Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

An innovative approach to automatically identify control point set for model deformation rectification

Huynh Cao Tuan [1,*], Do Nang Toan [2], Lam Thanh Hien [3], Thanh-Lam Nguyen [4]

[1]*Center of Information and Resources, Lac Hong University, Dong Nai, Vietnam*
[2]*Institute of Information Technology, Vietnam National University, Hanoi, Vietnam*
[3]*Board of Rectorate, Lac Hong University, Dong Nai, Vietnam*
[4]*Office of International Affairs, Lac Hong University, Dong Nai, Vietnam*

ARTICLE INFO

Article history:
Received 7 April 2019
Received in revised form
7 June 2019
Accepted 7 June 2019

Keywords:
Control point set
Model deformation
Radial basis function
Rectify deformation

ABSTRACT

Rectifying 3D model deformation based on control point set is one of the most important problems frequently applied in virtual reality fields, in which control point set is the key for implementing deformed manipulation. This paper presents a technique to automatically identify control point set by analyzing the changes of each point in a 3D model through its variations, then gathering clusters and selecting important points to become a control point set. Our proposed approach was tested and proved to be effective in our practical experiments with a deforming technique based on the interpolation of the radial basis function.

1. Introduction

Object deformation and rectifying the deformation has been one of the interesting research topics attracting the special attention of several researchers worldwide and there have been many different approaches existing. For instance, to test a design in a crash experiment, technicians tend to create its digital 3D models which can be used several times by adjusting its key parameters. The 3D models are then accordingly tested in the computational environment several times to measure their deformation before a real sample is experimented in practice. This will save a lot of resources and efforts. Or, cinematographers can usually create 3D computational scenes and environments, perform complex explosions and special tricks on computers and then merge scenes together as well as transform some actors/ actresses into other characters or even make certain transformations on the actors/ actresses themselves (Ansari and Abdel-Mottaleb, 2003; Hwang et al., 2011; Lee et al., 1995; Zhang et al., 2002; Fua, 1997; Blanz and Vetter, 1999; Akimoto et al., 1993; Ip and Yin, 1996; Lee et al., 2004; Lin et al., 2002). Moreover, there are still many professional applications relating to manipulating the object deformation and rectification (Fan et al., 2017; Rezende et al., 2016; Rock et al., 2015). Such practical problems have well resulted in the special research interests in computer graphics in general and rectifying 3D object deformation in particular.

One of the most frequently used to rectify the deformation of 3D models is control-point based approach (Cerveró et al., 2016; Jacobson et al., 2011; Hwang et al., 2012). The 3D object surface will be characterized and controlled by a set of points (control point sets), when we transform this set of points, the data of the object will change accordingly. This approach is usually done in two phases: the first phase is to determine the control point and the relationship between them and the entire data of the object, the second phase is to rectify the changes. In the rectification phase, a transformation function is firstly identified from the transformation of the control points; then, the function is applied to the entire object to obtain a rectified model. Hence, in the approach, control points play a crucial role. Consequently, it is really important to effectively select control points that are suitable for the type of objects that need to be modelled and modified.

Practically, the selection of suitable control points is based on the knowledge of experts who have made certain studies on the nature of the object of interest; for example, when studying scanned classes of medical photographs, the construction of points that show specific subjects such as bone sections, joints, brain, heart, etc. needs to have anatomically knowledgeable people and specialists who know

* Corresponding Author.
Email Address: caotuan@lhu.edu.vn (H. C. Tuan)
https://doi.org/10.21833/ijaas.2019.08.007
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-2051-4466

where specific positions need special attention. Thus, depending on practical problem, different information from the control set can be exploited. For example, in the study of building facial animation from videos, Cao et al. (2014) used face control points not only in 3D modelling problem, but also to calibrate the camera matrix to increase the accuracy of the system. Similar studies can be found in (Hwang et al., 2012; Hwang et al., 2011; Ansari and Abdel-Mottaleb, 2003). Specifically, a training data set should include a set of object images marked with control points describing the instance of the interest objects in the images. Experts will use a computational tool to perform the mark-up on all images existing in the training set. In fact, in addition to this manual way, there are other approaches that can be automatic or semi-automatic; for example, when a relatively well-trained model set is obtained, it can be used to identify relevant control points in new images which can be then tested for their accuracy by an expert and some proper corrections can be done to improve the performance.

Typically, the control point set of an object is understood as a set of points that are distinct from other points and have consistency on different observations of an object. For example, if we focus on eyes in a face model, the control points preferably selected include eye corners which can be easily identified and marked. The nature of control points on a specific type of object requires certain knowledge of the object itself; thus, the selection is usually based on the expertise of a knowledgeable person who has a good understanding of the selected object.

The control point set is stored in MPEG-4 format which is an object-based multimedia data compression standard (Pandzic and Forchheimer, 2003). In image processing of facial models, as shown in Fig. 1, MPEG-4 characterizes faces with 84 feature points along with the activation parameters which well correspond to the actual actions of the faces and result in deformation of the face model compared to their neutral statuses. Presenting the deformation procedure of the facial model with some activation parameters within a time frame results in a sequence of facial animations. Key facial features correspond to the main positions on a human face usually considered include the eye muscles, eye position, nose and mouth. This control point set is chosen to reflect the effective movement of the human face. The key features are often arranged in groups such as cheeks, eyes, mouth, etc.

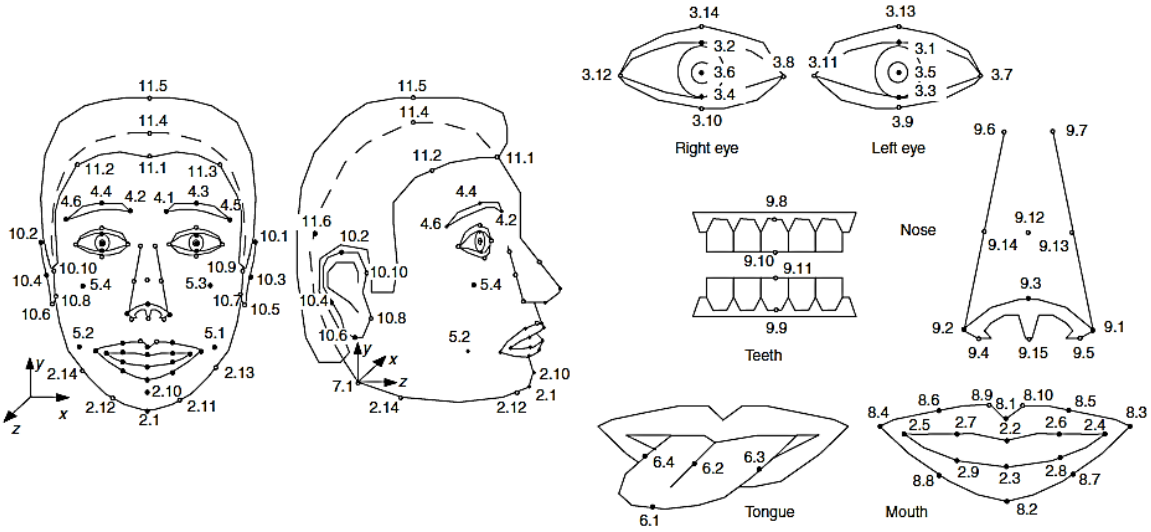

Fig. 1: Point system of MPEG-4

Another set of key features developed by Luxand (2019) including 70 facial features which are the coordinates of face components such as eyes, eye contours, eyebrows, lips, nose, cheeks and chin as shown in Fig. 2. Luxand's development kit has been used in many applications such as security monitoring, access control, building animation etc.

The proposed control point sets have been successfully applied in many studies and practical applications on model modification. However, the control points are usually marked manually. To overcome this shortcoming, this paper aims to propose an innovative approach to automatically select control points based on observational patterns of object transformation. Hence, the next section of this paper briefly presents our proposed technique in clustering and selecting control points based on their transformation while Section 3 shows our empirical results obtained from practical tests.

2. Identifying control point set for model deformation rectification

Our practical observations show that when there is a change in a model, the transformation of the control points is interdependent, i.e. there are some points to be significantly changed and there are some with little change. For example, when we grab our skin and pull it up, the points near to grabbed points are varied the most whereas those far away from the grabbed point are varied the least. Thus, it can be considered that the most varied points will pull other points and the least varied points will retain other ones. From such observation, our

approach will first find points with similar variations and thereby determine the strongest and weakest varied points to put into the control point set.

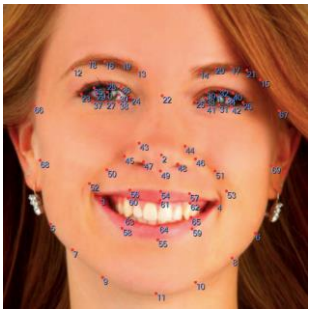

Fig. 2: Luxand's control points

2.1. Variable trajectory of points

The determination of the control point set of the model for deformation rectification is based on the analysis of a set of observation models of investigated objects. These models have the same number of points and side relations. Assume that model A of a 3D object consists of N points, and is represented as follows:

$A= \{a_0, a_1, ..., a_{N-1}\}$

where a_i denotes the i^{th} point of Model A, $a_i \in R^3$.

We collect M observations of that object, i.e. corresponding to M different 3D models. Then, the observed set of those points can be represented as:

$S= \{a_{ij} \mid i= 0, 1, ..., N-1, j= 0, 1, ..., M-1\}$

where a_{ij} denotes the i^{th} point of j^{th} observed sample of a 3D object of interest, $a_{ij} \in R^3$.

Thus, the i^{th} point of the 3D object will have M values corresponding to M observations. We call $T_i = \{a_{i0}, a_{i1}, ..., a_{iM-1}\}$ as the variable trajectory of the i^{th} point in the observation set S. Thus, T_i correspond to the variations of the i^{th} point in the observation set S of the 3D object. The determination of points with strong and weak changes will be done through the calculation of the trajectory of each point in the model.

2.2. Clustering transformation trajectory

From the set M, we calculate a set of variable trajectory of points in the model. In particular, each transformation trajectory respectively represents transformation of a point. We can then cluster the obtained trajectories into groups of points that have similar variations. Clustering techniques are built based on the algorithm K-means which is commonly used in clustering problems (Okabe and Yamada, 2018). Specifically, first, we calculate the distance between two transformation trajectories:

Function: euclidDistPointTrajectory
Input: 2 variable transformation trajectories T_1, T_2
Output: distance d
Process:

1. d: = 0
2. foreach i in [0, $|T_1|$)
3. d = d + euclid_distance(T_1[i], T_2[i])
4. endfor
5. d: = d/$|T_1|$

In each iteration, we need to recalculate the representation. To ensure the representative point be a point of the model, from the representative point calculated, for each cluster, we choose one point as a representative based on a specified criterion that it is the closest point to the calculated representative point. The distance is determined by the distance between the two mentioned trajectories. Specifically, the clustering algorithm of trajectories is as follows:

Function: kmeansTrajectories
Input: Trajectory set T, Number of clusters K
Output: Representative set delegates, cluster set clusters
Variables: Temporary variable storing clusters tmpClusters, temporary variable storing representatives tmpDelegates
Process:

1. delegates: = select_random (T, K)
2. resize (tmpClusters, K)
3. foreach l in [0, MAX_LOOP)
4. foreach k in [0, K)
5. tmpClusters[k] := ∅
6. endfor
7. foreach i in [0, |T|)
8. cid = get_nearest_cluster (T, i, delegates)
9. tmpClusters[cid] := tmpClusters[k] ∪ {i}
10. endfor
11. tmpDelegates: = calc_delegates (T, tmpClusters)
12. eps: = 0
13. foreach k in [0, K)
14. eps: = eps + euclidDistPointTrajectory(delegates[k], tmpDelegates[k])
15. endfor
16. eps: = eps/K
17. if eps<MIN_EPS
18. break
19. endif
20. delegates: = tmpDelegates
21. cluster: = tmpCluster
22. endfor

2.3. Selecting control points

After obtaining the clusters of model points, with each cluster, we choose points with the strongest and weakest changes as control points. The threshold is calculated based on a reference model. Thus, each point of the reference model corresponds to a variable trajectory. Corresponding to each point, we construct vector of deviation calculated by the deviation of each point in the trajectory against the

corresponding point on the reference model. The threshold of the changes of each point is calculated with Euclid standard of the deviation vector corresponding to the point.

2.4. Combining with deformation rectification algorithm

The goal of identifying the control points is to serve the deformation rectification. Specifically, to select a good control point set, we need to incorporate an algorithm for deformation rectification and a set of variable models to serve the quality evaluation of the deformation rectification. General algorithm is as follows:

Input: Reference model R, Variable trajectory set T, Model evaluation set M
Output: Control point set P;
Variables: list of values for K candidates for clustering KC
Temporary variables for storing clusters clusters
Temporary variables for storing representative's delegates
Process:

1. KC: = init_K_candidate ()
2. minErr: = MAX_VALUE
2. foreach K in KC
3. kmeansTrajectories (T, K, delegates, clusters)
4. tmpP: = select_controller_points (T_1, clusters, R)
5. err: = 0
6. foreach i in [0, |M|)
7. err: = err + err_morph(M[i], R,tmpP)
8. endfor
9. err: = err/|M|
10. if err < miner
11. minerr: = err
12. P: = tmpP
13. endif

In the algorithm, the function *errs morph* performs an error assessment between the target model and the transformation model from the reference model based on the given control set.

3. Experimental results

Before the practical experiments were conducted, the following components should be available: a rectification technique based on control point set, a 3D object for reference, a set of its variations to calculate and select control points and 1 set of object variations to evaluate the quality of selected control point set. The rectification technique used in our experiments is an approach using radial basic function (RBF) (Buhmann, 2003). And, we conducted experiments with many different values of K so that we can evaluate the model errors. Our experiments were conducted in two cases: (1) 3D objects which are spherical models created by computer graphical techniques, and (2) 3D models created from corresponding data of real human faces.

3.1. Spherical 3D models

3D object used in this tests is a spherical model with 482 vertices and 960 surfaces as shown in Fig. 3. Fig. 4 presents its distinct variants created by a 3D object transformation based on the random reference in the RBF rectification technique with a randomly generated control point set. Specifically, after a control point set is selected, coordinates of the control points are randomly transformed along the straight line connecting it to the center of the sphere in a certain proportional range. In our experiments, the proportion to radius of the sphere is chosen within [0.6, 1.4].

With the variant set for calculating and selecting control points, the trajectories of the points were first calculated before they are clustered with different K values to determine each region with similar variation. Fig. 5 briefly shows the clusters obtained in our experiments where points in the same cluster are marked with the same color.

With each cluster obtained from each K value, perform control point calculations and apply control points to transform the 3D model referenced by each sample in the set of variations for quality evaluation result. Each model rectified by RBF technique is then compared with the target model to calculate the model error. As the nature of the data in the 3D model and the trajectory of the points are all 3D point sets, the errors between the rectified models and the target one are also calculated through the distance function between the two variable trajectories. The error values of the control point set on the variable set for the model evaluation are calculated by the average error of each considered sample. Consequently, Fig. 6 presents the correlation between the errors and K values, where we can find that the errors are significantly reduced when the K is increased while Fig. 7 displays the rectification results based on our proposed approach with the spherical 3D object.

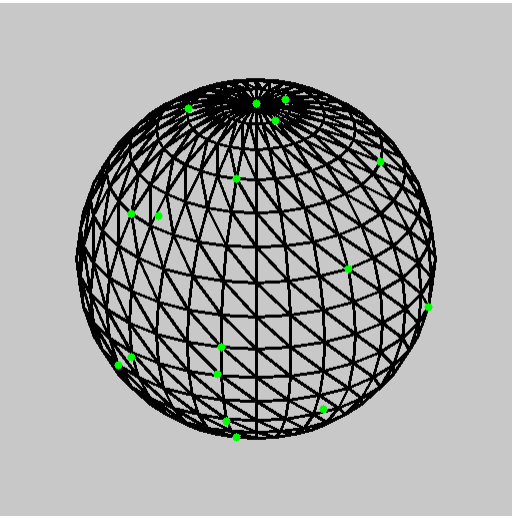

Fig. 3: Referenced 3D object and control point set

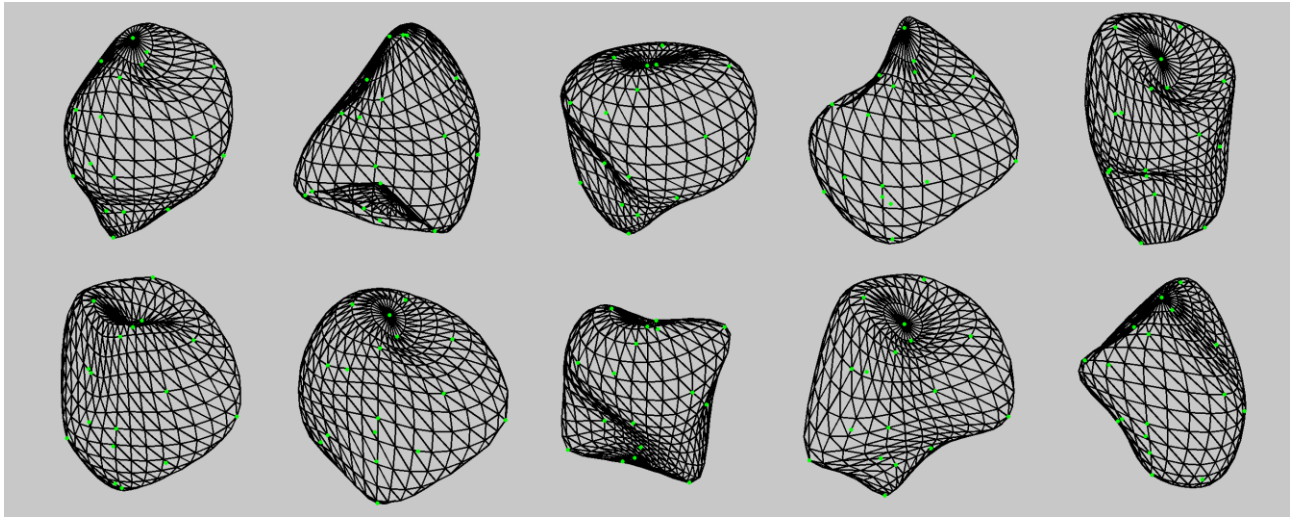

Fig. 4: Some variants of the referenced 3D object

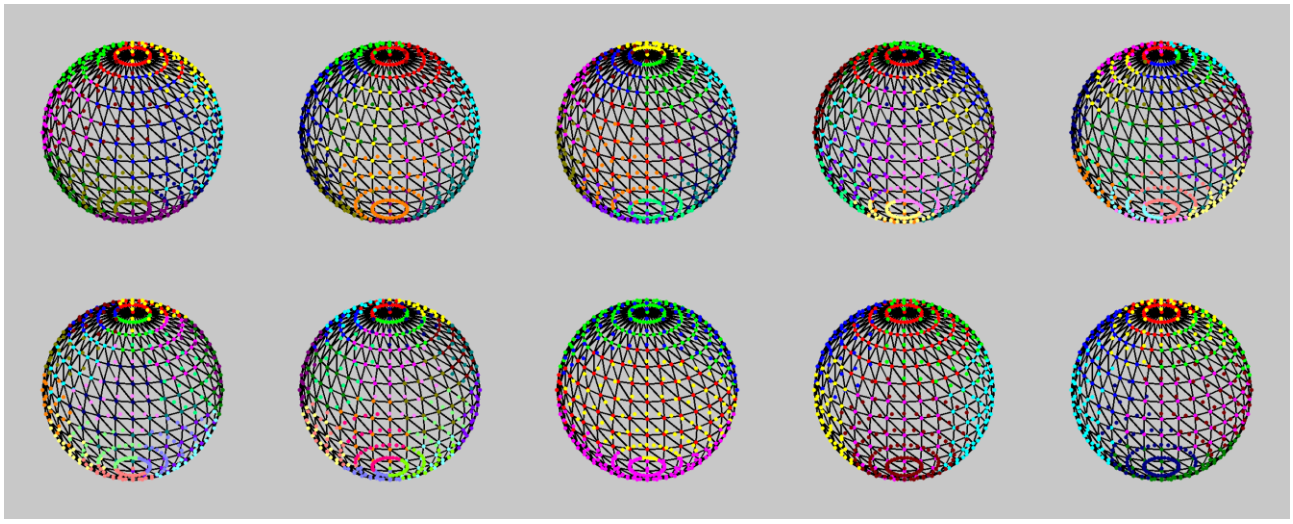

Fig. 5: Clustering results with different K values

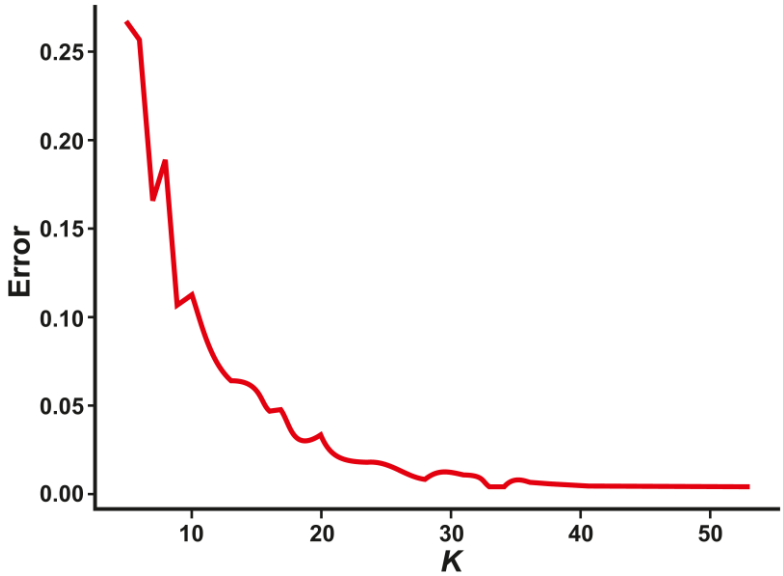

Fig.6: Correlation chart between error and K values

3.2. 3D faces

3D object used in these tests is a face model with 3448 vertices and 6736 surfaces created from the samples in JAFFE database by Lyons et al. (1998). A typical example of the 3D face model is shown in Fig. 8. JAFFE database contains 213 images of 7 different facial expressions (6 basic facial expressions including joy, sadness, surprise, anger, indignation, fear, and a neutral status) recorded on 10 Japanese female models and taken at the Department of Psychology at Kyushu University. 60 Japanese

students were asked to evaluate the images with 6 emotional states based on a 5-point scale where 5 denotes the highest level and 1 denotes for the lowest one.

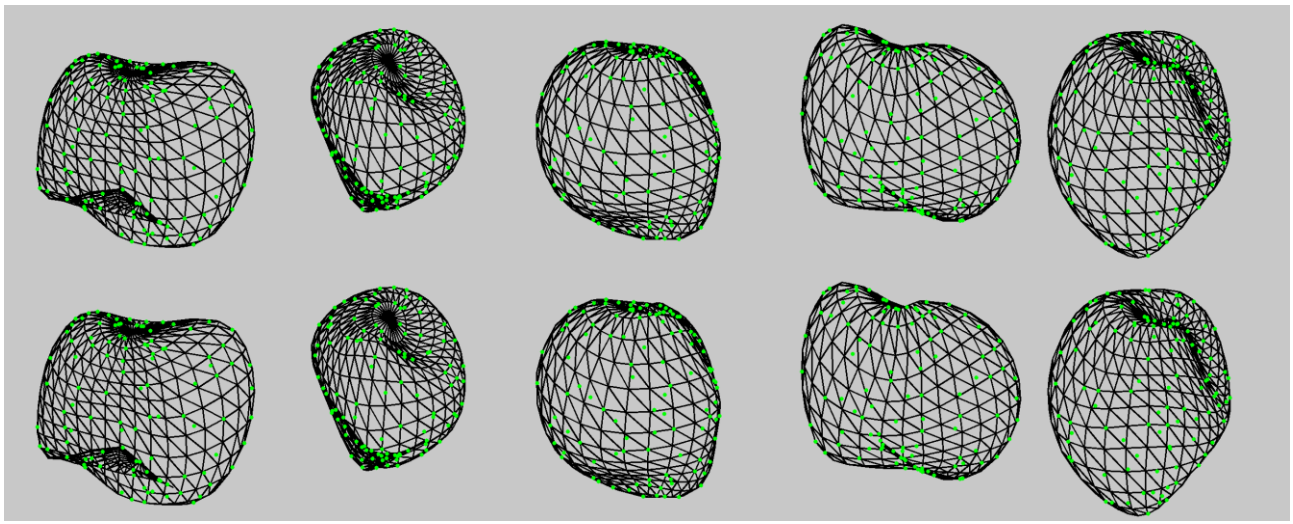

Fig. 7: Benchmark between target models and rectified ones

Corresponding to the 213 images in JAFFE dataset, we have 213 models of 3D faces. These models were then randomly divided into two distinct groups to be used in our tests which were carried out similarly to spherical 3D models mentioned in Section 3.1. Specifically, Fig. 9 shows some sample variants of the 3D faces created whereas Fig. 10 briefly shows the clusters obtained in our experiments where points in the same cluster are marked with the same color. In addition, Fig. 11 presents the correlation between the errors and K values while Fig. 12 displays the rectification results based on our proposed approach with the 3D faces.

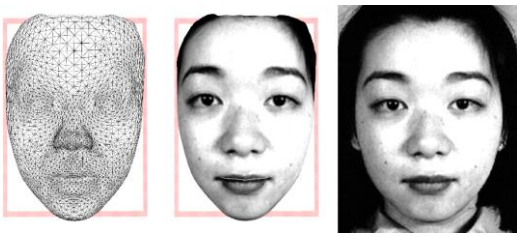

Fig. 8: Example of 3D face models

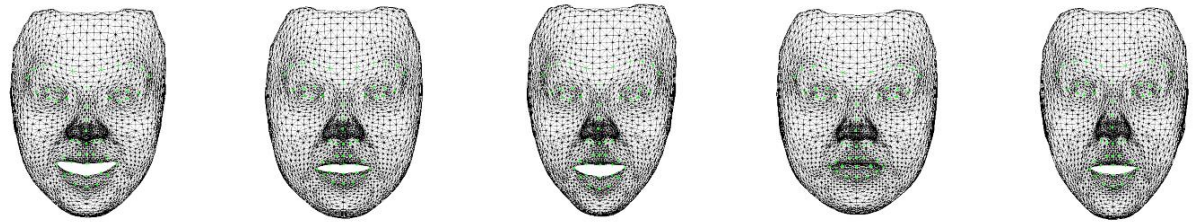

Fig. 9: Some sample variations of the 3D faces

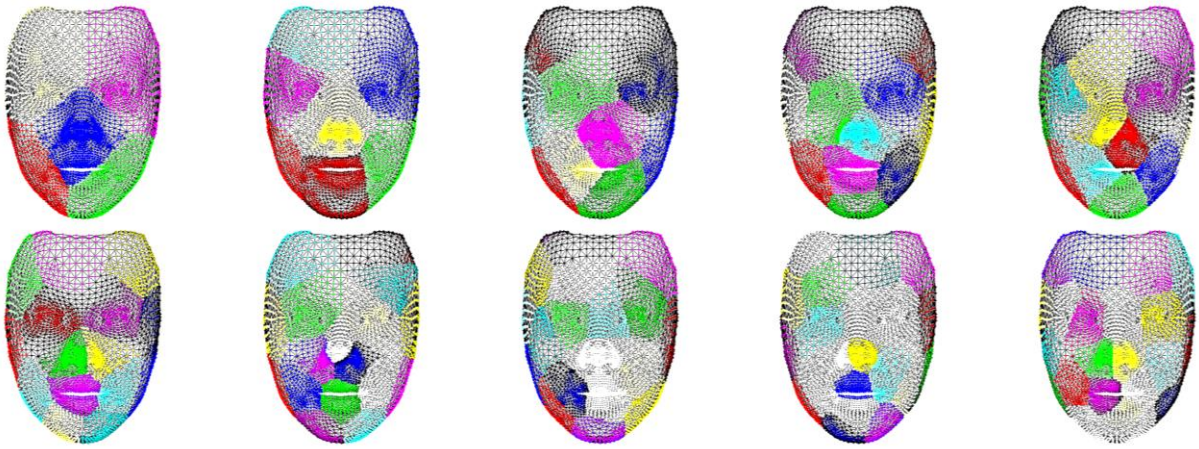

Fig. 10: Clustering results with different K values

From the above results, we can conclude that when increasing the value of K, the accuracy of our transformation and rectification is significantly improved regardless of simulated 3D spherical models or with 3D faces created from real-life photographs. As such, our proposed approach is not only effective in identifying reasonable control points and rectifying deformation within a given

error range but also applicable to different variations on different types of objects.

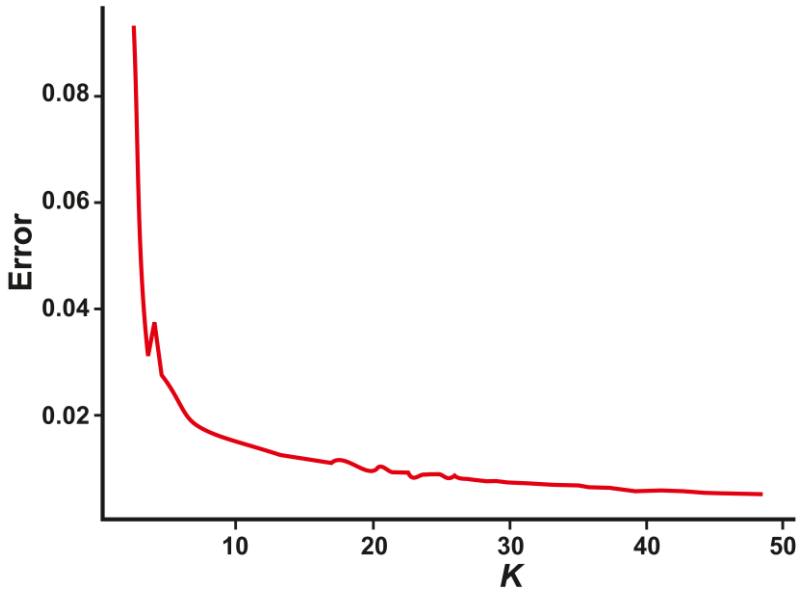

Fig. 11: Correlation chart between error and K values

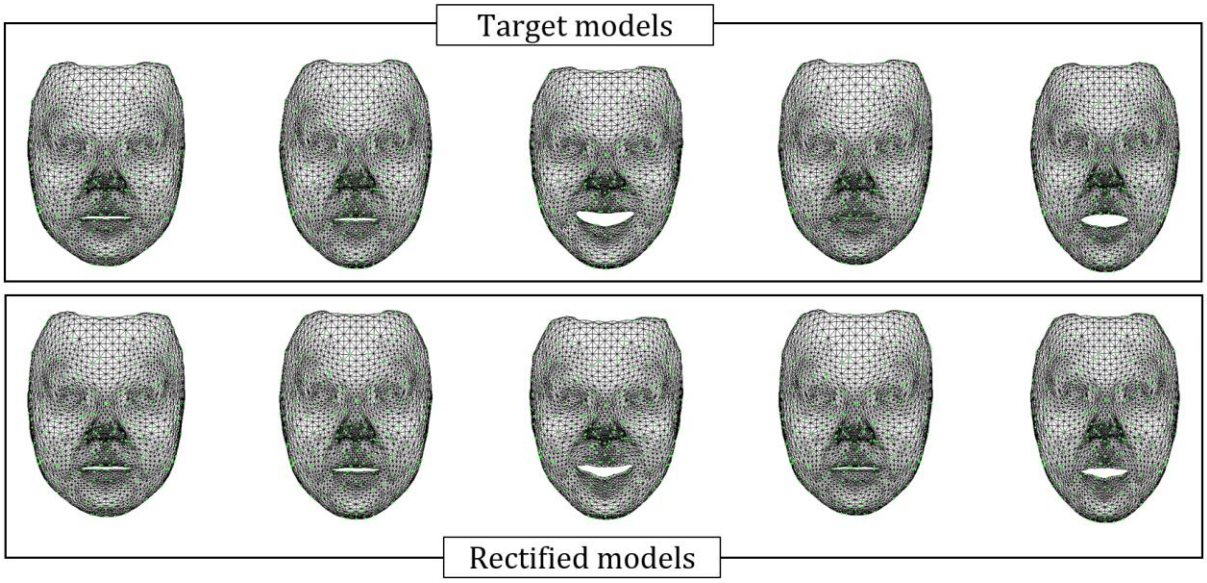

Fig. 12: Benchmark between target models and rectified ones

4. Conclusion

Transforming 3D models based on control points has been widely applied in several practical industries and how to effectively identify control points to serve the rectification of model deformation is still an open issue that has attracted the special attention of researchers and practitioners in the field of image processing and virtual reality. This paper proposes an innovative technique to automatically identify control points based on an algorithm incorporated with RBF interpolation technique. Our proposed approach proves its good performance in our practical experiments with different databases. Thus, our future research will further validate its performance with more collectable deformations of more types of objects that can be modelled in 3D objects and combine with some other interpolation techniques.

Acknowledgement

This paper shows some results from the technology science research project ID B2018-TNA-61. This paper is also partially supported by Lac Hong University under the Decision No. 879/QĐ-ĐHLH.

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

References

Akimoto T, Suenaga Y, and Wallace RS (1993). Automatic creation of 3D facial models. IEEE Computer Graphics and

Applications, 13(5): 16-22.
https://doi.org/10.1109/38.232096

Ansari AN, and Abdel-Mottaleb M (2003). 3D face modeling using two views and a generic face model with application to 3D face recognition. In the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003. IEEE, Miami, USA: 37-44. https://doi.org/10.1109/ICME.2003.1221305

Blanz V and Vetter T (1999). A morphable model for the synthesis of 3D faces. In the 26th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, USA: 187–194.
https://doi.org/10.1145/311535.311556

Buhmann MD (2003). Radial basis functions: Theory and implementations. Vol. 12, Cambridge University Press, Cambridge, UK.
https://doi.org/10.1017/CBO9780511543241

Cao C, Hou Q, and Zhou K (2014). Displaced dynamic expression regression for real-time facial tracking and animation. ACM Transactions on Graphics (TOG), 33(4).
https://doi.org/10.1145/2601097.2601204

Cerveró MÀ, Vinacua A, and Brunet P (2016). 3D Model deformations with arbitrary control points. Computers and Graphics, 57: 92-101.
https://doi.org/10.1016/j.cag.2016.03.010

Fan H, Su H, and Guibas LJ (2017). A point set generation network for 3d object reconstruction from a single image. In the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA: 605-613.
https://doi.org/10.1109/CVPR.2017.264

Fua P (1997). From multiple stereo views to multiple 3-d surfaces. International Journal of Computer Vision, 24(1): 19-35.
https://doi.org/10.1023/A:1007918123901

Hwang J, Kim W, Ban Y, and Lee S (2011). Robust 3D face shape estimation using multiple deformable models. In the 6th IEEE Conference on Industrial Electronics and Applications, IEEE, Beijing, China: 1953-1958.
https://doi.org/10.1109/ICIEA.2011.5975912

Hwang J, Yu S, Kim J, and Lee S (2012). 3D face modeling using the multi-deformable method. Sensors, 12(10): 12870-12889.
https://doi.org/10.3390/s121012870
PMid:23201976 PMCid:PMC3545547

Ip HH and Yin L (1996). Constructing a 3D individualized head model from two orthogonal views. The Visual Computer, 12(5): 254-266.
https://doi.org/10.1007/s003710050063

Jacobson A, Baran I, Popovic J, and Sorkine O (2011). Bounded biharmonic weights for real-time deformation. ACM Transactions on Graphics, 30(4).
https://doi.org/10.1145/2010324.1964973

Lee TY, Lin PH, and Yang TH (2004). Photo-realistic 3d head modeling using multi-view images. In the International Conference on Computational Science and Its Applications, Springer, Melbourne, Australia: 713-720.
https://doi.org/10.1007/978-3-540-24709-8_75

Lee Y, Terzopoulos D, and Waters K (1995). Realistic modeling for facial animation. In the 22nd Annual Conference on Computer Graphics and Interactive Techniques, ACM: 55-62.
https://doi.org/10.1145/218380.218407

Lin IC, Yeh JS, and Ouhyoung M (2002). Extracting 3D facial animation parameters from multiview video clips. IEEE Computer Graphics and Applications, 22(6): 72-80.
https://doi.org/10.1109/MCG.2002.1046631

Luxand (2019). Luxand FaceSDK-Detected facial features. Available online at:
https://bit.ly/2xkGJ3s

Lyons M, Akamatsu S, Kamachi M, and Gyoba J (1998). Coding facial expressions with gabor wavelets. In the Third IEEE international conference on automatic face and gesture recognition, IEEE, Nara, Japan: 200-205.
https://doi.org/10.1109/AFGR.1998.670949

Okabe M and Yamada S (2018). Clustering using boosted constrained k-means algorithm. Frontiers in Robotics and AI, 5: 18.
https://doi.org/10.3389/frobt.2018.00018

Pandzic IS and Forchheimer R (2003). MPEG-4 facial animation: The standard, implementation and applications. John Wiley and Sons Inc., Hoboken, USA.
https://doi.org/10.1002/0470854626

Rezende DJ, Eslami SA, Mohamed S, Battaglia P, Jaderberg M, and Heess N (2016). Unsupervised learning of 3d structure from images. In the 30th Advances in Neural Information Processing Systems, Barcelona, Spain: 4996-5004.

Rock J, Gupta T, Thorsen J, Gwak J, Shin D, and Hoiem D (2015). Completing 3d object shape from one depth image. In the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Boston, USA: 2484-2493.
https://doi.org/10.1109/CVPR.2015.7298863

Zhang Y, Prakash EC, and Sung E (2002). Constructing a realistic face model of an individual for expression animation. International Journal of information Technology, 8(2): 10-25.