BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab sebelumnya, penulis dapat menarik kesimpulan sebagai berikut:

1. Pemetaan ulang 33 titik fitur pada wajah model 3D berdasar pergerakan 33 titik *marker* citra wajah 2D membuktikan bahwa metode RBF dapat diterapkan pada transformasi ruang 2D ke 3D.
2. Penerapan metode RBF menghasilkan posisi titik AU yang sesuai untuk enam ekspresi dasar yang diujikan pada wajah model 3D yang memiliki morfologi berbeda dengan citra wajah 2D.

B. Saran

Penelitian ini tidak bisa berhenti hanya pada pemetaan ulang titik fitur pada wajah karakter animasi, namun dilanjutkan pada membangun permukaan wajah karakter animasi. Sehingga ekspresi yang diinginkan dapat terlihat dan diterapkan pada dunia animasi.

Selain pengembangan lebih lanjut dari hasil penelitian ini, sangat dimungkinkan transformasi ruang 2D ke 3D diterapkan menggunakan metode yang lain.

DAFTAR PUSTAKA

Bennett, J. & Carter, C., 2014. Adopting Virtual Production For Animated Filmmaking. In *7th Annual International Conference on Computer Games, Multimedia and Allied Technologies*. Singapore, 2014.

Chenoweth, M.E., 2012. *A Local Radial Basis Function Method for the Numerical Solution of Partial Differential Equations*. Theses, Dissertations and Capstones, Paper 243.

Chin, S., Lee, C.Y. & Lee, J., 2013. An automatic method for *motion capture-*based exaggeration of facial expressions with personality types. *Virtual Reality, Springer-Verlag London*, 17, pp.219-37.

Dutreve, L., Meyer, A. & Bouakaz, S., 2008. Feature Points Based Facial Animation Retargeting. In *ACM Symposium on Virtual Reality Software and Technology*. Bordeaux, France, 2008.

Ekman, P. & Friesen, W.V., 1978. *The Facial Action Coding System: A Technique For The Measurement of Facial Movement*. San Francisco, CA: Consulting Psychologists Press Inc.

Ekman, P. & Rosenberg, E.L., 2005. The Study of Spontaneous Facial. In *WHAT THE FACE REVEALS: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS) 2nd Edition*. Madison Avenue, New York: Oxford University Press. pp.3-18.

Gunanto, S.G., Hariadi, M. & Yuniarno, E.M., 2016. Feature-Points Nearest Neighbor Clustering on 3D Face Models. In *The 4th International*

Conference on Information Technology for Cyber IT Service Management (CITSM). Bandung, 2016.

Gunanto, S.G., Hariadi, M. & Yuniarno, E.M., 2016. Generating Weight Paint Area on 3D Cartoon-Face Models. *INFORMATION - An International Interdisciplinary Journal*, 19.

Gunanto, S.G., Hariadi, M. & Yuniarno, E.M., 2016. Improved 3D Face fp-NN Clustering Using Orthogonal Face Map. *Advanced Science Letters*, 22.

Havaldar, P., 2000. Performance Driven Facial Animation. In *ACM SIGGRAPH*. New York, 2000.

Hubbert, S., Gia, Q.T.L. & Morton, T.M., 2015. *Spherical Radial Basis Functions, Theory and Applications*. Cham Heidelberg New York Dordrecht London: Springer.

Kobayashi, H., Tange, K. & Hara, F., 1995. Real-time Recognition of Six Basic Facial Expressions. In *4th IEEE International Workshop on Robot and Human Communication*. Tokyo, Japan, 1995. IEEE.

Li, B., Zhang, Q., Zhou, D. & Wei, X., 2013. Facial Animation Based on Feature Points. *TELKOMNIKA*, 11(3), pp.1697-706.

Lorenzo, M.S., Edge, J.D., King, S.A. & Maddock, S., 2003. Use and Re-use of Facial *Motion capture* Data. *Vision, Video, and Graphics*, pp.1-8.

Lucey, P. et al., 2010. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010. IEEE.

Na, K. & Jung, M., 2004. Hierarchical retargeting of fine facial motions. *Eurographics*, 23(3).

NaturalPoint, Inc., 2008. *arena expression: 7-step quick start guide*. [Online] Available at: http://www.optitrack.com/static/documents/Expression Quick Start Guide.pdf [Accessed 8 April 2016].

Pighin, F. & Lewis, J.P., 2006. *Facial Motion Retargeting*. course notes Performance-driven Facial Animation. SIGGRAPH.

Seol, Y. et al., 2011. Artist Friendly Facial Animation Retargeting. *ACM Transactions on Graphics (Article 162)*, 30(6).

Song, J., Choi, B., Seol, Y. & Noh, J., 2011. Characteristic facial retargeting. *Comp. Anim. Virtual Worlds*, 22, pp.187-94.

Tian, Y., Kanade, T. & Cohn, J.F., 2001. Recognizing Action Units for Facial Expression Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), pp.97-115.

Umenhoffer, T. & Tóth, B., 2012. Facial animation retargeting framework using radial basis. In *Sixth Hungarian Conference on Computer Graphics and Geometry*. Budapest, 2012.

Zhu, L. & Lee, W.-S., 2007. Facial Animation with *Motion capture* Based on Surface Blending. *GRAPP*, AS/IE.

LAMPIRAN

Lampiran 1. Tabel AU tunggal dalam FACS

AU	*Descriptor*	*Muscular Basis*
1	*Inner Brow Raiser*	*Frontalis, Pars Medialis*
2	*Outer Brow Raiser*	*Frontalis, Pars Lateralis*
4	*Brow Lowerer*	*Depressor Glabellae, Depressor Supercilli; Corrugator*
5	*Upper Lid Raiser*	*Levator Palpebrae Superioris*
6	*Cheek Raiser*	*Orbicularis Oculi, Pars Orbitalis*
7	*Lid Tightener*	*Orbicularis Oculi, Pars Palebralis*
9	*Nose Wrinkler*	*Levator Labii Superioris, Alaeque Nasi*
10	*Upper Lip Raiser*	*Levator Labii Superioris, Caput Infraorbitalis*
11	*Nasolabial Fold Deepener*	*Zygomatic Minor*
12	*Lip Corner Puller*	*Zygomatic Major*
13	*Cheek Puffer*	*Caninus*
14	*Dimpler*	*Buccinator*
15	*Lip Corner Depressor*	*Triangularis*
16	*Lower Lip Depressor*	*Depressor Labii*
17	*Chin Raiser*	*Mentalis*
18	*Lip Puckerer*	*Incisivii Labii Superioris; Incisivii Labii Inferioris*
20	*Lip Stretcher*	*Risorius*
22	*Lip Funneler*	*Orbicularis Oris*
23	*Lip Tightener*	*Orbicularis Oris*
24	*Lip Pressor*	*Orbicularis Oris*
25	*Lips Part*	*Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris*
26	*Jaw Drop*	*Masetter; Temporal and Internal Pterygoid Relaxed*
27	*Mouth Stretch*	*Pterygoids; Digastric*
28	*Lip Suck*	*Orbicularis Oris*

Lampiran 2. Tabel AU yang lebih nyata dalam FACS

AU	*FACS*
8	*Lips Toward Each Other*
19	*Tongue Out*
21	*Neck Tightener*
29	*Jaw Thrust*
30	*Jaw Sideways*
31	*Jaw Clencher*
32	*Lip Bite*
33	*Blow*
34	*Puff*
35	*Cheek Suck*
36	*Tongue Bulge*
37	*Lip Wipe*
38	*Nostril Dilator*
39	*Nostril Compressor*
43	*Eyes Closure*
45	*Blink*
46	*Wink*

Lampiran 3. Tabel relasi titik *marker* dengan titik fitur pada ekspresi netral

Titik	x	y		x	y	z
1	510	431	↔	-0.013	0.203	0.116
2	688	429	↔	-0.071	0.209	0.094
3	398	436	↔	0.013	0.203	0.116
4	233	424	↔	0.071	0.209	0.094
5	683	496	↔	-0.061	0.203	0.099
6	620	513	↔	-0.048	0.195	0.097
7	244	485	↔	0.061	0.203	0.099
8	285	529	↔	0.048	0.195	0.097
9	618	585	↔	-0.042	0.175	0.095
10	650	681	↔	-0.051	0.146	0.093
11	296	592	↔	0.042	0.175	0.095
12	256	676	↔	0.051	0.146	0.093
13	459	517	↔	0.000	0.184	0.107
14	454	681	↔	0.000	0.144	0.126
15	550	681	↔	-0.021	0.156	0.101
16	592	744	↔	-0.031	0.139	0.094
17	363	685	↔	0.021	0.156	0.101
18	326	741	↔	0.031	0.139	0.094
19	541	802	↔	-0.015	0.120	0.094
20	461	795	↔	0.000	0.123	0.107
21	384	814	↔	0.015	0.120	0.094
22	587	854	↔	-0.032	0.113	0.088
23	711	737	↔	-0.066	0.136	0.077
24	328	849	↔	0.032	0.113	0.088
25	216	737	↔	0.066	0.136	0.077
26	536	886	↔	-0.022	0.102	0.093
27	464	893	↔	0.000	0.098	0.099
28	382	891	↔	0.022	0.102	0.093
29	569	982	↔	-0.030	0.078	0.071
30	454	1003	↔	0.000	0.072	0.082
31	349	979	↔	0.030	0.078	0.071
32	769	672	↔	-0.082	0.157	0.065
33	167	669	↔	0.082	0.157	0.065

Lampiran 4. Tabel pemetaan ulang titik *marker* pada ekspresi sedih

Titik	x	y		x	y	z
1	508	438	→	-0.012	0.201	0.116
2	685	434	→	-0.070	0.208	0.094
3	403	445	→	0.012	0.201	0.116
4	239	434	→	0.069	0.207	0.095
5	683	500	→	-0.060	0.202	0.099
6	619	527	→	-0.047	0.192	0.097
7	251	498	→	0.058	0.201	0.099
8	287	543	→	0.047	0.191	0.096
9	617	595	→	-0.041	0.172	0.095
10	656	689	→	-0.053	0.144	0.092
11	301	602	→	0.040	0.172	0.095
12	262	689	→	0.049	0.144	0.094
13	458	523	→	0.000	0.183	0.106
14	456	680	→	0.000	0.144	0.126
15	551	684	→	-0.021	0.156	0.101
16	592	748	→	-0.031	0.138	0.094
17	364	693	→	0.020	0.154	0.101
18	328	748	→	0.030	0.137	0.094
19	535	800	→	-0.013	0.120	0.095
20	458	800	→	0.001	0.122	0.106
21	387	814	→	0.014	0.120	0.095
22	587	859	→	-0.033	0.111	0.087
23	713	748	→	-0.066	0.135	0.076
24	328	857	→	0.033	0.111	0.087
25	221	748	→	0.064	0.134	0.077
26	537	887	→	-0.022	0.101	0.093
27	460	891	→	0.001	0.099	0.099
28	383	896	→	0.022	0.100	0.092
29	567	985	→	-0.030	0.077	0.071
30	458	991	→	-0.001	0.074	0.083
31	360	980	→	0.028	0.077	0.073
32	779	684	→	-0.083	0.155	0.063
33	178	686	→	0.079	0.151	0.067

Lampiran 5. Tabel pemetaan ulang titik *marker* pada ekspresi senang

Titik	x	y		x	y	z
1	514	433	→	-0.014	0.202	0.116
2	686	442	→	-0.070	0.207	0.095
3	393	435	→	0.014	0.203	0.116
4	227	431	→	0.072	0.208	0.094
5	677	503	→	-0.059	0.201	0.099
6	614	531	→	-0.045	0.190	0.096
7	241	485	→	0.062	0.203	0.099
8	286	537	→	0.048	0.193	0.097
9	611	574	→	-0.041	0.178	0.095
10	657	647	→	-0.053	0.155	0.092
11	291	583	→	0.044	0.178	0.094
12	241	644	→	0.057	0.156	0.090
13	457	519	→	0.001	0.184	0.107
14	454	683	→	0.000	0.143	0.126
15	552	660	→	-0.022	0.161	0.101
16	600	719	→	-0.034	0.145	0.095
17	357	663	→	0.023	0.161	0.101
18	313	719	→	0.034	0.143	0.095
19	559	767	→	-0.020	0.131	0.095
20	461	767	→	0.000	0.128	0.111
21	366	779	→	0.019	0.129	0.095
22	609	817	→	-0.036	0.121	0.087
23	725	704	→	-0.071	0.143	0.074
24	311	817	→	0.036	0.120	0.087
25	193	708	→	0.074	0.144	0.071
26	550	895	→	-0.026	0.100	0.090
27	463	906	→	0.000	0.095	0.097
28	372	897	→	0.024	0.100	0.091
29	575	983	→	-0.032	0.078	0.071
30	457	1024	→	-0.001	0.071	0.079
31	338	979	→	0.033	0.079	0.070
32	759	663	→	-0.080	0.157	0.067
33	163	660	→	0.083	0.159	0.065

Lampiran 6. Tabel pemetaan ulang titik *marker* pada ekspresi marah

Titik	x	y		x	y	z
1	506	448	→	-0.012	0.199	0.115
2	679	435	→	-0.069	0.208	0.095
3	410	455	→	0.010	0.199	0.115
4	253	432	→	0.065	0.207	0.097
5	667	503	→	-0.057	0.201	0.099
6	613	533	→	-0.045	0.190	0.096
7	259	494	→	0.056	0.202	0.100
8	298	540	→	0.044	0.192	0.097
9	601	574	→	-0.038	0.178	0.096
10	645	674	→	-0.050	0.148	0.094
11	316	583	→	0.037	0.178	0.096
12	268	672	→	0.047	0.148	0.095
13	462	521	→	-0.001	0.183	0.106
14	453	683	→	0.000	0.143	0.126
15	549	679	→	-0.021	0.157	0.101
16	585	740	→	-0.029	0.140	0.095
17	371	683	→	0.019	0.156	0.103
18	335	740	→	0.028	0.140	0.095
19	526	809	→	-0.012	0.118	0.096
20	462	818	→	0.000	0.118	0.104
21	392	822	→	0.014	0.118	0.095
22	581	848	→	-0.030	0.113	0.089
23	706	736	→	-0.065	0.136	0.078
24	339	848	→	0.029	0.113	0.089
25	230	733	→	0.061	0.136	0.081
26	531	879	→	-0.020	0.103	0.094
27	467	882	→	-0.001	0.102	0.100
28	394	884	→	0.018	0.103	0.095
29	570	977	→	-0.031	0.079	0.072
30	462	996	→	-0.002	0.073	0.083
31	360	975	→	0.028	0.078	0.074
32	777	665	→	-0.083	0.160	0.064
33	182	665	→	0.078	0.155	0.069

Lampiran 7. Tabel pemetaan ulang titik *marker* pada ekspresi jijik

Titik	x	y		x	y	z
1	495	441	→	-0.009	0.200	0.116
2	667	439	→	-0.066	0.207	0.096
3	408	451	→	0.010	0.199	0.115
4	249	435	→	0.066	0.207	0.097
5	665	503	→	-0.057	0.201	0.099
6	606	531	→	-0.043	0.190	0.097
7	258	496	→	0.056	0.202	0.100
8	293	543	→	0.046	0.191	0.096
9	604	580	→	-0.039	0.176	0.095
10	648	663	→	-0.051	0.151	0.094
11	307	585	→	0.039	0.178	0.096
12	253	663	→	0.052	0.150	0.093
13	460	522	→	0.000	0.183	0.106
14	453	679	→	0.000	0.144	0.126
15	543	689	→	-0.019	0.154	0.103
16	590	736	→	-0.030	0.141	0.095
17	364	684	→	0.021	0.156	0.101
18	324	733	→	0.031	0.141	0.094
19	550	794	→	-0.017	0.122	0.093
20	460	790	→	0.000	0.124	0.108
21	376	801	→	0.016	0.123	0.094
22	599	837	→	-0.034	0.117	0.087
23	714	724	→	-0.067	0.139	0.076
24	324	832	→	0.032	0.116	0.088
25	209	724	→	0.068	0.139	0.075
26	538	888	→	-0.022	0.101	0.092
27	463	898	→	0.000	0.097	0.098
28	380	891	→	0.022	0.102	0.093
29	571	983	→	-0.031	0.078	0.071
30	453	1011	→	0.000	0.071	0.080
31	347	978	→	0.031	0.078	0.071
32	768	667	→	-0.082	0.158	0.065
33	176	670	→	0.079	0.155	0.067

Lampiran 8. Tabel pemetaan ulang titik *marker* pada ekspresi takut

Titik	x	y		x	y	z
1	505	446	→	-0.012	0.200	0.115
2	671	446	→	-0.066	0.207	0.096
3	411	451	→	0.010	0.199	0.115
4	249	444	→	0.065	0.206	0.098
5	671	510	→	-0.057	0.200	0.099
6	612	540	→	-0.044	0.188	0.096
7	256	499	→	0.057	0.201	0.100
8	295	549	→	0.045	0.189	0.096
9	607	592	→	-0.039	0.173	0.095
10	650	681	→	-0.051	0.146	0.093
11	313	594	→	0.037	0.175	0.096
12	261	679	→	0.050	0.146	0.094
13	461	526	→	0.000	0.182	0.106
14	454	681	→	0.000	0.144	0.126
15	548	690	→	-0.020	0.154	0.101
16	587	747	→	-0.029	0.138	0.094
17	368	690	→	0.019	0.155	0.102
18	327	745	→	0.030	0.138	0.094
19	539	802	→	-0.014	0.120	0.094
20	463	802	→	0.000	0.121	0.106
21	386	813	→	0.014	0.120	0.095
22	589	847	→	-0.032	0.114	0.088
23	707	743	→	-0.065	0.135	0.077
24	329	847	→	0.032	0.113	0.088
25	217	743	→	0.065	0.135	0.076
26	534	879	→	-0.021	0.103	0.094
27	461	879	→	0.001	0.102	0.100
28	384	884	→	0.021	0.103	0.094
29	566	977	→	-0.030	0.078	0.073
30	459	991	→	-0.001	0.074	0.083
31	352	975	→	0.030	0.078	0.072
32	767	688	→	-0.081	0.153	0.064
33	174	681	→	0.080	0.153	0.066

Lampiran 9. Tabel pemetaan ulang titik *marker* pada ekspresi terkejut

Titik	x	y		x	y	z
1	515	431	→	-0.014	0.203	0.116
2	690	431	→	-0.072	0.208	0.094
3	398	433	→	0.013	0.203	0.117
4	233	422	→	0.072	0.209	0.094
5	688	494	→	-0.062	0.203	0.099
6	625	510	→	-0.050	0.196	0.098
7	244	485	→	0.061	0.203	0.099
8	286	527	→	0.048	0.196	0.098
9	622	585	→	-0.043	0.175	0.094
10	657	674	→	-0.053	0.147	0.092
11	296	592	→	0.042	0.175	0.095
12	254	672	→	0.052	0.147	0.093
13	459	520	→	0.000	0.183	0.107
14	452	681	→	0.000	0.144	0.126
15	552	681	→	-0.021	0.156	0.101
16	594	746	→	-0.031	0.138	0.094
17	363	681	→	0.021	0.157	0.101
18	324	741	→	0.031	0.139	0.094
19	548	797	→	-0.017	0.121	0.093
20	459	788	→	0.000	0.124	0.108
21	375	802	→	0.017	0.123	0.094
22	594	851	→	-0.034	0.114	0.087
23	718	734	→	-0.068	0.137	0.075
24	328	854	→	0.033	0.111	0.088
25	209	734	→	0.068	0.137	0.075
26	541	919	→	-0.024	0.092	0.087
27	464	937	→	-0.001	0.086	0.092
28	384	923	→	0.022	0.091	0.088
29	573	1008	→	-0.030	0.076	0.068
30	457	1040	→	-0.001	0.071	0.077
31	347	1000	→	0.030	0.076	0.069
32	774	674	→	-0.083	0.157	0.064
33	174	669	→	0.080	0.155	0.067

Lampiran 10. *Source Code*

```
int main(int iArgc, char** cppArgv) {
      char buffer[1024] ;
      char *record,*line,*csv;
      char folder[12];
      char address[15];
      char temp[15];
      int bar=0, indeks=0, koor=0;
      FILE *fstream;
      ulang:
      printf("Marker data folder: ");
      scanf("%s", folder);
      for(file=0; file<(ekspresi+1); file++){
            strcpy(address, folder);
            strcat(address, "\\");
            strcat(address, folder);
            strcpy(temp, folder);
            switch(file){
                  case 0 :strcpy(csv,
                        "neutral 3D face model\t\t");
                        strcat(address, "_model.csv"); break;
                  case 1 :strcpy(csv,
                        "human neutral expression\t");
                        strcat(address, "_netral.csv"); break;
                  case 2 :strcpy(csv,
                        "human sad expression\t\t");
                        strcat(address, "_sedih.csv"); break;
                  case 3 :strcpy(csv,
                        "human happy expression\t\t");
                        strcat(address, "_senang.csv"); break;
                  case 4 :strcpy(csv,
                        "human angry expression\t\t");
                        strcat(address, "_marah.csv"); break;
                  case 5 :strcpy(csv,
                        "human disgust expression\t");
                        strcat(address, "_jijik.csv"); break;
                  case 6 :strcpy(csv,
                        "human fear expression\t\t");
                        strcat(address, "_takut.csv"); break;
                  default:strcpy(csv,
                        "human surprise expression\t");
                        strcat(address, "_terkejut.csv");
                        break;}
            printf("\nReading %sfile: %s.", csv, address);
            fstream=fopen(address,"r");
            if(fstream==NULL){
                  printf("\nFailed opening %s\n\n", csv);
                  system("pause");
                  system("cls");
```

```
                goto ulang;}
            bar=0;
            while((line=fgets(buffer,sizeof(buffer),
                fstream))!=NULL){
                record=strtok(line,",");
                while(record !=NULL){
                    if (file==0){
                        if (indeks==0){
                            indeks=atoi(record);
                        } else {
                            model[0][indeks-1][koor]=
                                atof(record);
                            koor++;}
                    } else {
                        if (indeks==0){
                            indeks=atoi(record);
                        } else {
                            human[file-1]
                            [indeks-1][koor]=
                                atof(record);
                            koor++;}}
                    record=strtok(NULL,",");}
                indeks=0;
                koor=0;
            ++bar;}
            strcpy(folder, temp);}
        free(record);
        free(csv);
        training();
        hitungR();
        hitungT();
        qobj=gluNewQuadric();
        glutInit(&iArgc, cppArgv);
        glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
        glutInitWindowSize(winSize, winSize);
        glutInitWindowPosition(0, 0);
        winHum[0]=glutCreateWindow("1 Human ~ Neutral");
        glutDisplayFunc(markerHum0);
        inisialisasi();
        glutInitWindowSize(winSize, winSize);
        glutInitWindowPosition(0, winSize+35);
        winMod[0]=glutCreateWindow("1 Model ~ Neutral");
        glutDisplayFunc(markerMod0);
        inisialisasi();
        glutInitWindowSize(winSize, winSize);
        glutInitWindowPosition((winSize*1)+(30*1), 0);
        winHum[1]=glutCreateWindow("2 Human ~ Sad");
        glutDisplayFunc(markerHum1);
        inisialisasi();
        glutInitWindowSize(winSize, winSize);
        glutInitWindowPosition((winSize*1)+(30*1)+50, 50);
```

```
    winMod[1]=glutCreateWindow("2 Model ~ Sad");
    glutDisplayFunc(markerMod1);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition((winSize*2)+(30*2), 0);
    winHum[2]=glutCreateWindow("3 Human ~ Happy");
    glutDisplayFunc(markerHum2);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition((winSize*2)+(30*2)+50, 50);
    winMod[2]=glutCreateWindow("3 Model ~ Happy");
    glutDisplayFunc(markerMod2);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition((winSize*3)+(30*3), 0);
    winHum[3]=glutCreateWindow("4 Human ~ Angry");
    glutDisplayFunc(markerHum3);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition((winSize*3)+(30*3)+50, 50);
    winMod[3]=glutCreateWindow("4 Model ~ Angry");
    glutDisplayFunc(markerMod3);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition(
          (winSize*1)+(30*1), winSize+35);
    winHum[4]=glutCreateWindow("5 Human ~ Disgust");
    glutDisplayFunc(markerHum4);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition(
          (winSize*1)+(30*1)+50, winSize+95);
    winMod[4]=glutCreateWindow("5 Model ~ Disgust");
    glutDisplayFunc(markerMod4);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition(
          (winSize*2)+(30*2), winSize+35);
    winHum[5]=glutCreateWindow("6 Human ~ Fear");
    glutDisplayFunc(markerHum5);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition(
          (winSize*2)+(30*2)+50, winSize+95);
    winMod[5]=glutCreateWindow("6 Model ~ Fear");
    glutDisplayFunc(markerMod5);
    inisialisasi();
    glutInitWindowSize(winSize, winSize);
    glutInitWindowPosition(
          (winSize*3)+(30*3), winSize+35);
    winHum[6]=glutCreateWindow("7 Human ~ Surprise");
```

```
	glutDisplayFunc(markerHum6);
	inisialisasi();
	glutInitWindowSize(winSize, winSize);
	glutInitWindowPosition(
		(winSize*3)+(30*3)+50, winSize+95);
	winMod[6]=glutCreateWindow("7 Model ~ Surprise");
	glutDisplayFunc(markerMod6);
	glutMainLoop();
	return 0;}

void inisialisasi() {
	glEnable(GL_DEPTH_TEST);
	glClearColor(1, 1, 1, 1);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	glOrtho(0.0, 1.0, 0.0, 1.0, 0.0, 1.0);
	glutSpecialFunc(rotasi);}

void training() {
	printf("\nTraining Data: Neutral -> 3D Face Model\n");
	int sumber;
	for (sumber=0; sumber<titik; sumber++) {
		printf("%11.6f %11.6f\t ->
			%11.6f %11.6f %11.6f\n",
			human[0][sumber][0], human[0][sumber][1],
			model[0][sumber][0], model[0][sumber][1],
			model[0][sumber][2]);
		if (skalaSumber<fabs(human[0][sumber][0]))
			skalaSumber=fabs(human[0][sumber][0]);
		if (skalaSumber<fabs(human[0][sumber][1]))
			skalaSumber=fabs(human[0][sumber][1]);}
	skalaSumber-=skalaSumber/3;}

void hitungR() {
	for (baris=0; baris<titik; baris++) {
		for (kolom=0; kolom<titik; kolom++) {
			j[baris][kolom]=
				sqrt(pow(human[0][baris][0]-
				human[0][kolom][0],2)+
				pow(human[0][baris][1]-
				human[0][kolom][1],2));
			if (tambah==0) tambah=j[baris][kolom];
			if (j[baris][kolom]!=0 &&
				j[baris][kolom]<tambah) tambah=
				j[baris][kolom]; }}
	hitungH();}

void hitungH() {
	for (baris=0; baris<titik; baris++) {
		for (kolom=0; kolom<titik; kolom++) {
			h[baris][kolom]=
```

```
                    sqrt(pow(j[baris][kolom],2)+
                    pow(tambah,2));
               g[baris][kolom]=h[baris][kolom];}
          wx[baris]=model[0][baris][0];
          wy[baris]=model[0][baris][1];
          wz[baris]=model[0][baris][2];}
     hitungG();}

void hitungG() {
     int bawah, atas, lup;
     for (bawah=0; bawah<titik-1; bawah++) {
          for (atas=bawah+1; atas<titik; atas++) {
               m[atas][bawah]=-1*g[atas][bawah]/
                    g[bawah][bawah];
               for (lup=bawah; lup<titik; lup++) {
                    if (lup==bawah)
                         g[atas][lup]=0;
                    else
                         g[atas][lup]+=g[bawah][lup]*
                              m[atas][bawah];}
               wx[atas]+=wx[bawah]*m[atas][bawah];
               wy[atas]+=wy[bawah]*m[atas][bawah];
               wz[atas]+=wz[bawah]*m[atas][bawah];}}
     hitungW();}

void hitungW() {
     int lup, sub;
     for (lup=titik-1; lup>=0; lup--) {
          for (sub=titik-1; sub>=lup; sub--){
               if (lup==sub) {
                    wx[lup]=wx[lup]/g[lup][sub];
                    wy[lup]=wy[lup]/g[lup][sub];
                    wz[lup]=wz[lup]/g[lup][sub];}
               else {
                    wx[lup]=wx[lup]-g[lup][sub]*wx[sub];
                    wy[lup]=wy[lup]-g[lup][sub]*wy[sub];
                    wz[lup]=wz[lup]-g[lup][sub]*wz[sub];
               }}}}

void hitungT() {
     int target, lup;
     for(file=2; file<(ekspresi+1); file++){
          switch(file){
               case 2 :printf("\nRetargeting points:
                    Sad -> 3D Face Model\n"); break;
               case 3 :printf("\nRetargeting points:
                    Happy -> 3D Face Model\n"); break;
               case 4 :printf("\nRetargeting points:
                    Angry -> 3D Face Model\n"); break;
               case 5 :printf("\nRetargeting points:
                    Disgust -> 3D Face Model\n"); break;
```

```
                case 6 :printf("\nRetargeting points:
                    Fear -> 3D Face Model\n"); break;
                default:printf("\nRetargeting points:
                    Surprise -> 3D Face Model\n"); break;}
                for (target=0; target<titik; target++) {
                    for (lup=0; lup<titik; lup++) {
                        rh[lup]=
                            sqrt(pow(sqrt(pow(
                            human[file-1][target][0]-
                            human[0][lup][0],2)+
                            pow(human[file-1]
                            [target][1]-
                            human[0][lup][1],
                            2)),2)+pow(tambah,2));
                        model[file-1][target][0]+=
                            rh[lup]*wx[lup];
                        model[file-1][target][1]+=
                            rh[lup]*wy[lup];
                        model[file-1][target][2]+=
                            rh[lup]*wz[lup];}
                    printf("%11.6f %11.6f\t ->
                        %11.6f %11.6f %11.6f\n",
                        human[file-1][target][0],
                        human[file-1][target][1],
                        model[file-1][target][0],
                        model[file-1][target][1],
                        model[file-1][target][2]);
                    if (skalaTarget<fabs(model[file-1]
                        [target][0]))
                        skalaTarget=
                        fabs(model[file-1][target][0]);
                    if (skalaTarget<fabs(model[file-1]
                        [target][1]))
                        skalaTarget=
                        fabs(model[file-1][target][1]);
                    if (skalaTarget<fabs(model[file-1]
                        [target][2]))
                        skalaTarget=
                        fabs(model[file-1][target][2]);
    }}
    skalaTarget-=skalaTarget/3;}

void rotasi(int key, int x, int y) {
    if (key==GLUT_KEY_RIGHT) rotasix+=15;
    else if(key==GLUT_KEY_LEFT) rotasix-=15;
    else if(key==GLUT_KEY_UP) rotasiy+=15;
    else if(key==GLUT_KEY_DOWN) rotasiy-=15;
    else if(key==GLUT_KEY_PAGE_UP) rotasiz+=15;
    else if(key==GLUT_KEY_PAGE_DOWN) rotasiz-=15;
    for(file=0; file<ekspresi; file++){
        glutSetWindow(winHum[file]);
```

```
            glutPostRedisplay();
            glutSetWindow(winMod[file]);
            glutPostRedisplay();}}

void koordinat() {
      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
      glLoadIdentity();
      glRotatef(rotasix,1,0,0);
      glRotatef(rotasiy,0,1,0);
      glRotatef(rotasiz,0,0,1);
      glColor3f(0, 0, 1);}

void koorHum(int eks){
      float xo, yo;
      koordinat();
      int lup;
      for (lup=0; lup<titik; lup++) {
            glPushMatrix();
            xo=(-1)*(human[eks][lup][0]-
                  human[eks][13][0])/skalaSumber;
            yo=(-1)*(human[eks][lup][1]-
                  human[eks][13][1])/skalaSumber;
            glTranslatef(xo, yo, 0);
            glutSolidSphere(0.02, 32, 32);
            glPopMatrix();}
      glFlush();}

void koorMod(int eks){
      float xo, yo, zo;
      koordinat();
      int lup;
      for (lup=0; lup<titik; lup++) {
            glPushMatrix();
            xo=(model[eks][lup][0]-
                  model[eks][13][0])/skalaTarget;
            yo=(model[eks][lup][1]-
                  model[eks][13][1])/skalaTarget;
            zo=(model[eks][lup][2]-
                  model[eks][13][2]/2)/skalaTarget;
            glTranslatef(xo, yo, zo);
            glutSolidSphere(0.02, 32, 32);
            glPopMatrix();}
      glFlush();}

void markerHum0(){koorHum(0);}void markerMod0(){koorMod(0);}
void markerHum1(){koorHum(1);}void markerMod1(){koorMod(1);}
void markerHum2(){koorHum(2);}void markerMod2(){koorMod(2);}
void markerHum3(){koorHum(3);}void markerMod3(){koorMod(3);}
void markerHum4(){koorHum(4);}void markerMod4(){koorMod(4);}
void markerHum5(){koorHum(5);}void markerMod5(){koorMod(5);}
void markerHum6(){koorHum(6);}void markerMod6(){koorMod(6);}
```

Lampiran 11. **Sertifikat *Author in The 6th InAES 2016***

CERTIFICATE

This is to certify that

Troy

Author

Has contributed in

The 6th International Annual Engineering Seminar (InAES) 2016

on 1 - 3 August, 2016

at Eastparc Hotel, Yogyakarta, Indonesia

Jointly organized by :

Faculty of Engineering, Universitas Gadjah Mada, Universiti Tun Hussein Onn Malaysia (UTHM) and Universitas Muhammadiyah Surakarta (UMS)

Dean of the Faculty of Engineering
Universitas Gadjah Mada

Prof. Ir. Panut Mulyono, M.Eng., D.Eng.

General Chair, InAES 2016

Ir. Lukito Edi Nugroho, M.Sc., Ph.D.

Lampiran 12. *Proceeding Paper*

2D to 3D Space Transformation for Facial Animation Based on Marker Data

Troy
Magister Teknik Informatika
Universitas Atma Jaya Yogyakarta
Sleman, Indonesia
155302474@students.uajy.ac.id

Pranowo
Magister Teknik Informatika
Universitas Atma Jaya Yogyakarta
Sleman, Indonesia
pran@mail.uajy.ac.id

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

***Abstract*—Computer facial animation aims to create an animated character expression as natural as possible as well as human facial expressions. Using the data marker catches facial motion capture, will be determined the location of the feature points of 3D face models to follow the motion of the marker points of human faces. To overcome the morphological differences between the face of the source with the character's face, then applied with radial basis retargeting process mapping so that the character's face can still display the natural expression. Using the data marker 2D, Radial Basis Function (RBF) transformation was applied to determine the position of the feature points on the 3D face models. RBF space transformation has good ability in determining the appropriate facial motion marker points on a human face to the character's face. Motion that occurs in 3D face models is scaled according to the relative scale between the source and the target.**

Keywords*—*facial animation; radial basis function; marker data

I. Introduction

Human beings are very easy to recognize the non-neutrality of other human excretion especially the animated character. Changes in motion on the face for displaying an expression or movement of the chin and lips when talking points to consider in creating realistic animation. Facial expressions have played a key role in the field of nonverbal communication [1]. To approach this naturality, the facial motion capture applied to the human face. Then 3D face models represent the human face movements.

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] [3]. The two greatest challenges for the use of motion capture data is lie in the application of this discrete surface sampling discontinues to drive a fine mesh, and its retargeting to face meshes of different shape and scale [4].

General approaches for facial expression synthesis can be classified into the following three categories: interpolation, muscle-based expression, and performance-driven animation and retargeting [1].

The process of retargeting is the process of transformation of space-based markers, retargeting can be defined as the process of mapping marker dots on the face to 3D face animation source models (target face). Retargeting seeks to create expressions on the characters according to the source animation expression by correlating the marker points on the source animation to 3D face models [5]. There are three techniques for performing mapping:

1. *Linear mapping*

The simplest approach to the retargeting process. The motion of marker position on the source animation will be followed in a linear manner by 3D face models. Linear mapping vulnerable if 3D face models have different size and shape (morphology) with a source face.

2. *Scatter data interpolation*

This is a non-linear approach that is capable of handling a wide range of problems in the process of retargeting. This interpolation process can estimate the location of a new points on a 3D face model of if there is a change in the face of the source by taking into account the weight of the marker points source between feature points on the face with 3D face models. The most common techniques applied in this mapping process is Radial Basis Function (RBF) interpolation, also known as radial basis mapping.

3. *Art direction*

Unlike the above two techniques, art direction needs a trigger from animator to determine the points of change in the 3D face models. Correlation between the source animation with animation targets defined by the animator.

With the condition of the human face and 3D face models which have different sizes and shapes, RBF space transformation has a good ability in determining the appropriate facial motion marker points on a human face to the character's face. Motion that occurs in 3D face models is scaled According to the relative scale between the source and the target [4] [6]. Marker points of the source face define the source space while the feature points of the targets face defining the space targets [7] [8].

Radial base mapping has an advantage in the speed for determining the points of retargeting, especially if the radial base mapping is done on the data marker or motion capture. The marker base (feature points) concept is used to relieve computing, rather than doing a computing on the surface that should calculate the whole points to building 3D face models.

This research uses a 2D facial image as input and doing the retargeting process on 3D face models using space transformation. The process of retargeting using RBF interpolation method. By using space transformation from 2D to 3D, is expected to use a single camera in facial motion capture at source face while delivering the natural expression and emotion in the 3D face models.

II. State of the Art

A. Facial Animation

Facial animation concentrate on the creation of realistic 3D face models by showing the right emotion [9]. There are two techniques performed in facial animation, which is based marker and markerless. Marker-based means of facial animation is done by automation approach such as retargeting feature points. Automation should be able to minimize human intervention in the process [10]. While the markerless can mean automation facial animation by using the surface as a comparison, they can be interpreted facial animation is done manually by the animator. Reference [11] explained that manual editing is always necessary: 1) the movement is expected from the 3D model is not necessarily the same/similar to the movement of people, 2) 3D face model has a proportion of/morphology that is different from the human face, 3) the placement of markers in motion capture is not always fixed and precisely at a points from day to day. Therefore, generating good facial animation is one of the most important factors in the creation of appealing, believable, and attractive characters [6].

B. Feature Points

The motion capture data consists of motions for the sparse feature points. Feature points aim to simplify the process of facial animation. The challenges in feature points are creating an expression as natural as possible with the number of points less than the surface blending techniques [12]. On the other hand, the use of marker means lightens computing compared to a surface algorithm that should calculate the whole points of the face. If it is deemed necessary to reuse the facial animation with a different model [13].

C. RBF Methods

In the 1970s, RBF methods were developed to overcome the structure requirements of existing numerical methods. RBF methods have become a well-established tool for reconstructing functions and for solving partial differential equations based on data prescribed at scattered locations [14].Four common RBFs that are globally supported and infinitely differentiable are Gaussian (1), Inverse Quadratic (2) Inverse Multiquadric (3) Multiquadric (4). The Multi Quadric is arguably the most popular RBF that used in application [15].

$$\phi(r,\varepsilon) = e^{-\varepsilon^2 r^2} \quad (1)$$

$$\phi(r,\varepsilon) = 1/(1+\varepsilon^2 r^2) \quad (2)$$

$$\phi(r,\varepsilon) = 1/\sqrt{1+\varepsilon^2 r^2} \quad (3)$$

$$\phi(r,\varepsilon) = \sqrt{1+\varepsilon^2 r^2} \quad (4)$$

III. Experimental Design

This paper proposes a program of facial animation based on the marker data using RBF space transformation, from the 2D image as source face to 3D character model as target face.

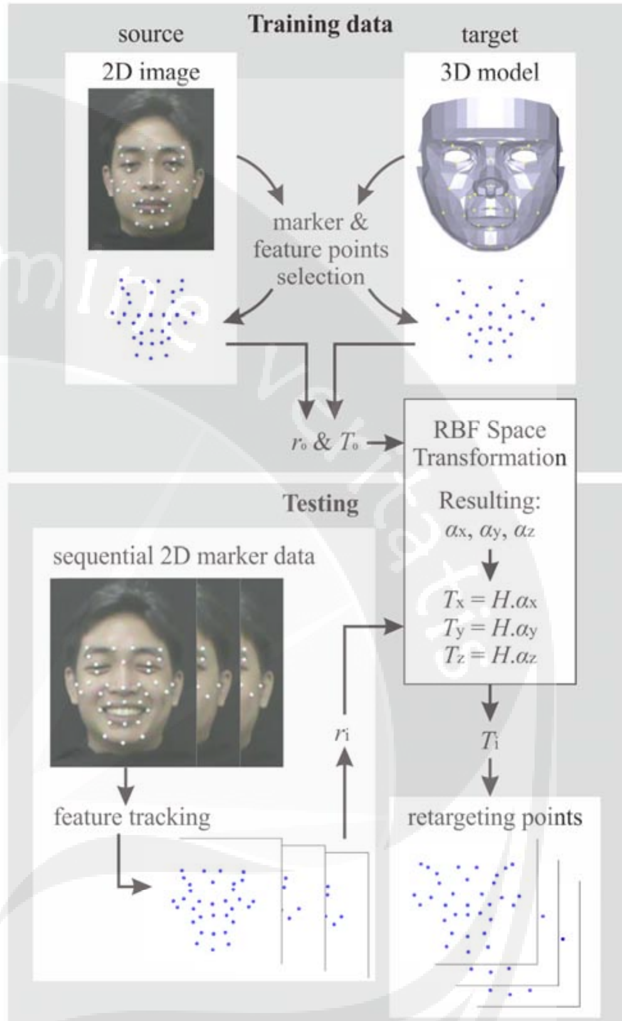


Fig 1. Schematic overview of RBF space transformation

The RBF approach constructs an interpolant as a linear combination of basis function [4].

The RBF equation is:

$$F(x) = \sum_{i=1}^{n} \alpha_i \,.\, \phi(\|x - x_i\|) \quad (5)$$

The value of function ϕ depends on only upon the distance from its center and thus is called radial (6). This radial is the distance between marker points on the face of a 2D facial image.

$$\phi(\|xy_i - xy_j\|) = \sqrt{(\|xy_i - xy_j\|)^2 + r^2} \quad (6)$$

Equation (6) is an RBF multi-quadric. Value *xy* is a marker position in 2D coordinates, so use the Pythagoras theorem to calculate the distance between the marker points. Value *r* determined based on the shortest distance from all marker points on the source face (7).

$$r = min_{i \neq j}(\|xy_i - xy_j\|) \quad (7)$$

Value of $\phi(\|xy_i - xy_j\|)$ is used to build the matrix H. Then the weight value (α) for each coordinate x, y, z on the face of 3D face model obtained by:

$$T_x = H.\alpha_x, T_y = H.\alpha_y, T_z = H.\alpha_z \quad (8)$$

Let $T_x = (t_1^x, t_2^x, t_3^x, \dots, t_n^x)$, $T_y = (t_1^y, t_2^y, t_3^y, \dots, t_n^y)$ and $T_z = (t_1^z, t_2^z, t_3^z, \dots, t_n^z)$, so by applying the Gaussian elimination with back substitution, from (4) is obtained:

$$\alpha_x = H^{-1}.T_x, \alpha_y = H^{-1}.T_y, \alpha_z = H^{-1}.T_z \quad (9)$$

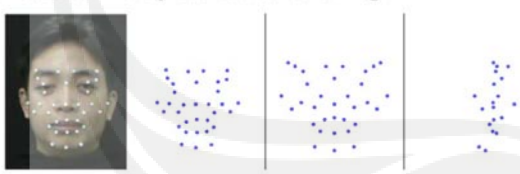
Once the matrix H (6) and the weight of each coordinate x, y, z (9) are obtained, the retargeting feature points can be calculated quickly for each marker position from the motion of source face using (10).

$$F(x) = \sum_{i=1}^{n} \alpha_i^x . \phi(\|xy - xy_i\|),$$
$$F(y) = \sum_{i=1}^{n} \alpha_i^y . \phi(\|xy - xy_i\|),$$
$$F(z) = \sum_{i=1}^{n} \alpha_i^z . \phi(\|xy - xy_i\|) \quad (10)$$

In the testing process, sequential 2D image is extracted to locate the marker points in the source face. After obtained the value of the radial distance from the training data, the RBF space transformation will determine the retargeting points (T) in the 3D character models to calculate the weight (α) and the matrix H (ϕ).The estimation results of the feature points still consider the z coordinates of the 3D face models. So 3D face models still have depth and maintain its 3D shape.

IV. Evaluation Result

The testing of RBF space transformation for facial animation based on the markers data from 2D source to 3D face models will be presented as following.



a b c d

Fig 2. Neutral expression (a) 2D image from motion capture with markers information as inputs (b) marker points from source face (c) feature points from a 3D face models viewed from the front of the target face (d) feature points from the target face that has a depth (z value) viewed from the side.

Fig 2. showing a pair of training data that consists of a source face (2D facial image), in which the image has a marker information in coordinates (x, y) and the feature points of the 3D face model in the coordinates (x, y, z). At the training data, the source face and the target face shows a blank or flat expression.

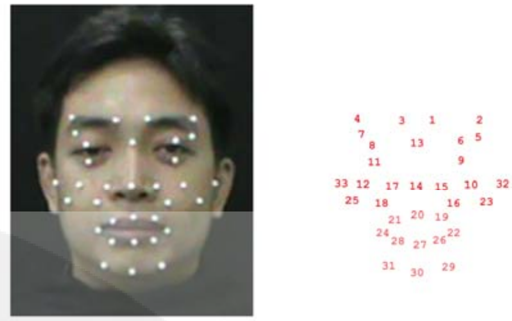


Fig 3. Thirty-three marker points on 2D facial image.

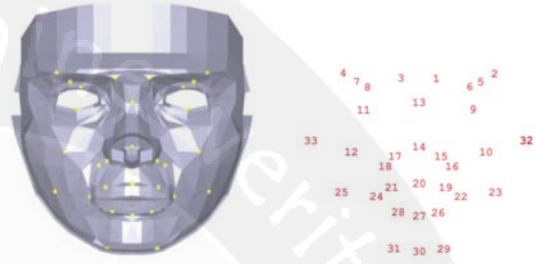


Fig 4. Thirty-three feature points on 3D face model.

There are 33 markers used to mark the feature points in the 2D facial image (Fig 3). This number is adjusted on the camera setup facial motion capture of OptiTrack [16]. The 33 markers in source face will correlate with the 33 feature points on the 3D face models (Fig 4). Motion area was able to be adaptively generated based on 3D face models, it shows the amount of percentage area formed based on feature points [17].

After performing the weighting of the training data, then testing against of 3D face models. Testing is done by the inserting image of faces that have different expressions with a training data image. Marker tracking applied to the image source to determine the coordinates of the new marker. Then RBF space transformation will estimate the 3D coordinates of the feature points for 3D face model.

A. Result

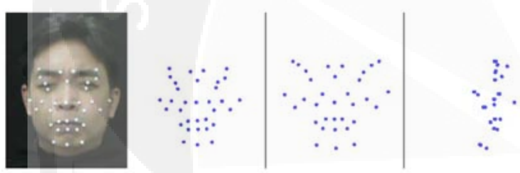
Results obtained are presented in Table 1 which contains the value of the coordinates (x, y) of the marker points of source face and coordinates (x, y, z) of the feature points of 3D face model for training data, and the result of retargeting of angry expression (Fig 5). Coordinates on the training data (source face's marker and 3D face model's feature points) are known, while the retargeting coordinates are calculated results of RBF space transformation.

TABLE I. Feature Points Result Using RBF Space Transformation for Angry Expression

Points	Training Data		Testing	
	Source (x,y)	*Target* (x,y,z)	*Source* (x,y)	*Retargeting* (x,y,z)
1	510,431	-1.30,20.27,11.64	3506,448	-1.19,19.92,11.47
2	688,429	-7.14,20.85,9.39	679,435	-6.90,20.76,9.49
3	398,436	1.30,20.27,11.64	410,455	1.00,19.86,11.49
4	233,424	7.14,20.85,9.39	253,432	6.46,20.74,9.73
5	683,496	-6.08,20.29,9.93	667,503	-5.73,20.10,9.93

6	620,513	-4.83,19.51,9.75	613,533	-4.50,18.97,9.64
7	244,485	6.08,20.29,9.93	259,494	5.61,20.20,10.01
8	285,529	4.83,19.51,9.75	298,540	4.43,19.17,9.71
9	618,585	-4.21,17.52,9.46	601,574	-3.83,17.80,9.56
10	650,681	-5.13,14.62,9.32	645,674	-4.98,14.83,9.40
11	296,592	4.21,17.52,9.46	316,583	3.69,17.82,9.64
12	256,676	5.13,14.62,9.32	268,672	4.73,14.81,9.53
13	459,517	0.00,18.40,10.67	462,521	-0.07,18.31,10.64
14	454,681	0.00,14.38,12.60	453,683	0.02,14.34,12.59
15	550,681	-2.08,15.63,10.11	549,679	-2.06,15.67,10.14
16	592,744	-3.07,13.89,9.39	585,740	-2.87,14.02,9.47
17	363,685	2.08,15.63,10.11	371,683	1.88,15.62,10.31
18	326,741	3.07,13.89,9.39	335,740	2.81,13.98,9.47
19	541,802	-1.49,11.98,9.42	526,809	-1.17,11.77,9.63
20	461,795	0.00,12.28,10.68	462,818	0.01,11.76,10.41
21	384,814	1.49,11.98,9.42	392,822	1.35,11.79,9.51
22	587,854	-3.22,11.25,8.77	581,848	-3.01,11.33,8.86
23	711,737	-6.59,13.63,7.65	706,736	-6.47,13.61,7.78
24	328,849	3.22,11.25,8.77	339,848	2.92,11.26,8.90
25	216,737	6.59,13.63,7.65	230,733	6.13,13.58,8.08
26	536,886	-2.17,10.15,9.28	531,879	-1.97,10.31,9.40
27	464,893	0.00,9.85,9.85	467,882	-0.08,10.15,9.95
28	382,891	2.17,10.15,9.28	394,884	1.83,10.32,9.49
29	569,982	-3.03,7.79,7.15	570,977	-3.07,7.88,7.20
30	454,1003	0.00,7.21,8.16	462,996	-0.22,7.29,8.26
31	349,979	3.03,7.79,7.15	360,975	2.78,7.77,7.37
32	769,672	-8.19,15.66,6.48	777,665	-8.33,15.96,6.42
33	167,669	8.19,15.66,6.48	182,665	7.76,15.52,6.93

The coordinate table above shows that the RBF space transformation able to handle repositioning the feature points of the 3D face models even though the source of motion derived from a marker on the 2D image.



a b c d

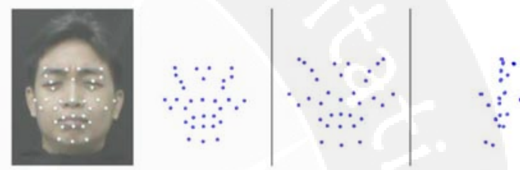
Fig 5. RBF transformation is applied to angry expression (a) 2D image from motion capture which provides information on the new marker (b) marker points from source face (c) feature points for 3D face model that obtained from RBF space transformation (d) feature points seen from the side.

B. Evaluation

Reference [18] developed the Facial Action Coding System (FACS) for describing facial expressions. The FACS is a human-observer-based system designed to describe subtle changes in facial features. FACS consists of 44 action units (AU), including those for head and eye positions. AUs are anatomically related to contraction of specific facial muscles [19].

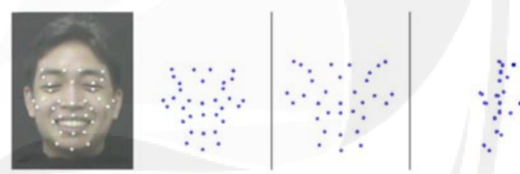
There are six basic expressions on a human face: angry, disgust, happy, sadness, fear, and surprise [20]. Based on [21] an angry expression has AU criteria: a combination of lip tightened and lip pressor. In the feature points (Fig 5.b-d) is shown with a dots in the circle lip (19, 20, 21, 22, 24, 26, 27, 28) which docked/shrink. Disgust expression has AU criteria: wrinkled nose or upper lip raised. In the feature points (Fig 6.b-d) indicated by dots on the nose (14, 15, 16, 17, 18) wrinkles and the upper lip (19, 20, 21) which is raised. The Happy expression has AU criteria: lip corner puller. In the feature points (Fig 7.b-d) indicated by dots on the tip of the lips (22, 24) is widened and rose. Sadness expression has AU criteria: a combination of inner brow raiser + brow lowered + lip depressor or nasolabial deepener corner. In the feature points (Fig 8.b-d) indicated by dots on the brow (1, 2, 3, 4, 5, 7) is lowered and the tip of the lip (22, 24) is depressor. The fear expression has AU criteria: a combination of inner brow raiser + outer brow raiser + brow lowered. In the feature points (Fig 9.b-d) indicated by dots on the inner brow (6, 8) raise up and eyebrows (1, 2, 3, 4, 5, 7) is lower. The surprise expression has AU criteria: a combination of inner brow raiser + outer brow raiser or upper lip raised. In the feature points (Fig 10.b-d) indicated by dots on the brow (1, 2, 3, 4, 5, 7) and the inner brow (6, 8) are raised, as well as the upper lip (19, 20, 21, 22, 24).

Five following pictures (Fig 6-Fig 10) are the other result of the calculation of the RBF space transformation on six basic expressions, which (a) 2D image from motion capture with marker information as input (b) marker points from source face (c) feature points from a 3D face models viewed from the front of the target face (d) feature points from the target face that has a depth (*z* value) viewed from the side.



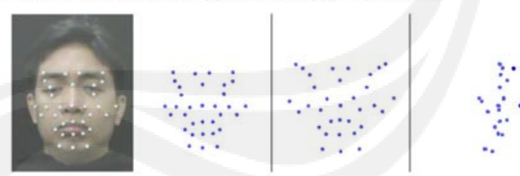
a b c d

Fig 6. RBF transformation is applied to a disgust expression.



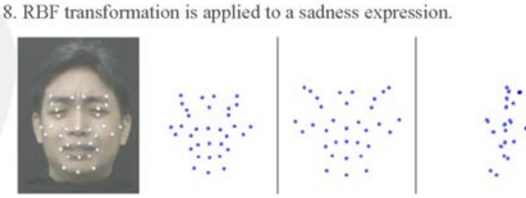
a b c d

Fig 7. RBF transformation is applied to a happy expression.



a b c d

Fig 8. RBF transformation is applied to a sadness expression.



a b c d

Fig 9. RBF transformation is applied to a fear expression.

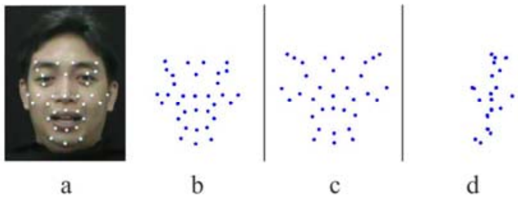


Fig 10. RBF transformation is applied to a surprise expression.

V. CONCLUSION

Referring to Table 1 and introduction to the RBF interpolation, RBF space transformation can relocate the feature points of 3D face model even if the source face is only captured by one camera which can only provide 2D information of marker points.

Based on the evaluation results on six basic expressions, RBF space transformation can reposition feature points on a 3D face model according to the motion of marker on 2D facial image and morphology of the target face. Space transformation experiment aims to automate the process of retargeting feature points on the facial animation.

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 Filmmaking", 7th Annual International Conference on Computer Games, Multimedia and Allied Technologies, 2014.

[3] K. Na and M. Jung, "Hierarchical retargeting of fine facial motions", Eurographics, Vol. 23 No. 3, 2004.

[4] M. S. Lorenzo, J. D. Edge, S. A. King, and S. Maddock, "Use and Re-use of Facial Motion Capture Data", Vision, Video, and Graphics, 2003, pp.1-8.

[5] F. Pighin and J. P. Lewis, "Facial Motion Retargeting", course notes Performance-driven Facial Animation, Siggraph, 2006.

[6] J. Song, B. Choi, Y. Seol, and J. Noh, "Characteristic facial retargeting", Comp. Anim. Virtual Worlds, Vol. 22, 2011, pp.187-194.

[7] L. Dutreve, A. Meyer, and S. Bouakaz, "Feature Points Based Facial Animation Retargeting", ACM symposium on Virtual reality software and technology, 2008, pp.197-200

[8] B. Li, Q. Zhang, D. Zhou, and X. Wei, "Facial Animation Based on Feature Points", TELKOMNIKA, Vol.11 No. 3, 2013, pp.1697-1706.

[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] S. G. Gunanto, M. Hariadi and E. M. 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.

[11] P. Halvadar, "Performance Driven Facial Animation", ACM SIGGRAPH, 2006.

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

[13] S. G. Gunanto, M. Hariadi and E. M. Yuniarno, "Improved 3D Face fp-NN Clustering Using Orthogonal Face Map", Advanced Science Letters, Vol. 22, 2016.

[14] S. Hubbert, Q. Thong Le Gia and T. M. Morton, "Spherical Radial Basis Functions, Theory and Applications", Springer, 2015.

[15] M. E. Chenoweth, "A Local Radial Basis Function Method for the Numerical Solution of Partial Differential Equations", Theses, Dissertations and Capstones, Paper 243, 2012.

[16] "arena expression; 7-step quick start guide". 8 April 2016. NaturalPoint, Inc. http://www.optitrack.com/static/documents/ Expression Quick Start Guide.pdf.

[17] S. G. Gunanto, M. Hariadi and E. M. Yuniarno, "Generating Weight Paint Area on 3D Cartoon-Face Models", INFORMATION - An International Interdisciplinary Journal, Vol. 19, 2016.

[18] P. Ekman and W. V. Friesen, "The Facial Action Coding System: A Technique For The Measurement of Facial Movement", Consulting Psychologists Press Inc., San Francisco, CA, 1978.

[19] Y. Tian, T. Kanade and J. F. Cohn, "Recognizing Action Units for Facial Expression Analysis" IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(2): 97-115.

[20] H. Kobayashi, K. Tange and F. Hara, "Real-time Recognition of Six Basic Facial Expressions", 4th IEEE International Workshop on Robot and Human Communication, 1995, Proceedings, pp.179-186.

[21] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih and Z. Ambadar, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp.94-101.

Lampiran 13. Rangkaian Surat Elektronik InAES 2016

Subject:	[InAES 2016] Your paper #1570273909 ('2D to 3D Space Transformation for Facial Animation Based on Marker Data') is accepted in InAES 2016
From:	inaes2016@ugm.ac.id.edas.info (inaes2016@ugm.ac.id.edas.info)
To:	155302474@students.uajy.ac.id; pran@mail.uajy.ac.id; gandang@isi.ac.id;
Date:	Monday, May 2, 2016 7:06 AM

Dear Mr. Troy Troy:

Congratulations - your paper #1570273909 ('2D to 3D Space Transformation for Facial Animation Based on Marker Data') has been accepted for presentation in InAES 2016, and will be published in the Proceedings of InAES 2016. All presented papers in International Symposium of InAES 2016 within the scope of IEEE will be submitted to IEEE Xplore.

Please reflect the reviewers' comments and make the necessary changes into your final manuscript, the detail reviews are below or can be found at http://edas.info/showPaper.php?m=1570273909, using your EDAS login name 155302474@students.uajy.ac.id.

Please also consider 4 mandatory steps for author in submitting the camera-ready manuscript:

1. Registration and Payment
Please be reminded that the due date for early bird registration is 1 June 2016. At least one author has to register for the conference. Regular or student registration category is based on the 1st author of the paper. Payment and registration details can be found at https://edas.info/r22107 (for International participants) and http://inaes2016.ugm.ac.id/registration/ (for National participants). After payment, please complete the registration confirmation at the bottom part of this link: http://inaes2016.ugm.ac.id/registration/ (for National participant only). Letter of Acceptance can be requested after completing the registration. Note that the Gala Dinner and Full-day Tour have been included in the registration for International Participants.

2. Final Manuscript Preparation
Papers of International Symposium must be written on A4-format page according to the IEEE template Manuscripts of minimum four (4) pages and up to six (6) pages in length. Submissions exceeding 6 pages will be charged extra page length. Maximum accepted pages are eight (8) pages. Papers should NOT have page numbers, headers, and footers. Papers of National Symposium must be written on A4-format page according to template provided in http://inaes2016.ugm.ac.id/author-information/

3. PDF eXpress Plus(IEEE Xplore compatible)
The revised manuscript in International Symposium on Electronics, Control, Computer, and Information and Communication Technology, International Symposium on Geoscience, Geomatic Engineering, and Remote Sensing, International Symposium on Mechanical and Industrial Engineering, International Symposium on Nuclear Engineering and Engineering Physics, and International Symposium on Power and Energy must be compatible with IEEE Xplore. Please check with PDF eXpress Plus at http://www.pdf-express.org/plus/ using the Conference ID: 38737XP. Minimum IEEE PDF requirements can be found at http://inaes2016.ugm.ac.id/camera-ready-submission-guideline/

4. Camera-Ready Paper Upload and Copyright Transfer
After passing the IEEE Xplore compatible, the camera-ready manuscript should be submitted in the same manner as the review manuscript using EDAS at http://edas.info/showPaper.php?m=1570273909 before

15 June 2016. Electronic copyright form for paper in the scope of IEEE should also be submitted electronically through EDAS.

The detail of all steps above can be found at http://inaes2016.ugm.ac.id/camera-ready-submission-guideline/

Should you have any question, do not hesitate to contact us. We look forward to seeing you at the Conference.

Sincerely,

Chair, InAES 2016
http://inaes2016.ugm.ac.id

Detail Reviews:

======= Review 1 =======

> *** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.
Good (4)

> *** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the
analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.
Solid work of notable importance. (4)

> *** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.
Some interesting ideas and results on a subject well investigated. (3)

> *** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.
Readable, but revision is needed in some parts. (3)

> *** Recommendation: How do you rate your recommendation?
Accept. (3)

> *** Detailed comments: Please justify your recommendation and suggest improvements in technical content or presentation.

It is important to describe all parameters in the used equations.
It is necessary to analyze in detail the Figure 3.
RBF stands for Radial Basis Function just needs to be written at once.
There are some mistakes in English grammar and sentence structure.

======= Review 2 =======

> *** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper

within its area of research.
Acceptable (3)

> *** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the
analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.
Marginal work and simple contribution. Some flaws. (2)

> *** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.
Minor variations on a well investigated subject. (2)

> *** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.
Substantial revision work is needed. (2)

> *** Recommendation: How do you rate your recommendation?
Accept. (3)

> *** Detailed comments: Please justify your recommendation and suggest improvements in technical content or presentation.

First of all, the authors need to clearly state the source of the 2D marker points (how to get or from which database) used in this research. I believe it should be taken from other works or some form of sensors. Then, you need to differentiate how the training and testing data are taken, and how the ground truth data was prepared.

The pseudocode in the evaluation section (IVA) can be replaced with an algorithm form.

======= Review 3 =======

> *** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.
Acceptable (3)

> *** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the
analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.
Valid work but limited contribution. (3)

> *** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.
Minor variations on a well investigated subject. (2)

> *** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.
Substantial revision work is needed. (2)

> *** Recommendation: How do you rate your recommendation?

Possible Accept. (2)

> *** Detailed comments: Please justify your recommendation and suggest improvements in technical content or presentation.

Some miss typos and confusing statements are found. The article needs to be carefully proofread.
The results obtained must be systematically evaluated in terms of performance evaluation. The results need to be further analysed. How good the results and what is the significance of these results? Statistical analysis may also be presented as an additional to performance analysis.

Subject:	[InAES 2016] #1570273909 has been uploaded
From:	inaes2016@ugm.ac.id.edas.info (inaes2016@ugm.ac.id.edas.info)
To:	155302474@students.uajy.ac.id; pran@mail.uajy.ac.id; gandang@isi.ac.id;
Date:	Saturday, June 25, 2016 1:52 AM

Dear Mr. Troy Troy:

Thank you for uploading your paper 1570273909 (*2D to 3D Space Transformation for Facial Animation Based on Marker Data*) to **2016 6th International Annual Engineering Seminar (InAES)**. The paper is of type application/pdf and has a length of 1202574 bytes.

You can modify your paper at https://edas.info/showPaper.php?m=1570273909 and see all your submissions at https://edas.info/index.php?c=22107 using the EDAS identifier 155302474@students.uajy.ac.id

Regards,
Chair of InAES 2016

Subject:	InAES 2016 Final Program
From:	inaes2016@ugm.ac.id.edas.info (inaes2016@ugm.ac.id.edas.info)
To:	155302474@students.uajy.ac.id;
Date:	Tuesday, July 5, 2016 8:55 AM

Dear Mr. Troy Troy,

Congratulations - your paper #1570273909 ('2D to 3D Space Transformation for Facial Animation Based on Marker Data') has been accepted for presentation in InAES 2016, and will be published in the Proceedings of InAES 2016. All presented papers in International Symposium of InAES 2016 within the scope of IEEE will be submitted to IEEE Xplore. IEEE reserves the right to exclude a paper from distribution after the conference (e.g., removal from IEEE Xplore) if the paper is not presented at the conference.

InAES 2016 Final Program has just been released. You may find the program at http://inaes2016.ugm.ac.id/final-program/

Thank you for your contribution in InAES 2016. Should you have any question, please do not hesitate to contact us. We look forward to seeing you at the Conference.

Sincerely,

Chair, InAES 2016
http://inaes2016.ugm.ac.id

Subject:	[InAES 2016] #1570273909 has been uploaded
From:	inaes2016@ugm.ac.id.edas.info (inaes2016@ugm.ac.id.edas.info)
To:	155302474@students.uajy.ac.id; pran@mail.uajy.ac.id; gandang@isi.ac.id;
Date:	Wednesday, July 13, 2016 3:35 PM

Dear Mr. Troy Troy:

Thank you for uploading your paper 1570273909 (*2D to 3D Space Transformation for Facial Animation Based on Marker Data*) to **2016 6th International Annual Engineering Seminar (InAES)**. The paper is of type application/vnd.openxmlformats-officedocument.presentationml.presentation and has a length of 9857443 bytes.

You can modify your paper at https://edas.info/showPaper.php?m=1570273909 and see all your submissions at https://edas.info/index.php?c=22107 using the EDAS identifier 155302474@students.uajy.ac.id

Regards,
Chair of InAES 2016

Subject:	InAES #38737 - ATTENTION REQUIRED
From:	inaes2016@ugm.ac.id.edas.info (inaes2016@ugm.ac.id.edas.info)
To:	155302474@students.uajy.ac.id;
Date:	Sunday, January 8, 2017 9:18 AM

Dear Authors,

I am glad to inform that InAES 2016 proceedings passed standard quality for conference proceedings and will be accepted for publication in IEEE Xplore. Thank you for your kindly patience and look forward for your contribution in InAES 2017.

Best regards,

General Chair, InAES 2016

===

Message from IEEE

Dear Dr. Lukito Edi Nugroho,

IEEE has reviewed your conference materials and supplemental Technical Program Questionnaire for the 2016 6th International Annual Engineering Seminar (InAES) (38737) and determined that the proceedings from this conference will be accepted for publication in IEEE Xplore.

IEEE is committed to maintaining the highest quality standards for our events and conference proceedings and we greatly appreciated your patience while the Technical Program Integrity Committee completed their assessments.

As reference for future events, please review page 9 of the following document to help provide guidance on IEEE's current scope of coverage that reflect how content is categorized in IEEE Xplore® *:
http://www.ieee.org/conferences_events/conferences/publishing/paper_acceptance_criteria.pdf

Thank you,

Conference Quality Specialist