An Automatic 3D Face Model Segmentation for Acquiring Weight Motion Area

Rio Caesar
Magister Informatics Engineering
Atma Jaya Yogyakarta University
Sleman, Indonesia
rio.caesar.88@gmail.com

Suyoto
Magister Informatics Engineering
Atma Jaya Yogyakarta University
Sleman, Indonesia
suyoto@mail.uajy.ac.id

Samuel Gandang Gunanto
Department of Animation
Indonesian Institute of the Arts
Yogyakarta, Indonesia
gandang@isi.ac.id

Abstract*— Inside facial animation works there is an animator that need to be skilled enough to produce detailed animation, so the facial animation can be smooth when doing facial expressions. Every animated character requires special handling based on the characteristics of the size and location of the bone. This process, where every face model need special handling were time consuming and tedious work. For that issue this research propose method for using motion capture marker data in 3D face model for automatically segment weight motion area based on the feature point. Marker data that came from motion capture of human model will be used to represent a centroid of vertex cluster that forming expressions in animated character. The data grouping process will be spherical coordinate result calculation between feature point and vertices using modified nearest neighbor algorithm. The result obtained in this research will show the weight motion area that generated automatically from the feature point based on nearest neighbor algorithm in a 3D face model.

***Keywords*—facial animation; segmentation; weight motion area; nearest neighbor; feature point**

I. INTRODUCTION

Human being can easily recognize the non-natural expressions of others human excretion especially from the animated character, changes in motion on the face for displaying an expression or movement of the chin and lips when talking is considered when creating realistic facial animation. Meanwhile, with facial expressions human can do some short of non-verbal communication with others [1]. To approach this naturalness, facial motion capture applied to the human face. The motion capture is aimed at capturing the position and orientation of an object in physical space then record that information to be used and developed in the virtual world [2]. Then 3D face models represent the human face movement.

Main issues that arise from producing detailed animation is the time that consumed in the work process and in the recent days this process is still done manually by the animator. This time consumed process implicate to the cost of the production [3]. Naturally, every movement in the human face always move as one group, one point affect other point and every region affect another region. In 3D face character, after those face model finished it need to be processed again, before delivered to animator for animation to determine the joint and movement controller of the 3D face model. This process is called facial rigging [3], and this process implicate from pre-determining cluster point of the 3D face model.

This research proposes an automatic method on the process of facial animation especially when determining regions that affected from the movement of facial rigging on the 3D face character. This phase proposes approach of feature point with nearest neighbor method as solution when segment the face based on feature point marker, and in this case is marker position on face.

This feature point with nearest neighbor method on 3D face character case using geodesic distance instead Euclidian distance for calculate the distance of every vertex on 3D face model to the feature point because the nearest distance in 3D face model is not straight line but curve that come along with the surface [3], as we assume that 3D face model just like sphere. So, every coordinate of the 3D face model that stored as Cartesian coordinate need to be converted first to the spherical coordinate.

II. RELATED WORK

Segmentation in image processing is a process that aimed to retrieve the object from the image or to divide the image to regions with every object or regions that have attribute similarity. Segmentation itself is can be used for divide shape or color. As for this research the segmentation is used for clustering the vertices of the 3D face model with nearest neighbor algorithm with feature point approach to automatically grouped the member from every vertex.

The main rule from clustering process of nearest neighbor is to identify category of unknown data using already established nearest neighbor data group. This principle already used in many cases, such as pattern recognition [4,5], text categorization [6], and object recognition [7]. This method already gone through many developments to simplified the computation and adaptation to the problem.

Generally, nearest neighbor technique is come to two categories: 1) structure less and 2) based on structure [7]. On the first category, data is grouped into training data and sample data. Distance calculation is performed on the entire training data to the sample data, and if the distance between those point is minimum those point is expressed as the nearest neighbor. As for the second category, based on the name, a data structure is used as reference for computing the nearest neighbor. Both algorithm is still focuses on the data domain of face recognition, meanwhile in this research, structure less technique will be used for determining movement area on 3D face model that have association with the location of the marker.

Data in motion capture consist of movement for the sparse feature points. Aim from using feature point is to simplified the process of facial animation and the challenge in feature point is to produce facial animation as natural as possible with the number point that used is less than the number of point that make up surface of 3D face model [8]. On the other side, using feature point can help to lighten the calculation done by computer than using an algorithm that calculate all the surface point of the 3D face model.

Facial animation is concentrated in creating realistic expression in 3D face model [9]. There are two techniques that used in the making of facial animation 1) based on marker and 2) marker less. By using marker that mean facial animation can be done automatically by calculate the feature point on the 3D face model. While in the marker less, facial animation automation is done by animator that using the surface as comparison.

3D face model can be sees as sphere which is have lots of hills and valley. Because of that, different approach is need to be done to calculate the distance of every vertex back to the feature points. Unlike in the flat surface where shortest distance between two points is straight line and calculated using Euclidean distance, in the round surface the shortest distance between two point is a curve and is calculated using Geodesic distance. By using geodesic distance before the distance can be calculated, the coordinates need to be convert first. As for Euclidean use Cartesian coordinate and Geodesic use spherical coordinate.

III. EXPERIMENTAL DESIGN

Our research conducted on low polygonal 3D human face model data (fig.1) and processed with reference from 33 feature points (fig. 2) acquired from human face motion capture marker data.

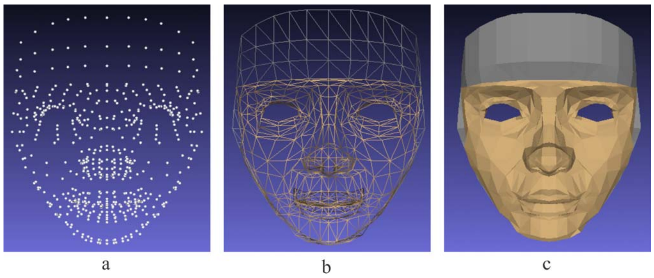

Fig 1. a) 3D face vertices; b) 3D polygonal line; c) low-poly 3D face model

Fig 2. Feature Point

The observation is done by synthesis approach of clustering directly across the surface vertices on 3D models of human face with centroid at the point of the motion features. The aim of this research is for divide member of the vertices of 3D face model into feature point cluster to conclude local deformation in the region that influenced by movement of the feature point.

The early phase of the research is to synthesis the vertices data extracted from 3D human face model and the feature point that mapped based on human face. After that, clustering process is conducted to form a grouping vertex area on the face that will able to become a cluster area of the weight for each feature point on the face motion feature. Grouping method with clustering techniques based on the location of the feature-points on the face is a novelty that we propose in this experiment to lead to the automation of adaptive grouping to any form of 3D face models. As we can see in fig 3, two data input is processed to get synthesis result from the 3D face model vertices and 33 feature points and then next phase is conducted until the segmentation based nearest neighbor of vertices to feature point so can be assigned weight paint for the 3D face model.

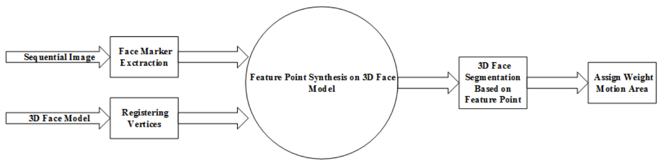

Fig 3. Research Flow

Approach that used in this research when grouping vertex that assemble 3D face model based on feature point is nearest neighbor method where the distance is calculated using spherical coordinates with great circle distance haversine formula.

Base conversion from Cartesian into spherical coordinates that is used for the distance calculation is start by calculated the distance (1), then continue to calculate the latitude (2) and the longitude (3) for each vertex.

$$\rho = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

$$\emptyset = arc \cos\left(\frac{z}{\rho}\right) \quad (2)$$

$$\theta = arc \cos\left(\frac{x}{\rho \sin \emptyset}\right) \quad (3)$$

After we get the right coordinate for calculation, the distance of each vertex to the feature point is calculated using great circle distance haversine formula (9). “a” (7) is the result of calculation using latitude and longitude parameter and is the square of half the cord length between points. While “c” (8) is the angular distance.

$$a = sin^2\left(\frac{\Delta\emptyset}{2}\right) + \cos\emptyset_1 \times \cos\emptyset_2 \times sin^2\left(\frac{\Delta\theta}{2}\right) \quad (7)$$

$$c = 2\ atan2\left(\sqrt{a}, \sqrt{(1-a)}\right) \quad (8)$$

$$d = \rho\ .\ c \quad (9)$$

TABLE I. CONVERSION AND DISTANCE CALCULATION EXAMPLE

	Cartesian			Geodesic			distance
	X	Y	Z	ρ	θ	$\emptyset$	
A	4	5	6	8.774964	0.896055	0.817889	9.127189
B	1	2	3	3.741657	1.107149	0.640522	

As we can see in table 1 as calculation example, coordinate cartesian is converted into spherical coordinate and then the distance is calculated. From the distance of each vertex that we get and compare it with the feature point location, if a point has the closest or minimum distance, those points are expressed as nearest neighbor and will be grouped as member for feature point accordingly.

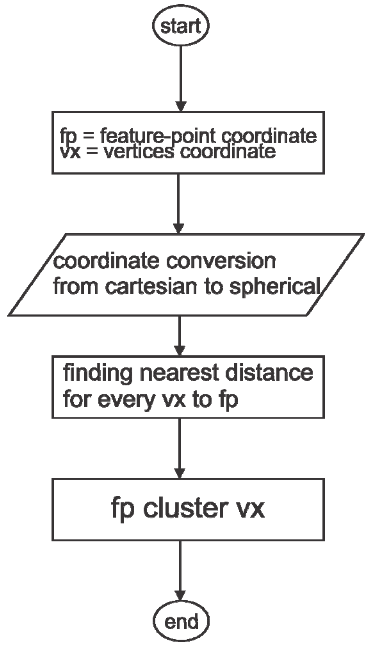

Fig 4. Flowchart fp-NN Clustering (proposed method)

Clustering process used in this research is refer to clustering algorithm *k*-Nearest Neighbor with some modification in the process of Definition value of *k* [3]. Algorithm k-Nearest Neighbor (k-NN) is a method to perform the clustering of objects based on the learning process from data that were located closest to the object. In this case, learning data is the data of vertices which are located close to the point features. The process of modified clustering using k-NN to find a feature point cluster can be seen in the following flowchart (Fig 4), namely feature point Nearest Neighbor(fp-NN) Clustering.

IV. RESULT AND DISSCUSION

Segmentation process in this research is using manual process because this research is focused on the segmentation itself. In the Fig 5, the two data input is processed before to get the 33 feature point data and 3324 vertices that form 3D face model. Those two major data synthesized and calculated to get the distance between 3324 vertices to 33 feature points. This distance then will be observed with nearest neighbor algorithm to group the corresponding vertex from the feature point and form weigh paint of the 3D face model.

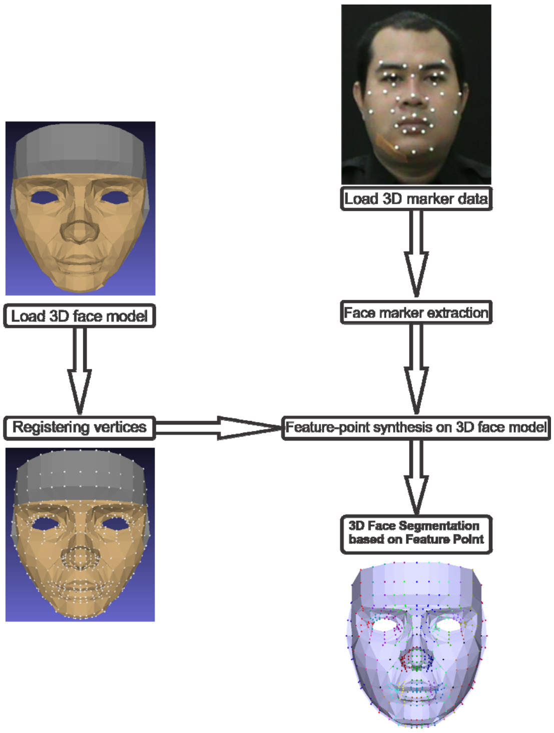

Fig 5. Schematic overview

At first, vertices data from the 3D face model (table 2) and feature point (table 3) need to be extracted and converted into spherical coordinate from originally Cartesian coordinate before calculation of the distance and grouping the vertices.

TABLE II. FACE MODEL COORDINATE

	X	Y	Z
1	-0.0748519	0.173604	0.0779839
2	-0.0841419	0.17433	0.05629
3	-0.0847029	0.161148	0.0449887
4	-0.0748519	0.173604	0.0779839
5	-0.0847029	0.161148	0.0449887
6	-0.0818852	0.15655	0.0647916
7	-0.0127411	0.167827	0.100667
8	-0.0226703	0.167376	0.0964337
9	-0.0207969	0.156281	0.101071
10	-0.0127411	0.167827	0.100667
⋮	⋮	⋮	⋮
3314	-0.0842711	0.213026	0.0673735
3315	-0.0728902	0.220582	0.0957139
3316	-0.0842711	0.233026	0.0653735
3317	-0.0728902	0.220582	0.0957139
3318	-0.0728902	0.235582	0.0947139
3319	-0.0572479	0.187655	0.0858353
3320	-0.0582105	0.179475	0.088362
3321	-0.0576436	0.182243	0.088103
3322	-0.0572479	0.187655	0.0858353
3323	-0.0576436	0.182243	0.088103
3324	-0.0562258	0.189032	0.0817365

TABLE III. FEATURE POINT COORDINATE

	X	Y	Z
1	-0.08414	0.1743	0.05629
2	-0.06985	0.2034	0.08725
3	-0.06173	0.1663	0.08874
4	-0.04735	0.1149	0.07787
5	-0.04721	0.1995	0.09649
6	-0.04205	0.1752	0.09462
7	-0.0391	0.1961	0.09997
8	-0.03842	0.08712	0.06429
9	-0.03477	0.1469	0.09699
10	-0.02803	0.1116	0.08813
⋮	⋮	⋮	⋮
23	0.01811	0.1144	0.102
24	0.02673	0.1113	0.0883
25	0.03477	0.1469	0.09699
26	0.03842	0.08712	0.06429
27	0.0391	0.1961	0.09997
28	0.04205	0.1752	0.09462
29	0.04721	0.1995	0.09649
30	0.04735	0.1149	0.07787
31	0.06173	0.1663	0.08874
32	0.06985	0.2034	0.08726
33	0.08414	0.1743	0.05629

From this point, the vertices data need to be convert from Cartesian to spherical using basic spherical Cartesian conversion method and then continued by calculate the distance using great circle distance haversine formula between two point of each vertex to the feature point (table 4).

TABLE IV. CONVERSION AND DISTANCE CALCULATION

	VERTEX			CENTROID			D
	ρ	θ	Ø	ρ	θ	Ø	
1	0.20450	1.9778	1.1795	0.2015	2.0205	1.2877	0.01286
2	0.20159	2.0204	1.2878	0.2320	1.901	1.1853	0.00001
3	0.18752	2.0547	1.3285	0.1983	1.9262	1.1069	0.01623
4	0.20450	1.9778	1.1795	0.1466	1.9616	1.0110	0.01286
5	0.18752	2.0547	1.3285	0.2265	1.8031	1.1308	0.01623
6	0.18817	2.0527	1.2192	0.2035	1.8063	1.0872	0.02691
7	0.19611	1.6465	1.0317	0.2235	1.7676	1.1071	0.00842
8	0.19449	1.7054	1.0520	0.1148	1.9861	0.9769	0.01508
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3314	0.2387	1.947	1.284	0.154	1.413	0.848	0.0129
3315	0.2512	1.889	1.179	0.144	1.335	0.913	0.0141
3316	0.2562	1.917	1.312	0.179	1.338	0.999	0.0135
3317	0.2512	1.889	1.179	0.114	1.155	0.976	0.0141
3318	0.2641	1.870	1.204	0.223	1.373	1.107	0.0078
3319	0.2141	1.866	1.158	0.203	1.335	1.087	0.0162
3320	0.2083	1.884	1.132	0.226	1.338	1.130	0.0140
3321	0.2104	1.877	1.138	0.146	1.179	1.011	0.0170
3322	0.2141	1.866	1.158	0.198	1.215	1.106	0.0091
3323	0.2104	1.877	1.138	0.232	1.240	1.185	0.0091
3324	0.2134	1.859	1.177	0.201	1.121	1.287	0.0170

Observation in this research is conducted in the process of determining minimum distance for each vertex to the feature point, so it can be assumed as a center for the vertex and become one segment on the location of those cluster centroid. It assumed that after all 33 cluster is done processed, there will be 33 point of segmentations on the face that will be similar with the weight paint or region affected motion. This weigh paint result as propose by researcher in this paper is can be visually shown as in figure 6 where each vertex that correspondent to nearest feature point is specifically colored to some membership.

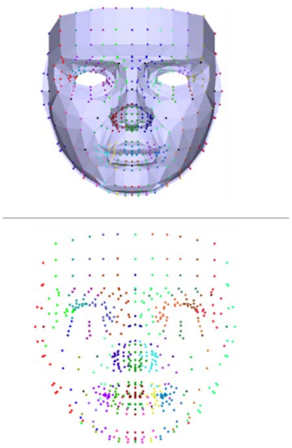

Fig 6. Clustering result for 3D human-face model with 3324 vertexes (low polygonal models)

Table 5 show us sum of member for each feature point correspondently and percentage of each feature point cluster region on surface of the 3D face model.

TABLE V. MEMBERSHIP OF EACH VERTEX TO THE FEATURE POINT

FP	n	%
1	100	3.008424
2	122	3.670277
3	55	1.654633
4	44	1.323706
5	98	2.948255
6	68	2.045728
7	127	3.820698
8	17	0.511432
9	55	1.654633
10	206	6.197353
11	133	4.001203
12	175	5.264741
13	55	1.654633
14	135	4.061372
15	25	0.752106
16	40	1.203369
17	159	4.783394
18	250	7.521059
19	55	1.654633
20	162	4.873646
21	55	1.654633
22	169	5.084236
23	133	4.001203
24	200	6.016847
25	55	1.654633
26	22	0.661853
27	127	3.820698
28	68	2.045728
29	98	2.948255
30	45	1.353791
31	55	1.654633
32	116	3.489771
33	100	3.008424

The result of this research that for acquiring motion influenced region using minimum distance in 3D face model great circle distance haversine formula, if compared with previous research that using Euclidean distance formula [3] for determining distance between vertex and the centroid is have approximate data trend (fig 7 and fig 8) which is going up.

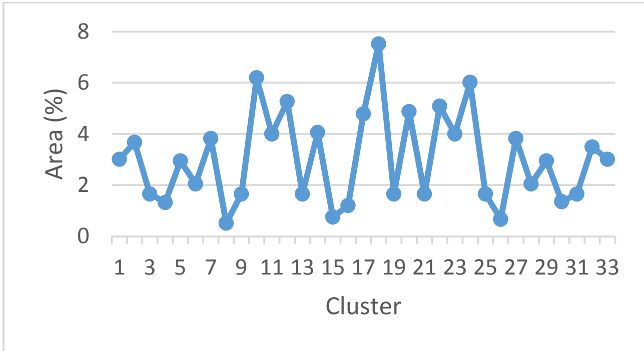

Fig 7. Geodesic data trend

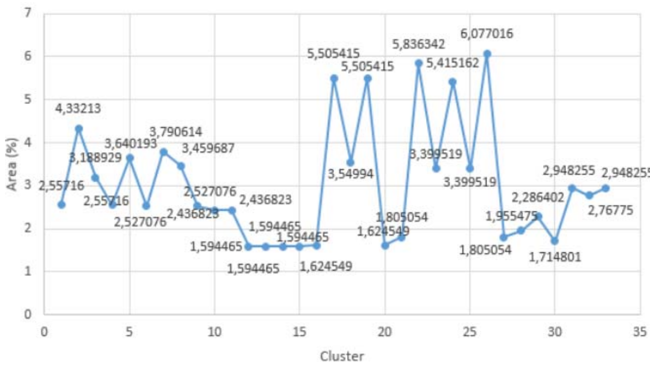

Fig 8. Euclidean data trend

V. CONCLUSION

The experiments were conducted to search for automation process using different distance calculation approach in generating weighted area which affected by the movement of the feature-points on the 3D face model. By compare this research and previous research although different distance calculation method is used, this research also still proves that feature point approach with nearest neighbor algorithm is still able to simplified the process to acquiring motion affected region from centroid deformation.

Even though the clustering process using geodesic distance is work well in low-poly 3D face model as from previous research that using Euclidean distance, performance in this research can be improved and tested with other 3D face model like cartoon character face for acquiring more detailed result from automatic segmentation using feature point or marker.

VI. REFERENCES

[1] S. Chin, C. Yeon Lee, and J. Lee, “An automatic method for motion capture-based exaggeration of facial expressions with personality types”, Virtual Reality, Vol. 17 Springer-Verlag London, 2013, pp.219-237.

[2] J. Bennett and C. Carter, “Adopting Virtual Production For Animated Filmaking”, 7th Annual International Conference on Computer Games, Multimedia and Allied Technologies, 2014

[3] Samuel Gandang Gunanto, Mochamad Hariadi and Eko Mulyanto Yuniarno. Feature-Points Nearest Neighbor Clustering on 3D Face Models. The 4th International Conference on Information Technology for Cyber IT Service Management(CITSM). Bandung, 2016

[4] T. Bailey and A. K. Jain, “A note on Distance weighted k-nearest neighbor rules”, IEEE Trans. Systems, Man Cybernatics, Vol.8, pp 311-313, 1978.

[5] V.Vaidehi, S. Vasuhi, “Person Authentication using Face Recognition”, Proceedings of the world congress on engg and computer science, 2008.

[6] G. Toker, O. Kirmemis, “Text Categorization using k Nearest Neighbor Classification”, Survey Paper, Middle East Technical University.

[7] Bhatia, N and Vandana, “Survey of Nearest Neighbor Techniques”, International Journal of Computer Science and Information Security(IJCSIS) Vol. 8 No. 2, pp. 302-305, 2010.

[8] L. Zhu and W. Sook Lee, “Facial Animation with Motion Capture Based on Surface Blending”, GRAPP, Vol. AS/IE, 2007.

[9] Y. Seol, J. Seo, P. Hyunjin Kim, J. P. Lewis, and J. Noh, “Artist Friendly Facial Animation Retargeting”, ACM Transactions on Graphics, Vol. 30 No. 6 Article 162, 2011.

[10] F. Angiulli, “Fast Condensed Nearest Neighbor”, ACM International Conference Proceedings, Vol 119, pp 25-32.

[11] E. Alpaydin, “Voting Over Multiple Condensed Nearest Neighbors”, Artificial Intelligent Review 11:115-132, 1997.

[12] Geoffrey W. Gates, “Reduced Nearest Neighbor Rule”, IEEE Trans Information Theory, Vol. 18 No. 3, pp 431-433.

[13] T. Liu, A. W. Moore, A. Gray, “New Algorithms for Efficient High Dimensional Non-Parametric Classification”, Journal of Machine Learning Research, 2006, pp 1135-1158.

[14] K. Chidananda and G. Krishna, “The condensed nearest neighbor rule using the concept of mutual nearest neighbor”, IEEE Trans. Information
Theory, Vol IT- 25 pp. 488-490, 1979.

[15] G. Guo, H. Wang, D. Bell, “KNN Model based Approach in Classification”, Springer Berlin Vol 2888.