vertex has its specific label.

We introduce two binary variables- y_k and x_{ik} , $k = 1 \dots n$. The first variable indicates whether a label k is used ($y_k = 1$) or not ($y_k = 0$), and the second one indicates if a given vertex v_i has received the label k . The model with all variables is established as follows:

$$\text{Min } \sum_{k=1}^n y_k$$

Subject to:

- (1) $\sum_{k=1}^n x_{ik} = 1 \forall i = 1, \dots, n$
- (2) $x_{ik} - y_k \leq 0 \quad \forall i, k = 1, \dots, n$
- (3) $x_{ik} - x_{jk} \leq 0 \quad \forall k = 1, \dots, n; \forall (v_i, v_j) \in E; v_i, v_j \in F(G) \vee B(G),$
- (4) $x_{ik} \in \{0, 1\}; y_k \in \{0, 1\}$

The constraints (4) and (5) ensure that x_{ik} and y_k are binary variables. The constraints (1)–(3) guarantee (in this order) that each vertex is labelled; vertex v_i receives the label k only if this label is used, and any two adjacent vertices have different label if they belong to different classes. The optimal solution of such problem is a graph where each group of connected vertices belonging to the same class receives the same label.

We present a heuristic algorithm to solve the ILP problem. The idea of such algorithm is to start from an empty solution and construct a solution iteratively using expansion criteria that allows one to take a choice at each iteration subject to our problem constraints. The expansion criteria are controlled by the score function $\text{Score}(v_i, v_j)$ defined as

$$\text{Score}(v_i, v_j) = \begin{cases} 0 & \text{if } v_j \in C(v_i) \\ 1 & \text{otherwise,} \end{cases} \quad (7)$$

where $C(v_i)$ is a set of vertices that combines the colour properties and the similarity measure between vertices and defined as

$$C(v_i) = \{v_j \in N(v_i) \mid \hat{v}_j \in F(G) / \omega(v_i, v_j) \leq \alpha\}, \quad (8)$$

where $\alpha \in [0.5..1]$ based on experimental results observations. We introduced this set of vertices to handle the case where some vertices are interpreted as non-skin regions due to lighting conditions and the presence of small objects such as hair, beauty marks, and moles as shown in Figure 5.

Algorithm 1 describes the label propagation algorithm starting from v_i . This algorithm is illustrated in (Figure 6).

Algorithm 1: LabelConnectedSimilar

//This algorithm makes call to this procedure for each non selected vertex.

Procedure Propagation

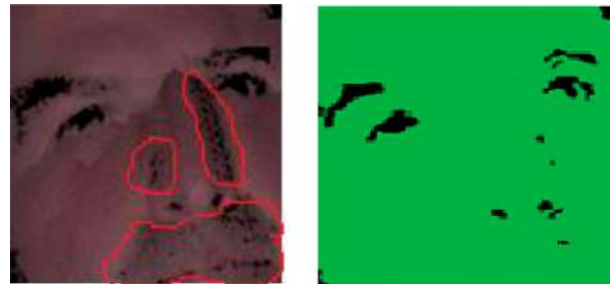


Figure 5. Influence of colour information and similarity measure on vertices labelling process. Please see the image notes section for further details.

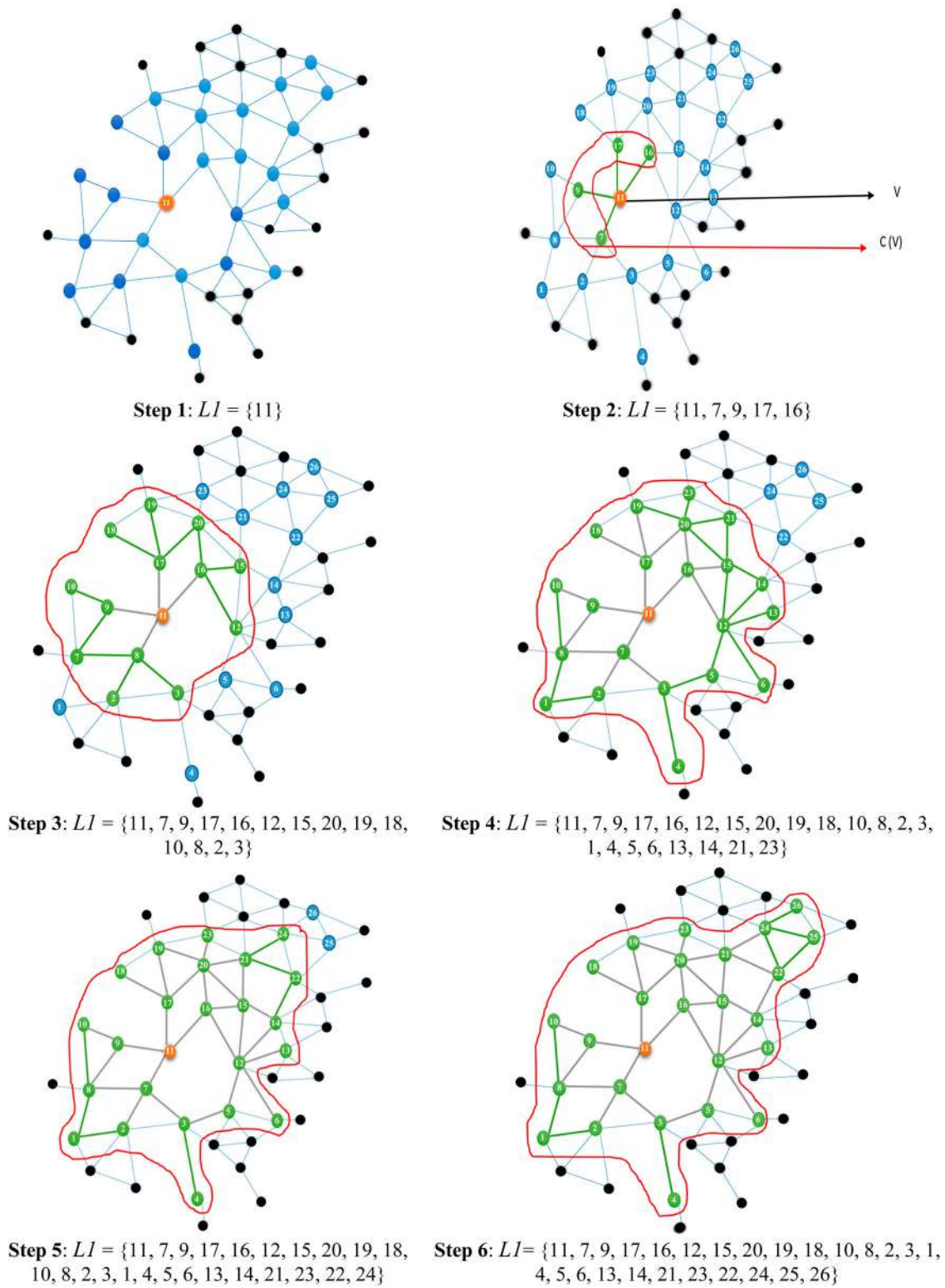


Figure 6. Illustration of labelling process. Please see the image notes section for further details.

Input
 Vertex start
 Integer K , NB_Label_Used /* $NB_Label_Used =$
 NB of vertices as initial value*/

Output:
 Connected vertices have the same Label L
 Let '[L1]' an empty list of vertices
 Let '[L2]' an empty list of vertices
 If (notSelected (start)) then

Add start to '[L1]'

End if

While (true)

Begin

If ('[L1]' is $\emptyset$) then

Quit Procedure

End if

For each (Vertex v_i in '[L1]')

Begin

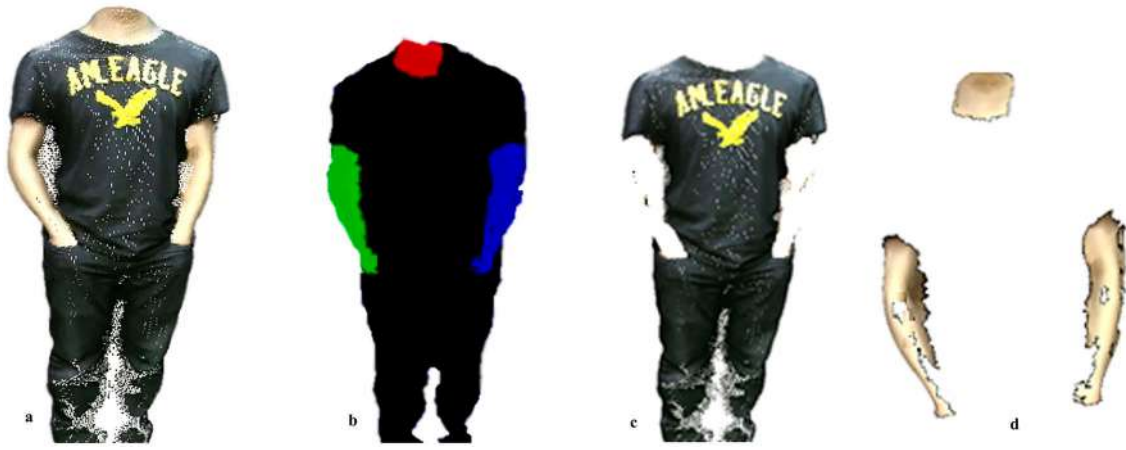


Figure 7. Segmentation process. Please see the image notes section for further details.

```

Set  $v_i$  as Selected
Assign  $k$  to  $v_i$ 
End if
For each (Vertex  $v$  in '[L1]')
Begin
  For each (Vertex  $n$  in  $N(v_i)$ )
  Begin
    If (not Selected ( $v_j$ )) then
    If (Score( $v_i v_j$ ) = 0) then
    Add  $v_j$  to '[L2]'
    Remove duplicate ('[L2]')
    NB_Label_Used = NB_Label_Used-1
    End if
  End if
  End for
End for
Clear ('[L1]')
Replace '[L1]' by '[L2]'
Clear ('[L2]')
End While
End Procedure

```

Algorithm 1 results a graph where each set of vertices with the same properties has the same label. Finally, we traverse all graph vertices to assign a colour for each label, and then we create a segment from each set of vertices having the same label as shown in Figure 7.

Figure 7 shows the set of segments produced by applying our method on 3D point cloud. Image (a) shows the original point cloud, image (b) presents labelled graph in which each group of connected vertices has a different colour, and image (c,d) shows the segments

Face detection

Once the graph vertices are labelled and segmented into skin and non-skin regions, we affirm whether a skin region has the face properties based on the most important characteristics of a human face such

as the eyes, nose, and mouth. We define these characteristics as a set of non-skin vertices surrounded by a set of skin vertices, as shown in Figure 8. To do so, we refine the list of faces candidates using some proposed refinement rules. These rules can be classified into three categories: refinement based on the number of non-skin regions contained in candidate region, refinement based on the size and ratios, and refinement based on the distance between face parts.

Refinement based on the number of non-skin regions

A skin region must contain a determined number of non-skin regions to be considered as a face. Therefore, candidate face regions are defined using the following conditions:

- For each face region R , $\exists$ at least three non-skin regions B_1 , B_2 , and B_3 such as $B_1 \subset R$, $B_2 \subset R$, $B_3 \subset R$ and the maximum number of non-skin regions in R should be less than or equal to $[NBFormsMax]$.

Algorithm 2 presents the procedure of defining the set of non-skin regions in a given candidate region.

Algorithm 2: GapLocator

Procedure Check_Boundary

Input:

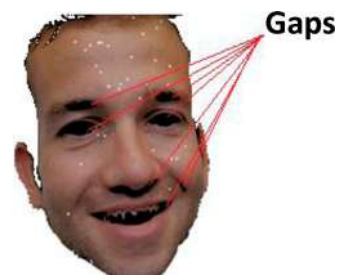


Figure 8. Skin region-containing gaps. Please see the image notes section for further details.

Non-Skin Region S_b

Output:

Label of Skin region Container

Let $[L]$ an empty list of vertices

Let $[R_v]$ a list of vertices $= \{v\} / v \in S_b$ and

$\nabla(N(v)) = 0$

While (**Card** (R_v) $\neq \nabla$ (R_v) or **Card** (R_v) increased)

Begin

For each (Vertex x in $[R_v]$)

If ($x \in B(G)$) then

Add $N(x)$ to $[L]$

if $[L] \neq \emptyset$ replaces x with the vertices of $[L]$

End if

End for

End While

If (**Card** (R_v) $= \nabla$ (R_v)) Return Label of any element

Else return -1

End procedure

Refinement based on the size

The size of the human face has specific characteristics in which there is a proportional relation between its height and width. This proportional relation obeys the golden ratio of human faces (Figure 9), which is approximately 1.6 [15]. Meanwhile, the candidate regions that are smaller than the minimum size are eliminated. To determine the size of a given candidate region, we define the minimum oriented bounding box of candidate region vertices as shown in Figure 10. Once, the box width, height, and depth are determined, the following rules should be verified:

1. $GR - th \leq \frac{HB}{WB} \leq GR + th$,
2. $HB \geq h_{min}$ and $WB \geq w_{min}$,

where HB is the height of bounding box, WB is the width, and th is a parameter controlled by the user to increase the range of the face ratio.

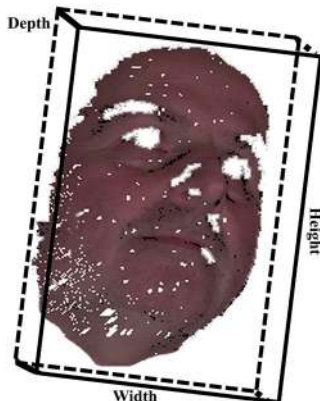


Figure 9. Minimum bounding box of candidate region. Please see the image notes section for further details.

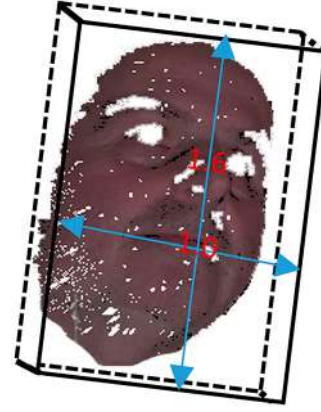


Figure 10. Human face golden ratio. Please see the image notes section for further details.

Refinement based on the distance between face parts

For each face region R , $\exists$ At least three non-skin regions B_1 , B_2 , and B_3 such as $B_1 \subset R$, $B_2 \subset R$, $B_3 \subset R$ such as

- $d(\theta(B_1), \theta(B_2)) - d(\theta(B_2), \theta(B_3)) \leq \varepsilon$,
- $d(\theta(B_1), \theta(B_2)) - d(\theta(B_1), \theta(B_3)) \leq \varepsilon$,
- The sum of ratios $\frac{\varphi(B_1)}{\varphi(R)}$, $\frac{\varphi(B_2)}{\varphi(R)}$ and $\frac{\varphi(B_3)}{\varphi(R)} \in [0.05, 0.2]$,

where $\varphi(R)$ is the area of the region R , $\theta(R)$ is the gravity centre of the region R , and ε is a distance threshold. Figure 11 shows the selection of face characteristic based on the distance between non-skin regions. Note that the values of interval $[0.05, 0.2]$ are chosen based on the observations done in our experiments.

Experimental results

In our experiments, we used a sample of 145 coloured objects and 250 coloured images from existing datasets, such as 'Helen' dataset [12], 'Faces in the Wild' dataset [16], 'Caltech' dataset [17], and others. The selected objects and images contain faces of different skin colours, ages, expressions, and positions. To evaluate our method, we show visual and statistical results for both 3D point clouds, meshes, and 2D images.

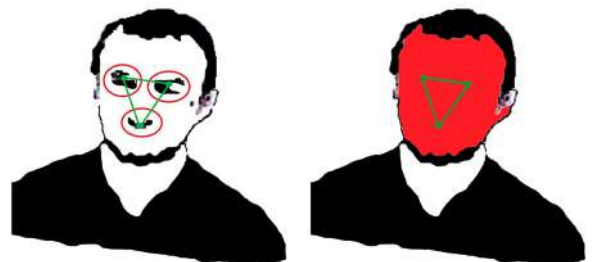


Figure 11. Distance between face parts. Please see the image notes section for further details.

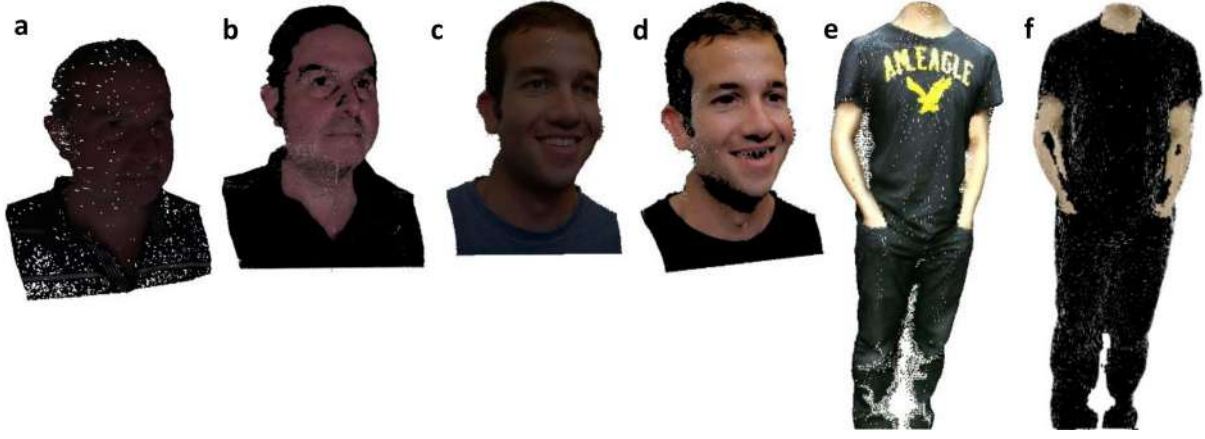


Figure 12. Skin detection on 3D coloured point clouds. Please see the image notes section for further details.

3D point clouds and 3D meshes

In this section, we show the efficiency and robustness of our proposed method by applying it on many coloured 3D point clouds models and 3D meshes. We will show and discuss the influence of different parameters on our approach, such as minimal face size (h_{min} , w_{min}), presented in section 4.2, k -NN threshold k , the parameter $NBFormsMax$ existing in section 'Refinement based on the number of non-skin regions', the parameter th defined in section 'Refinement based on the size', and the distance threshold ε presented in section 'Refinement based on the distance between face parts'.

Figure 12 presents the results of our skin detection model with a coloured point cloud with the following parameter values ($k=0.01363$, $h_{min}=5$, $w_{min}=8$, $NBFormsMax=6$, $th=0.5$, $\varepsilon=0.6$). Images (a,c, and e) show the original 3D point clouds and images (b,d,f) show the detected skin regions where the black regions are not skin.

Figure 13 presents the influence of the neighbouring distance k (k -NN threshold) on 3D point clouds. Image (a) shows the original 3D point clouds. Image (b) shows skin regions containing gaps using the same parameters presented in (Figure 12). Image (c) presents the influence of k with the value 1. One can see that the number of gaps decreases when we increase the neighboring distance due to neighbours' information.

Figure 14 presents the results of our face detection method applied on a coloured 3D point cloud. Images (a,c,e,g,m) show the original 3D point cloud, images (b, d,f,h,l, n) show the detected faces using our method with the following parameters ($k=0.01363$, $h_{min}=5$, $w_{min}=8$, $NBFormsMax=6$, $th=0.5$, $\varepsilon=0.6$). The fuchsia areas on the 3D point clouds object represent the detected faces. We can visually confirm the face detection. The image (n) presents an additional face detected, which is a failure case. This failure case is due to the characteristics of human face that are verified in this skin region.

Figure 15 presents the result of our face detection method applied on a 3D coloured point clouds with multifaces and expressions. We can visually confirm the detection of all faces in these point clouds.

Figure 16 presents the result of our face detection method applied on a noisy 3D coloured point clouds. One can see that our method produces good results with noisy surfaces.

After presenting a sample of visual results of our proposed method on 3D meshes and 3D point clouds, Table 2 shows some statistical results of our experiments in terms of detection accuracy.

The accuracy of our proposed method is measured as follows:

$$\text{Accuracy} = 100 - (\text{FPR} + \text{FNR}), \quad (9)$$

where false-positive rate (FPR) and false-negative rate

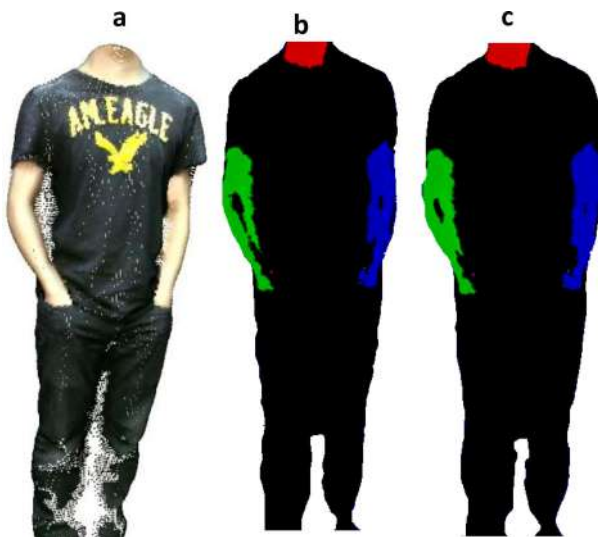


Figure 13. Influence of vertices neighbours on labelling process. Please see the image notes section for further details.

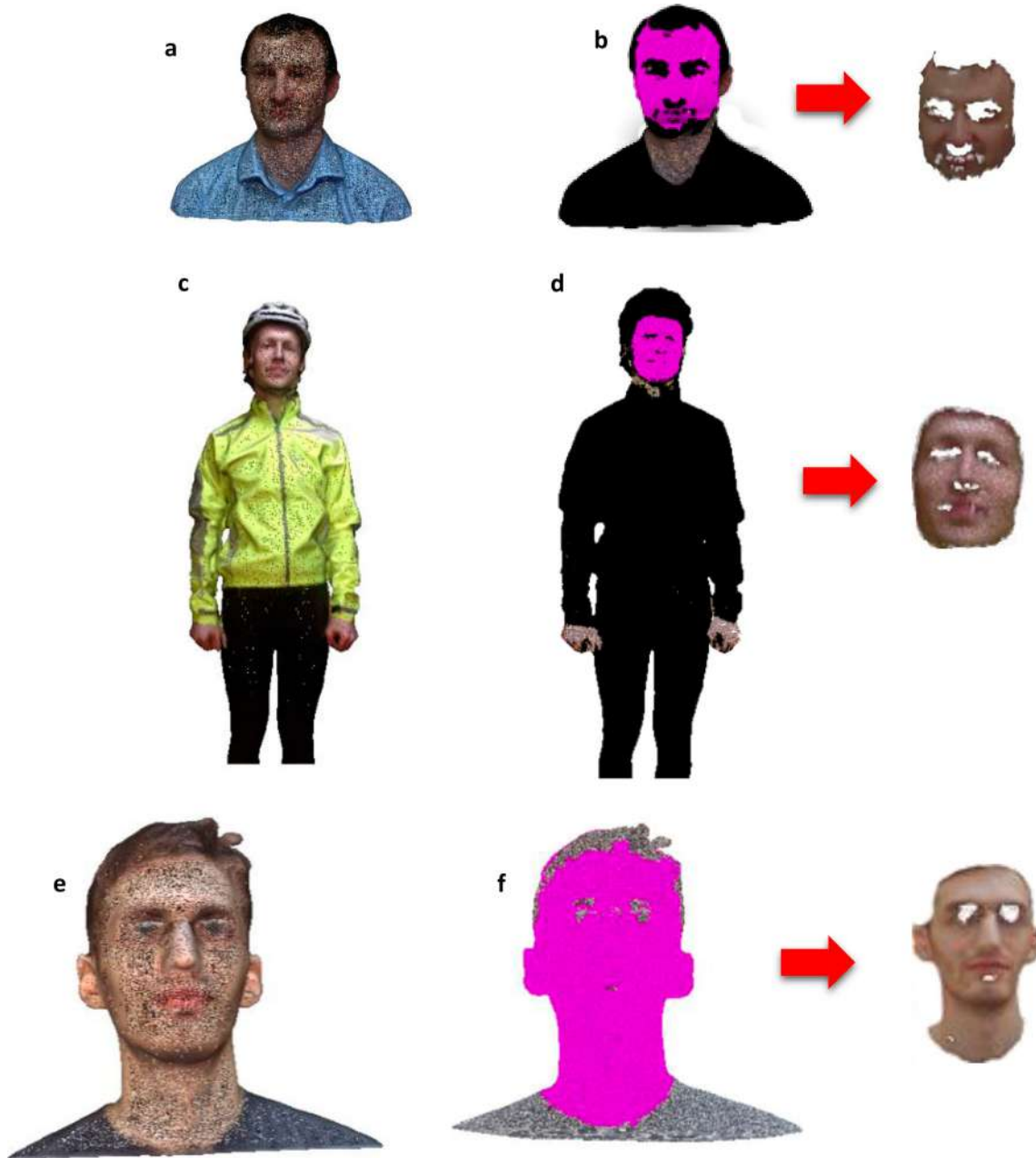


Figure 14. Detected faces on noisy 3D point clouds. Please see the image notes section for further details.

(FNR) can be computed as follows:

$$FPR = \frac{\text{number of false alarms}}{\text{number of detected faces}} \times 100, \quad (10)$$

and

$$FNR = \frac{\text{number of missed faces}}{\text{number of all faces}} \times 100. \quad (11)$$

In our experiments, we noted that the variation in the parameters *NBFormsMax* and *th* can affect the accuracy of our system. The following charts show the influence of these parameters on detection accuracy observed on 50 objects of our dataset [Figure 17](#).

Furthermore, when the value of parameters (h_{min} , w_{min}) increases, the number of missed faces relatively

increases. In contrast, when the value of parameter ε increases, the number of false alarms relatively increases.

2D images

As mentioned above, our method can deal with any type of data that can be represented by a weighted graph, such as 2D coloured images and 3D coloured point clouds. The figures show the result of applying our method on 2D coloured images. Note that a rectangle is drawn around the detected face region, as shown in [\(Figure 18\)](#).

[Figure 18](#) presents the results of our face detection method applied on 2D images with the following parameter values ($k=1$, $h_{min}=5$, $w_{min}=8$, *NBFormsMax*=6,

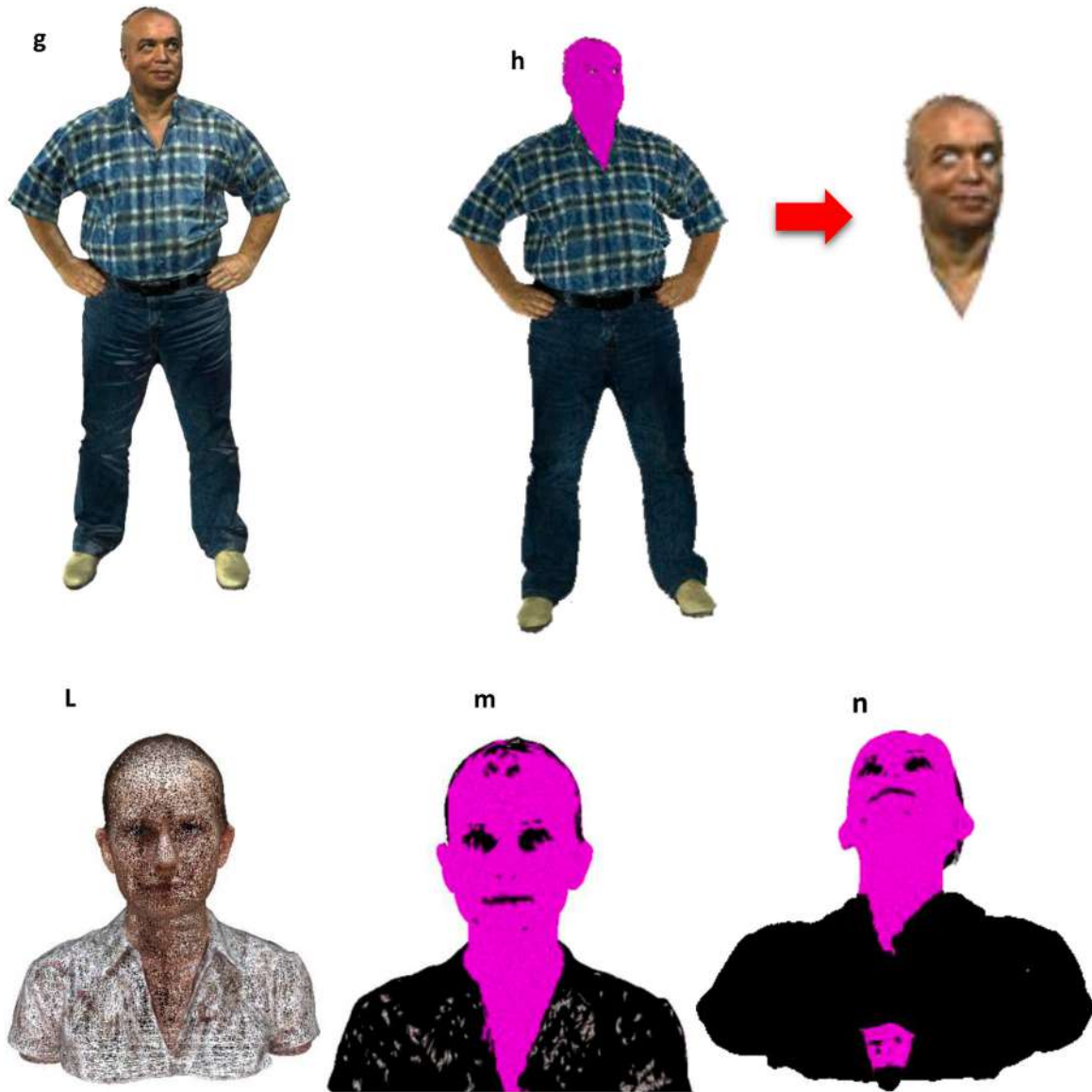


Figure 14. Continued.

$th = 0.5$, $\varepsilon = 0.6$). Images a_1 – e_1 show the original 2D images, images a_2 – e_2 present the detected skin regions, and images a_3 – e_3 present the detected faces by our method. Image (b_3) presents a face detection in an image with a complicated background, but Image (d_3) shows a failure case where an additional face is detected due to the colour of this region that is interpreted as skin colour and all face rules are verified.

Figure 19 presents the results of our face detection method on an image containing many faces of a person with different expressions. Image (a) shows the original image and Image (b) shows the accuracy of our method where all faces are detected.

Finally, a numerical summary of our experiments is shown in Table 3.

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 point clouds, 3D meshes, and 2D images based on skin colour. In contrast, most of the methods described in the state of the art section deal with 2D images or 3D meshes that lie on geometry features.

As another evaluation of our work, we compare the accuracy of our method to those of some methods described in the literature. Figure 20 shows the numerical data:

In addition to the statistical evaluation, we choose some visual examples of failure case presented by some related works and we apply our method on the same images. Figure 21 presents a comparison of the skin detection with [18]. One can see that the result of our method is better than that of the method proposed in [18] as shown in Figure 21. As shown in image (a), a non-skin colour is interpreted as skin colour. In contrast, it is detected correctly by our method, as shown in image (b).

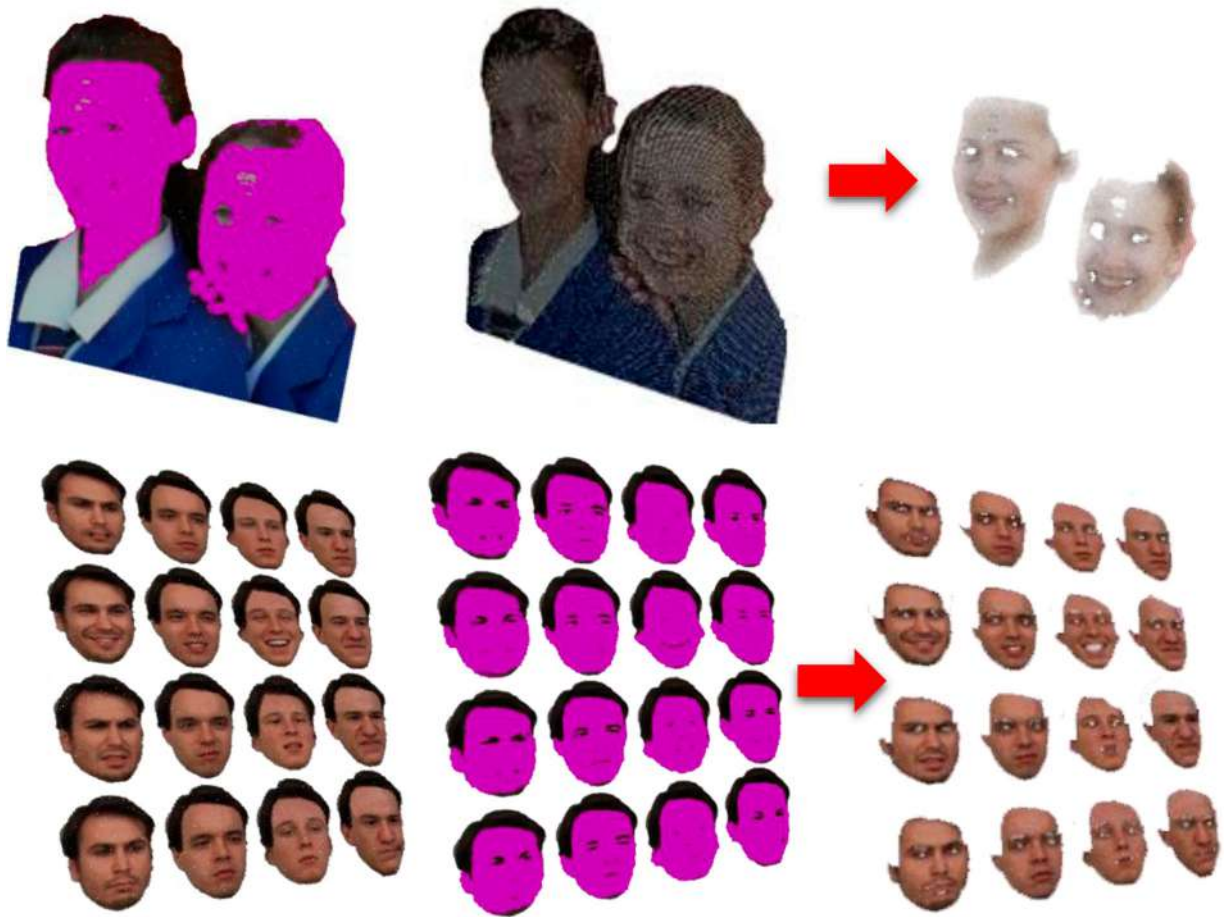


Figure 15. Detected faces with different expressions. Please see the image notes section for further details.

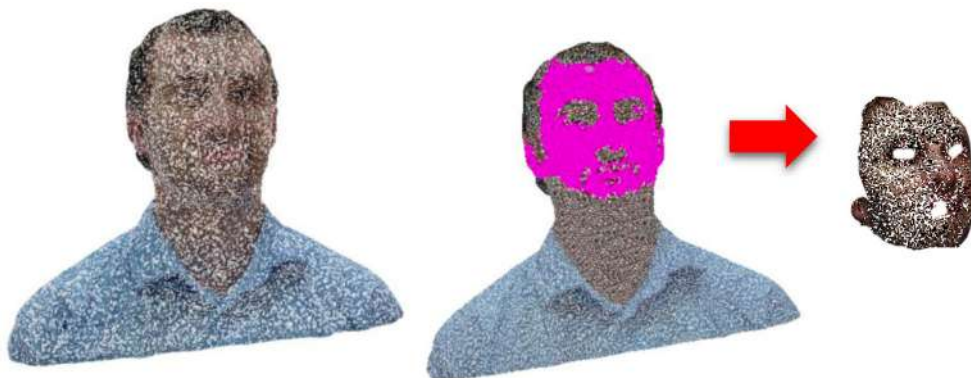


Figure 16. Detected face on a noisy 3D coloured point cloud. Please see the image notes section for further details.

Table 2. Results obtained from 3D point clouds and 3D meshes.

Objects count	145
Number of all faces	385
Number of truly detected faces	341
FPR	8.8%
FNR	3.63%
Accuracy	87.57%

Conclusion

In this paper, we have presented a new approach for skin and face detection for 3D coloured point clouds

using the weighted graph representation. To illustrate the robustness of our approach, we showed that our proposed method could be used to solve two signal-processing problems. The first one is related to the vertices classification based on the skin information. The second one is the face detection problem. We showed also that our method can deal with any data that can be represented by weighted graphs, such as 2D images, 3D surfaces, and 3D point clouds. We presented then a comparison with some other related face detection works to prove the efficiency of our method.

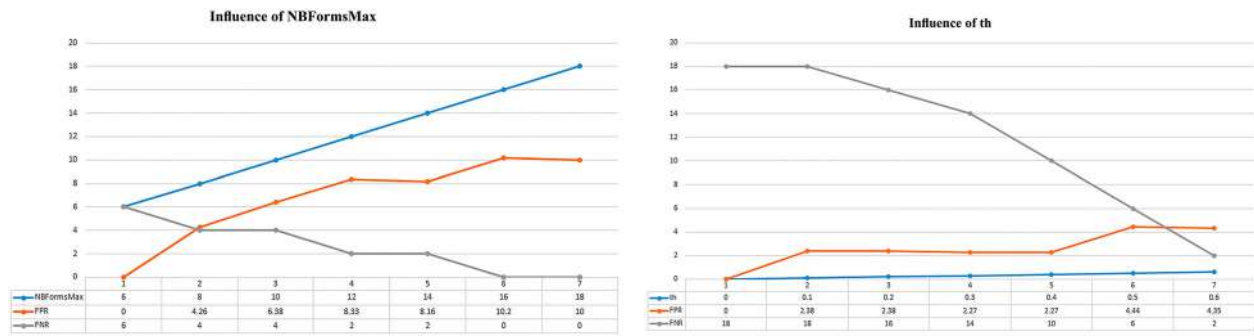


Figure 17. Influence of parameters *NBFormsMax* and *th* on accuracy.

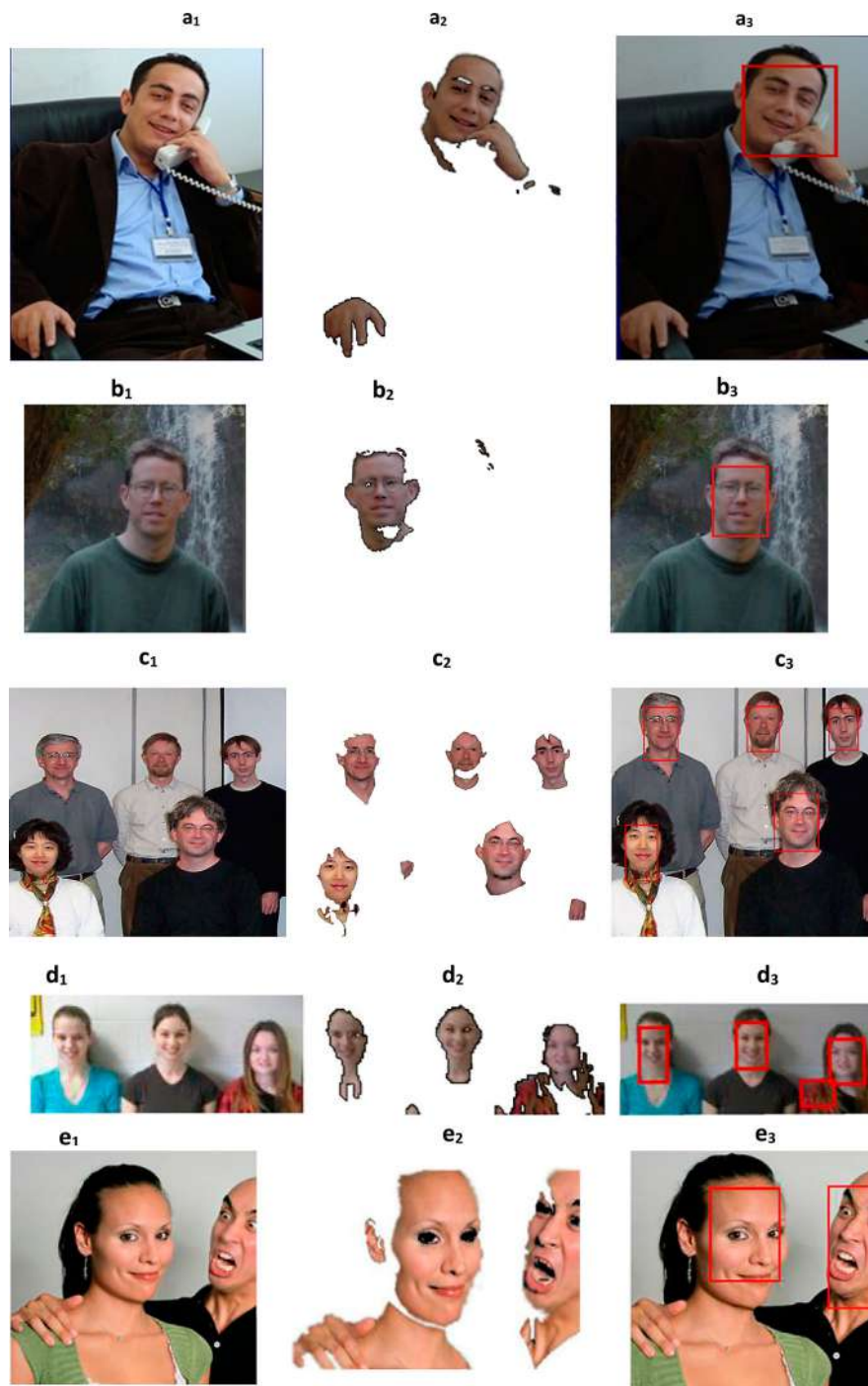


Figure 18. Detected faces on coloured 2D images. Please see the image notes section for further details.



Figure 19. Detected faces from an image with different expressions. Please see the image notes section for further details.

Table 3. Results obtained from 2D images.

Objects count	250
Number of all faces	304
Number of truly detected faces	278
FPR	7.01%
FNR	1.97%
Accuracy	91.02%

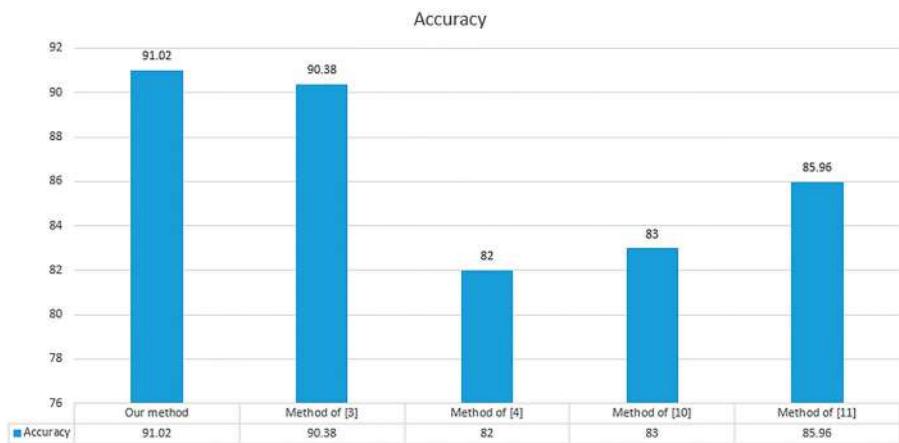


Figure 20. Accuracy comparison with other methods. Please see the image notes section for further details.

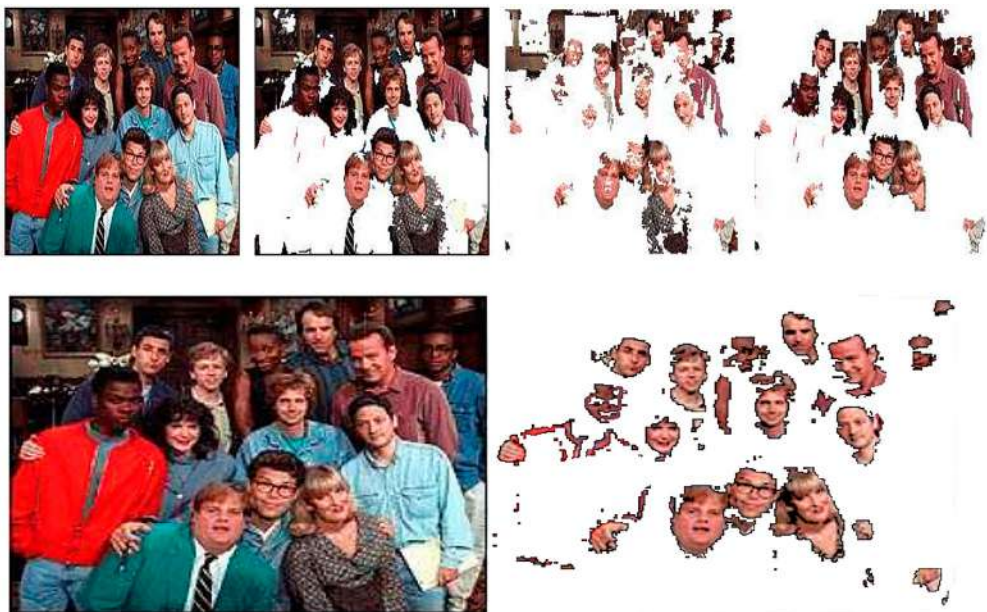


Figure 21. Result obtained by Yang et al. [18] (a) and our method (b). Please see the image notes section for further details.

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, Centre 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 image processing in 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 from the 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.

Images Notes

Figures 4, 8, 12 and 14. 3D Objects, which can be obtained from <https://sketchfab.com/models>
<https://www.artec3d.com/3d-models>

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

<https://www.ronitbaras.com/family-matters/parenting-family/three-is-a-crowd/>
<http://www.bytedigital.co.uk/who-we-are/our-company/>

Figure 18:

Image (b1) Used from dataset in Reference [1] (Moudani W, Sayed AR, 2011)

Image (c1) obtained from:

<http://www.aii.ed.ac.uk/project/plan/peapod/2002-05-13.html>

Image (d1) Used from dataset in Reference [1] (Moudani W, Sayed AR, 2011)

Image (e1) Used from dataset in Reference [14]

Figure 19 Image obtained from:

https://www.google.com.lb/imgres?imgurl=http%3A%2F%2Ftopfonpack.ru%2Fuploads%2Fimages%2Fk%2Fa%2Fr%2Fk%2Fartinki_s_emotsijami_1.jpg&imgrefurl=http%3A%2F%2Ftopfonpack.ru%2Fwallpaper%2Fopredelenie-emotsij-cheloveka&docid=znHp-lwDo3TZ_M&tbid=rwOgkoyCLEuv-M%3A&vet=10ahUKEwjXyJjVgvzVAhVQkRQKHZFUBkQQMwgIKAAwAA..i&w=1610&h=1200&itg=1&hl=en-LB&safe=strict&bih=602&biw=1242&q=facial%20expression%20children&ved=0ahUKEwjXyJjVgvzVAhVQkRQKHZFUBkQQMwgIKAAwAA&iact=mr&uact=8

References

- [1] Moudani W, Sayed AR. Efficient image classification using data mining. *Int J Comb Optim Probl Inform.* 2011;2 (1):38–55.
- [2] Bin Ghazali KH, Ma J, Xiao R. An innovative face detection based on skin color segmentation. *Int J Comput Appl.* 2011;34(2):6–10.
- [3] Colombo A, Cusano C, Schettini R. 3D face detection using curvature analysis. *Pattern Recognit.* 2006;39 (3):444–455.
- [4] Mian A, Bennamoun M, Owens R. An efficient multimodal 2D-3D hybrid approach to automatic face recognition. *IEEE Trans Pattern Anal Mach Intell.* 2007;29 (11):1927–1943.
- [5] Niese R, Al-Hamadi A, Michaelis B. A novel method for 3D face detection and normalization. *J Multimed.* 2007;2(5):1–12.
- [6] Payet N, Todorovic S. From contours to 3D object detection and pose estimation. *Int Conf Comput Vis.* 2011: 983–990.
- [7] Su H, Sun M, Fei-Fei L, et al. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. *Int Conf Comput Vis.* 2009: 213–220.
- [8] Kheirikhah E, Tabatabaie Z. A hybrid face detection approach in color images with Complex background. *Indian J Sci Technol.* 2015;8(1):49–60.
- [9] Lin HJ, Wang SY, Yen SH. Face detection based on skin color segmentation and neural network. *IEEE.* 2015;2:1144–1149.
- [10] Wang J, Yang H. Face detection based on template matching and 2DPCA algorithm. *Congr Image Signal Process.* 2008;4:575–579.
- [11] Huang H, Mu Z, Zeng H, et al. 3D facial point cloud pre-processing based on skin color detection using SVM. *10th International Conference on Natural Computation (ICNC).* 2014:524–528.
- [12] Helen dataset. <http://www.ifp.illinois.edu/~vuongle2/helen/>
- [13] Rajput A, Aharwal RP, Dubey M, et al. J48 and JRIP rules for E-governance data. *Int J Comput Sci Secur.* 2011; 5(2):201.
- [14] Smith T, Frank E. Statistical genomics: methods and protocols, chapter introducing machine learning concepts with WEKA. New York (NY): Springer; 2016, 353–378.
- [15] Hjelmas E, Low B. Face detection: a survey. *Comput Vis Image Underst.* 2001;83(3):236–274.

- [16] Fergus R. The Caltech face database. Caltech. Feb 2004. <http://www.vision.caltech.edu/html-files/archive.html>
- [17] Berg T, Berg A, Edwards J, et al. Forsyth Neural Information Processing Systems (NIPS), 2004. <http://tamaraberg.com/faceDataset/>
- [18] Yang MH, Ahuja N. Gaussian mixture model for human skin colour and its applications in image and video databases. SPIE/EI&T Storage and Retrieval for Image and Video Databases. 1999;3656: 458–466.