

Pengendalian Model 3 Dimensi Wajah Melalui Pendeteksian dan Tracking Titik Fitur Wajah

Sub Judul : Facial Feature Tracking Based Template Matching on Image Sequence

Bitaria Citra Dewi ¹⁾, Zulfan Hakim ²⁾, Sritrusta Sukaridhoto ³⁾, Dadet Pramadihanto ⁴⁾
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
E-mail: ¹⁾ bitaria@kebo.vlsm.org, ²⁾ zulfan@kebo.vlsm.org

Abstrak

Belakangan ini banyak dilakukan penelitian terhadap obyek berupa wajah untuk mendeteksi gerakan wajah terutama ekspresi wajah. Dari pendeteksian wajah maka banyak informasi yang dapat diperoleh, misalnya identifikasi wajah dan analisa ekspresi wajah.

Untuk menampilkan ekspresi wajah yang diambil dari sample wajah melalui kamera digunakan sebuah model animasi 3D . Dengan model ini , user dapat mengendalikan sebuah sistem animasi wajah melalui tahapan proses clustering menggunakan algoritma kmeans, pendeteksian wajah dengan metode integral proyeksi, pendeteksian fitur-fitur wajah dan titik-titik fitur wajah pada image hasil gabungan inversi daerah kulit dan deteksi tepi menggunakan operator sobel. Kemudian dilakukan tracking pada titik-titik tersebut dari frame satu ke frame yang lain secara berurutan untuk menghasilkan gerakan yang dinamis.

Setelah ditentukan titik-titik wajah dan fitur-fitur wajah maka untuk mendapatkan sebuah model wajah 3D, titik-titik tersebut digambarkan menjadi suatu model wireframe, kemudian tekstur mapping bisa dilakukan terhadap gambar wajah pada model wireframe tersebut.

Dalam makalah ini, telah berhasil dibuat model wajah 3D dari gambar 2D berdasarkan hasil pendeteksian titik fitur wajah.

Kata Kunci: *pendeteksian wajah, pendeteksian fitur-fitur wajah, pendeteksian titik-titik fitur wajah, clustering, kmeans, integral proyeksi, inversi, deteksi tepi, sobel operator, wireframe, texture mapping.*

1. Pendahuluan

1.1 Latar Belakang

Tracking atau sistem penjejakan wajah merupakan salah satu pengembangan dari pengolahan citra (*image processing*) dimana metode ini telah diaplikasikan dalam banyak sistem, sebagai contohnya aplikasi ini dapat digunakan untuk melacak posisi wajah seseorang sekaligus mengenali pergerakan wajah beserta fitur-fiturnya. Sehingga dengan dikenali wajah tersebut dapat diaplikasikan pula untuk pengenalan ekspresi wajah seseorang.

Pengenalan ekspresi adalah salah satu riset yang menarik dalam hal mengenali tingkah laku manusia. Sejumlah eksperimen akan dilakukan dengan tujuan menentukan emosi manusia, sehingga komputerpun mampu mengenali ekspresi manusia seperti manusia mampu memahami manusia lainnya saat melakukan interaksi. Seakan-akan komputer dapat merasakan juga apa yang dirasakan oleh manusia dan nantinya dimungkinkan akan membentuk suatu aksi dan interaksi antara keduanya.

Dalam hal ini akan dilakukan pemrosesan video yang diambil dari kamera. Video tersebut akan diproses tiap framenya, yaitu dengan langkah awal membuat template menggunakan frame pertama. Template inilah yang nantinya akan ditracking, sehingga akhirnya dapat diketahui perpindahan titik-titik fitur tersebut pada frame kedua dan seterusnya .

Dari perpindahan titik-titik hasil proses tracking tersebut untuk selanjutnya digambarkan dalam bentuk wireframe lalu dipetakan tekstur gambar wajah yang sesuai dengan letaknya untuk dijadikan pemodelan wajah 3D.

1.2 Permasalahan

Perumusan masalah yang dapat disimpulkan dari pembuatan proyek akhir ini adalah :

- Bagaimana melakukan pengambilan gambar bergerak dari suatu wajah
- Bagaimana mengkonversi file yang berupa video tersebut menjadi file frame atau static image
- Bagaimana membuat template yang nantinya akan dicari pergeserannya dengan frame selanjutnya dari frame pertama
- Bagaimana melakukan tracking pada titik-titik fitur wajah yaitu dengan menghitung korelasi diantara keduanya yaitu antara template dan current frame
- Bagaimana membaca data-data wireframe hingga bisa digambarkan model wireframnya.
- Bagaimana menentukan koordinat-koordinat fitur wajah yang belum diketahui.
- Bagaimana Memodelkan suatu gambar 2D menjadi bentuk gambar 3D.

1.3 Batasan Masalah

Dalam tugas akhir ini permasalahan hanya difokuskan pada pembahasan perangkat lunak yang meliputi metode dan program-program untuk membangun suatu model wajah 3D melalui pendeteksian titik-titik fitur wajah dengan batasan sebagai berikut :

Dalam hal ini, wajah model yang akan diidentifikasi:

- Pengidentifikasian hanya terfokus pada wajah dan hanya dengan tanpa atau sedikit pergerakan
- Hanya satu wajah yang di capture tiap prosesnya
- Pada wajah tidak ada kacamata, kumis dan jenggot
- Hanya melakukan pengambilan video wajah dalam ruangan dengan keadaan pencahayaan yang baik.

2. Dasar Teori

2.1 Pemrograman OpenGL di Linux

OpenGL adalah software (perangkat lunak) interface perangkat keras grafik. Interface ini terdiri dari sekitar 150 command atau perintah yang berbeda yang biasa digunakan untuk menentukan objek dan operasi yang dibutuhkan untuk menghasilkan aplikasi tiga dimensi yang interaktif.

OpenGL didesain untuk memperlancar, di implementasikan pada banyak jenis perangkat keras yang berbeda. OpenGL tidak menyediakan command (perintah) untuk menggambarkan model dari objek tiga dimensi, kita harus membangun model yang diinginkan dari satu set kecil dari geometri seperti titik, garis, dan poligon.

Library yang menyediakan fitur ini tentunya bisa dibuat dengan OpenGL. The OpenGL Utility Library (GLU) menyediakan beberapa fitur modeling, seperti permukaan kuadrik, kurva, dan bentuk-bentuk lainnya. GLU adalah bagian standar dari setiap implementasi OpenGL.

Fungsi-fungsi OpenGL dimuat didalam 2 *library* yang disebut dengan *gl* dan *glu* (atau *GL* dan *GLU*). *Library* yang pertama, adalah fungsi utama dari OpenGL, berisi semua fungsi OpenGL yang dibutuhkan sedangkan yang kedua, *the OpenGL Utility Library* (GLU) memuat fungsi yang ditulis menggunakan fungsi dari *library* utama dan sangat membantu bagi pengguna. Fungsi utama mempunyai nama yang diawali dengan “gl” seperti “glVertex3f()”, sedangkan fungsi didalam GLU mempunyai nama yang diawali dengan “glu” seperti “gluOrtho2D()”.

2.2 PLIB

PLIB merupakan singkatan dari Portable Game Library. PLIB merupakan kumpulan pustaka yang mendukung untuk pembuatan sebuah aplikasi grafik yang menggunakan bahasa pemrograman C++ dan platform grafik openGL. Tujuan dari pembuatan PLIB ini untuk lebih memudahkan proses pembuatan sebuah aplikasi dalam menangani proses grafik seperti manajemen window, penanganan sistem suara, *joystick*, dan perangkat lainnya. PLIB dilengkapi dengan pustaka efek suara, 3D engine, Font Rendering, pustaka window sederhana, script bahasa pemrograman game, pustaka matematis 3D, dan koleksi pustaka lainnya. Seluruh API yang terdapat dalam pustaka PLIB dibangun dengan arsitektur pemrograman berorientasi objek, hal ini akan lebih memudahkan ketika penanganan atau pemanggilan objek yang telah dibuat. PLIB berjalan dalam platform sistem operasi Windows dan Linux dengan lisensi open source. Saat ini PLIB versi terbaru adalah 1.8.4, kode source bisa di dapat di alamat situs <http://plib.sourceforge.net/download.html>. nama dan penjelasan mengenai pustaka yang terdapat di dalam PLIB sebagai berikut.

2.3 Video For Linux Device

Video4Linux merupakan kumpulan API yang berfungsi untuk penanganan video pada sistem operasi linux. Bagi yang telah familiar dengan sistem operasi linux, Video4Linux biasanya sering dikenal dengan “/dev/bttv”. /dev/bttv ini juga membentuk symlink ke /dev/video0.

Klasifikasi perangkat video pada sistem operasi linux berdasarkan type Minor Range adalah sebagai berikut :

Nama Device:	jarak minor:	Fungsi:
/dev/video	0-63	Interface penangkapan video
/dev/radio	64-127	peralatan radio AM/FM
/dev/vtx	192-223	chip untuk perantara Teletext
/dev/vbi	224-239	Raw VBI data (Intercast/teletext)

Pada dasarnya program Video4Linux melakukan proses pembukaan dan pencarian sebuah perangkat yang terkoneksi dengan sebuah PC yang berjalan dengan sistem operasi linux. Kemampuan untuk pencarian ini sepenuhnya ditangani oleh fungsi API yang ada dalam Video4Linux, namun tidak semua fungsi API yang ada mampu berjalan pada sebuah perangkat video. Hal ini bergantung pada jenis perangkat yang digunakan.

2.4 Template Matching

Template matching adalah metode pemfilteran yang sederhana dari pendeteksian fitur tertentu dalam sebuah gambar. Disediakan tampilan dari fitur dalam gambar yang diketahui secara tepat, salah satunya untuk mencoba mendeteksi dengan sebuah operator yang disebut template. Template ini pengaruhnya yaitu subgambar yang terlihat seperti gambar objek. Menghitung kemiripan diperhitungkan bagaimana supaya data gambar cocok dengan template untuk masing-masing lokasi template yang memungkinkan. Titik maksimal yang dicocokkan dapat diseleksi sebagai lokasi dari fitur.

2.5 Korelasi

Salah satu standart untuk pengukuran kemiripan antara fungsi $f(x)$ dan template $t(x)$ adalah jarak Euclidean $d(y)$ yang dikuadratkan, dijelaskan dengan persamaan sebagai berikut ;

$$d(y)^2 = \sum_x [f(x) - t(x-y)]^2 \tag{1}$$

dengan \sum_x artinya $\sum_{x=-M}^M$, untuk beberapa M, N didefinisikan ukuran dari luas template. Jika gambar pada titik y cocok maka $d(y)=0$, jika tidak maka $d(y)>0$. Perluasan ekspresi untuk d^2 dapat dilihat dengan persamaan;

$$d^2(y) = \sum_x [f^2(x) - 2f(x)t(x-y) + t^2(x-y)] \tag{2}$$

sebagai catatan bahwa $\sum_x t^2(x-y)$ adalah konstan dan dapat diabaikan. Ketika $\sum_x f^2(x)$ kurang lebih konstan ini juga dapat dipotong, hal ini disebut sebagai cross corelation antara f dan t .

$$R_{ft}(y) = \sum_x f(x) t(x-y) \tag{3}$$

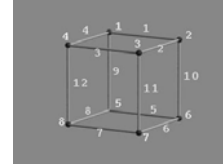
2.6 Wireframe Model

Wireframe model mungkin cara terlama dalam merepresentasikan gambar solid, tepatnya dikembangkan pada awal tahun 1960-an. Tetapi sebelum itu hanya ada system 2D yang membuat kita harus membuat sebuah model dengan titik per titik atau baris per baris. Wireframe data terdiri dari 2 tabel, yaitu tabel

vertex dan tabel edge. Tabel vertex berisi sebuah vertex (titik) dengan nilai koordinatnya. Sedangkan tabel edge berisi titik awal dan titik akhir dari tiap tepi (edge), tepi tersebut bisa berupa kurva atau garis.

Wireframe ini disimpan sebagai struktur data. Filenya kecil dan waktu aksesnya pendek.

Dalam komputer grafik 3D, vertex merupakan sebuah titik dalam ruang 3D, yang didefinisikan sebagai sumbu x, y, dan z.



Gambar 1: Model wireframe kubus

Tabel 1 : Vertex wireframe kubus

Vertex Table			
Vertex #	X	Y	Z
1	1	1	1
2	1	-1	1
3	-1	-1	1
4	-1	1	1
5	1	1	-1
6	1	-1	-1
7	-1	-1	-1
8	-1	1	-1

Tabel 2 : Edge wireframe kubus

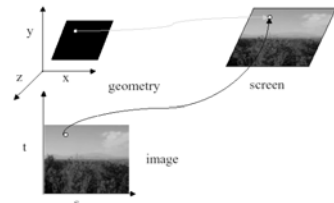
Edge Table		
Edge #	Start Vertex	End Vertex
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8
8	8	5
9	1	5
10	2	6
11	3	7
12	4	8

2.7 Texture Mapping

Texture mapping merupakan teknik pemetaan sebuah tekstur pada pola gambar wireframe, dimana wireframe yang telah dibuat akan ditampilkan memiliki kulit luar seperti tekstur yang diinginkan. Dalam pemberian tekstur, perlu diperhatikan dasarnya seperti:

- Menentukan tekstur
 - Membaca atau membangkitkan tekstur
 - Menandai tekstur
 - Mengenablekan tekstur

- Menandai koordinat tekstur pada vertek
- Menentukan parameter tekstur
 - wrapping , filtering.



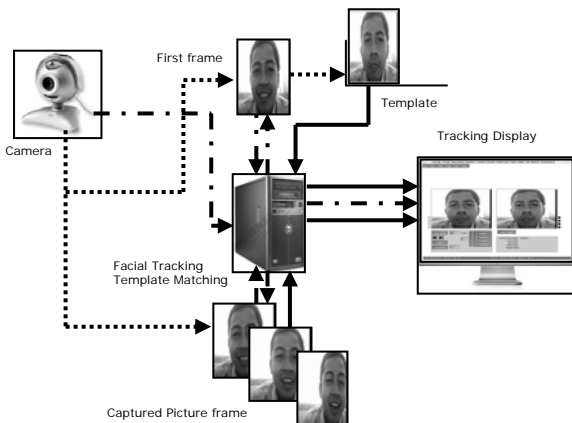
Gambar 2 : Strategi dasar Texture Mapping

Gambar tersebut merupakan tekstur dengan gambar 256x256 yang telah dimappingkan pada sebuah polygon persegi panjang pada perspective.

3. Perencanaan Dan Pembuatan

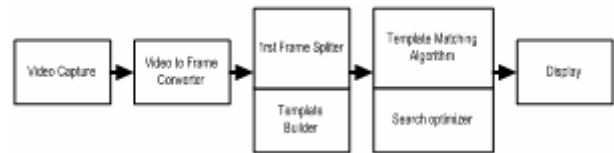
3.1. Diagram Sistem

Pada dasarnya sistem yang dibangun merupakan sebuah perangkat lunak yang memiliki fungsi melakukan tracking wajah pada gambar sekuen dari video. Algoritma yang digunakan adalah template matching dengan pendekatan korelasi. Sistem ini dibangun pada platform sistem operasi Linux. Berikut adalah blok diagram sistem face tracking;

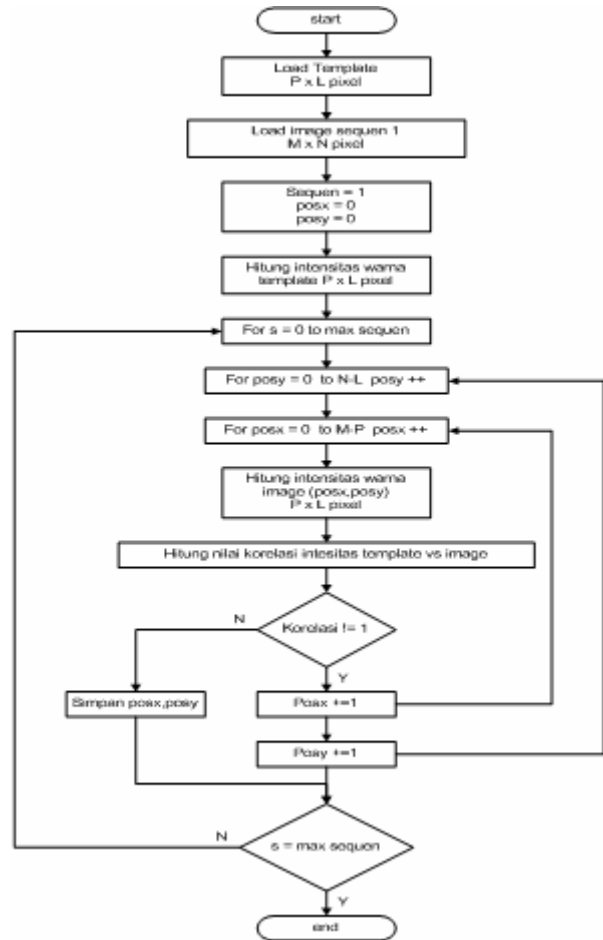


Gambar 3. Blok diagram sistem face tracking

Pada gambar 1 merupakan blok diagram sistem yang akan dirancang yang ditinjau dari masukan yang diolah dan target yang ingin dicapai. Bagian utama dari sistem ini adalah pada gambar 1 yaitu face tracking sistem. Bagian ini memiliki arsitektur seperti tampak pada gambar 2, sistem memiliki beberapa bagian yang meliputi bagian pengambilan data input yaitu berupa video capture subsistem, yang kedua adalah bagian subsistem konversi dari gambar video yang diambil menjadi bentuk frame. Dibagian ketiga terdapat sistem yang memisahkan frame pertama dengan frame ke-2 sampai dengan ke-n. Bagian selanjutnya adalah subsistem yang merupakan implementasi algoritma, dimana semua proses perhitungan dengan algoritma korelasi dilakukan. Pada bagian akhir adalah subsistem untuk menyimpan dan menampilkan hasil yang diperoleh dari proses algoritma.



Gambar 4. face tracking sistem



Gambar 5. Flowchart Face Tracking

Sistem perangkat lunak ini dibangun dengan menggunakan bahasa pemrograman C dengan dukungan pustaka OpenGL, PLIB, dan berjalan pada sistem operasi Linux versi kernel 2.6.8. detail spesifikasi teknis PC dan pustaka tertera pada tabel 1 sebagai berikut;

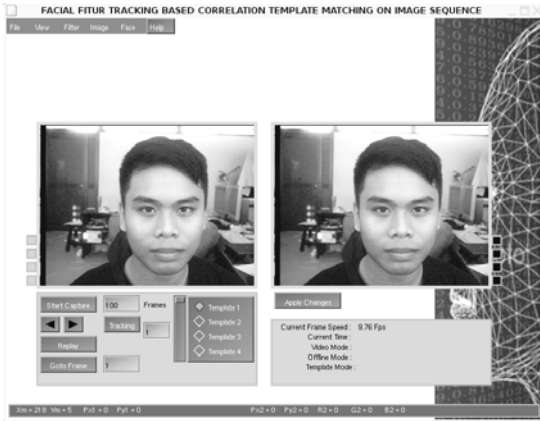
Tabel 3. Spesifikasi teknis perangkat lunak face tracking sistem.

No	Deskripsi	Spesifikasi
1	CPU	P IV 1700 MHz 40 Gb Hardisk
2	RAM	256 MB
3	Graphic Card	Nvidia MX2 400 64MB
4	GCC	Versi 3.3
5	PLIB	1.8.3
6	Video4Linux	Versi 1
7	OpenGL	Versi 2.0
8	Sistem Operasi	Linux/GNU Debian
9	Kernel	2.6.8-386

3.2 Window Utama

Window utama pada sistem ini memiliki ukuran awal 800x600 pixel. Kemungkinan untuk merubah ukuran window di sistem ini masih dimungkinkan. Window utama inilah yang akan menjadi ruang tampilan untuk seluruh proses yang dilakukan oleh sistem face tracking.

Berikut adalah gambar dari rancangan window utama;



Gambar 4. Window utama

4. Pengujian Dan Analisa Software

4.1 Pengambilan gambar



Gambar 5. proses pengambilan gambar

4.2 Pengujian fungsionalitas

Sebelum melakukan pengujian sistem terintegrasi, pada sistem ini dilakukan pengujian fungsi secara perbagian. Tujuan dari pengujian ini memastikan bagian-bagian dari sistem berfungsi dengan baik, sehingga pada saat diintegrasikan dapat memperkecil terjadinya kesalahan yang disebabkan oleh kegagalan sub proses dari sistem tracking.

4.2.1 Fungsi pembuatan template

Sistem ini dilengkapi dengan fasilitas fungsi yang berguna untuk membuat template berdasarkan hasil deteksi titik fitur wajah. Hasil dari pembuatan template yang dibuat tampak pada gambar 4 berikut.



Gambar 6. Hasil pembuatan template

4.2.2 Fungsi penangkapan gambar video

Penangkapan video dilakukan dengan menangkap gambar dari kamera. Data dari kamera ini kemudian disimpan menjadi file bitmap sesuai dengan urutan frame. Tampak pada gambar 5 hasil penangkapan gambar dari kamera.



Gambar 7. Hasil penangkapan gambar dari kamera

4.3 Pengujian performa

Pengujian performa dilakukan dengan sistem yang terintegrasi. Maksud dari pengujian ini adalah untuk mengetahui beberapa faktor yang mempengaruhi sistem, dan juga kecepatan sistem untuk melakukan tracking. Pada pengujian ini dilakukan dengan mode offline.

Pada mode offline gambar frame yang telah ditangkap dibaca kembali dan dibuat template untuk masing-masing kumpulan sekuen frame. Dari template yang dibuat ini kemudian dilakukan pengujian tracking untuk masing masing frame.

4.3.1 Tracking titik pada fitur wajah

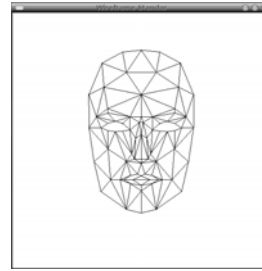
Dengan menggunakan algoritma yang telah dijelaskan, pada sistem ini juga diujikan pada sistem tracking titik fitur wajah yang meliputi titik-titik pada mata, alis, hidung dan, mulut. Tabel 4 berikut ini hasil pengujian tracking untuk beberapa frame sekuen.

Tabel 4. koordinat hasil tracking titik fitur

Titik	frame 1		frame 2		frame 3	
	x	y	x	y	x	y
48	114	113	114	114	115	114
49	108	107	108	108	108	108
51	113	113	113	113	113	113
50	112	112	112	112	112	112
17	115	115	115	115	115	115
16	107	107	107	107	107	107
18	111	111	111	111	111	111
15	112	111	111	112	111	112
3	116	115	116	116	116	116
53	127	126	127	127	127	127

56	131	131	132	131	132	131
52	121	121	121	121	121	121
57	132	132	132	132	132	132
23	130	129	130	130	130	130
20	126	126	127	126	126	126
19	120	120	120	120	120	120
24	133	132	133	133	133	133
5	160	160	160	160	160	160
6	167	167	167	167	167	167
59	169	169	169	169	169	169
26	169	169	169	169	169	169
76	164	164	165	164	165	164
75	163	163	163	163	164	163
64	189	188	189	189	189	189
31	188	188	188	188	188	188
7	183	183	183	183	183	183
8	201	201	201	201	201	201
83	191	191	191	191	191	191
81	190	190	190	190	190	190
89	190	189	190	190	190	190
88	189	189	189	189	189	189
10	222	222	222	222	223	222
0	73	73	74	73	73	73
62	123	123	124	123	124	123
29	121	121	122	121	121	121

Kemudian bila titik-titik tersebut digambarkan menjadi sebuah wireframe akan terlihat seperti gambar 14 berikut :



Gambar 14 : Model wireframe standar

4.4.1 Model Wajah 3D Berdasarkan Proses Pendeteksian Wajah dan Letak Bagian Wajah

Gambar wajah yang akan dimodelkan adalah sebagai berikut :

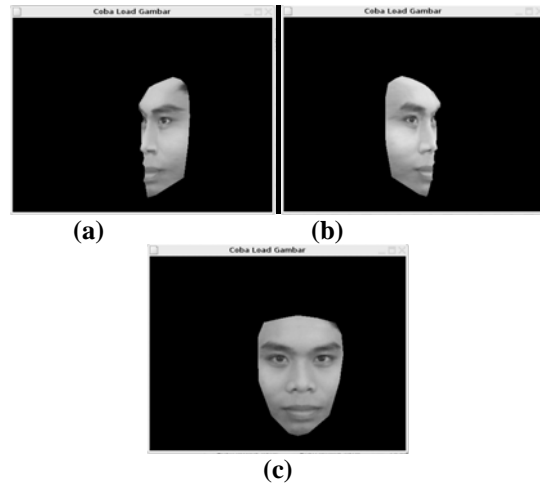


Gambar 15. Model wajah

Langkah terakhir adalah tekstur mapping gambar wajah pada wireframe, hasilnya dapat dilihat pada gambar 16 berikut :



Gambar 12. Hasil tracking titik fitur wajah



Gambar 16 : Hasil tekstur mapping (a) tampak samping kiri (b) tampak samping (c) tampak depan

4.4 Wireframe standar

Letak titik-titik wireframe standar yang digunakan ditunjukkan oleh gambar 13 berikut :



Gambar 13 : Letak titik-titik wireframe standar

5. Penutup

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan didapatkan beberapa kesimpulan untuk proses tracking dengan menggunakan metode korelasi.

1. Proses tracking dipengaruhi oleh faktor brightness. Jarak maksimum mean antara template dengan gambar yang ditracking adalah 138.

2. Kecepatan tracking dipengaruhi oleh besar template yang digunakan. Semakin besar ukuran template akan semakin lama proses tracking.
3. Untuk gambar template yang memiliki sifat dinamis seperti mata, mulut dapat ditracking menggunakan jenis template titik fitur sehingga mampu melacak perubahan pada fitur tersebut.
4. Nilai SSD tidak bergantung pada korelasi, tetapi sebaliknya jika nilai SSD sama dengan 0 maka nilai korelasi sama dengan 1.
5. Nilai korelasi merupakan konstanta pengali tingkat kecerahan pada gambar dengan template.
6. Tracking posisi secara presisi tidak ditentukan oleh nilai korelasi maksimum. Hal ini dikarenakan gambar yang ditracking dengan template mengalami kecenderungan perubahan orientasi.
7. Sistem yang telah dibangun ini masih belum sempurna, terbatasnya jumlah titik yang digunakan untuk membuat model jaring 3D (wireframe) menjadikan pembuatan model 3 D wajah masih terlihat kaku.
8. Titik yang digunakan sebagai acuan untuk membuat model jaring 3D belum memenuhi jumlah titik yang dibutuhkan, sehingga dilakukan perhitungan tiap-tiap titik yang belum diketahui.

Referensi

- [1] Helmut Prendinger and Mitsuru Ishizuka, *Introducing the Cast for Social Computing: Life-like Characters*, life-like.pdf, The University of Tokyo, 2003
- [2] Dana Harry Ballard & Cristopher M. Brown, *Computer Vision*, Prentice Hall Professional Technical Reference, 1982.
- [3] Vernon, David, "*Machine Vision*" *Automated Visual Inspection & Robot Vision*, Prentice Hall, 1991.
- [4] B S Venkatesh, S Palanivel and B Yegnanarayana, *Face Detection and Recognition in an Image Sequence using Eigenedginess*, Department of Computer Science and Engineering, Indian Institute of Technology, Madras.
- [5] Karl Schwerdt and James L. Crowley, *Robust Face Tracking using Color*, Projet PRIMA, INRIA Rh^{one}-Alpes, 655 ave. de l'Europe, 38330 Montbonnot St. Martin France.
- [6] Mikhail J. Atallah, Faster Image Template Matching in the Sum of the Absolute Value of Differences Measure, *IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 4, APRIL 2001*.
- [7] Ahmad, Usman, *Pengolahan Citra Digital dan Teknik Pemrogramannya*, Yogyakarta; Penerbit Graha Ilmu, 2005.
- [8] <http://linux.bytesex.org/v4l2/API.html> diakses terakhir 28 juni 2006
- [9] <http://plib.sourceforge.net/pui/index.html> diakses terakhir 10 juni 2006