



**DISERTASI - TE 143597**

**SEMI-AUTOMATIC VIDEO OBJECT EXTRACTION  
MENGUNAKAN ALPHA MATTING BERBASIS MOTION  
ESTIMASI**

**RURI SUKO BASUKI  
2210 301 008**

**DOSEN PEMBIMBING  
Prof. Ir, Mauridhi Hery Purnomo, M.Eng., Ph.D  
Mochamad Hariadi, ST., M.Sc., Ph.D..**

**PROGRAM DOKTOR  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2016**



DISSERTATION - TE 143597

**SEMI-AUTOMATIC VIDEO OBJECT EXTRACTION  
USING ALPHA MATTING WITH MOTION ESTIMATION BASED**

**RURI SUKO BASUKI**  
2210 301 008

**SUPERVISOR**

Prof. Ir, Mauridhi Hery Purnomo, M.Eng., Ph.D  
Mochamad Hariadi, ST., M.Sc., Ph.D..

**DOCTORAL PROGRAM  
ELECTRICAL ENGINEERING DEPARTMENT  
INDUSTRIAL TECHNOLOGY FACULTY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2016**

## LEMBAR PENGESAHAN

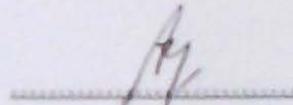
### *Semi-Automatic Video Object Extraction Menggunakan Alpha Matting Berbasis Motion Estimation*

Disertasi ini disusun untuk memenuhi salah satu syarat memperoleh gelar  
Doktor (Dr.)  
di Institut Teknologi Sepuluh Nopember Surabaya

Oleh :  
Ruri Suko Basuki  
NRP. 2210301008

Tanggal Ujian : 13 Januari 2016  
Periode Wisuda : Maret 2016

Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng  
NIP. 19580916 198601 1 001



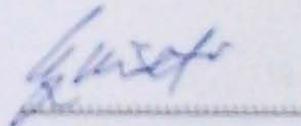
(Pembimbing I)

Mochamad Hariadi, ST., M.Sc., Ph.D  
NIP. 19691209 199703 1 002



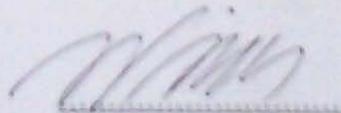
(Pembimbing II)

Dr. Wiseto Agung  
NIK. 632980



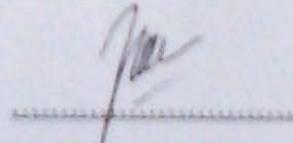
(Penguji I)

Dr. Ir. Wirawan, DEA  
NIP. 19631109 198903 1 011



(Penguji II)

Dr. I Ketut Eddy Purnama, ST., MT.  
NIP. 19690730 199512 1 001



(Penguji III)



Mengetahui,  
Direktur Program Pascasarjana ITS

  
Prof. Djauhar Manfaat, M.Sc., Ph.D  
NIP. 19601202 198701 1 001

## ***Semi-Automatic Video Object Extraction Menggunakan Alpha Matting Berbasis Motion Estimation***

Nama Mahasiswa : Ruri Suko Basuki  
NRP : 2210301008  
Pembimbing : Prof. Ir. Mauridhi Hery Purnomo, M.Eng., Ph.D.  
Co-Pembimbing : Mochamad Hariadi, S.T., M.Sc., Ph.D.

### **ABSTRAK**

*Ekstraksi objek merupakan pekerjaan penting dalam aplikasi video editing, karena objek independen diperlukan untuk proses compositing. Proses ekstraksi dilakukan dengan image matting diawali dengan mendefinisikan scribble manual untuk mewakili daerah foreground dan background, sedangkan daerah unknown ditentukan dengan estimasi alpha.*

*Permasalahan dalam image matting adalah piksel dalam daerah unknown tidak secara tegas menjadi bagian foreground atau background. Sedangkan dalam domain temporal, scribble tidak memungkinkan untuk didefinisikan secara independen di seluruh frame. Untuk mengatasi permasalahan tersebut, diusulkan metode ekstraksi objek dengan tahapan estimasi adaptive threshold untuk alpha matting, perbaikan akurasi image matting, dan estimasi temporal constraint untuk propagasi scribble. Algoritma Fuzzy C-Means (FCM) dan Otsu diaplikasikan untuk estimasi adaptive threshold.*

*Dengan FCM hasil evaluasi menggunakan Means Squared Error (MSE) menunjukkan bahwa rata-rata kesalahan piksel di setiap frame berkurang dari 30.325,10 menjadi 26.999,33, sedangkan dengan Otsu menjadi 28.921,70. Kualitas matting yang menurun akibat perubahan intensitas pada image terkompresi diperbaiki menggunakan Discrete Cosine Transform (DCT-2D). Algoritma ini menurunkan Root Means Squared Error (RMSE) dari 16.68 menjadi 11.44. Estimasi temporal constraint untuk propagasi scribble dilakukan dengan memprediksi motion vector dari frame sekarang ke frame selanjutnya. Prediksi motion vector yang*

*dilakukan menggunakan exhaustive search diperbaiki dengan mendefinisikan matrik yang berukuran dinamis terhadap ukuran scribble, motion vector ditentukan dengan Sum of Absolute Difference (SAD) antara frame sekarang dan frame berikutnya. Hasilnya ketika diaplikasikan pada ruang warna RGB dapat menurunkan rata-rata kesalahan piksel setiap frame dari 3.058,55 menjadi 1.533,35, sedangkan dalam ruang waktu HSV menjadi 1.662,83.*

*KiMoHar yang merupakan framework yang diusulkan meliputi tiga hal sebagai berikut. Pertama adalah image matting dengan adaptive threshold FCM dapat meningkatkan akurasi sebesar 11.05 %. Kedua, perbaikan kualitas matting pada image terkompresi menggunakan DCT-2D meningkatkan akurasi sebesar 31.41%. Sedangkan yang ketiga, estimasi temporal constraint pada ruang warna RGB meningkatkan akurasi 56.30%, dan dalam ruang HSV 52.61%.*

**Kata kunci :** *Video Object Extraction, Alpha Matting, Temporal Constraint Estimation, Motion Estimation*

## ***Semi-Automatic Video Object Extraction Using Alpha Matting with Motion Estimation-Based***

Name : RuriSukoBasuki  
NRP : 2210301008  
Supervisor : Prof. Ir. MauridhiHeryPurnomo, M.Eng., Ph.D.  
Co-Supervisor : MochamadHariadi, S.T., M.Sc., Ph.D.

### **ABSTRACT**

*It is important to have object extraction in video editing application because compositing process is necessary for independent object. Extraction process is performed by image matting which is defining manual scribble to represent the foreground and background area, and alpha estimation to determine the unknown area.*

*In image matting, there are problem which are pixel in unknown area is not firmly being the part of foreground or background, whereas, in temporal domain, it is not possible to define the scribble independently in whole frame. In order to overcome the problem, object extraction model with adaptive threshold estimation phase for alpha matting, accuracy improvement for image matting, and temporal constraint estimation for scribble propagation is proposed. Fuzzy C-Means (FCM) Algorithm and Otsu are applied for adaptive threshold estimation.*

*By FCM, the evaluation result by using Means Squared Error (MSE) shows that the average error of pixels in each frame is reduced from 30.325,10 to 26.999,33, while in the use of Otsu, the result shows 28.921,70. The matting quality is reducing since the intensity changing in compressed image improved by Discrete Cosine Transform (DCT-2D). The algorithm reduces Root Means Squared Error (RMSE) value from 16.68 to 11.4. Temporal constraint estimation for scribble propagation is performed by predicting motion vector from recent frame and forward. Motion vector prediction performed using exhaustive search is improved by defining the matrix in dynamic size to scribble; motion vector is determined by Sum of Absolute Difference (SAD)*

*between recent frame and forward. In its application to RGB space, it results the average error of pixels in each frame from 3.058,55 to 1.533,35, and 1.662,83 in HSV time space.*

*KiMoHar, the proposed framework, includes three things which are: First, image matting by adaptive threshold FCM increases the accuracy to 11.05%. Second, matting quality improvement in compressed image by DCT-2D increases the accuracy to 31,41%. Three, temporal constraint estimation in RGB space increases the accuracy to 56.30%, and 52.61% in HSV space.*

**Keywords:** *Video Object Extraction, Alpha Matting, Temporal Constraint Estimation, Motion Estimation*

## KATA PENGANTAR

Puji syukur kepada Allah SWT, atas rahmat, nikmat dan hidayah-Nya serta salawat dan salam senantiasa tercurahkan kepada junjungan kita Rasulullah SAW, sehingga kami dapat menyelesaikan disertasi ini. Disertasi ini disusun untuk memenuhi sebagian persyaratan dalam mencapai derajat Doktor Program Pascasarjana Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya.

Sepenuhnya kami menyadari bahwa selama penulisan disertasi ini banyak pihak yang telah memberikan bantuannya, sehingga pada kesempatan ini kami ingin menyampaikan terima kasih yang tulus kepada:

1. Bapak Prof. Ir. Mauridhi Hery Purnomo, M.Eng, Ph.D, dan Bapak Mochamad Hariadi, S.T, M.Sc., Ph.D. selaku Promotor dan Co-Promotor yang telah memberikan bimbingan dan dorongan semangat dengan tulus dan penuh kesabaran.
2. Bapak Prof. Ir. Djauhar Manfaat, M.Sc, Ph.D, selaku Direktur Program Pascasarjana Institut Teknologi Sepuluh Nopember Surabaya.
3. Bapak Ir. Djoko Purwanto, M.Eng., Ph.D. selaku Koordinator Pengelola Program Studi Pascasarjana Jurusan Teknik Elektro ITS Surabaya.
4. Beasiswa Program Pasca Sarjana (BPPS) DIKTI periode 2010 – 2013 dan Hibah Bersaing DP2M DIKTI (2014-2015) serta pihak Universitas Dian Nuswantoro yang telah membantu pembiayaan dalam kuliah program Doktor pada Program Studi Teknik Multimedia dan Jaringan Jurusan Elektro ITS.
5. Bapak Dr. Ir. Edi Noersasongko, M.Kom selaku Rektor Universitas Dian Nuswantoro yang telah memberikan ijin, motivasi serta dukungan untuk melanjutkan studi di Teknik Elektro ITS

6. Bapak Dr. Abdul Syukur, M.M, selaku Dekan Fak. Ilmu Komputer yang telah memberikan kesempatan dan motivasi untuk menimba ilmu lebih lanjut.
7. Segenap pengelola Program Pascasarjana (PPs) ITS, dosen dan karyawan PPs Jurusan Teknik Elektro ITS yang telah memberikan dukungan dan bantuannya selama kami menempuh program pendidikan ini.
8. Bapak serta ibuku tercinta (Bapak Sutarno, Bapak Hartono, Ibu Sri Darini, Ibu Lidia Indriyani) yang senantiasa memberikan doa, dukungan serta nasehat-nasehat yang selalu menenangkan hati.
9. Istriku tercinta Agustina "*Tina*" Fitri Anulia dan putriku tersayang Gayatri Ghazia "*Cia*" Gilda, yang telah memberikan dukungan, doa dan sabar mendampingi dalam menyelesaikan studi dan disertasi ini.
10. Adindaku tersayang (Om Joko, Tante Irfa, Tante Niken, Om Yus, Tante Anin, Om Yan, Tante Novi) atas bantuan dan doanya.
11. Sahabat-sahabat, khususnya buat Pak Akok yang telah menularkan ilmunya dan strateginya dalam melakukan penelitian.
12. Rekan-rekan mahasiswa S3, Pak Pulung, Pak Puji, Pak Arief, Pak Rusmono, Pak Max, Pak Fanani, Pak Arifin, Pak Budi Harjo, Pak Huda serta rekan-rekan lainnya atas bantuan diskusi, dukungan dan do'anya.
13. Pihak-pihak lain yang tidak dapat kami sebutkan satu-persatu dan telah memberikan bantuan, inspirasi serta do'anya selama ini.

Kami menyadari bahwa karya ini masih banyak kekurangan dan keterbatasan, sehingga karya ini masih jauh dari kesempurnaan. Untuk itu, saran dan kritik yang membangun akan penulis terima dengan tangan terbuka.

Surabaya, Desember 2015  
Ruri Suko Basuki

# DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>ii</b>
<b>PERNYATAAN KEASLIAN DISERTASI .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b><i>ABSTRACT</i> .....</b>	<b>vii</b>
<b>KATA PENGANTAR .....</b>	<b>ix</b>
<b>DAFTAR ISI .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xix</b>
<b>DAFTAR SIMBOL .....</b>	<b>xx</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	7
1.3. Tujuan dan Manfaat Penelitian .....	8
1.4. Posisi dan Kontribusi Penelitian .....	9
1.4.1. Posisi Penelitian .....	9
1.4.2. Kontribusi Penelitian .....	9
1.5. Susunan Penulisan Disertasi .....	10
<b>BAB 2 KAJIAN PUSTAKA TENTANG EKSTRAKSI OBJEK VIDEO .....</b>	<b>17</b>
2.1. Video .....	17
2.2. <i>Matting</i> .....	19
2.2.1. Domain Spasial .....	20
2.2.2. Domain Temporal .....	32

2.3. Komponen <i>Matting</i> .....	39
2.4. <i>Analysis Spectral</i> .....	40
2.4.1. <i>Matting Laplacian</i> .....	42
2.4.2. <i>Linear Transformation</i> .....	43
2.4.3. <i>Grouping Component</i> .....	44
<b>BAB 3 METODOLOGI PENELITIAN .....</b>	<b>45</b>
3.1. Terminologi.....	45
3.2. Blok Diagram Penelitian.....	47
<b>BAB 4 ESTIMASI THRESHOLD UNTUK ALPHA MATTING .....</b>	<b>51</b>
4.1. <i>Alpha Matting</i> .....	51
4.2. <i>Threshold</i> dalam <i>Alpha Matting</i> .....	52
4.3. Metodologi <i>Alpha Matting</i> .....	53
4.4. Ekstraksi <i>Matte</i> .....	54
4.4.1. Algoritma Otsu .....	56
4.4.2. Algoritma FCM .....	59
4.5. Evaluasi Ekstraksi Objek dengan <i>Alpha Matting</i> .....	60
4.6. Kesimpulan .....	61
<b>BAB 5 DCT-2D UNTUK PERBAIKAN KUALITAS KEYFRAME .....</b>	<b>65</b>
5.1. Blok Diagram Perbaikan Kualitas <i>Matte</i> .....	65
5.2. Pengujian <i>Image Matting</i> pada Image Terkompresi .....	67
5.3. Kesimpulan .....	70
<b>BAB 6 PENELUSURAN OBJEK PADA VIDEO SEQUENCES .....</b>	<b>73</b>
6.1. Konstruksi Keyframe .....	74
6.2. Mekanisme Tracking.....	74
6.3. Pengujian <i>Background Subtraction</i> untuk Penelusuran Objek .....	75
6.4. Kesimpulan .....	78

<b>BAB 7</b>	<b><i>MOTION VECTOR PREDICTION</i> UNTUK ESTIMASI</b>	
	<b><i>TEMPORAL CONSTRAINT</i></b> .....	<b>79</b>
7.1.	Blok Diagram Estimasi <i>Temporal Constraint</i> .....	79
7.2.	Pengujian <i>Temporal Constraint</i> .....	82
7.2.1.	<i>Manual Constraint</i> .....	83
7.2.2.	Estimasi <i>Constraint</i> dengan <i>Exhaustive Search</i> .....	83
7.3.	Permasalahan dalam <i>Exhaustive Search</i> .....	88
7.4.	Perbaikan Algoritma <i>Block Matching</i> .....	88
7.5.	Ruang Warna HSV .....	89
7.6.	Evaluasi Estimasi <i>Temporal Constraint</i> .....	90
7.7.	Kesimpulan .....	92
<b>BAB 8</b>	<b>KESIMPULAN &amp; RENCANA SELANJUTNYA</b> .....	<b>97</b>
8.1.	Kesimpulan .....	97
8.2.	Rencana selanjutnya .....	99
<b>DAFTAR PUSTAKA</b>	.....	<b>101</b>

# DAFTAR GAMBAR

Gambar 1.1. Hasil Segmentasi pada <i>Natural Image</i> .....	4
Gambar 1.2. Sistem Segmentasi Video .....	7
Gambar 1.3. Posisi Penelitian .....	11
Gambar 1.5. Keterkaitan Penelitian Dengan Sistematika Penulisan .....	14
Gambar 2.1. Objek Video Dalam Komunikasi Visual.....	18
Gambar 2.2. <i>Bayesian Video Matting</i> .....	35
Gambar 2.3. <i>Keyframe</i> Berbasis Sistem <i>Rotoscoping</i> .....	36
Gambar 2.4. Derajat Matrik .....	40
Gambar 3.1. <i>Input Image</i> dan <i>Matte</i> yang Terekstraksi.....	46
Gambar 3.2. Blok Diagram Penelitian .....	47
Gambar 3.3. <i>Input Image</i> dan <i>Trimap</i> .....	48
Gambar 3.4. <i>Trimap</i> Berbasis <i>Scribble</i> .....	48
Gambar 3.5. Proses Ekstraksi <i>Matted</i> dengan <i>Trimap Scribble-based Interface</i> ..	49
Gambar 3.6. Konstrai Statis Terhadap Pergerakan <i>Scene</i> .....	49
Gambar 4.1. Metodologi <i>Alpha Matting</i> .....	53
Gambar 4.2. Ekstraksi <i>Matte Simple Color</i> .....	57
Gambar 4.3. Perbedaan Visual Ekstraksi <i>Matte</i> .....	62
Gambar 4.4. Grafik Hasil Perbandingan Ekstraksi <i>Matte</i> .....	63
Gambar 5.1. Blok Diagram Perbaikan Kualitas <i>Matte</i> .....	67
Gambar 5.2. Ekstraksi <i>Matte</i> dalam <i>Raw Imagedan Image</i> Terkompresi.....	69
Gambar 5.3. Perubahan Intensitas Warna .....	71
Gambar 5.4. Grafik Perbandingan Hasil Ekstrak <i>Matte Image</i> Terkompresi ....	72
Gambar 6.1. Perbedaan <i>Scene</i> pada <i>Video Sequences</i> .....	75

Gambar 6.2. Perbedaan Intensitas Dua Frame .....	76
Gambar 6.3. <i>Otsu Thresholding</i> .....	76
Gambar 6.4. Ekstraksi Objek dengan Konstrain <i>Background Subtraction</i> .....	77
Gambar 6.5. Grafik MSE Evaluasi Ekstraksi Objek dengan <i>Background Subtraction</i> .....	79
Gambar 7.1. Blok Diagram Ekstraksi Objek Video Semi Otomatis .....	80
Gambar 7.2. Dataset <i>Video Matting</i> .....	81
Gambar 7.3. Pendefinisian <i>Manual Constraint</i> .....	83
Gambar 7.4. <i>Macro Block</i> pada Frame Selanjutnya .....	84
Gambar 7.5. Proses Pencocokan Blok .....	84
Gambar 7.6. Titik Pusat Perhitungan Jarak .....	85
Gambar 7.7. Evaluasi Dataset <i>Artem</i> .....	90
Gambar 7.8. Evaluasi Dataset <i>Alex</i> .....	91
Gambar 7.9. Evaluasi Dataset <i>Dmitriy</i> .....	91
Gambar 7.10. Evaluasi Dataset <i>Vitaliy</i> .....	92
Gambar 7.11. Hasil Propagasi Konstrain.....	94
Gambar 7.12. Perbandingan Visual Hasil Ekstraksi Matte Dengan ES, DRGB, DHSV .....	95
Gambar 7.13. Perbandingan Visual Hasil Ekstraksi Objek Dengan ES, DRGB, DHSV .....	95

# DAFTAR TABEL

Tabel 1.1. Penelitian Terkait dan Kontribusi Penelitian .....	12
Tabel 4.1. Perbandingan Jumlah Perbedaan Pixel dalam <i>Image Matting</i> antara <i>Closed-Form Solution</i> , <i>Otsu</i> , dan <i>FCM</i> .....	63
Tabel 5.1. Perbandingan Nilai RMSE Ekstraksi <i>Matte</i> .....	70
Tabel 7.1. Hasil <i>Motion Vector</i> pada <i>Frame Sequences</i> .....	87
Tabel 7.2. Rata-Rata Perbedaan Pixel Dataset .....	92

# DAFTAR SIMBOL

$\alpha$	<i>Alpha</i>
$\lambda$	<i>Lambda</i>
$\sigma$	<i>Sigma</i>
$\pi$	<i>Phi</i>
$\varepsilon$	<i>Epsilon</i>
$\mu$	<i>Mu</i>
$\delta$	<i>Delta</i>
$\theta$	<i>Theta</i>

# DAFTAR LAMPIRAN

Lampiran A	.....	107
Lampiran B	.....	108
Lampiran C	.....	109

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Munculnya standar televisi digital seperti *Digital Television* (DTV) di Amerika, *Digital Video Broadcasting-Terrestrial* (DVB-T) di Eropa dan *Integrated Services Digital Broadcasting-Terrestrial* (ISDB-T) di Jepang membuat permintaan teknologi multimedia meningkat pesat. Transformasi data multimedia melalui media penyimpanan seperti CD atau DVD, serta video streaming yang disebarkan melalui internet menjadi sangat populer. Hal ini mendorong munculnya standard video coding untuk peningkatan efisiensi dan kualitas layanan. Sebelumnya, standar video coding yang telah didefinisikan dalam teknologi H.264/MPEG-4 AVC [1] yang disebarkan dalam *High Definition – Television* (HD-TV) dan ditransmisikan melalui sinyal satelit, kabel dan sistem transmisi terrestrial. Selain itu, H.264/MPEG-4 AVC juga menyediakan layanan aplikasi untuk sistem editing dan akuisisi konten, aplikasi keamanan, internet dan mobile network video, dll.

Namun popularitas *Ultra High Definition* (UHD) serta munculnya video dengan resolusi yang lebih tinggi dari *Full High Definition* (FHD) mendorong permintaan standar video coding yang memiliki kinerja lebih efisien dibandingkan dengan H.264/MPEG-4 AVC. Peningkatan penggunaan aplikasi berbasis mobile, PC tablet serta kebutuhan transmisi untuk layanan *video on demand* merupakan

tantangan tersendiri dalam bidang jaringan komputer. Di samping itu permintaan layanan display resolusi tinggi juga meningkat secara signifikan, hal ini membuat standar video coding H.264/MPEG-4 AVC sudah tidak dapat memenuhi kebutuhan user. Untuk meningkatkan kualitas dan efisiensi H.264/MPEG-4 AVC, saat ini didefinisikan High Efficiency Video Coding (HEVC) / H.265 [2] yang memberikan layanan representasi berbasis objek dan kompresi video yang pengembangannya mengikuti variasi baru dari aplikasi berbasis konten yang fokus pada peningkatan resolusi dan arsitektur parallel processing.

Dalam pemrosesan video berbasis objek, peningkatan fungsi diperlukan agar mendapatkan keuntungan lebih sehingga menambah manfaat dalam aplikasi visi komputer seperti *gesture understanding*, *imagerecognition*, *augmented reality*, *video segmentation*, dll. Permasalahan yang muncul dalam segmentasi objek video adalah *ill-posed problem*, yaitu informasi semantik tidak secara eksplisit disediakan dalam data video, sehingga tidak mungkin segmentasi objek video di seluruh *frame* dilakukan secara otomatis. Secara umum pekerjaan terkait dengan segmentasi objek video diklasifikasikan menjadi 2 tipe, yaitu *automatic segmentation* dan *semi-automatic segmentation*. Pada metode *automatic segmentation*[3], [4], [5], pemisahan objek dilakukan berbasis fitur khusus seperti *gesture*, *color*, *texture* dan *movement*[6]. Hal ini sangat sulit diaplikasikan untuk memisahkan objek sesuai dengan persepsi *user*. Hingga saat ini tidak ada jaminan bahwa segmentasi objek secara otomatis dapat memberikan kepuasan pada *user*[7]. Sedangkan pada *semi-automatic segmentation*[8], [7], [9], ekstraksi objek dilakukan dengan melibatkan *user* untuk menyediakan *constraint* secara manual pada *image* yang

mengindikasikan area *foreground* objek dan *background* yang sesuai dengan video context.

Segmentasi objek pada *image* dan video memiliki peran penting dalam dalam aplikasi editing. Hal ini membuat penelitian terkait dengan segmentasi objek *image/ video* banyak dilakukan secara *ekstensif* lebih dari dua puluh tahun terakhir ini. Tujuan utama dari proses segmentasi adalah untuk memisahkan objek *foreground* dari *background* dengan kualitas tinggi sehingga memiliki kemampuan mendekati mata manusia dalam memberikan semantik pada daerah yang diobservasi. Karena tingkat akurasi dari proses segmentasi akan menentukan hasil yang berkualitas maka hal ini akan memiliki dampak yang positif dalam *post-processing*. Aplikasi dari hasil segmentasi objek dapat digunakan untuk mengkombinasikan objek *foreground* dengan *background* yang berbeda sehingga dapat digunakan untuk efisiensi produksi dalam industri film, berita maupun periklanan, sehingga dapat mengurangi waktu dan biaya pengambilan adegan. Di samping itu dalam industri pertelevisian hasil segmentasi objek dapat diaplikasikan untuk pembuatan berita, dalam hal ini objek dapat dikombinasi dengan beberapa *background* yang berbeda sehingga dapat mengurangi jumlah properti studio.

Untuk melakukan proses pemisahan objek *foreground* dari *background* pada *image*, operasi ini dilakukan dengan melibatkan sebagian atau seluruh piksel dalam sebuah *image*. Sebagai dasar untuk proses ekstraksi, Porter dan Duff pada tahun 1984 [10] memperkenalkan *alpha channel* yang digunakan untuk mengontrol interpolasi linear pada warna *foreground* dan *background*. Penelitian yang terkait segmentasi objek dilakukan dengan ekstraksi “*matte*” yang disebut dengan “*pulling matte*” atau “*digital matting*”. Pada umumnya pendekatan yang telah dilakukan

diklasifikasikan berdasarkan pendekatan *color-sampling* dan pendekatan *defining-affinity*. Dalam pendekatan berbasis *color-sampling* nilai piksel yang berdekatan dengan *foreground* dan *background* dikumpulkan dan digunakan sebagai sampel warna untuk estimasi nilai *alpha*. Pendekatan *color-sampling* ini dapat bekerja dengan baik ketika karakteristik input *image* terdiri dari daerah yang smooth dan trimap didefinisikan dengan baik oleh *user*. Keterbatasan dari pendekatan ini adalah terjadinya kesalahan klasifikasi sampel warna pada scene yang kompleks. Oleh karena itu, untuk meningkatkan kemampuan pendekatan ini, diperkenalkan pendekatan berbasis *defining-affinity* yang dilakukan dengan menggunakan model statistik pada *local image*. Perhitungan nilai *alpha* dilakukan dengan estimasi *gradient matte* yang secara intrinsik tidak dihitung secara langsung, namun dimodelkan di seluruh kisi-kisi *image* dengan menentukan kemiripan antara berbagai piksel yang bertetangga. Dibandingkan dengan pendekatan sebelumnya, pendekatan berbasis *defining-affinity* lebih kuat (*robust*), karena afinitas ditentukan pada *local windows*, sehingga asumsi tersebut dapat dijadikan basis untuk *image* yang kompleks.



Gambar 1.1. Hasil segmentasi pada *natural image*[11].

Peningkatan kualitas *matte* pada segmentasi objek dalam spatial domain dilakukan dengan menghitung nilai *threshold* pada channel *alpha* diestimasi dengan algoritma FCM (*Fuzzy C-Means*). Penyediaan *userconstraint* dilakukan dengan memberikan *scribble* warna putih untuk obyek *foreground* dan hitam untuk *background*. Teknik ini telah berhasil diimplementasikan dalam *natural image* pada warna yang kompleks (seperti yang ditunjukkan dalam Gambar 1.1). Dalam konteks segmentasi objek pada aplikasi video, proses dilakukan dengan mempartisi *framevideo sequences* ke dalam bentuk objek dan *background* yang memiliki semantik, hal ini dapat dilakukan dengan model *intra-frame (spatial)* maupun *inter-frame (temporal)*. Ekstraksi obyek video pada komputer vision seperti *human pose estimation*, *event recognition*, dan *video annotation* dianggap sebagai teknik *pre-processing* tingkat tinggi sehingga hasilnya akan membantu mesin dalam menterjemahkan konten data video.

Permasalahan dalam *alpha matting*, piksel dalam *unknown region* menjadi bagian dari *foreground* dan *background*, sehingga untuk ekstraksi objek diperlukan estimasi *threshold* dalam *alpha channel*. Dalam ekstraksi objek video, *scribble* dalam *alpha matting* diperlukan untuk inisialisasi ekstraksi objek pada frame selanjutnya, namun *scribble* tidak mungkin dilakukan secara independen di seluruh frame, sehingga diperlukan *scribble* yang dipropagasi secara otomatis agar dapat mengikuti pergerakan objek.

Dalam penelitian sebelumnya, *alpha channel*[11], [12] diekstrak dengan mendefinisikan nilai  $\epsilon$  (*epsilon*) dalam range  $0.1^7$  hingga  $0.1^5$ , sehingga user harus mendefinisikan nilai tersebut sesuai dengan karakteristik image. Untuk estimasi *alpha channel* pada *alpha matting* dalam penelitian ini digunakan estimasi adaptive

threshold. Sedangkan *motion estimation*[13] yang menggunakan algoritma *exhaustive search* diperbaiki dengan estimasi *temporal constraint*.

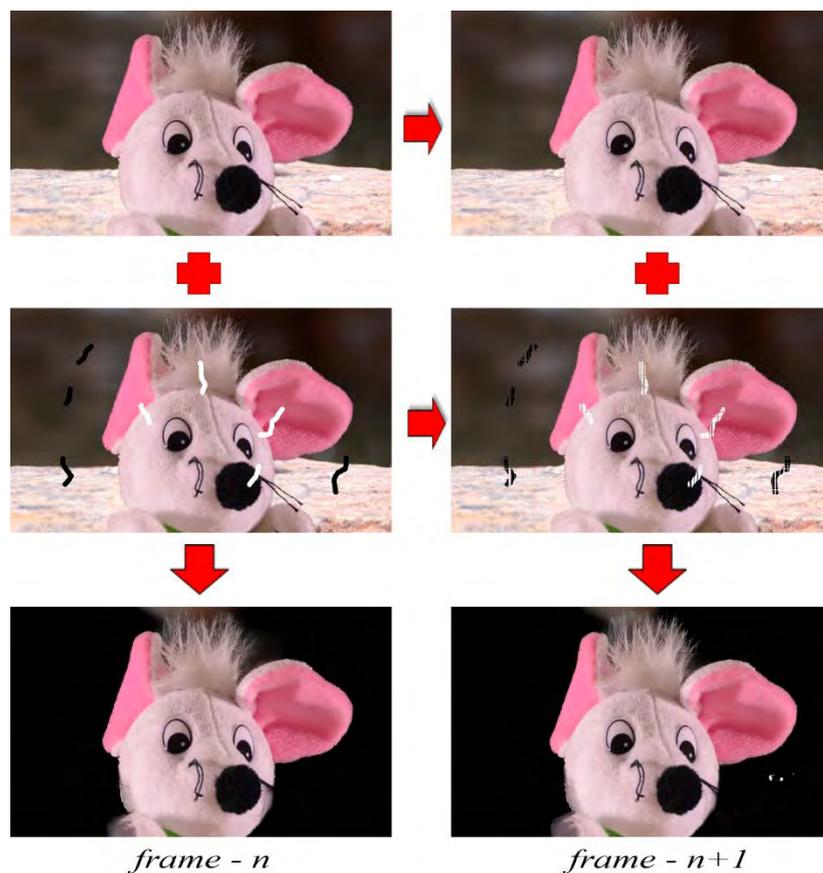
Untuk segmentasi objek video sebagai tujuan dalam penelitian ini, diusulkan teknik segmentasi objek semi otomatis. Proses segmentasi diawali dengan memberikan *scribble (supervised)* untuk daerah yang akan diestimasi, (daerah objek dan daerah *background*). Agar hasil segmentasi mendapatkan kualitas yang optimal, proses pemisahan objek dilakukan dengan algoritma *matting*[11]. Pendefinisian *constraint* secara manual oleh *user* pada input *image* tidak mungkin dilakukan secara independen di seluruh *frame*, oleh karena itu untuk memberikan *constraint* pada frame selanjutnya dilakukan dengan mendefinisikan *constraint* baru secara otomatis pada frame selanjutnya.

Dengan asumsi bahwa urutan frame memiliki karakteristik *temporal coherence*, maka pergerakan objek dalam *video sequence* tidak bergerak secara cepat atau tiba-tiba, namun bergerak secara halus (*coherence*) dari frame sekarang ke frame selanjutnya. Piksel dari suatu objek akan menempati koordinat tertentu dan akan bergerak pada koordinat yang terdekat terlebih dahulu sebelum bergerak ke koordinat yang jauh, sehingga untuk ekstraksi *matte* pada frame selanjutnya secara *temporal* dapat dilakukan menggunakan *imagematting* dengan mempropagasi *constraint* dari frame sekarang.

Propagasi *constraint* (seperti yang diilustrasikan dalam Gambar. 1.2) dapat dilakukan dengan estimasi gerakan dengan memprediksi *motion vector*. Karena *constraint* dapat dipropagasi, maka *constraint* baru yang dihasilkan secara otomatis dapat digunakan sebagai *input image* untuk *image matting* pada frame selanjutnya. Estimasi *constraint* secara *temporal* merupakan kontribusi dalam penelitian ini.

## 1.2. Rumusan Masalah

Kualitas ekstraksi matte dalam *alpha matting* dapat dicapai apabila pada daerah batas antara *foreground* dan *background* (*unknown region*) dapat diestimasi secara akurat. Permasalahan dalam alpha matting adalah setiap piksel dalam *unknown region* menjadi bagian dari *foreground* dan *background* sehingga pada batas tepi objek terdiri dari campuran piksel *foreground* dan *background*. Estimasi *adaptive threshold* pada *alpha matting* dilakukan dalam *unknown region* digunakan untuk memisahkan piksel sehingga menjadi bagian utuh dari *foreground* atau *background*.



Gambar 1.2. Sistem segmentasi video [14].

Ekstraksi objek video yang berbasis *alpha matting* dilakukan diseluruh frame dalam *scene* yang sama. Dengan data video yang memiliki volume yang besar tidak memungkinkan *scribble* didefinisikan secara independen oleh user di seluruh frame. Sehingga secara temporal konstrain perlu diestimasi untuk menggerakkan *scribble* dari frame sekarang ke frame selanjutnya. Pergerakan *scribble* yang dilakukan berbasis *motion estimation* menggunakan *exhaustive search* tidak akurat untuk propagasi *scribble*. Hal ini terjadi karena terdapat perbedaan nilai *motion vector* pada beberapa blok yang berkorelasi dengan piksel *scribble*.

### **1.3. Tujuan Penelitian dan Manfaat Penelitian**

Tujuan penelitian ini adalah meningkatkan akurasi *alpha matting* dan *motion estimation* untuk ekstraksi objek video semi otomatis. *Matte* akurat dalam *alpha matting* diperoleh dengan memisahkan campuran piksel dalam *unknown region*. Pemisahan dilakukan dengan estimasi *adaptive threshold* pada *unknown region*. Ekstraksi *matte* pada frame selanjutnya diinisialisasi oleh *scribble* yang dipropagasi dari frame sebelumnya. Peningkatan akurasi pada pergerakan *scribble* dari frame sebelumnya ke frame selanjutnya diperoleh dari *motion estimation* yang diestimasi dengan *temporal constraint* yang kami usulkan.

Adapun beberapa manfaat dari penelitian ini, diharapkan hasil penelitian dapat digunakan pada bidang *computer vision* seperti *imagerecognition*, *video surveillance*, *video annotation* maupun *motion understanding* yang dapat digunakan berbagai aplikasi seperti sport, produksi film, iklan, berita dalam industri perfilman, periklanan maupun produksi berita televisi.

## 1.4. Posisi dan Kontribusi Penelitian

Penelitian yang telah dilakukan akan mempunyai kontribusi terhadap masyarakat akademik secara langsung. Selain itu secara tidak langsung akan memberikan kontribusi terhadap penerapan ilmu pengetahuan bidang visi komputer dalam kehidupan sehari-hari. Untuk memperjelas kontribusi yang ada dalam penelitian ini, diuraikan posisi penelitian dalam bentuk diagram penelitian serta matrik kontribusi penelitian.

### 1.4.1. Posisi Penelitian

Tabel 1.1 menunjukkan detail dari posisi penelitian yang dilakukan. Dari penelitian-penelitian yang telah dilakukan sebelumnya, tahapan utama dalam *video matting* adalah ekstraksi *matte* dan *trimap propagation*. Dalam penelitian ini, *adaptive threshold* digunakan untuk menentukan nilai alpha dalam ekstraksi *matte*, sedangkan estimasi konstrain secara temporal dilakukan untuk menggerakkan *scribble* yang digunakan sebagai konstrain dalam domain temporal.

### 1.4.2. Kontribusi Penelitian

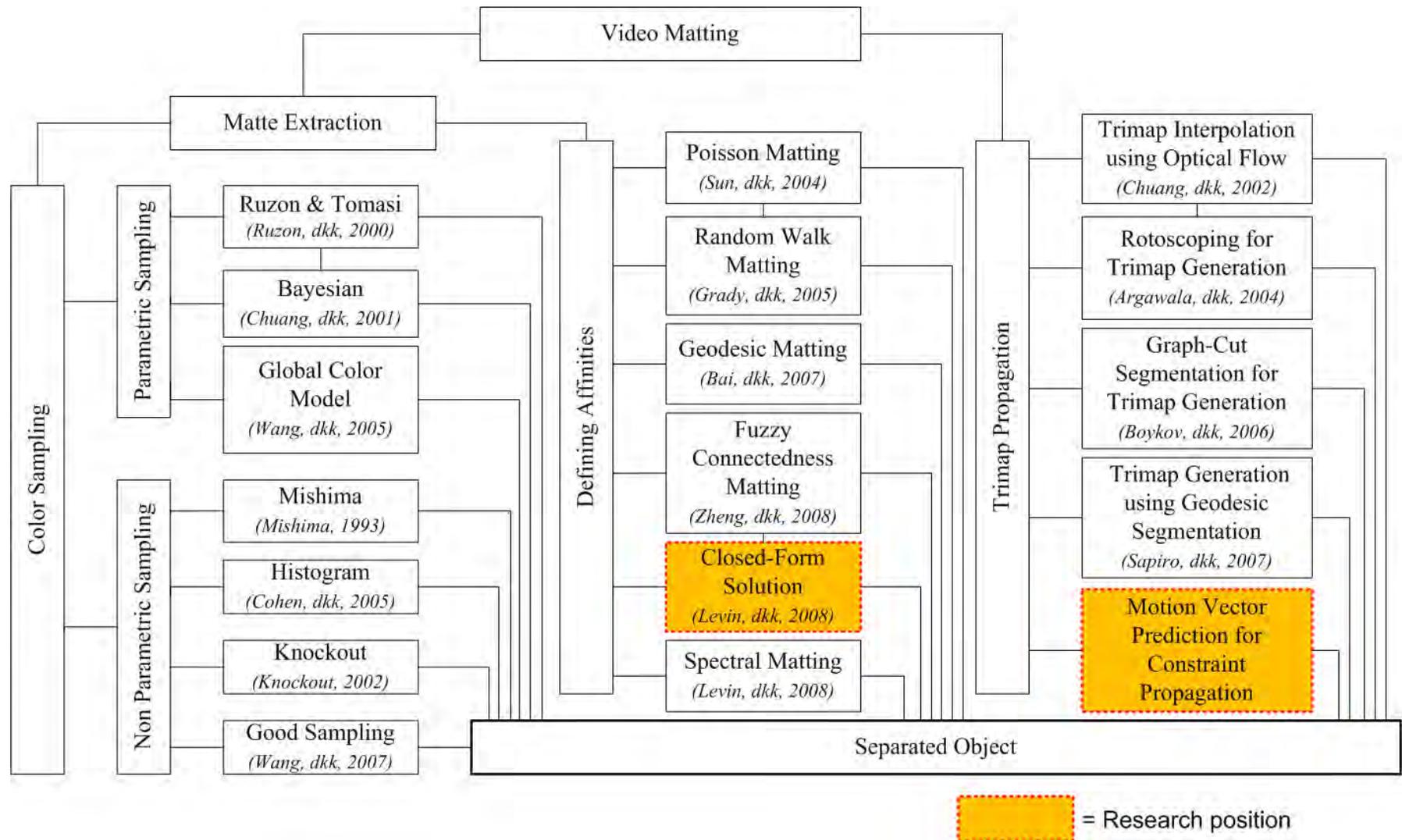
Tabel 1.1 disajikan pemetaan penelitian dengan yang dilakukan dengan peneliti-peneliti sebelumnya. Penulis memberikan kontribusi terhadap penelitian yang ada secara keilmuan seperti Gambar 1.3. Metode *closed-form solution* ditingkatkan akurasi dengan estimasi *adaptive threshold* pada *image matting*. Selain itu estimasi konstrain temporal untuk propagasi konstrain dilakukan dengan memprediksi *motion vector*. Kontribusi dalam tataran aplikasi meliputi bidang *computer vision* yang terkait dengan *sinematografi*, deteksi objek,

*imagerecognition* atau *image analysis* lainnya yang terkait dengan pemisahan objek.

### **1.5. Susunan Penulisan Disertasi**

Penelitian ini terdiri dari empat tahapan, yaitu: kajian pustaka, *matting* objek dalam *spasial domain*, dan propagasi trimap pada *temporal domain*. Setiap tahapan terdiri dari beberapa tugas rinci yang harus dilakukan. Detail tahapan penelitian, tugas dan hubungannya dengan setiap bab disajikan dalam Gambar 1.5. Seperti yang diilustrasikan dalam Gambar 1.5, penelitian ini dilakukan dengan mengkaji beberapa literatur sebagai titik awal.

Pengkajian beberapa literatur didefinisikan sebagai proses identifikasi, menelaah dan menterjemahkan semua bukti penelitian yang tersedia dengan tujuan untuk memberikan jawaban untuk permasalahan spesifik penelitian. Berdasarkan kajian yang dilakukan, *outline* penelitian ini dikonstruksi dari latar belakang yang berisi permasalahan *imagematting* pada *spatial-domain* dan estimasi *constraint* pada *temporal-domain*. Bagian tinjauan pustaka secara sistematis disajikan dalam Bab 2.

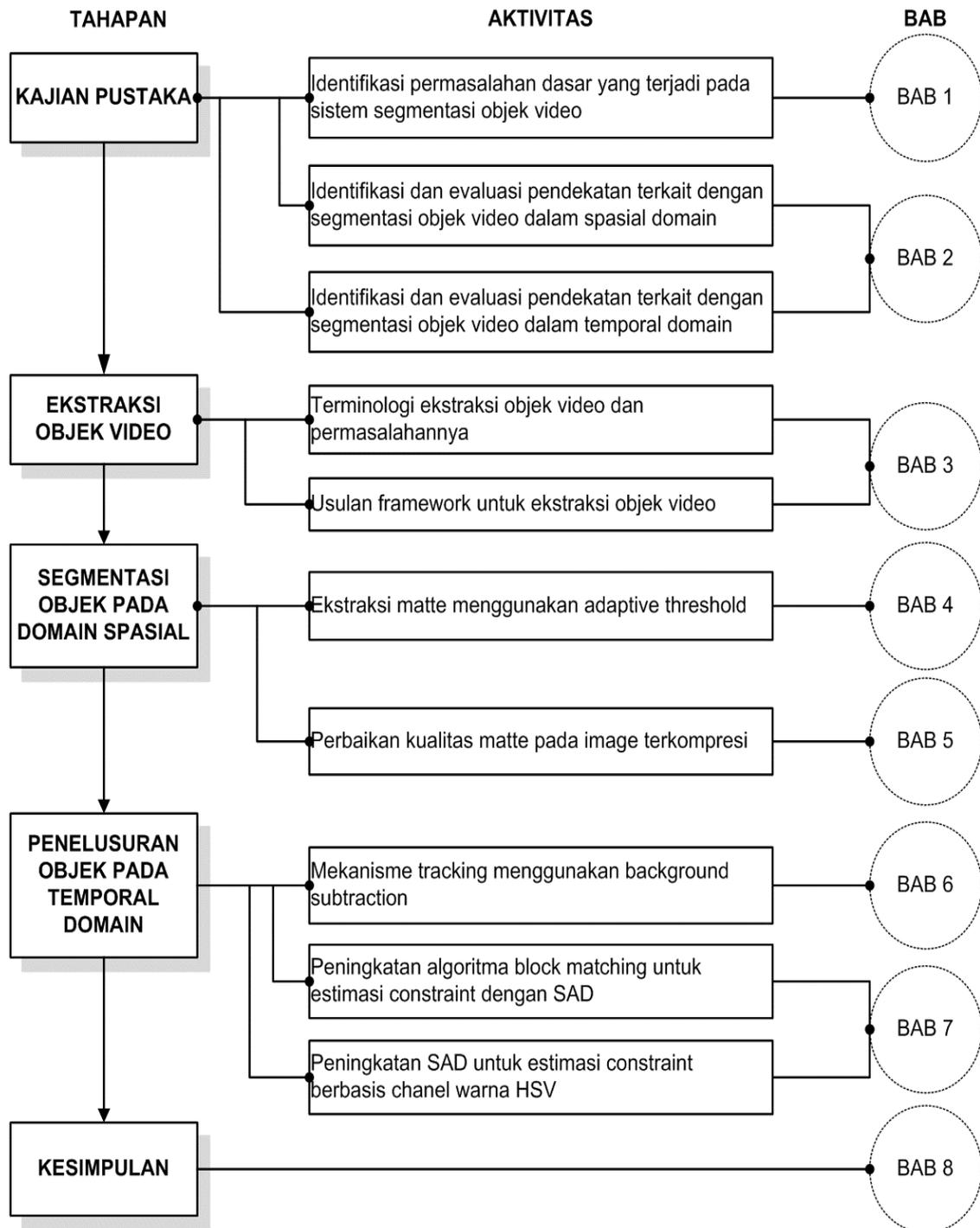


Gambar 1.3. Posisi penelitian

Tabel 1.1: Penelitian terkait dan kontribusi penelitian

Topik Riset <i>Matting</i>	Metode	Kontribusi Peneliti Lain	Kontribusi Peneliti
Image matting	Blue screen matting untuk estimasi nilai alpha. (Ruzon, dkk, 2000)	<i>Mixture of Gaussian</i> (MOG) untuk memodelkan warna <i>foreground</i> dan <i>background</i> serta <i>sliding continues window</i> untuk mendefinisikan daerah tetangga. (Chuang, dkk, 2001)	Menghitung <i>threshold</i> untuk <i>alphamatting</i> secara <i>adaptive</i> menggunakan <i>Fuzzy C-Means</i> . Seminar Internasional CITEE (Basuki, Hariadi, dan Pramunendar, 2012).
		Estimasi <i>gradient matte</i> dengan <i>poisson matting</i> . (Sun, dkk, 2004)	Perbaikan kualitas matting pada <i>keyframe</i> menggunakan DCT-2D. Jurnal nasional KURSOR (Basuki, Hariadi, dan Purnomo, 2013)
		Optimasi <i>iterative matting</i> menggunakan pendekatan berbasis <i>scribble</i> . (Wang, dkk, 2005)	
		Estimasi probabilitas <i>random-walker</i> pada piksel <i>unknown</i> . (Grady, dkk, 2005)	
		Meminimisasi energi iteratif untuk <i>image matting</i> dengan memodelkan MRF ( <i>Markov Random Field</i> ) pada <i>known region</i> serta menentukan bobot secara otomatis selama iterasi. (Guan, dkk, 2006)	

		<i>Closed-form solution</i> untuk mengekstrak <i>matte</i> dari <i>natural image</i> . (Levin, dkk, 2008)	
Propagasi trimap	Estimasi pergerakan inter-frame setiap piksel dalam <i>video sequences</i> menggunakan <i>optical flow</i> . (Chuang, dkk, 2002).	Sistem <i>keyframe</i> berbasis <i>rotoscoping</i> . (Agarwala, dkk, 2004)	Penelusuran objek pada <i>video sequences</i> menggunakan <i>background subtraction</i> . Seminar Internasional IEVC (Basuki, Soeleman, Hariadi, Purnomo, Pramunendar, dan Yogananti, 2014)
		<i>Graph-cut</i> untuk segmentasi image interaktif. (Rother, dkk, 2004)	<i>Motion vector prediction</i> untuk estimasi <i>temporal constraint</i> . Jurnal Internasional IRECOs (Basuki, Hariadi, Yuniarno, dan Purnomo, 2015)
		<i>Object-cut-and-paste system</i> untuk <i>trimap generation</i> . (Li, dkk, 2005)	
		<i>Interactive Cutout System</i> untuk <i>trimap generation</i> . (Wang, dkk, 2005)	
		<i>Geodesic distance</i> untuk menghitung jarak pada daerah <i>spatio-temporal</i> untuk mempropagasi <i>scribble</i> . (Bai, dkk, 2008)	



Gambar 1.5. Keterkaitan blok diagram penelitian dengan sistematika penulisan

Tahapan berikutnya setelah kajian pustaka adalah pengumpulan data dan usulan framework penelitian. Pada bagian ini dataset yang digunakan dalam proses segmentasi video diidentifikasi. Dataset yang digunakan diperoleh dari ucf sport

action dan *video matting* dataset, yang terdiri dari data video yang telah ekstrak dalam bentuk *image* yang didefinisikan ke dalam ruang warna RGB dan *groundtruth* sebagai *image* referensi dalam proses pengujian. Setelah data dipersiapkan, usulan penelitian dilakukan berdasarkan permasalahan yang diperoleh dari kajian pustakan. Usulan penelitian disajikan dalam Bab 3.

Nilai biner '1' atau warna putih yang dihasilkan berada di luar batas tepi objek, sehingga ketika didefinisikan sebagai input *image* mengalami *misclassification* (objek yang terekstraksi seperti mengalami dilasi) dalam proses ekstraksi objek. Perbaikan terhadap algoritma penelusuran objek dilakukan dengan estimasi jarak vektor, karena diasumsikan jarak vektor dapat memprediksi pergerakan objek. Nilai jarak vektor ini dapat dijadikan parameter untuk propagasi konstrain dari frame sekarang ke frame selanjutnya.

Estimasi jarak pergerakan vektor (*motion vector*) awalnya dilakukan menggunakan *exhaustive search* pada algoritma *block matching* (BMA), namun karena ada beberapa kesalahan propagasi, pendekatan ini diperbaiki dengan mendefinisikan blok yang berukuran dinamis terhadap ukuran *scribble* dan hanya berkorelasi dengan daerah *scribble*. *Motion vector* dua blok dari dua frame yang berurutan (frame sekarang dan frame berikutnya) diestimasi menggunakan *sum of absolute difference* (SAD) yang nilainya terkecil. Algoritma ini dilakukan pada ruang warna RGB dan HSV. Ruang warna HSV diaplikasikan karena memiliki komputasi yang sederhana sehingga waktu proses yang dibutuhkan lebih efisien serta telah diaplikasikan secara luas dalam *image processing*[15]. Pembahasan

secara detail dari framework ini diuraikan dalam Bab 7. Tahapan terakhir dari semua langkah penelitian disimpulkan dan diringkas dalam Bab 8.

## BAB 2

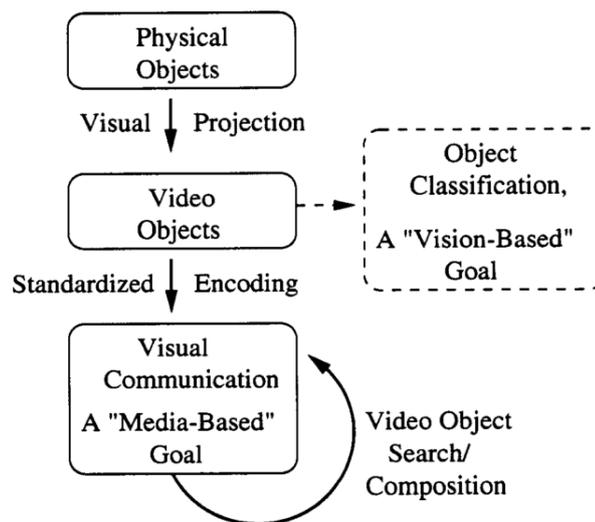
# KAJIAN PUSTAKA TENTANG EKSTRAKSI OBJEK VIDEO

### 2.1. Video

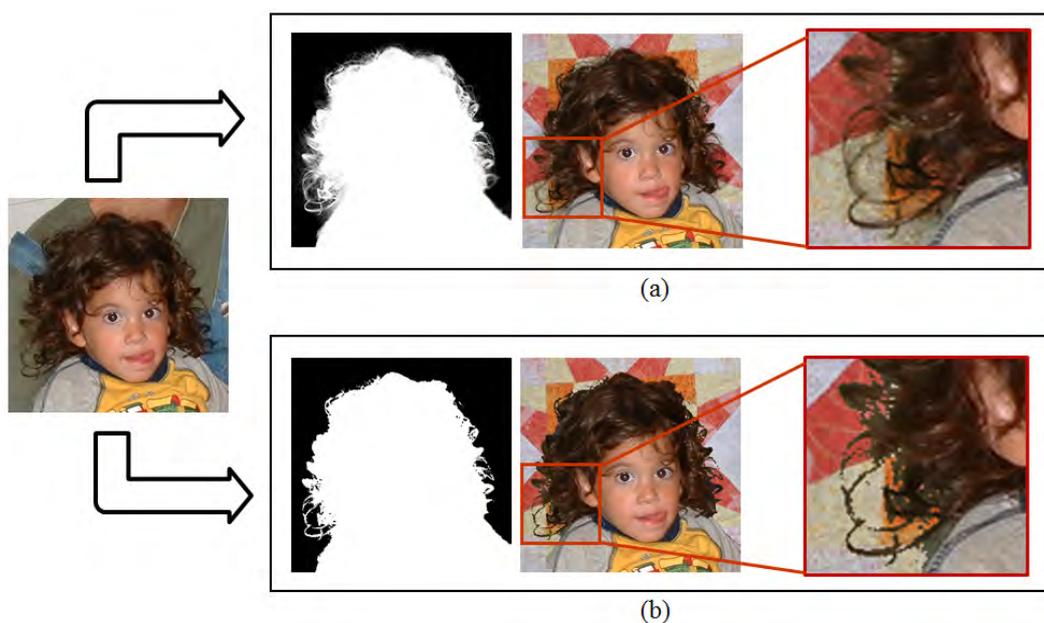
Video [16] adalah sejenis film yang diproduksi menggunakan rekaman *image* elektronik dan diproduksi menggunakan teknik elektronik. Data video terdiri dari *framesequences* yang memiliki koherensi secara *temporal*. Frame adalah image tunggal yang ditunjukkan sebagai bagian dari video atau film yang memiliki ukuran besar. Beberapa image tunggal dijalankan secara berurutan untuk menghasilkan video atau film yang halus. Setiap frame dalam video memiliki informasi objek yang merepresentasikan scene dalam video. Objek video adalah daerah *spatio-temporal* 3D yang berisi proyeksi dari objek fisik dalam *video sequences*. Sedangkan objek fisik adalah objek yang biasa dimanipulasi dalam ruang 3D [17].

Untuk melakukan ekstraksi objek dari frame video dalam video matting diperlukan trimap sebagai starting point. Trimap adalah segmentasi kasar berupa gambar kasar yang menunjukkan tiga area: foreground digambarkan dalam warna putih, background dalam warna hitam dan unknown dalam warna abu-abu [11]. Segmentasi objek video [18] dilakukan dalam dua tahapan, yaitu ekstraksi objek pada domain spasial dan tracking objek pada domain *temporal* (*spatio-temporal*). Ekstraksi objek dapat dilakukan dengan teknik *hard segmentation* dan *matting*.

Pada *hard segmentation* setiap piksel menjadi bagian *foreground* atau *background* secara tegas (*strict*), sedangkan pada teknik *matting* piksel pada *unknown region* setiap piksel dapat menjadi bagian dari *foreground* dan *background*[19]. Kualitas *matting* dibandingkan *hard segmentation* secara visual dideskripsikan dalam Gambar 2.2.



Gambar 2.1. Objek video dalam komunikasi visual [17].



Gambar 2.2. Ekstraksi dan *compositing*. (a) *image matting*, (b) *hard segmentation*[11].

## 2.2. *Matting*

*Matting* berasal dari kata “*matte*” yang biasanya digunakan dalam dunia fotografi dan *sinematografi*. Pada awal 1900, proses penggabungan film dilakukan menggunakan printer *optic* dan *art of optical compositing*[20]. Untuk menciptakan kombinasi dalam proses *optical compositing*, tiga bagian film diperlukan untuk menciptakan *composite*[21], yaitu:

- *Foreground plate*: film yang berisi elemen *foreground* yang akan digabungkan dengan *background* yang berbeda.
- *Matte*: film monokrom yang menentukan tingkat gradasi atau (*opacity*) pada *foreground plate*. Area di luar *foreground plate* didefinisikan dalam warna hitam merupakan *background* yang tidak diinginkan oleh *foreground* sehingga elemen *foreground* terlihat jelas, daerah inilah yang disebut dengan “*matte*”.
- *Background plate*: *background* diambil dari film, model miniatur, atau *matte painting* (lukisan fotorealistik dari scene yang real maupun imajiner).

Proses menarik “*matte*” dari keseluruhan *image* disebut dengan “*pulling matte*” atau “*matting*”.

Proses pemisahan objek *foreground* dari *image* atau *video sequences* memiliki peranan yang sangat penting dalam aplikasi image maupun *video editing*, sehingga penelitian terkait dengan hal tersebut telah dilakukan lebih dari dua puluh tahun terakhir ini [19]. Permasalahan utama dalam proses pemisahan objek adalah *ill-posed problem*[6], yaitu objek dalam citra maupun *frame* video

tidak memiliki informasi semantik, sehingga untuk memisahkannya memerlukan intervensi manusia. Awalnya permasalahan ini dimodelkan secara matematika oleh Porter dan Duff [10] pada tahun 1984 dengan memperkenalkan channel *alpha* yang dapat digunakan untuk mengontrol interpolasi linear pada objek *foreground* dan *background* seperti dalam persamaan (2.1).

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i, \text{ dimana } 0 \leq \alpha \leq 1 \quad (2.1)$$

di mana  $I_i$  adalah *image compositing*,  $F_i$  adalah *definite foreground*,  $B_i$  adalah *definite background* dan  $\alpha_i$  adalah *channel alpha*. Channel *alpha* diperoleh dari koordinat setiap piksel yang dimodelkan sebagai *convex combination* dari objek *foreground* dan *background* yang didefinisikan sebagai gradasi warna (*opacity color*) dengan nilai intensitas piksel antara 0 hingga 1.

Dalam pendekatan ini diasumsikan bahwa informasi semantik sebuah citra deklasifikasikan menjadi 3 (tiga) yaitu *foreground(F)* sebagai *definite foreground* yang bernilai 1, *background(B)* sebagai *definite background* yang bernilai 0, dan *alpha( $\alpha$ )* sebagai piksel campuran yang bernilai antara 0 dan 1. Karena *alpha* berupa gradasi, maka nilai *alpha* terdiri elemen dari ( $F$ ) dan elemen dari ( $B$ ), oleh karena itu penentuan nilai batas (*threshold*) diantara ( $F$ ) dan ( $B$ ) menjadi pekerjaan utama yang sangat mempengaruhi akurasi dalam proses pemisahan objek pada citra maupun *frame* video. Teknik ini disebut dengan “*pulling matte*” atau “*matting*”.

### 2.2.1. Domain Spasial

Dalam *spatial-domain* proses segmentasi yang berbasis *matting* dikelompokkan menjadi 2 (dua) metode, yaitu metode *color sampling* dan *affinities*[19]. Metode *color sampling* dilatarbelakangi asumsi bahwa diantara piksel image yang terdekat (*pixel neighborhood*) memiliki korelasi yang sangat kuat, sehingga bisa dimanfaatkan untuk mengurangi kesulitan dalam pemecahan *ill-posed problem*. Secara statistik kemiripan warna diantara piksel terdekat sering memiliki parameter *matting* yang sama. Korelasi lokal ini dapat digunakan untuk berbagai aplikasi seperti *imagedenoising*, *superresolution*, *colorization* dan *segmentation*.

Pendekatan ini diklasifikasikan menjadi metode *parametric sampling* dan *non parametric sampling*. Langkah awal dalam metode *parametric sampling* adalah mengumpulkan sampel *foreground* dan *background*, kemudian model statistik parametrik seperti *Gaussian* digunakan untuk mencocokkan sampel yang sama. Model ini selanjutnya digunakan untuk mengukur jarak antara *unknown pixel* dengan distribusi *foreground* dan *background* yang secara langsung dapat digunakan untuk estimasi nilai *alpha*.

#### 2.2.1.1. Metode *color sampling*

Algoritma ini awalnya diusulkan oleh Ruzon dan Tomazi pada tahun 2000 [22]. Dalam pendekatan ini nilai *alpha* diukur di sepanjang *manifold* yang menghubungkan batas di setiap distribusi warna objek. Secara implisit pendekatan ini mengasumsikan bahwa daerah *unknown* adalah kanal sempit yang berada di sekitar batas *foreground* sedangkan *skeleton* pada daerah *unknown* dapat

direpresentasikan dalam bentuk jaringan piksel. Model konstruksi dan prosedur estimasi  $\alpha$  dilakukan dengan:

- 1) Membagi jaringan piksel (*skeleton* pada daerah *unknown*) dalam interval dengan membagi *anchor points* yang diseleksi;
- 2) Pusat dari setiap *anchor points* mendefinisikan local spatial window yang meliputi daerah *unknown*, daerah local *foreground* dan *background*;
- 3) Piksel *foreground* dan *background* dalam *local window* digunakan untuk estimasi *foreground* dan *background* yang berorientasi pada distribusi *Gaussian*, atau masing-masing *point mass* dalam ruang warna CIE-Lab [23].
- 4) Membangun *manifold* dengan mengumpulkan *foreground Gaussian* dan *background Gaussian*, selain itu juga menolak beberapa koneksi untuk memastikan kriteria “interseksi” dan “sudut”.
- 5) Warna yang diamati dari piksel *unknown* dimodelkan dari distribusi menengah antara distribusi *foreground* dan *background*. Distribusi menengah didefinisikan sebagai jumlah dari *Gaussian* yang mana setiap *Gaussian* telah diinterpolasi rata-rata secara linear antara pasangan *foreground* dan *background* menurut estimasi nilai  $\alpha$ . Nilai optimal  $\alpha$  adalah salah satu yang menghasilkan distribusi menengah dengan warna yang diamati memiliki probabilitas maksimum.

- 6) Pada *unknown* piksel, setelah *alpha* diestimasi, warna *foreground* dikalkulasi dengan interpolasi pasangan *Gaussian* pada *foreground* dan *background*.

Dari alur di atas, daerah *unknown* diasumsikan sebagai kanal sempit yang dapat diciptakan dari dilasi jaringan piksel. Pemilihan *anchor point* dan pembuatan *local windows* agar tidak overlap sifatnya hanya sementara. Warna *local window* dimodelkan dengan non-oriented *Gaussian* dengan matrik kovarian diagonal ini menghasilkan kesalahan yang besar pada saat *fitting* daerah bertekstur.

Berbasis algoritma yang diusulkan oleh Ruzon dan Tomasi [22], [24] mengusulkan pendekatan *Bayesian matting*. Algoritma ini memiliki kemiripan dengan algoritma yang diusulkan Ruzon dan Tomasi, yaitu algoritma ini juga menggunakan model warna *mixture Gaussian* pada *foreground* dan *background*, namun dengan sejumlah perbaikan. *Bayesian matting* menggunakan *sliding window* untuk mendefinisikan kemiripan piksel yang bergerak dari daerah *foreground* ke daerah *background*.

Sebagai tambahan dari algoritma ini adalah penggunaan sampel *foreground* dan *background* untuk membangun distribusi warna dengan menghitung *foreground*, *background* dan *alpha* sehingga setiap piksel yang bertetangga akan memberikan kontribusi pada *Gaussian foreground* dan *background*. Permasalahan *matting* diformulasikan dalam *framework Bayesian* dan *matte* diselesaikan menggunakan teknik *Maximum a Posteriori* (MAP). Secara matematis estimasi pada *unknown pixel*  $F_i, \alpha_i$ , dan  $B_i$  diperoleh dari observasi  $I_i$  dengan memaksimalkan distribusi probabilitas  $P$  dan menggunakan aturan Bayes untuk mengekspresikan hasil sebagai maksimalisasi atas jumlah *log likelihoods*:

$$\begin{aligned}
(F^*, B^*, \alpha^*) &= \arg \max_{F_i, B_i, \alpha_i} P(F_i, B_i, \alpha_i | I_i) \\
&= \arg \max_{F_i, B_i, \alpha_i} P(I_i | F_i, B_i, \alpha_i) P(F_i) P(B_i) P(\alpha_i) / P(I_i) \\
&= \arg \max_{F_i, B_i, \alpha_i} L(I_i | F_i, B_i, \alpha_i) + L(F_i) + L(B_i) + L(\alpha_i) \quad (2.2)
\end{aligned}$$

dimana  $L(\cdot)$  adalah *log likelihood*  $L(\cdot) = \log P(\cdot)$ , dan  $P(I_i)$  adalah nilai yang di drop karane konstan terhadap parameter optimasi. Permasalahan telah dapat dikurangi dengan mendefinisikan  $L(I_i | F_i, B_i, \alpha_i)$ ,  $L(F_i)$ ,  $L(B_i)$ , dan  $L(\alpha_i)$ . Model untuk mengukur perbedaan antara warna yang diamati dan warna yang diprediksi dengan estimasi  $L(F_i, B_i, \alpha_i) = -\|I_i - \alpha_i F_i - (1 - \alpha_i) B_i\|^2 / \sigma_i^2$ , dengan variabel  $\sigma$  diukur secara lokal.

Kesalahan model *log-likelihood* dalam pengukuran  $I_i$ , dan sesuai dengan distribusi probabilitas Gaussian berpusat pada  $\overline{I_i} = \alpha F_i + (1 - \alpha) B_i$  dengan standard deviasi  $\sigma_i$ . Secara sederhana kesalahan *fitting* mengikuti persamaan compositing (2.1). Untuk menghitung  $L(F_i)$ , warna *foreground* yang terdekat, pertama kali dipartisi kedalam beberapa kelompok, dan setiap kelompok berorientasi *Gaussian* diestimasi dengan menghitung rata-rata  $\overline{F}$  dan kovarian  $\Sigma_F$ .  $L(F_i)$  didefinisikan sebagai  $-(F_i - \overline{F})^T \Sigma_F^{-1} (F_i - \overline{F}) / 2$ .  $L(B_i)$  dihitng dengan cara yang sama menggunakan sampel *background*  $L(\alpha)$  yang diperlakukan sebagai konstanta. Persamaan (2.2) diselesaikan dengan menghitung  $F_i$ ,  $B$ , dan  $\alpha_i$  secara iteratif dengan tahapan:

1. Memperbaiki  $\alpha_i$  untuk menyelesaikan  $F_i$  dan  $B_i$  sebagai

$$\begin{bmatrix} \sum_F^{-1} + I\alpha_i^2 / \sigma_i^2 & I\alpha_i(1 - \alpha_i) / \sigma_i^2 \\ I\alpha_i(1 - \alpha_i) / \sigma_i^2 & \sum_B^{-1} + I\alpha_i^2 / \sigma_i^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \sum_F^{-1} \overline{F} + I\alpha_i^2 / \sigma_i^2 \\ \sum_B^{-1} \overline{B} + (1 - \alpha_i) / \sigma_i^2 \end{bmatrix} \quad (2.3)$$

Dengan  $I$  adalah matrik identik berukuran  $3 \times 3$ .

2. Memperbaiki  $F_i$  dan  $B_i$  untuk menyelesaikan  $\alpha_i$  sebagai

$$\alpha_i = \frac{(I_i - B_i)(F_i - B_i)}{\|F - B\|^2} \quad (2.4)$$

Penerapan algoritma ini dalam *foreground* dan *background* yang memiliki banyak *cluster*, proses ini dilakukan dengan memilih pasangan *clusterforeground* dan *background* yang memberikan kemungkinan maksimum. *Bayesianmatting* akan menghasilkan *matte* yang akurat dengan asumsi diberikan citra input dengan definisi trimap yang presisi. Namun jika asumsi ini tidak dipenuhi, misalnya citra input mengandung daerah yang sangat bertekstur dengan penggunaan *Gaussian* yang secara statistik tidak cukup dalam memodelkan distribusi warna atau pendefinisian trimap yang kasar, hal ini akan mengakibatkan korelasi antara piksel *unknown*, sampel *foreground* dan *background* menjadi lemah dan cenderung menghasilkan banyak *noise*.

Pendekatan Ruzon dan Tomasi [22] serta *Bayesianmatting*[24] menggunakan model warna lokal dengan asumsi bahwa daerah *unknown* merupakan kanal sempit di sepanjang daerah batas *foreground*, sehingga dalam pusat *local window* terdapat piksel *foreground* dan *background* yang dianggap sebagai piksel *unknown*. Kelemahan dari pendekatan ini adalah apabila *user* memberikan trimap kasar dengan menggunakan beberapa goresan maka sampel *foreground* dan *background* serta piksel *unknown* akan saling overlap. Untuk mengatasi permasalahan ini beberapa sistem dikembangkan untuk estimasi “*good matte*” dengan trimap yang kasar. Beberapa pendekatan dilakukan dengan menggunakan alternatif metode *global sampling*.

Pendekatan *iterative matting*[25] dilakukan pertama kali dengan *training GaussianMixture Models (GMMs)* pada warna *foreground* dan *background* secara global. Pada piksel *unknown*, sampel diambil dari semua *Gaussian* untuk menutup semua kemungkinan yang dapat memiliki warna *foreground*. Sedangkan dalam pendekatan *geodesicmatting*[26] penggunaan *Mixture ofGaussian (MoG)* untuk memodelkan distribusi warna global pada *foreground* dan *background* dalam ruang LUV, sedangkan metode estimasi fast kernel density [27] digunakan untuk mengurangi kompleksitas komputasi dalam mengkonstruksi PDFs.

Model statistik diperlukan untuk menyelesaikan daerah yang halus serta perbedaan distribusi warna pada *foreground* dan *background*. Teknik ini akan menghasilkan kesalahan *fitting* ketika distribusi warna secara significant non-*Gaussian*. Untuk menghindari permasalahan ini beberapa metode *non-parametric* digunakan untuk menangani sampel warna. Mishima [28] mengembangkan teknik *blue screen* yang berbasis pada representasi sampel *foreground* dan *background*. Karena piksel *background* hanya memiliki satu *cluster* warna, maka semua piksel *background* dapat ditutup dengan bulatan kecil yang diaproksimasi dengan *polyhedra* (jaring segitiga) dalam ruang warna, sedangkan semua bentuk piksel *foreground* selain *polyhedral* dikeluarkan dari *background*. Nilai *alpha* dari piksel *unknown* diestimasi dengan menghitung posisi relative terhadap dua *polyhedra*.

Sistem Knockout [29] pertama kali dilakukan dengan ekstrapolasi warna *foreground* dan *background* ke dalam daerah *unknown*. Piksel *unknown* dari citra dengan warna *foreground* dihitung sebagai jumlah tertimbang warna *foreground*

terdekat dan memiliki bobot yang proporsional dengan jarak spasial terhadap piksel *unknown*. Sedangkan warna *background* dihitung dengan cara inversi terhadap warna *background*, selanjutnya diperbaiki dengan mempertimbangkan posisi relative terhadap piksel *unknown* dan *foreground*. Untuk menghitung nilai *alpha* dilakukan tiga kali di masing-masing channel warna dengan melakukan estimasi jumlah tertimbang dari nilai di mana bobot sebanding dengan perbedaan *foreground* dan *background* dalam channel warna yang sesuai.

Beberapa pendekatan *matting* memperlakukan nilai *alpha* sebagai variabel random dan menggunakan sampel *foreground* dan *background* untuk menghitung PDFs. Dalam sistem *iterative matting*[25], PDFs diaproksimasi sebagai sebuah histogram dari tingkatan diskrit *alpha*. Nilai *alpha* berupa nilai continue dengan range antara 0 hingga 1. Setiap pasangan sampel dari *foreground* dan *background* memiliki kemungkinan nilainya digunakan untuk menghasilkan warna sintetis. Tingkat *alpha* yang menghasilkan warna yang paling dekat dengan piksel *unknown* dipilih dan kotak pada grafik histogram dinaikkan satu.

Konstruksi histogram selanjutnya digunakan untuk mendukung algoritma *Belief Propagation* untuk menentukan nilai optimal *alpha* yang secara lokal konsisten. Metode sampling yang mirip dengan ini juga digunakan dalam pendekatan *EasyMatting*[30]. Pada umumnya algoritma sampling menggunakan *local window* untuk mengumpulkan sampel warna, akibatnya ketika *foreground* dan *background* berisi pola yang kompleks atau trimap yang kasar maka kemungkinan besar sampel warna menjadi beragam sehingga berakibat hanya sedikit sampel yang valid untuk estimasi nilai *alpha* pada piksel *unknown*. Untuk

menutupi kekurangan ini prosedur *optimized color sampling* pada sistem Robust *Matting*[31] diusulkan. Pendekatan ini menyajikan prosedur pemilihan sampel untuk menguji ulang sampel yang dikumpulkan dan hanya memilih sampel yang baik yang digunakan sehingga menghasilkan hasil yang akurat dan robust.

#### 2.2.1.2. Metode Affinitas

Kelemahan dari metode berbasis sampling adalah keterbatasan dalam menangani klasifikasi sampel warna ketika citra memiliki warna yang kompleks. Metode berbasis statistik [sun, rother dkk] diusulkan untuk mengatasi keterbatasan dalam metode color sampling. Hal ini dilakukan dengan mendefinisikan beragam kedekatan (*affinities*) diantara piksel yang berdekatan yang mana secara intrinsik model gradien *matte* akan melintasi struktur citra, bahkan secara langsung dapat melakukan estimasi nilai *alpha* di setiap piksel tunggal.

Dibandingkan dengan pendekatan yang berbasis sampling, pendekatan berbasis affinitas memiliki banyak keuntungan. Pertama, metode berbasis affinitas selalu didefinisikan dalam *small window* (umumnya berukuran 3 x 3 piksel). Dengan *small window* biasanya korelasi antar piksel cukup kuat, sehingga local smoothnes assumption dapat dipenuhi, termasuk ketika diaplikasikan dalam citra yang cukup kompleks. Sebaliknya, ketika input trimap kasar dalam pendekatan berbasis sampling terpaksa harus mengumpulkan sampel yang jauh dari target piksel, sehingga dimungkinkan sampel yang diperoleh tidak berguna sama sekali. Di lain sisi, aturan yang didefinisikan dalam affinitas menghasilkan *matte* dengan

“*locally smooth*”, sehingga secara fundamental terhindar dari “*matte discontinuities*”.

*PoissonMatting*[32] merumuskan permasalahan dalam *natural imagematting* (umumnya citra masukan cukup kompleks) yang mengatasi persamaan *Poisson* dengan bidang *gradientmatte*. Pendekatan ini memiliki beberapa keuntungan, yaitu: pertama, *matte* secara langsung dibangun dari bidang *continuousgradient* pada *matte* dengan menyelesaikan persamaan *Poisson* menggunakan batasan informasi dari trimap yang disediakan oleh *user*. Kedua, dengan memanipulasi bidang *gradientmatte* menggunakan sejumlah filter, *user* dapat meningkatkan *Poissonmatting* yang hasilnya secara local lebih memuaskan. Hasil yang memuaskan seringkali dapat dicapai dengan menggunakan pendekatan ini, namun seringkali memerlukan waktu proses yang tidak efisien.

Pendekatan klasik berbasis affinitas didefinisikan dalam beberapa segmentasi citral spectral [33]. Afinitas ini mengadopsi sistem *Random walkmatting*[34] dengan memodifikasi yang tidak diukur dari ruang warna RGB, tetapi menciptakan ruang warna menggunakan teknik *Local Preserving Projection* (LPP). Hasil dari ruang warna yang diciptakan bekerja lebih baik dibandingkan ruang RGB jika digunakan untuk membedakan batas *foreground* dari *background*. Untuk efisiensi dalam implementasi algoritma ini diusulkan menggunakan *Graphical Processing Unit* (GPU) yang dapat digunakan untuk menyelesaikan permasalahan *matting* pada citra yang berukuran 1024 x 1024 piksel dalam waktu 0.5 detik dengan trimap yang didefinisikan secara akurat. Implementasi ini terbukti lebih cepat dibandingkan dengan pendekatan yang ada.

*Random walkmatting*[34], perhitungan probabilitas *Random walker* hingga mencapai *foreground* dimulai dari piksel *unknown*. Pendekatan *Geodesicmatting*[26] mengukur bobot jarak *geodesic* dimana *Random walker* akan mencapai *foreground* berawal dari piksel *unknown*. Dalam pendekatan ini yang dihitung jarak *geodesic* adalah waktu linear dengan dengan kebutuhan memori minimal namun memiliki kecepatan komputasi serta hasil segmentasi menggunakan *matting* dengan *user-scribble* memperoleh hasil dengan kualitas tinggi ketika afinitas antar piksel diaplikasikan dengan benar.

Fungsi jarak tertimbang (*geodesic*) yang dapat menyelesaikan persamaan geometri *Hamilton-Jacobi* dengan waktu linear komputasi optimal menjadi keunggulan dari pendekatan ini. Hal ini sangat menguntungkan untuk *video matting* karena memiliki permasalahan waktu komputasi yang cukup serius. Sedangkan kelemahan dari pendekatan ini bobot yang ditetapkan terlalu sederhana sehingga tidak dapat bekerja dengan baik ketika distribusi warna *foreground* dan *background* memiliki overlap yang besar, karena PDFs tidak dapat diestimasi dengan baik.

Pendekatan *matting* yang memiliki kemiripan dengan *geodesicmatting*[26] adalah sistem *FuzzyMatte*[35]. Algoritma ini dilakukan dengan menghitung jarak *geodesic* dari piksel *unknown* ke daerah *foreground* dan *background* yang dihitung menggunakan *Fuzzy connectedness* (FC) [36] dari piksel *unknown* ke *foreground* atau *background*. FC merupakan sebuah konsep yang secara efektif dapat menangkap kedekatan dan kesamaan diantara elemen citra. Algoritma ini dapat dihitung secara efisien menggunakan algoritma pencarian terkuat *connected*

*path*. Nilai akhir *alpha* di setiap piksel selanjutnya dihitung dari FC nya. Dalam pendekatan sebelumnya, perhitungan ulang terhadap *matte* harus benar-benar dilakukan ketika terdapat input citra yang di-update.

*FuzzyMatte* secara efektif mengintegrasikan input baru dengan estimasi *matte* sebelumnya dengan menghitung ulang nilai FC untuk subset kecil pada piksel. Hal ini menyebabkan biaya komputasi untuk perbaikan masing-masing piksel berkurang secara signifikan. Hasilnya menunjukkan bahwa *FuzzyMatted* dapat memperbaiki *matte* yang dilakukan interatif secara online dan dapat menghasilkan *matte* dengan kualitas tinggi untuk citra yang kompleks.

Baik pendekatan *Poissonmatting* maupun *Geodesicmatting*, keduanya masih melibatkan teknik estimasi distribusi warna pada *foreground* dan *background*. Sampai dengan batas tertentu teknik ini kinerjanya akan menurun ketika estimasi tidak akurat. Untuk mengatasi hal ini, pendekatan *Closed-formmatting*[11] diusulkan untuk menghindari keterbatasan teknik estimasi distribusi warna pada *foreground* dan *background*.

Algoritma ini dilakukan dengan menurunkan *cost function* dari asumsi local smoothness pada warna *foreground* dan *background* dan menunjukkan bahwa dalam persamaan yang dihasilkan sangat memungkinkan untuk menghilangkan *foreground* dan *background* secara analitis sehingga menghasilkan fungsi kuadrat dalam *alpha* sehingga dengan mudah diselesaikan dengan persamaan sparse linear system. Formula *Closed-form* dapat digunakan untuk memprediksi sifat-sifat solusi yang diinginkan dengan menganalisa *eigenvector sparse matrix* yang berkorelasi erat dengan matrik yang digunakan

dalam algoritma segmentasi citra *spectral*. Dengan sedikit intervensi *user* pada citra input, algoritma ini dapat menghasilkan *matte* dengan kualitas tinggi. Analisis selanjutnya telah dilakukan pada *matting* Laplacian yang menghasilkan pendekatan *matting* otomatis yang disebut dengan Spectral *matting*[12].

Matting pada *natural image* secara otomatis dilakukan dengan menghitung basis set *Fuzzymatting* component dari eigenvector terkecil yang sesuai dengan yang didefinisikan oleh *Laplacian matrix*. Algoritma ini dapat dikembangkan dengan teknik segmentasi spectral yang tujuannya untuk ekstraksi *hard segmentation* maupun ekstraksi komponen soft *matting*. Komponen-komponen ini selanjutnya dapat digunakan sebagai bangunan blok untuk memudahkan dalam mengkonstruksi *matteforeground* yang secara semantic lebih memiliki arti, baik dalam mode *unsupervised* atau dengan sedikit intervensi *user*.

### 2.2.2. Domain Temporal

Penelitian yang dilakukan secara parallel dengan *imagematting* adalah *video matting*, yang mana juga menjadi perhatian tersendiri. Metode ini dilakukan dengan menarik *matte* dari *video sequences* dengan elemen *dynamic foreground* yang berhadapan dengan *natural background*. *Video matting* memiliki peranan penting dalam produksi konten dalam televisi komersial maupun produksi film seperti memasukkan unsur-unsur baru dalam scene, memasukkan actor ke dalam lingkungan untuk menciptakan artefak baru. Segmentasi objek *foreground* yang dinamis dari *video sequences* merupakan pekerjaan yang kemungkinan memiliki banyak keuntungan meskipun cukup menantang, diantaranya:

1. Ukuran data video yang besar sehingga algoritma harus dapat memroses piksel dalam jumlah besar dalam *video sequences*, sehingga algoritma yang digunakan harus memiliki waktu komputasi yang efisien sehingga interface dapat muncul responsive dan *user* dapat terlibat dalam proses ekstraksi.
2. Karakteristik *video sequences* adalah *temporal coherence*, hal ini telah terbukti bahwa human visual system (HVS) sangat sensitive terhadap inkonsistensi *temporal* yang disajikan dalam *video sequences* [37], namun penerapan algoritma *imagematting* yang diaplikasikan secara konstan di setiap *frame* sering mengakibatkan inkonsistensi *temporal*. Tantangan terbesar adalah bagaimana estimasi *matte* pada seluruh *frame* secara konsisten.
3. Umumnya camera video hanya mampu menangkap objek 30 kali dalam 1 detik, sehingga sampel rate terlalu jauh dari ideal dibanding kecepatan gerakan. Sehingga membangun hubungan korespondensi pada *frame* dan mempertahankan *temporal coherence* menjadi pekerjaan yang sulit.

Sejumlah teknik telah diusulkan dalam pendekatan *video matting* untuk mengatasi permasalahan dan memberikan kontribusi dalam *computer vision*. Untuk mengatasi permasalahan *temporal coherence* tidak dilakukan dengan menciptakan trimap secara independen, tetapi trimap diciptakan dengan pendekatan yang mengikuti fitur *temporal coherence* dengan optimasi *spatio-temporal*. Teknik ini mengijinkan trimap dapat menyebar dari sejumlah batasan

yang didefinisikan *user* pada *keyframe* diseluruh *frame*, sehingga hasilnya dapat mengurangi intervensi *user* secara dramatis.

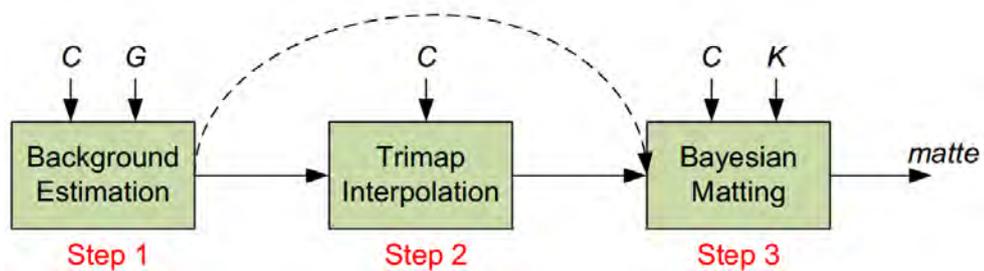
#### 2.2.2.1. Interpolasi *Trimap* dengan *Optical flow*

Teknik estimasi pergerakan *inter-frame* pada setiap piksel dalam *video sequences* dengan *optical flow* telah secara luas digunakan dalam sistem *video matting Bayesian*[38] untuk propagasi *trimap* (dideskripsikan dalam Gambar 2.3). Ide dasar dari interpolasi *trimap* adalah meminta *user* untuk mendefinisikan *trimap* pada beberapa *keyframe* dalam *input sequences*, kemudian *optical flow*[39] digunakan untuk propagasi *trimap* ke semua *frame*. Pada pergerakan objek yang besar estimasi *optical flow* sering terjadi kesalahan, hal ini menjadi kelemahan dari algoritma ini.

Untuk menjaga kestabilan propagasi *trimap*, diperlukan beberapa metode tambahan. Langkah pertama adalah mengharuskan *user* untuk menentukan beberapa “*garbagematte*” untuk mengeliminasi *foreground* pada beberapa *frame*. Hal ini memungkinkan layer *dynamicbackground* direkonstruksi dari komposisi mosaik *background* yang tersisa di setiap *frame*. Sehingga baik pada propagasi *trimap* maupun estimasi *matte* dapat memanfaatkan layer *background* yang dikonstruksi. Selanjutnya aliran *bi-directional* (*forward* dan *backward*) dikalkulasi untuk memandu *trimap* yang didefinisikan oleh *user* dari *keyframe* menuju *intermediateframe*.

Tahapan ketiga dalam algoritma ini adalah mengaplikasikan *Bayesianmatting* pada setiap *frame* dengan *trimap* yang dipropagasi. Estimasi *layer background* juga diperlukan untuk meningkatkan kualitas *matte* dan juga

untuk mempercepat proses komputasi *matting*. Implementasinya dalam sistem, *optical flow* digunakan dengan cara yang sangat konservatif, kinerjanya masih dibatasi pada akurasi dan estimasi *flow* sehingga sering mengalami kesalahan. Estimasi lapisan *background* mengasumsikan bahwa *background* mengalami transformasi *planar-perspective*, sehingga jika *background* mengandung objek yang bergerak, urutan *background* dimungkinkan menjadi tidak presisi.



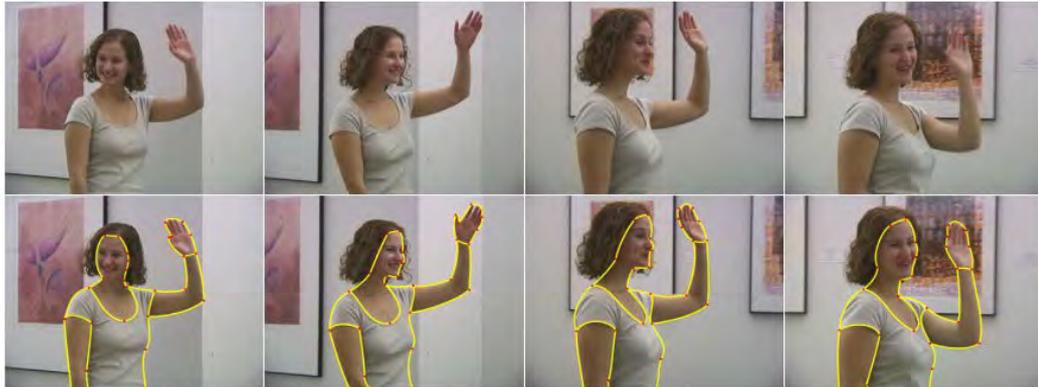
Gambar 2.3. *Bayesianvideo matting*[19].

#### 2.2.2.2. *Rotoscoping* untuk *Trimap Generation*

Dalam produksi film secara tradisional teknik *rotoscoping* mengacu pada proses penelusuran bentuk secara manual yang dilakukan pada saat menangkap urutan *image* satu per satu. Saat ini teknik optimasi telah diperkenalkan pada proses *rotoscoping* dan sistem *keyframe* berbasis *rotoscoping* telah diusulkan [40] yang secara dramatis telah mengurangi keterlibatan campur tangan manusia dalam menelusuri bentuk.

Sistem *rotoscoping* (Gambar 2.4), dideskripsikan sebagai tool segmentasi biner yang dapat digunakan untuk menghasilkan *trimap* untuk kelanjutan *matting* batas objek yang lebih akurat dengan menciptakan kanal sempit di sepanjang batas objek menggunakan kurva interpolasi dengan benar. Namun kelemahan dari pendekatan ini adalah hanya dapat menangani gerakan-gerakan yang sederhana,

karena keterkaitan antar batas yang valid tidak pernah terjadi ketika terjadi oklusi, sehingga diperlukan jaminan untuk menjaga objek berada dalam topologi yang sama tanpa ada oklusi.



Gambar 2.4. *Keyframe* berbasis sistem *rotoscoping*[40].

### 2.2.2.3. Segmentasi *Graph-cut* untuk *Trimap Generation*

*Graph-cut* telah didemonstrasikan tool optimisasi yang efisien pada segmentasi *image* interaktif [41], [33], [42]. Awalnya *Graph-cut* digunakan untuk menangani *image* tunggal, namun karena memiliki hasil yang menjanjikan, algoritma ini dikembangkan untuk *spatio-temporal* segmentasi objek video. Sistem ini telah berhasil diaplikasikan dan diklasifikasikan menjadi dua, yaitu: *Object-cut-and-Paste System*[36] dan *Interactive Cutout System*[43].

#### 2.2.2.3.1. *Object-cut-and-Paste System*

Terdapat tiga tahapan utama dalam sistem video *Object-cut-and-Paste System*[36]: *3D graph-cut segmentation*, *tracking-based local refinement* dan *generate finalmatte* dengan metode *coherentmatting*.

- a. *3D graph-cut segmentation*

Awalnya *user* menyediakan objek *foreground* yang sudah disegmentasi untuk panduan awal. Selanjutnya pasangan urutan *keyframe*, grafik 3D dikonstruksi daerah *atomic* (diperoleh dengan melakukan segmentasi awal menggunakan algoritma *watershed*[44]. Algoritma *graph-cut* dilakukan dalam grafik 3D untuk menyelesaikan permasalahan labeling. Optimisasi *graph-cut* akan menjaga konsistensi warna pada distribusi warna *foreground* atau *background* yang awalnya dikonstruksi oleh segmentasi *user* pada *keyframe*, di saat yang sama perbedaan warna antara daerah yang melewati batas objek dimaksimalkan. Untuk menggambarkan distribusi warna pada sistem digunakan *Gaussianmixture models* (GMMs).

b. *Tracking-based local refinement*

Distribusi warna pada *foreground/background* secara global dari *keyframe*, segmentasi *graph-cut* mengalami kesalahan pada daerah yang warna *foreground* mirip dengan warna *backgroundnya* pada bagian yang berbeda dari data video atau sebaliknya. Untuk mengatasi permasalahan tersebut diusulkan algoritma *tracking-based local refinement*. Algoritma *tracking* ini mirip dengan algoritma *rotoscoping*, namun didefinisikan dalam bentuk persegi panjang, bukan kubus. Energi didefinisikan untuk memastikan berbagai proses dalam tabung video berjalan lancar baik warna maupun lokasi window, dan juga dapat memaksa *intermediate window* untuk menangkap bagian *image* yang memiliki kemiripan terhadap window pertama dan window pada *frame* ke-*T*. Setelah dilakukan tracking, tingkat piksel

konstrain 2D pada algoritma graph-cut diaplikasikan di setiap window secara individual menggunakan model warna lokal pada *foreground* dan *background* yang dikonstruksi dari window dalam *keyframe*. Hasil segmentasi disempurnakan di setiap window sehingga dapat terhubung ke daerah batas yang berada di luar window.

c. *Coherent matting*

Untuk menciptakan hasil akhir dari *matte*, batas segmentasi biner di dilasi dengan 10 piksel untuk menciptakan kanal sempit sebagai daerah *unknown* untuk *matting* dengan menggunakan algoritma *coherent matting*[45] yang diperbaiki dengan *Bayesianmatting*[24] dengan memperkenalkan istilah regularisasi untuk *alpha*. Dengan teknik ini dapat menghasilkan *mattealpha* yang sesuai dengan batas segmentasi biner pada *frame* sebelumnya dan dapat melakukan *matting* yang lebih baik dibandingkan dengan *Bayesianmatting* ketika *foreground* dan *background* memiliki kemiripan warna.

2.2.2.3.2. *Interactive Cutout System*

Teknik segmentasi berbasis min-cut yang sebelumnya hanya diaplikasikan pada *image* diperluas ke dalam domain video [43]. *User* diberikan fasilitas untuk memberikan coretan pada *userinterface* objek yang melintasi domain ruang dan waktu. Untuk meminimalkan jumlah node pada operasi *min-cut* dilakukan preprocessing dengan algoritma *mean-shift*. Dalam *min-cut* juga didefinisikan *local cost function* baru untuk menambah *global cost* yang telah didefinisikan dalam pendekatan sebelumnya. Desain metode *matting* 2D diperluas pada *image*

diperluas agar dapat bekerja dalam domain 3D. Hasilnya pendekatan ini mampu menjaga tingkat akurasi dalam domain ruang dan waktu.

#### 2.2.2.4. Segmentasi *Geodesic*

Pendekatan segmentasi *geodesic* dan *matting*[26] juga diaplikasikan pada *video sequences*. Dalam pendekatan ini video dimodelkan sebagai kubus 3D dari piksel yang mana setiap piksel memiliki enam tetangga, empat secara spasial dan dua secara *temporal*. Pertama kali *user* menentukan *scribble* pada *foreground* dan *background* dalam beberapa *keyframe*, dan *scribble* dipropagasi ke seluruh video dengan menghitung bobot jarak dalam ruang *spatio-temporal*. Permasalahan dalam pendekatan sederhana ini adalah adanya waktu backward propagation yang tidak diinginkan.

### 2.3. Komponen *Matting*

Ekstraksi obyek baik pada *image* maupun video menjadi perhatian menarik untuk diteliti. Porter and Duff [10], [11], [12] memperkenalkan channel *alpha* yang digunakan sebagai alat untuk mengontrol linear interpolation dari warna *foreground* dan *background*. Selanjutnya channel *alpha* didefinisikan sebagai algoritma *matting* dengan mengasumsikan bahwa setiap piksel  $I_i$  pada input *image* merupakan kombinasi linear dari warna *foreground*  $F_i$ , dan warna *background*  $B_i$ , sedangkan  $\alpha_i$  adalah tingkat keburaman pada piksel *foreground*. Persamaan compositing (2.1) selanjutnya diasumsikan bahwa setiap piksel merupakan convex combination layer  $K$  image  $F^1, \dots, F^k$  yang dinotasikan seperti dalam persamaan 2.2

$$I_i = \sum_{k=1}^K \alpha_i^k F_i^k \quad (2.5)$$

Vektor  $K$  dari  $\alpha^k$  merupakan komponen *matting* dari *image* yang menentukan kontribusi fractional dari setiap layer pada warna yang diamati di setiap piksel.

#### 2.4. Spectral Analysis

*Graph* merupakan sekumpulan benda-benda yang disebut *vertex* atau *node* yang terhubung oleh *edge*. Umumnya *graph* dideskripsikan sekumpulan titik-titik yang melambankan *vertex* dan dihubungkan oleh garis-garis yang merepresentasikan *edge*. *Edge* dapat menghubungkan *node-node* yang sama yang disebut dengan *loop*[46].

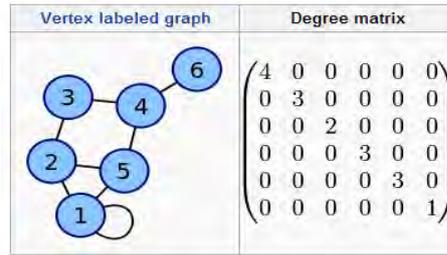
Metode segmentasi spektral umumnya berasosiasi dengan *image* dengan afinitas matrik  $A$  berukuran  $N \times N$ , yang diasumsikan sebagai  $A_{(i,j)} = e^{-d_{ij}/\sigma^2}$  dengan  $d_{ij}$  merupakan ukuran jarak diantara piksel (seperti warna dan jarak *geodesic*), yang selanjutnya didefinisikan sebagai

$$L = D - A \quad (2.6)$$

Dengan  $D$  merupakan derajat matrik dari *graph* (diilustrasikan dalam Gambar 2.5)

$$G = (V, E) \text{ dengan } \|V\| = n \quad (2.7)$$

dimana  $V$  adalah *vertex* atau *node*, dan  $E$  adalah *edge* atau sisi.



Gambar 2.5. Derajat matrik

Merupakan matrik diagonal  $D_{(i,j)} = \sum_j A(i,j)$  yang berisi informasi derajat setiap vertex (node) dengan  $D$  untuk  $G$  sebagai matrik bujur sangkar berukuran  $n \times n$  didefinisikan

$$d_{i,j} = \begin{cases} \deg(v_i) & \text{if } i = j \\ 0 & \text{Otherwise} \end{cases} \quad (2.8)$$

Sehingga  $L$  merupakan matrik semidefinite positif simetris yang eigenvector nya menangkap banyak struktur *image*. Matrik afinitas  $A$  dapat menangkap informasi bahwa sebuah *image* terdiri dari beberapa *cluster* yang berbeda atau connected components. Subset  $C$  pada piksel *image* merupakan connected component dari *image*  $A_{(i,j)} = 0$  untuk setiap  $(i,j)$  sehingga  $i \in C$  dan  $j \notin C$ , sehingga tidak ada subset  $C$  yang dapat memenuhi properti ini. Jika vektor indikator komponen  $C$  dinotasikan sebagai  $m^C$  maka

$$m_i^C = \begin{cases} 1 & i \in C \\ 0 & i \notin C \end{cases} \quad (2.9)$$

$m^C$  selanjutnya merepresentasikan 0-eigenvector (eigenvector dengan eigenvalue 0) dari  $L$ . Dengan asumsi bahwa *image* terdiri dari connected components  $K, C_1, \dots, C_K$  sehingga  $\{1, \dots, N\} = \cup_{k=1}^K C_k$  dengan  $C_k$  disjoint subset pada piksel. Vektor indikator  $m^{C_1}, \dots, m^{C_K}$  semua independen dan orthogonal 0-eigenvector dari  $L$ . Vektor indikator yang dihasilkan dari perhitungan eigenvector pada  $L$

hanya sampai rotasi, karena rotasi matrik  $R$  dengan ukuran  $K \times K$ , dan vektor  $[m^{c_1}, \dots, m^{c_K}]R$  merupakan basis nullspace pada  $L$ . Ekstraksi komponen-komponen berbeda dari eigenvector terkecil disebut dengan ‘‘Spectral Rounding’’ dan menjadi perhatian dalam beberapa penelitian [47], [48], [49], [50], [51]. Pendekatan sederhana untuk *clustering* piksel *image* menggunakan algoritma *K-Means*[47] dan analisis perturbation untuk membatasi kesalahan algoritma sebagai fungsi konektivitas dalam dan antar *cluster*.

#### 2.4.1. *Matting Laplacian*

*Matting Laplacian* [11]digunakan untuk mengevaluasi kualitas *matte* tanpa mengestimasi warna *foreground* dan *background* seperti dalam persamaan 2.1, namun menggunakan *local window* yang membentuk dua garis yang berbeda dalam domain RGB. Selanjutnya nilai  $\alpha$  dalam  $w$  dinyatakan sebagai kombinasi linear dari chanel warna.

$$\alpha_i = \alpha^R I_i^R + \alpha^G I_i^G + \alpha^B I_i^B + b \quad (2.8)$$

Permasalahan ekstraksi *matte* menjadi salah satu temuan dalam *alphamatte* yang meminimalkan deviasi dari model linear (persamaan 2.1) di seluruh *image windows*  $w_k$  :

$$J(\alpha, a, b) = \sum_{q \in i} \sum_{i \in w_k} (\alpha_i - \alpha^R I_i^R + \alpha^G I_i^G + \alpha^B I_i^B + b_k)^2 + \varepsilon \|a_k\|^2 \quad (2.9)$$

$\|a_q\|^2$  adalah persyaratan regularisasi dari  $\alpha$ . Koefisien model linear  $a, b$  memungkinkan untuk dieliminasi dari persamaan (2.9), dan menghasilkan *quadratic cost* dalam  $\alpha$

$$J\alpha = \alpha^T L\alpha \quad (2.10)$$

$J\alpha$  didefinisikan sebagai nilai trivial minimum dari  $J(\alpha, b)$ , dengan  $L$  adalah *matting* Laplacian yang berupa matrik berukuran  $N \times N$  dengan  $(i, j)$

$$\sum_{k|(i,j) \in w_k} \left( \delta_{ij} - \frac{1}{|w_k|} \left( 1 + \frac{1}{\frac{\varepsilon}{|w_k|} + \sigma_k^2} (I_i - \mu_k)(I_j - \mu_k) \right) \right) \quad (2.11)$$

$\delta_{ij}$  merupakan Kronecker delta,  $\mu_k$  dan  $\sigma_k^2$  adalah rata-rata dan varian intensitas pada window  $|w_k|$  disepanjang piksel  $k$ ,  $\Sigma_k$  adalah matrik kovarian berukuran  $3 \times 3$  pada window yang sama,  $|w_k|$  merupakan jumlah piksel dalam window, dan  $I$  adalah matrik identitas berukuran  $3 \times 3$  dengan nilai  $\varepsilon$  berkisar antara  $0.1^7$  hingga  $0.1^5$ . Dengan munculnya eigenvector terkecil, kegunaan lain properti *matting* Laplacian (2.10), adalah untuk menangkap informasi job *Fuzzycluster* pada piksel *image*, termasuk sebelum penentuan batasan dengan *user-specified* juga diperhitungkan [11].

#### 2.4.2. Linear Transformation

Pencarian transformasi linear pada eigenvector akan menghasilkan satu set vektor yang nilainya mendekati biner. Formulasinya dinotasikan sebagai  $E = [e^1, \dots, e^k]$  menjadi matrik  $N \times K$  dari eigenvector. Selanjutnya untuk menemukan satu set dari kombinasi linear  $K$ , vektor  $y^k$  meminimalkan

$$\sum_{i,k} |\alpha_i^k|^\gamma + |1 - \alpha_i^k|^\gamma, \text{ where } \alpha^k = Ey^k \text{ subject to } \sum_k \alpha_i^k = 1. \quad (2.12)$$

Jika  $0 < \gamma < 1$  maka nilai  $\gamma = 0,9$ , selanjutnya  $|\alpha_i^k|^\gamma + |1 - \alpha_i^k|^\gamma$  adalah nilai pengukuran yang robust pada komponen *matting* [13]. Karena *cost function* (2.12.) tidak convex, hasil dari proses Newton bergantung pada proses inisialisasi. Untuk menginisialisasi dapat dilakukan dengan menerapkan algoritma *K-means* pada eigenvector yang terkecil dalam *matting* Laplacian dan memproyeksikan vektor indicator dari *cluster* yang dihasilkan dari eigenvector  $E$

$$\alpha^k = EE^T m^{C^k} \quad (2.13)$$

Hasil *matting* komponen selanjutnya dijumlahkan sehingga memberikan solusi untuk persamaan (2.12).

### 2.4.3. Grouping Component

Hasil ekstraksi *foregroundmatte* secara lengkap selanjutnya ditentukan dengan penambahan sederhana komponen pada *foreground*. Misalkan  $\alpha^{k_1}, \dots, \alpha^{k_n}$  dirancang sebagai komponen *foreground*, maka

$$\alpha = \alpha^{k_1} + \dots + \alpha^{k_n} \quad (2.14)$$

Jika eigenvector yang terkecil tidak sama dengan nol, pengukuran kualitas hasil *alpha-matte* dilakukan dengan  $\alpha^T L \alpha$ , yang mana  $L$  adalah *matting* Laplacian. Kalkulasi awal korelasi diantara komponen *matting* dengan  $L$  dan penyimpanan dalam matrik  $\Phi K \times K$  didefinisikan

$$\Phi(k, l) = \alpha^{kT} L \alpha^l \quad (2.15)$$

Selanjutnya *matte cost* dihitung sebagai

$$J(\alpha) = b^T \Phi b \quad (2.16)$$

Dimana  $b$  adalah vektor biner  $K$ -dimensional yang mengindikasikan komponen yang dipilih.

## BAB 3

# METODOLOGI PENELITIAN

### 3.1. Terminologi

Ekstraksi pada video merupakan proses pengambilan objek dalam sebuah *cene* data video. Video diekstrak menjadi *image sequence*, selanjutnya ekstraksi objek dilakukan di setiap *image*. Ekstraksi objek dilakukan dengan cara segmentasi atau dengan mengekstrak *matte* yang disebut dengan “*matting*”. Dalam proses ekstraksi, segmentasi atau *matting* bertujuan untuk memisahkan objek *foreground* dari *image*, namun dalam teknik segmentasi setiap piksel secara tegas menjadi bagian dari *foreground* atau *background*.

Sedangkan dalam teknik *matting*, ada tiga parameter dalam *image* yang diperhitungkan dalam ekstraksi *matte*, yaitu:  $F$  (piksel yang secara tegas menjadi bagian dari *definite foreground*),  $B$  (piksel yang secara tegas menjadi bagian dari *definite background*), dan  $\alpha$  (*unknown pixel*, yaitu piksel yang menjadi bagian dari *foreground* dan *background*). Karena dalam *unknown pixel* terdapat piksel yang terdiri warna campuran dari *foreground* dan *background*, maka proses komputasi dalam teknik ini memerlukan waktu yang lebih lama dari pada teknik segmentasi. Namun karena proses pemisahan dilakukan secara mendetail hingga tingkat *sub-pixel*, maka hasil ekstraksi menjadi lebih realistis seperti hasil tangkapan kamera ketika objek *dicomposite*.

*Image matting* didahului dengan mendefinisikan konstrain secara manual. Konstrain didefinisikan karena dalam ekstraksi objek harus menyelesaikan permasalahan *ill-posed*, yaitu informasi semantic dalam *image* tidak tersedia, sehingga memerlukan intervensi *user* dalam mendefinisikan informasi semantic dalam sebuah *image*. *User* mendefinisikan *trimap* pada *image* sebagai konstrain yang merepresentasikan tiga parameter, yaitu *hand-drawn* warna putih berkorelasi dengan *definite foreground* dengan nilai  $\alpha = 1$ , *hand-drawn* warna hitam berkorelasi dengan *definite background* dengan nilai  $\alpha = 0$ , sedangkan *hand-drawn* warna abu-abu berkorelasi dengan *unknown region* dengan nilai  $\alpha$  yang harus diestimasi.

Estimasi nilai  $\alpha$  sangat menentukan kualitas *matte* yang diekstrak (ilustrasi dari *matte* yang terekstraksi dideskripsikan dalam gambar 3.1). Kualitas ekstraksi objek yang sangat dipengaruhi oleh *matte* yang diekstrak. *Matte* yang memiliki kualitas baik untuk ekstraksi objek adalah *matte* yang memiliki distribusi warna yang merata pada batas tepi objek.



Gambar 3.1. Input *image* dan *matte* yang terekstraksi[12].

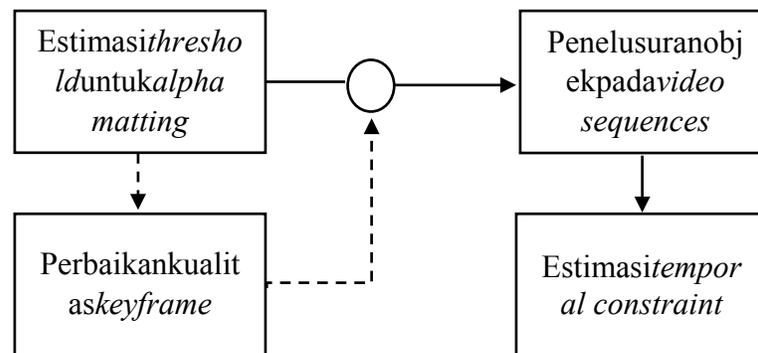
Ekstraksi objek dalam video dilakukan di seluruh *image sequences* pada video, sehingga *trimap* harus mengikuti pergerakan *scene* dalam *image sequences*.

Diasumsikan *image sequences* dalam video memiliki *temporal coherence*, sehingga untuk mengatasi permasalahan *temporal coherence* tidak dilakukan dengan menciptakan *trimap* secara independen, tetapi *trimap* diciptakan dengan pendekatan yang mengikuti fitur *temporal coherence*.

Teknik ini memungkinkan *trimap* dapat terpropagasi ke *next image* dengan sejumlah batasan yang didefinisikan *user* pada *current image*, sehingga hasilnya dapat mengurangi intervensi *user* secara dramatis. Selanjutnya *matting* pada *next image* dilakukan dengan konstrain berupa *image* dengan *trimap* yang terpropagasi.

### 3.2. Blok Diagram Penelitian

Sistem ekstraksi video objek yang diusulkan diilustrasikan dalam Gambar 3.2. *Matting* pada *still image* dilakukan dengan input berupa *source image* dan *image constraint* menggunakan *scribble-based interface* (seperti diilustrasikan Gambar 3.4) yang merupakan perbaikan dari konstrain *trimap* [24], [32] seperti dalam Gambar 3.3. Dengan *alpha matting* nilai *threshold* dalam  $\alpha$  (*alpha*) diestimasi secara adaptif menggunakan *FuzzyC-Means*[52]. Nilai *threshold* dalam *alpha* digunakan sebagai batas antara *foreground* dan *background* dalam proses ekstraksi *matte* (dideskripsikan dalam Gambar 3.5). Proses ekstraksi objek ini selanjutnya dibahas secara rinci dalam Bab 4.



Gambar 3.2. Blok diagram penelitian

Dalam ekstraksi objek video, *keyframe* diperlukan sebagai referensi untuk penelusuran objek dalam *video sequence*. *Keyframe* yang berasal dari image terkompresi dalam *alpha matting* mengalami penurunan kualitas di sekitar batas tepi objek. Perbaikan kualitas dilakukan dengan mereduksi *noise* di sekitar batas tepi objek dengan *Discrete Cosine Transform-Two Dimensional (DCT-2D)*, tahapan ini secara rinci dibahas dalam Bab 5.

Ekstraksi objek pada frame selanjutnya dilakukan dengan *alpha matting* seperti dalam Bab 4 atau Bab 5. Karena *matting* dalam video memerlukan input *scribble* yang dinamis terhadap pergerakan objek, maka *scribble* perlu diciptakan sesuai dengan pergerakan objek. *Scribble* diciptakan secara otomatis dengan estimasi *background subtraction* antara frame sekarang dan frame selanjutnya. Secara rinci tahapan ini dibahas dalam Bab 6.



Gambar 3.3. *Input image dan trimap.*  
Gambar diambil dari (*video matting dataset*)

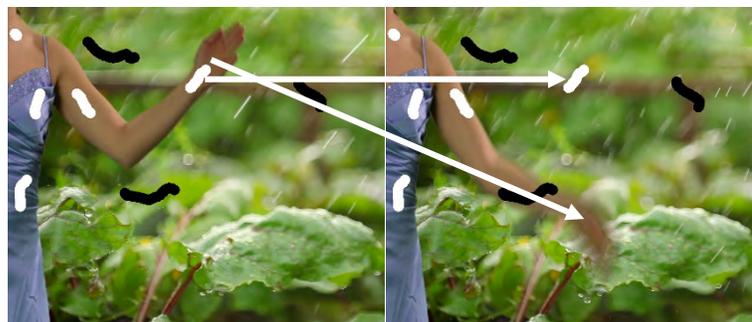


Gambar 3.4. *Trimap berbasis scribble*  
Gambar diambil dari (*video matting dataset*)

Pergerakan frame dalam *video sequence* dari waktu ke waktu mengalami perubahan, sedangkan konstrain manual yang didefinisikan oleh *user* bersifat statis atau tidak mengikuti pergerakan *scene image* (Gambar 3.6). Akurasi ekstraksi *matte* akan turun secara dramatis jika konstrain tidak mengikuti pergerakan *scene*. Oleh karena itu *estimasi temporal constrain* tsangat diperlukan agar konstrain dapat mengikuti pergerakan objek. Secara detail *estimasi temporal constraint* dibahas dalam Bab 7.



Gambar 3.5. Proses ekstraksi *matte* dengan *trimap scribble-based interface*  
Gambar diambil dari (*video matting dataset*)



Gambar 3.6. Konstrain statis terhadap pergerakan scene  
Gambar diambil dari (*video matting dataset*)

# BAB 4

## ESTIMASI *THRESHOLD* UNTUK *ALPHA* *MATTING*

### 4.1. *Alpha Matting*

Ekstraksi objek *foreground* dari *image* secara keseluruhan merupakan tugas penting dalam aplikasi *image* maupun *video editing*. Agar hasil ekstraksi objek *foreground* mendapatkan hasil terbaik, pemisahan dilakukan pada piksel *image* yang disebut dengan “*pulling matte*” atau “*digital matting*”. Penarikan *matte* dilakukan dengan menggabungkan warna *semi-transparent* dari *foreground* dengan warna *background* sehingga menghasilkan warna campuran baru. Tingkat gradasi warna *foreground* berkisar antara full transparan hingga full buram. Apabila warna *foreground* full transparan warna campuran akan menjadi warna *background* sebaliknya jika warna *foreground* full buram warna campuran akan menjadi warna *foreground*.

Warna campuran merupakan rata-rata tertimbang antara warna *foreground* dan *background*. *Alpha matting* merupakan kombinasi *convex* dari dua warna yang memungkinkan dapat memberikan efek transparansi dalam komputer grafis. Nilai *alpha* berkisar antara 0.0 – 1.0 dengan nilai full transparan bernilai 0.0 sedangkan full buram bernilai 1.0.

#### 4.2. *Threshold dalam Alpha Matting*

Hasil ekstraksi *matte* yang baik adalah *matte* yang memiliki distribusi warna yang merata, tidak terlalu putih atau tidak terlalu buram. Apabila piksel *matte* terlalu putih maka akan dominan bagian piksel yang berkorelasi dengan *foreground* yang diekstrak, sebaliknya jika terlalu buram maka *matte* yang terekstrak akan dominan dengan piksel *background*. Perbedaan pemisahan objek antara segmentasi dan *matting* terletak pada bagaimana memperlakukan piksel *foreground* dan *background*.

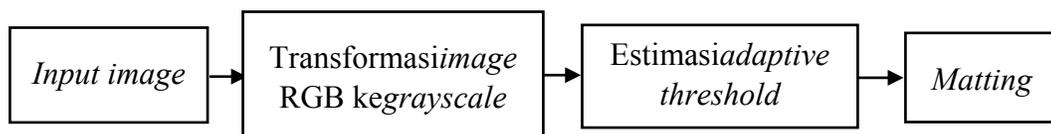
Dalam teknik segmentasi proses pemisahan dilakukan secara tegas terhadap piksel, sehingga dalam sebuah piksel hanya menjadi bagian dari *foreground* atau *background* saja. Sebaliknya dalam teknik *matting*, piksel *unknown region* ( $\alpha = \text{alpha}$ ) menjadi bagian dari *foreground* dan *background*. Proses penarikan *alpha* yang disebut dengan “*alpha matting*” dilakukan dengan membedakan piksel yang menjadi bagian dari *foreground* dan *background*. Untuk mendefinisikan *unknown region* dilakukan dengan mendefinisikan nilai ambang (*threshold*).

Dalam penelitian sebelumnya [11], *threshold alpha* didefinisikan antara  $0.1^7$  -  $0.1^5$  dengan asumsi bahwa nilai *noise* dalam sebuah *image* berkisar dalam range tersebut. Pendefinisian *threshold* dilakukan berdasarkan persepsi *user* dengan mempertimbangkan karakteristik *image* yang akan diekstrak, sehingga *user* memerlukan keahlian tertentu dalam menentukan *threshold* agar mendapatkan *matte* yang baik. Kesalahan *user* dalam menentukan *threshold* akan berpengaruh terhadap kualitas *matte*. Untuk mengatasi permasalahan ini, algoritma berbasis *adaptive*

*threshold* diusulkan untuk menentukan nilai *threshold* secara adaptif mengacu pada karakteristik *image* yang digunakan sebagai *threshold alpha*.

### 4.3. Metodologi *Alpha Matting*

Gambar 4.1 mendeskripsikan tahapan-tahapan dalam ekstraksi *matte* dengan mengestimasi *alpha* dengan menghitung *threshold* secara adaptif.



Gambar 4.1. Metodologi *alpha matting*

#### a. *Input image*

*Input image* merupakan *image* yang digunakan sebagai data masukan dalam proses *matting*. *Image* yang digunakan berupa *image* RGB yang belum dikompresi.

#### b. Transformasi *image* RGB ke *grayscale*

Untuk menyederhanakan proses komputasi pada *image*, nilai piksel *image* yang terdiri dari chanel *Red*, *Green*, *Blue* (RGB) ditransformasi ke satu dimensi dalam bentuk *image grayscale*.

#### c. Estimasi *adaptive threshold* nilai *alpha*

Kualitas ekstraksi sangat ditentukan oleh nilai *threshold* pada *alpha* (*unknown region*), karena dalam piksel pada *unknown region* setiap piksel menjadi bagian *foreground* dan *background*. Nilai *threshold* dalam *alpha*

adaptive terhadap perubahan intensitas piksel pada image, dapat digunakan untuk membedakan piksel yang menjadi bagian dari *foreground* dan *background*, dan pekerjaan ini sangat menentukan kualitas *matte*.

#### d. *Matting*

*Matting* adalah proses mengekstrak *matte* dari *image* secara keseluruhan. Dalam algoritma yang diusulkan, teknik *matting* dilakukan dengan mengaplikasikan pendekatan *Closed-form solution* [11]

#### 4.4. Ekstraksi *Matte*

Untuk merumuskan metode *digital matting*, *image* ( $I$ ) diasumsikan sebagai kombinasi antara *foreground* ( $F$ ) dan *background* ( $B$ ). Warna pada piksel pada *image* ke  $i$  ( $I_i$ ) diasumsikan merupakan kombinasi linear yang saling berhubungan antara warna *foreground* dan warna *background* seperti dalam persamaan(2.1).

*Matting* dalam *natural image* diasumsikan bahwa perhitungan pada sebelah kanan persamaan *compositing* (2.1) merupakan permasalahan estimasi *unknown region* yang melibatkan tiga ruang warna *image* di setiap piksel. Pada umumnya permasalahan *matting* adalah keterbatasan *constraint*, sehingga diperlukan intervensi *user* untuk mendapatkan *matte* yang baik. Beberapa penelitian menggunakan *trimap* sebagai inisialisasi atau titik awal.

Pada persamaan (2.1), setiap piksel diasumsikan sebagai *convex combination* dari *layer*  $K$  pada *image*  $F^1, \dots, F^K$

$$I_i = \sum_{k=1}^K \alpha_i^k F_i^k \quad (4.1)$$

Dalam sistem yang dikembangkan, pertama kali *user* mendefinisikan *trimap* berbasis *user scribble* sebagai *constraint* pada *matte*. Untuk mendefinisikan *scribble* digunakan *pencil tool* yang digoreskan pada *region* yang merepresentasikan *foreground* dan *background*.

*Scribble* warna hitam menunjukkan piksel *background* yang berkorelasi dengan  $\alpha = 0$  sedangkan *scribble* warna putih menunjukkan piksel *foreground* yang berkorelasi dengan  $\alpha = 1$ . Untuk ekstraksi *alpha matte* yang sesuai dengan *user scribble* dilakukan dengan menyelesaikan persamaan (4.1).

$$\alpha = \arg \min \alpha^T L \alpha + \lambda (\alpha^T - b_s^T) D_s (\alpha - b_s) \quad (4.2)$$

Dengan  $L$  adalah matrik berukuran  $N \times N$ ,  $\lambda$  merupakan nilai yang didefinisikan, adalah matrik diagonal yang elemen-elemen diagonalnya bernilai 1 untuk piksel *constraint*, dan 0 untuk piksel yang lain. Sedangkan  $b_s$  adalah vektor yang terdiri nilai *alpha* yang secara khusus digunakan untuk piksel *constraint*, sedangkan nilai 0 untuk piksel yang lain.

Nilai *global minimum* yang diestimasi dalam *quadratic cost function* dalam persamaan (4.2) ditemukan adanya perbedaan, sehingga system *sparse linear* persamaan (4.3) diaplikasikan untuk mengatur persamaan hingga bernilai 0.

$$(L + \lambda D_s) \alpha = \lambda b_s \quad (4.3)$$

Persamaan ini didemonstrasikan pada *natural image* (Gambar 4.2a) sebagai input *image* yang menunjukkan area *foreground* dan *background* yang memiliki warna

sederhana. Dengan memberikan dua coretan titik hitam dan putih (Gambar 4.2b), dapat digunakan untuk mengekstrak *matte* dengan kualitas yang baik (Gambar 4.2c), karena distribusi warna pada piksel *unknown* merata, sehingga batas tepi objek terlihat halus dan ketika dikorelasikan dengan *foreground* pada *image input* hasil pemisahan objek terlihat lebih natural (Gambar 4.2d).

Hasil ekstraksi *matte* yang dilakukan dengan pendekatan *Closed-form solution* [11] terlihat halus ketika digunakan untuk ekstraksi *matte* dari *input image* pada Gambar 4.2(a). Kualitas *matte* dipengaruhi oleh tingkat gradasi pada *unknown region*, yang mana *matte* yang baik memiliki distribusi yang merata.

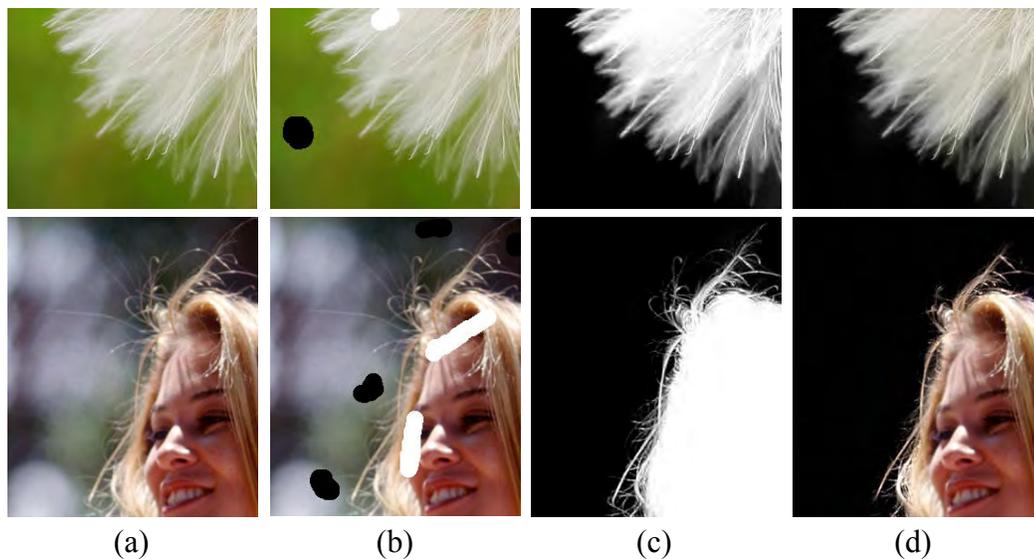
Algoritma ini diujicoba diaplikasikan untuk ekstraksi *matte* pada *image* yang memiliki warna yang lebih kompleks. Gambar 4.3(a) merupakan *input image* dengan kombinasi warna yang lebih kompleks. Hasil ekstraksi *matte* dengan *closed-form solution* ditunjukkan dalam Gambar 4.3(b). *Matte* yang terekstraksi dilihat dari potongan gambar yang diperbesar (Gambar 4.3c) menunjukkan distribusi warna pada daerah batas tepi tidak merata (terlalu besar tingkat keburamannya) sehingga ketika dikorelasikan dengan input *image*, maka akan terjadi banyak piksel *foreground* yang dieliminasi.

*Adaptive threshold* diusulkan dalam penelitian ini diperlukan agar pendefinisian *threshold* dapat dilakukan secara adaptif sehingga lebih presisi dalam menentukan nilai *threshold* sehingga dapat meningkatkan kualitas *alpha*. Untuk meningkatkan kualitas *matte*, dalam penelitian ini nilai *alpha* di *generate* secara

adaptif. Algoritma *adaptive threshold* Otsu dan *Fuzzy C-Means* (FCM) diaplikasikan dan hasilnya dievaluasi untuk menentukan *threshold* yang terbaik.

#### 4.4.1. Algoritma *Otsu*

Algoritma *Otsu*[53] adalah algoritma *adaptive threshold* berbasis histogram yang menunjukkan perubahan nilai intensitas setiap piksel dalam *image* satu dimensi. Sumbu -  $x$  menyatakan tingkat perbedaan intensitas, sedangkan sumbu -  $y$  digunakan untuk menyatakan sejumlah piksel yang memiliki nilai intensitas. *Threshold* pada piksel *image* dilakukan menggunakan *histogram clustering*, dengan *optimal threshold* diperoleh dari perbedaan intensitas piksel. Hal ini dapat digunakan untuk pemisahan kelas. Informasi yang diperoleh dari histogram adalah jumlah perbedaan intensitas (dinotasikan dengan  $L$ ), dan jumlah piksel untuk setiap tingkatan intensitas yang dinotasikan dengan  $n(k)$  dengan  $k = 0 \dots, 255$ . Proses pencarian *threshold* dalam algoritma Otsu dilakukan dengan tahapan sebagai berikut:



Gambar 4.2. Ekstraksi pada distribusi warna sederhana, (a). input *image*, (b). *scribbleimage*, (c). *matte* extraction, (d). object extraction

1. Menghitung histogram *image* normal yang dilambangkan  $p_i$  dengan  $i = 1, 2, 3, \dots, L - 1$

$$p_i = \frac{n_i}{MN} \quad (4.4)$$

Dengan  $n_i$  adalah jumlah intensitas piksel dan  $MN$  adalah jumlah  $n_i$  mulai dari  $n_0 - n_{L-1}$

2. Menghitung jumlah komulatif dari  $p_i(k)$  dengan  $k = 1, 2, 3, \dots, L - 1$

$$P_1(k) = \sum_{i=0}^k p_i \quad (4.5)$$

3. Menghitung rata-rata komulatif (*cumulative mean*),  $m(k)$  untuk  $k = 1, 2, 3, \dots, L - 1$

$$m(k) = \sum_{i=0}^k ip_i \quad (4.6)$$

4. Menghitung rata-rata intensitas global,  $m_G$  menggunakan formula:

$$m_g = \sum_{i=0}^{L-1} ip_i \quad (4.7)$$

5. Menghitung varian antar kelas (*between-class variance*),  $\sigma_B^2(k)$  untuk  $k = 1, 2, 3, \dots, L - 1$  dengan rumus:

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (4.8)$$

6. Memilih *threshold*  $k^*$  yang merupakan indeks dari nilai varians antar kelas maksimum,  $\sigma_B^2(k) \rightarrow \max$ . Jika ternyata terdapat lebih dari satu

nilai  $k^*$ , maka nilai *threshold* ditentukan dengan menghitung rata-rata nilai  $k^*$ .

7. Menghitung *separability measurement*,  $\eta^*$  pada  $k = k^*$  menggunakan formula:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_B^2} \quad (4.9)$$

Sedangkan,

$$\sigma_B^2 = \sum_{i=0}^{L-1} (1 - m_G)^2 p_i \quad (4.10)$$

Catatan: nilai  $k$  terjadi pada saat nilai  $\sigma_B^2(k)$  maksimum.

Ketika diaplikasikan pada input *image* (Gambar 4.3a), hasil *threshold* dari algoritma *Otsu* digunakan sebagai *threshold alpha* dalam ekstraksi *matte*. Hasil ujicoba menunjukkan *matte* yang dihasilkan cenderung ke warna putih (diilustrasikan dalam *matte* yang ditunjukkan dalam Gambar 4.3d dan versi diperbesar dalam Gambar 4.3e), sehingga ketika dikorelasikan dengan input *image* akan banyak mengeliminasi piksel *background*.

#### 4.4.2. Algoritma *Fuzzy C-Means*

*Fuzzy C-Means* (FCM) yang dikemukakan Bezdek pada tahun 1973 merupakan metode *clustering hard c-means* yang ditingkatkan [54], dan didefinisikan sebagai dataset  $X = \{x_1, x_2, \dots, x_n\}$ . Clustering dataset  $X$  dikelompokkan menjadi beberapa cluster  $c$  dengan ( $2 \leq c \leq n$ ) dan  $V = \{v_1, v_2, \dots, v_c\}$  yang merupakan pusat cluster dari  $c$ . Setiap sampel dalam partisi Fuzzy tidak secara tegas

dikelompokkan dalam jenis tertentu, namun derajat keanggotaan tertentu menjadi criteria tertentu. Kriteria dalam FCM clustering didefinisikan dalam fungsi objektif:

$$J(U, V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m (d_{ik})^2 \quad (4.11)$$

dimana  $J(U, V)$  menunjukkan berbagai jenis sampel pusat cluster dengan jumlah bobot dari jarak kuadrat tertimbang;  $u_{ik}$  mengindikasikan kategori keanggotaan  $i$  untuk sampel  $x_k$ ;

$$d_{ik} = d \|x_k - v_j\| \quad (4.12)$$

dimana  $d_{ik}$  adalah jarak Euclidian dari urutan  $k$  terhadap pusat  $I$ ;  $m$  merupakan indek bobot yang memainkan peranan penting dalam mengatur derajat fuzzy dalam clustering, dan  $m \in [1, \infty)$ ;

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}} \forall i, \forall k} \quad (4.13)$$

dengan pusat cluster  $V$

$$v_i = \frac{\sum_{k=1}^N (u_{ik})^m x_k}{\sum_{k=1}^N (u_{ik})^m}, (i = 1, 2, \dots, c), (i < m) \quad (4.14)$$

Langkah-langkah dalam algoritma ini diaplikasikan dalam input *image* (gambar 4.3a).

Nilai *threshold* yang dihasilkan dari FCM diaplikasikan dalam ekstraksi *matte*

menggunakan metode *Closed-form solution*. Hasil ekstraksi secara visual ditunjukkan dalam Gambar 4.3f yang bagian *imagenya* diperbesar dalam Gambar 4.3g. Secara visual terlihat bahwa dengan *threshold* FCM distribusi tingkat gradasi lebih merata sehingga ketika dikorelasikan dengan input *image* hasil ekstraksi objek *foreground* lebih natural.

#### 4.5. Evaluasi Ekstraksi Objek dengan *Adaptive Threshold*

Untuk menguji estimasi *threshold* pada *alpha channel* yang diusulkan, *input image* dari *video matting dataset* diaplikasikan pada algoritma *Otsu* maupun FCM, selanjutnya *matte* yang terekstrak dikomparasi dengan *ground truth*. Secara detail dataset yang digunakan ditunjukkan dalam lampiran A. Secara kuantitatif hasil ekstraksi *matte* dievaluasi dengan MSE (*Mean Squared Error*).

$$MSE = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \left( \frac{abs(Grt\ Im\ g - MatteExt)}{M \times N} \right)^2 \quad (4.24)$$

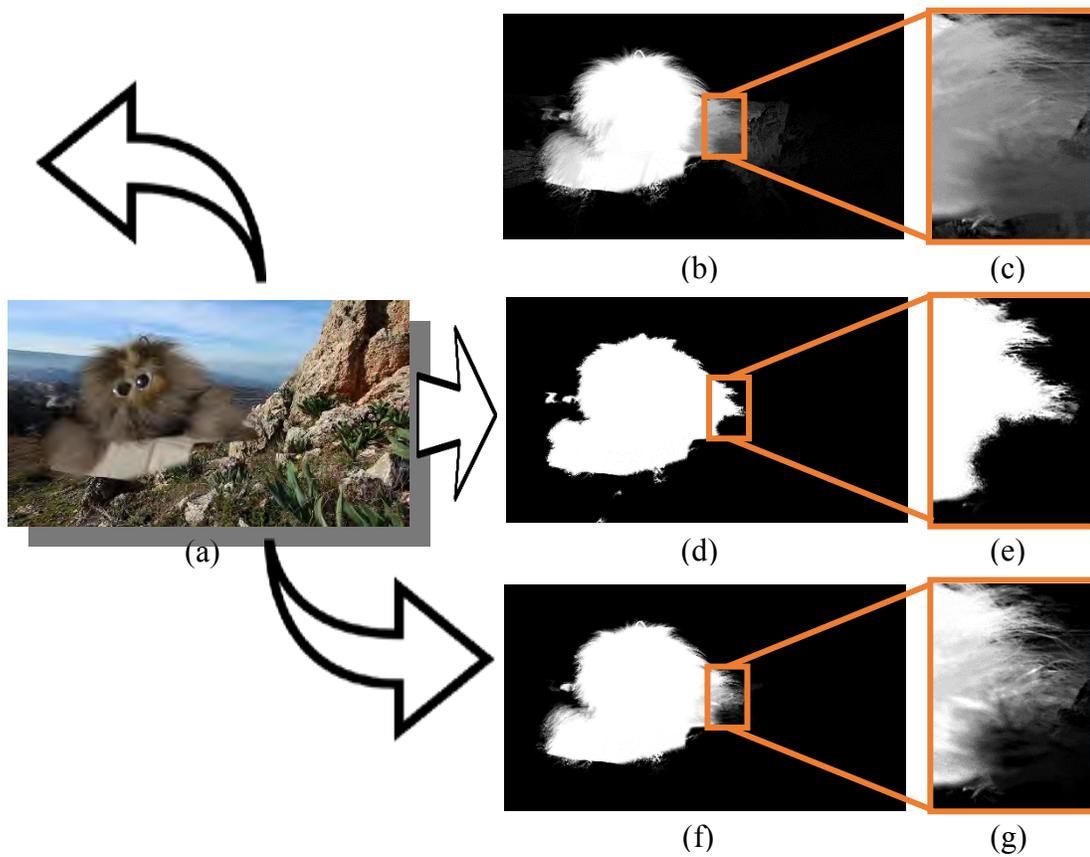
Dengan *GrtImg* adalah *ground truth image* sebagai *image* referensi, *MatteExt* adalah *matte* yang terekstraksi oleh system dan  $M \times N$  adalah ukuran *image* yang dieksekusi. Hasil perbandingan ekstraksi *matte* secara kuantitatif ditunjukkan dalam tabel 4.1 dan gambar 4.4. Sedangkan hasil perbandingan secara visual dideskripsikan dalam lampiran B.

#### 4.6. Kesimpulan

Tujuan *matting* adalah mengekstrak *matte* dari *image* secara keseluruhan. Kualitas *matte* yang baik adalah *matte* yang memiliki distribusi warna yang merata

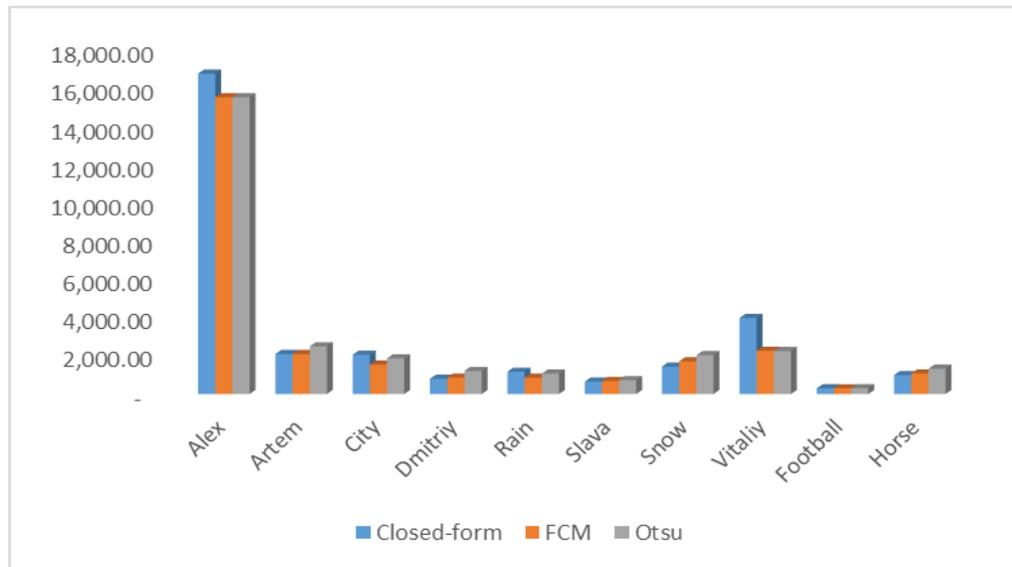
(tidak terlalu putih atau tidak terlalu buram). *Matte* yang memiliki distribusi warna yang cenderung buram akan banyak mengeliminasi piksel *foreground* sedangkan yang terlalu putih akan mengeliminasi piksel *background*. Untuk menjaga agar *matte* memiliki distribusi warna yang merata, estimasi *threshold* dalam *alpha* ditentukan secara adaptif, yaitu dihitung berdasarkan distribusi warna dari *input image*.

Pengujian yang dilakukan dengan algoritma *Otsu* dan FCM, secara visual hasil *alpha matting* (penarikan *matte* dalam *region alpha*) menunjukkan bahwa *threshold* dalam algoritma FCM memiliki tingkat akurasi yang lebih baik daripada algoritma *Otsu* maupun *Closed-form solution* ketika diaplikasikan dalam warna yang kompleks, karena distribusi warna *matte* pada *unknown region* lebih merata. Selain itu, ketika hasil ekstraksi tersebut dikomparasi dengan *ground truth* (*matte* hasil ekstraksi manual), *threshold* FCM dapat menurunkan error sebesar 11.05% dibandingkan dengan *Closed-form solution*, sedangkan *Otsu* dapat menurunkan error sebesar 4.72%.



Gambar 4.3. Perbedaan visual ekstraksi matte (a). *input image*, (b,c). *matte* terekstraksi dengan *Closed-form solution*, (d, e). *matte* terekstraksi dengan *threshold Otsu*, (f, g). *matte* terekstraksi dengan *threshold FCM*.

Gambar diambil dari (*video matting dataset*)



Gambar 4.4. Grafik hasil perbandingan ekstraksi *matte*

Tabel 4.1. Perbandingan jumlah perbedaan piksel dalam *image matting* antara *closed-form*, FCM dan *Otsu*

<b>Image</b>	<b>Closed-form</b>	<b>FCM</b>	<b>Otsu</b>
<i>Alex</i>	16,817.00	15,578.00	15,578.00
<i>Artem</i>	2,102.50	2,101.70	2,503.30
<i>City</i>	2,064.00	1,545.00	1,873.60
<i>Dmitriy</i>	804.56	863.29	1,213.10
<i>Rain</i>	1,170.80	857.19	1,074.40
<i>Slava</i>	657.12	683.56	729.36
<i>Snow</i>	1,443.90	1,719.90	2,044.20
<i>Vitaliy</i>	3,993.20	2,272.20	2,264.00
<i>Football</i>	302.74	301.40	309.94
<i>Horse</i>	997.29	1,077.10	1,331.80

# **BAB 5**

## **DCT-2D UNTUK PERBAIKAN KUALITAS**

### ***KEYFRAME***

Ekstraksi objek menggunakan *image matting* yang merupakan salah teknik dalam pemisahan objek seperti yang disajikan dalam Bab 4 yang telah diaplikasikan dalam *image original (raw image)*. Meningkatnya penggunaan teknologi jaringan sebagai media transformasi data multimedia serta memori untuk penyimpanan data secara otomatis memerlukan teknologi kompresi untuk memampatkan data multimedia.

Penggunaan teknologi kompresi pada data multimedia khususnya pada *image* atau video membuat informasi yang berupa nilai intensitas warna mengalami distorsi. Meskipun distorsi nilai intensitas ini secara visual tidak terlihat, namun hal ini akan sangat berpengaruh ketika digunakan dalam *image matting* (proses memisahkan *matte* dari *image*), karena kualitas hasil ekstraksi *matte* menurun secara dramatis. Perbaikan akurasi *image matting* diusulkan dengan mengurangi *noise* di sekitar batas tepi objek menggunakan DCT-2D.

#### **5.1. Blok Diagram Perbaikan Kualitas *Matte***

Gambar 5.1 mendeskripsikan blok diagram tahapan dari perbaikan kualitas pemisahan objek dengan ekstraksi *matte* menggunakan *input image* yang terkompresi dalam bentuk JPG.

a. ***Input image***

Input *image* dalam *framework* ini menggunakan *image* yang didapatkan dari proses pemecahan *frame* video test standar MPEG-4 yang diperoleh dataset UCF Sport Action. *Image* terdiri dari ***playing golf, football, running, diving, skateboarding, lifting, walk front, karate, dan riding horse.***

b. ***Scribble image***

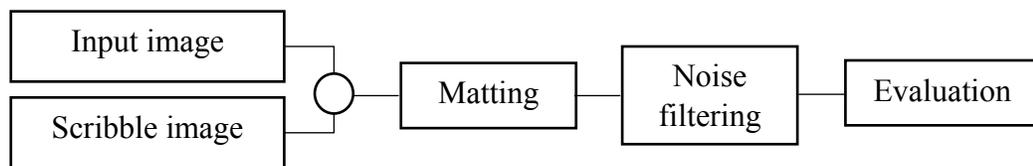
Informasi semantik dalam *image* merupakan permasalahan *ill-posed*, sehingga informasi semantik yang akan diekstraksi memerlukan intervensi *user*. Informasi semantic dilakukan untuk membedakan daerah yang membedakan daerah objek *foreground* dan *background*. Inisialisasi dilakukan dengan mendefinisikan *scribble* (coretan) pada daerah yang mewakili *foreground* dan *background* sebagai *constraint* dalam ekstraksi *matte*. *Scribble* didefinisikan sebagai coretan pensil berwarna putih untuk merepresentasikan daerah yang mewakili region *foreground* ( $\alpha = 1$ ) dan hitam untuk daerah yang merepresentasikan region *background* ( $\alpha = 0$ ).

c. ***Matting***

*Matting* merupakan proses menarik *matte* dari *image* secara keseluruhan, sehingga dapat digunakan untuk mengekstrak piksel yang berkorelasi dengan *foreground* dan mengeliminasi piksel yang berkorelasi dengan *background*. Proses ekstraksi *matte* lebih rinci telah dilakukan dalam Bab 4.

**d. Noise filtering**

Kualitas *image* seringkali dirusak oleh sejumlah *noise* yang diakibatkan oleh proses pemindaian yang tidak sempurna atau turunan dari proses pembentukan *image* itu sendiri. Untuk meningkatkan kualitas *image* salah satu teknik yang digunakan adalah mereduksi *noise* yang dilakukan dengan *noise filtering* menggunakan filter *Discrete Cosine Transform-2D* (DCT-2D).



Gambar 5.1. Blok diagram perbaikan *kualitas matte*

**5.2. Pengujian *Image Matting* Pada *Image* Terkompresi**

Eksperimen diaplikasikan pada video tes standar MPEG-4 yang diperoleh dari dataset UCF sport action (disajikan dalam lampiran C). Pertama kali data video dipecah menjadi beberapa *frame* dan *frame* pertama diperlakukan sebagai *image* yang dianalisa. *Image* diekstraksi menggunakan algoritma *Closed-form solution* [11] seperti yang dilakukan dalam Bab 4.

Percobaan menunjukkan hasil yang cukup impresif ketika diaplikasikan pada *raw image* (ditunjukkan dalam Gambar 5c). Namun algoritma ini tidak begitu *robust* ketika diaplikasikan dalam *image* yang terkompresi dalam JPG (ilustrasi gambar 5.2d). Hal ini terjadi karena ketika *image* dikompresi, nilai intensitas piksel di setiap ruang warna mengalami perubahan. Perubahan nilai intensitas piksel antara *raw*

*image* dengan *image* yang terkompresi ditunjukkan dalam bentuk histogram yang membandingkan frekuensi nilai intensitas diantara *raw image* dan *image* terkompresi dalam masing-masing ruang warna. Image terkompresi menunjukkan warna yang lebih kontras dibandingkan dengan *raw image* (digambarkan dalam Gambar 5.3).

Untuk meningkatkan kemampuan dari algoritma yang telah dilakukan dalam penelitian sebelumnya (*closed-form solution*), algoritma FCM [52] seperti yang telah diaplikasikan dalam Bab 4 digunakan sebagai *adaptive threshold* dalam *alpha matting*. Sedangkan *noise* hasil ekstraksi yang berada di sekitar batas tepi objek direduksi menggunakan filter DCT 2D [55], dengan tahapan sebagai berikut:

$$C(u, v) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x, y) \cos \left[ \frac{\pi(2x+1)u}{2n} \right] \cos \left[ \frac{\pi(2y+1)v}{2n} \right] \quad (5.1)$$

Untuk  $u, v = 1, 2, 3, \dots, N - 1$ ,  $a(u)$  dan  $a(v)$  didefinisikan dengan:

$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u=0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0 \end{cases} \quad (5.2)$$

Dan transformasi invers didefinisikan dengan:

$$f(x, y) = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} \alpha(u) \alpha(v) C(u, v) \cos \left[ \frac{\pi(2x+1)u}{2n} \right] \cos \left[ \frac{\pi(2y+1)v}{2n} \right], \quad (5.3)$$

Dengan  $x, y = 1, 2, 3, \dots, N - 1$ .

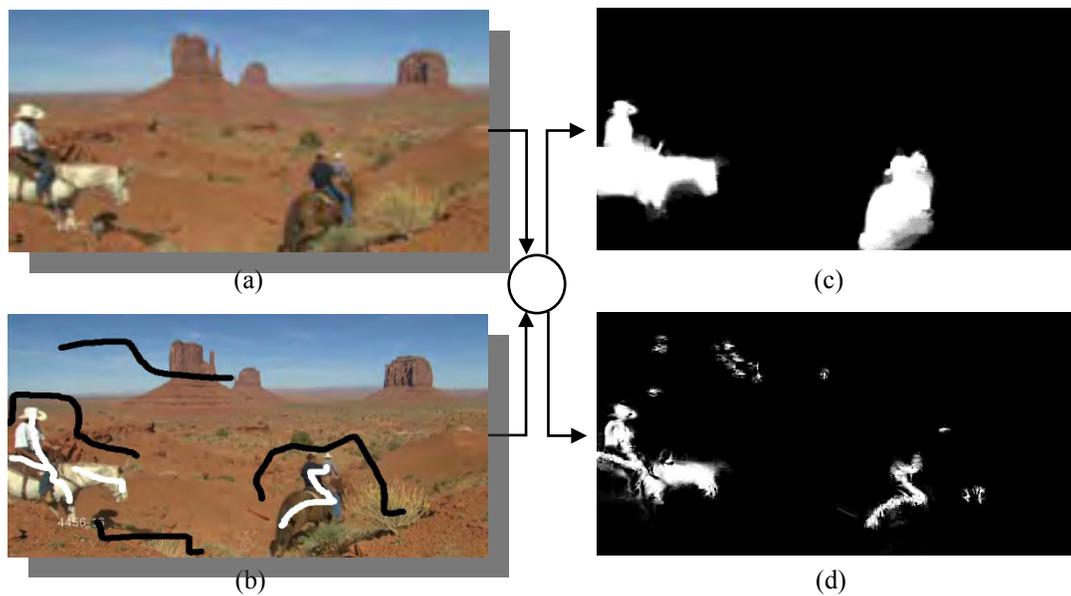
Setelah diimplementasikan dalam 10 *image* yang terkompresi, hasil ekstraksi dievaluasi menggunakan RMSE (*Root Mean Squared Error*)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M-1} \sum_{j=1}^{N-1} (Grd.img_{(i,j)} - Ext.Obj_{(i,j)})^2}{N \times M}} \quad (5.4)$$

Dengan *Grd. img* *groundtruth image* sebagai referensi, *Ext. Obj* adalah objek hasil ekstraksi dan  $N \times M$  adalah ukuran dari *image* yang dianalisa. Hasil ekstraksi *matte* antara *closed-form solution matting* dengan peningkatan *closed-form solution matting* dengan DCT-2D disajikan dalam grafik yang ditunjukkan dalam Gambar 5.4. Meskipun secara visual tidak terlihat perbedaan, namun hasil perhitungan secara kuantitatif menunjukkan peningkatan dengan adanya penurunan nilai RSME seperti yang disajikan dalam Tabel 5.1.

### 5.3. Kesimpulan

Kualitas hasil ekstraksi obyek dengan input *image* terkompresi pada algoritma *Closed-form solution* menurun secara dramatis karena adanya beberapa perubahan nilai intensitas pada *channel* RGB. Batas tepi obyek hasil ekstraksi terlihat kasar baik secara *visual* maupun hasil pengukuran dengan *RootMean Squared Error* (RMSE) yang membandingkan antara obyek hasil ekstraksi dengan *groundtruth*. Hal ini merupakan *noise* yang ditimbulkandari proses ekstraksi



Gambar 5.2. Ekstraksi *matted* dalam *raw image* dan *image* terkompresi. (a) *input image*, (b) *scribble image*, (c) hasil ekstraksi *matted* dari *raw image*, (d) hasil ekstraksi *matted* dari *image* terkompresi

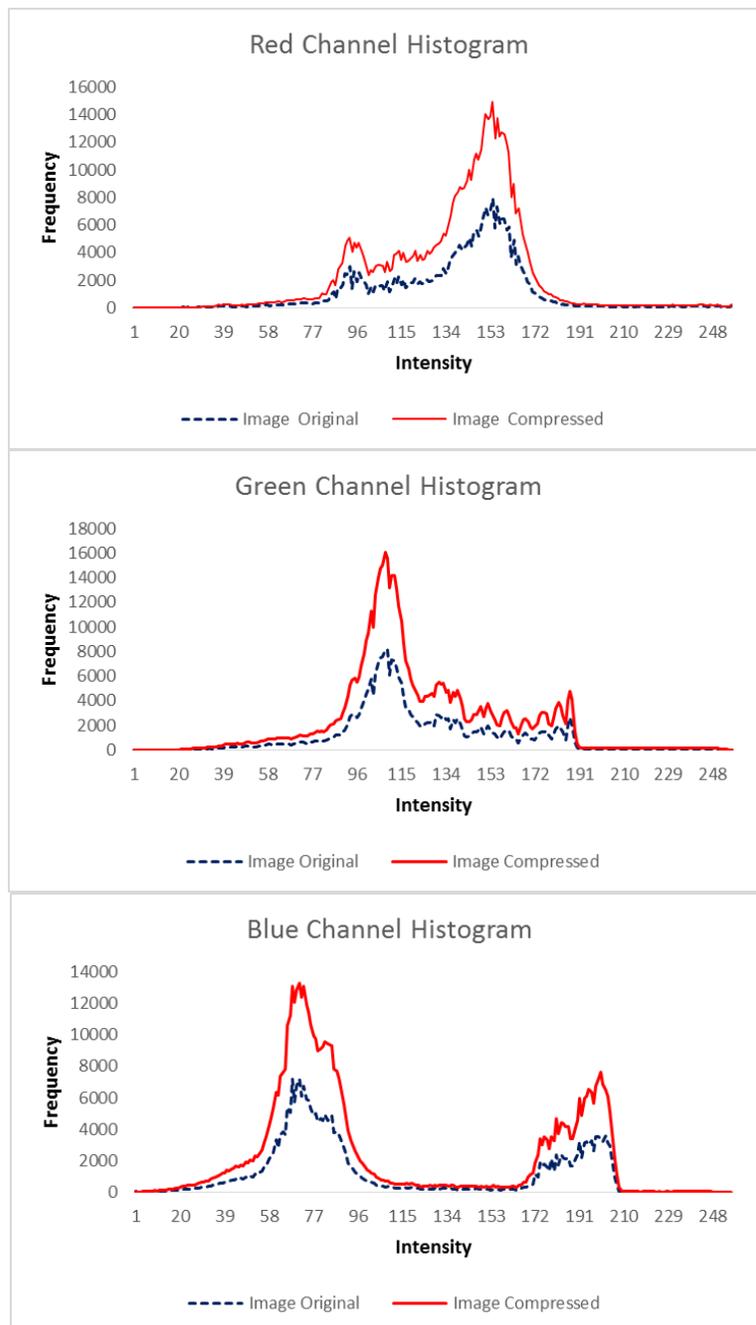
Gambar diambil dari (*ucf sport action dataset*)

Tabel 5.1. Perbandingan nilai RMSE hasil ekstraksi *matte* antara *closed-form solution matting* dan peningkatan *closed-form solution matting* dengan DCT-2D

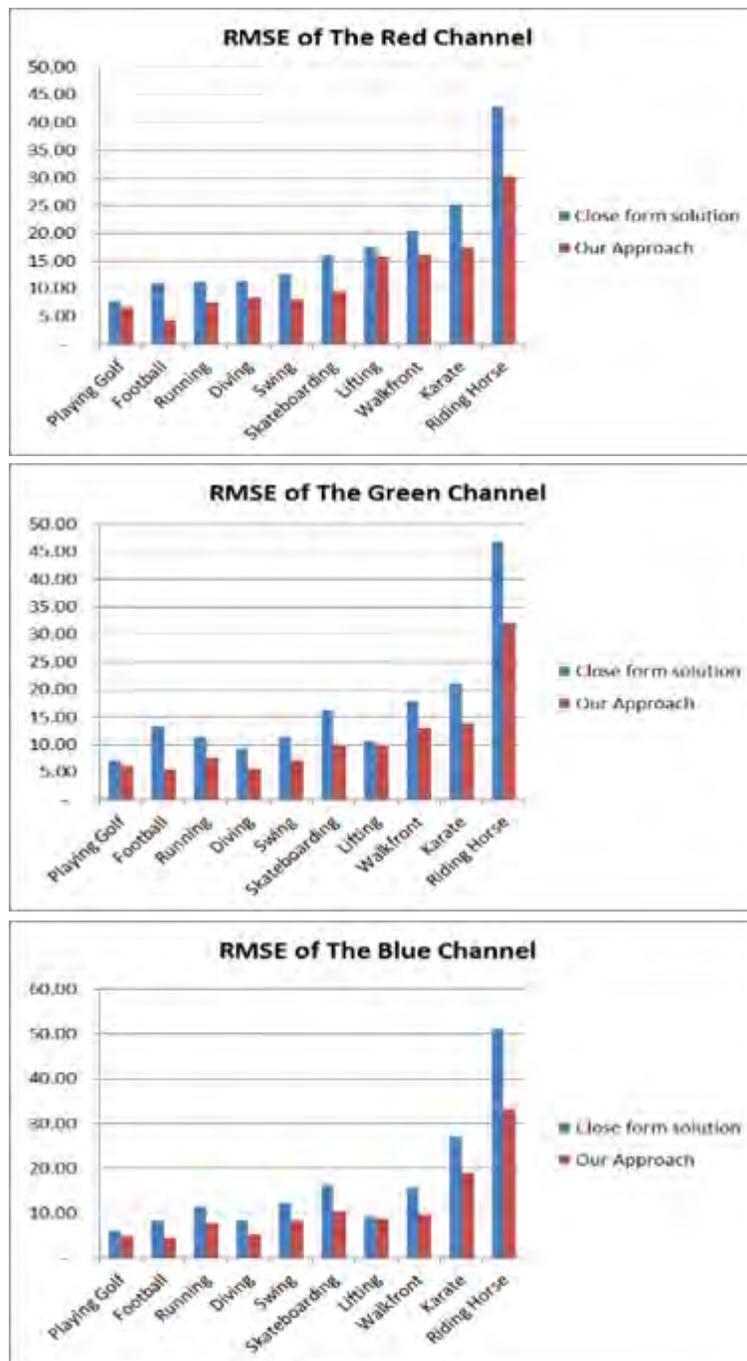
Image	Closed-Form Solution			Our Approach		
	Red Channel	Green Channel	Blue Channel	Red Channel	Green Channel	Blue Channel
Playing Golf	7.79	7.17	5.94	6.69	6.07	4.85
Football	11.07	13.42	8.34	4.49	5.57	4.43
Running	11.37	11.45	11.37	7.69	7.76	7.71
Diving	11.49	9.36	8.47	8.52	5.69	5.23
Swing	12.74	11.50	12.13	8.07	7.21	8.19
Skateboarding	16.12	16.27	16.10	9.65	9.85	10.41
Lifting	17.54	10.56	9.28	15.99	9.80	8.76
Walkfront	20.53	17.95	15.51	16.30	13.14	9.71
Karate	25.22	21.19	27.03	17.51	13.90	19.00
Riding Horse	42.97	46.99	51.18	30.40	32.13	33.33

Untuk meningkatkan kualitas hasil ekstraksi objek, nilai *threshold* pada *unknown region* ditentukan melalui *adaptive threshold* yang diperoleh dengan mengaplikasikan algoritma *Fuzzy C-Means* (FCM). Algoritma FCM dipilih dalam

penelitian ini Karena dalam penelitian sebelumnya menunjukkan hasil yang lebih *robust* dibandingkan algoritma Otsu untuk mendapatkan nilai *threshold* yang optimal. Sedangkan untuk menghaluskan obyek di sekitar daerah batas tepi digunakan filter *Discrete Cosine Transform* (DCT) – 2D. Dari 10 obyek yang digunakan dan dievaluasi dengan RMSE menunjukkan peningkatan rata-rata sebesar 31.55%.



Gambar 5.3. Perubahan nilai intensitas warna



Gambar 5.4. Grafik hasil ekstraksi *matte* antarclosed-form solution matting dengan peningkatan closed-form solution matting dengan DCT-2D

# **BAB 6**

## **PENELUSURAN OBJEK PADA *VIDEO SEQUENCES***

Ekstraksi bentuk informasi dari semantic objek dari data video merupakan pekerjaan yang sulit, karena informasi semantic tidak secara eksplisit disediakan dalam data video. Informasi semantic pada suatu objek dalam data video hanya dapat diidentifikasi oleh manusia, misalnya pemberian efek khusus dalam film dihasilkan oleh segmentasi manual. Teknik ini memerlukan waktu yang lama ketika diaplikasikan untuk segmentasi dengan data video yang memiliki volume besar. Beberapa algoritma telah dikembangkan dalam beberapa tahun terakhir untuk mengatasi permasalahan ini [56], yang diklasifikasikan menjadi 2 (dua) tipe, yaitu:

- Segmentasi otomatis

Segmentasi otomatis seperti yang dilakukan [4], [3], dan [5] dilakukan tanpa intervensi dari manusia dengan mempertimbangkan karakteristik khusus seperti fitur warna, tekstur, dan gerakan.

- Segmentasi semi otomatis

Segmentasi semi-otomatis dilakukan oleh [9], [7], dan [57] merupakan kombinasi antara segmentasi manual dengan segmentasi otomatis.

Karena tidak memiliki konstrain, permasalahan utama dalam segmentasi otomatis adalah kesulitan dalam memisahkan objek sesuai dengan keinginan user, oleh karena itu dalam penelitian ini metode segmentasi semi-otomatis yang

Mengkombinasikan segmentasi manual dan segmentasi otomatis diusulkan untuk ekstraksi objek data video.

### 6.1. Konstruksi *Keyframe*

Tahapan awal dalam proses ekstraksi objek video dilakukan dengan mengkonstruksi frame awal dalam video sequence menjadi *key frame*. Karena objek tidak memiliki informasi semantik, maka user mendefinisikan *trimap* sebagai konstrain yang berbasis *scribble interface*. Konstrain didefinisikan dengan *scribble* warna putih untuk merepresentasikan daerah *foreground* dan warna hitam untuk merepresentasikan daerah *background*. Ekstraksi *matte* dilakukan dengan teknik *matting* seperti yang telah dibahas dalam Bab 4.

### 6.2. Mekanisme Penelusuran Objek

Segmentasi objek pada frame selanjutnya dilakukan menggunakan mekanisme penelusuran berbasis transformasi temporal. Mekanisme *tracking* mengacu pada *keyframe* yang telah dikonstruksi sebelumnya. *Tracking object* dilakukan menggunakan *background subtraction* untuk mengidentifikasi perbedaan intensitas antara *current frame*. *Frame difference* merupakan teknik yang digunakan dalam *background subtraction* yang dilakukan secara rekursif. Model *background subtraction* (BF) diasumsikan sebagai nilai biner dari *foreground* dengan:

$$BF(x, y, n) = \begin{cases} 1, & \text{if } |I(x, y, n) - I(x, y, n-1)| \geq \alpha \\ 0, & \text{otherwise} \end{cases} \quad (6.1)$$

dengan  $\alpha$  adalah *threshold* untuk mengklasifikasi *foreground* dan *background*.

### 6.3. Pengujian *Background Subtraction* untuk Penelusuran Objek

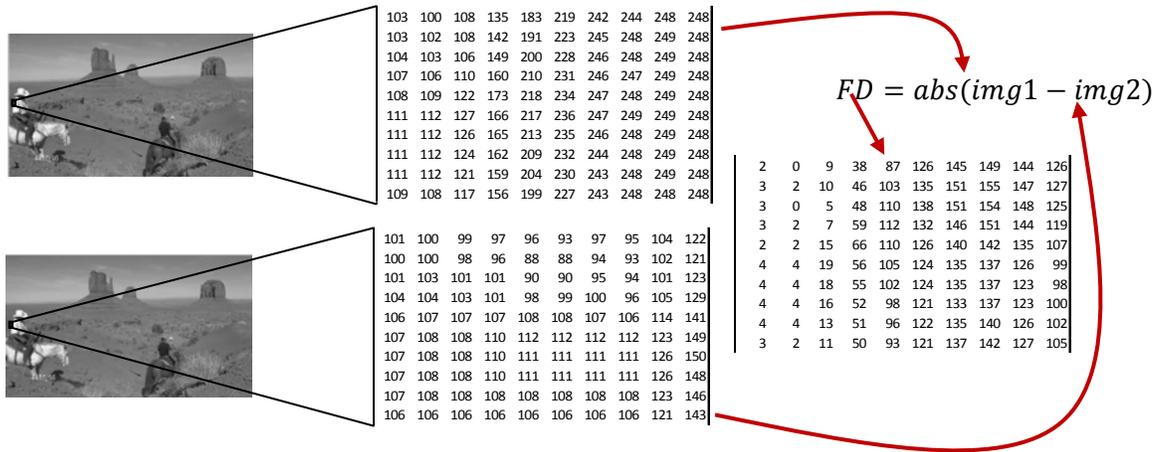
Eksperimen dilakukan dengan mengevaluasi tes *video standard* yang diperoleh dari *UCF Sport Action Dataset* yang secara detail disajikan dalam lampiran C. Tracking objek dilakukan padadua frame yang berurutan, frame awal disebut dengan *current frame* ( $I(x, y, n) - 1$ ) seperti diilustrasikan dalam gambar 6.1a, frame berikutnya disebut dengan *next frame* ( $I(x, y, n)$ ) disajikan dalam gambar 6.1b.



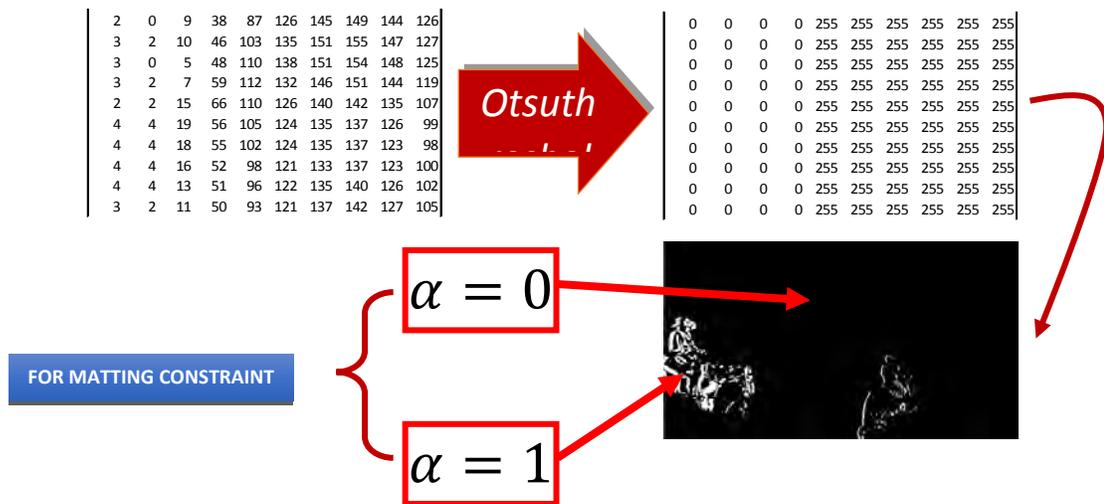
Gambar 6.1. Perbedaan *scene* pada *video sequences*. (a,b). *frame sequences*, (c,d). *grayscale image* dari *frame sequence*.  
Gambar diambil dari (*UCF Sport Action Dataset*)

Diasumsikan bahwa perbedaan *scene* antara *current frame* dan *next frame* dapat diperlakukan sebagai konstrain untuk ekstraksi objek. Sebelum diestimasi, frame yang berurutan ditransformasi menjadi *grayscale image* (Gambar 6.1c dan 6.1d). Nilai *frame difference* antara frame selanjutnya (Gambar 6.1b) dan frame sekarang

(Gambar 6.1a) yang dihitung menggunakan persamaan (6.1). Estimasi perbedaan intensitas dua frame disajikan dalam Gambar 6.2.

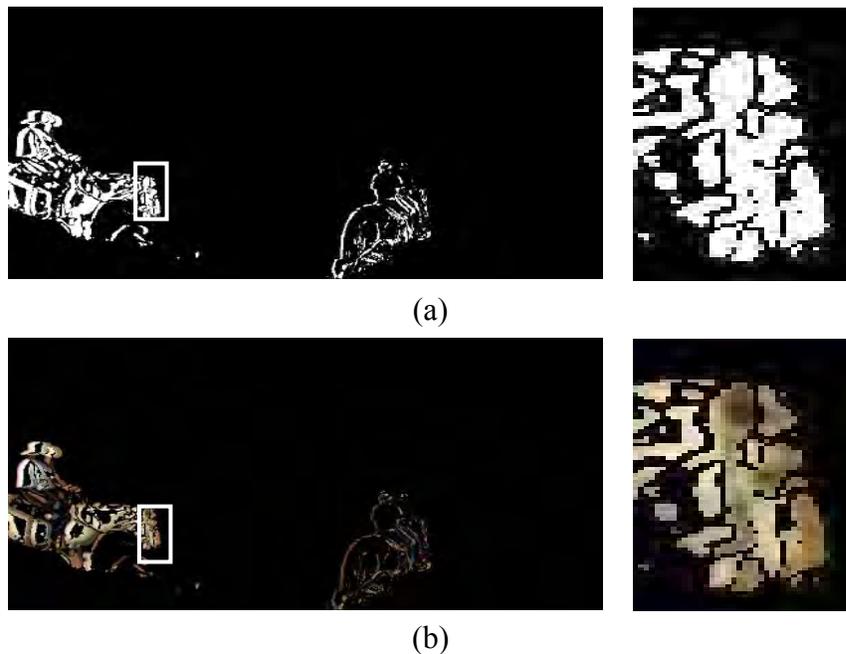


Gambar 6.2. Perbedaan intensitas dua frame



Gambar 6.3. Otsu thresholding

Setelah diperoleh konstrain dari *background subtraction*, ekstraksi objek dilakukan menggunakan *image matting* seperti yang telah dibahas dalam Bab 4. Gambar 6.4a menunjukkan hasil ekstraksi *matte* dengan konstrain yang diperoleh dari *frame difference*, sedangkan Gambar 6.4b menunjukkan hasil ekstraksi objek yang berkorelasi dengan *matte* hasil ekstraksi dalam Gambar 6.4a.

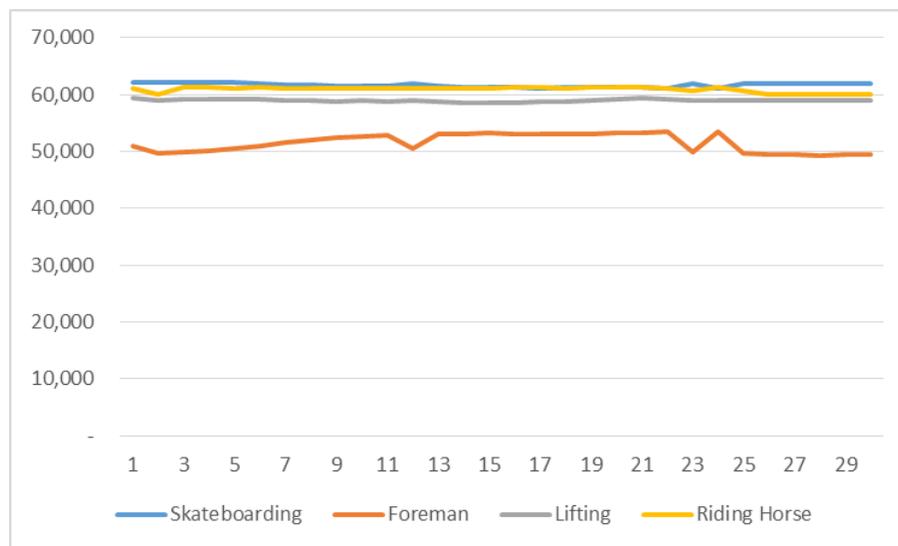


Gambar 6.4. Ekstraksi objek dengan konstrain *background subtraction*

Akurasi objek yang diekstraksi diukur menggunakan MSE (*Mean Squared Error*) seperti dalam persamaan (6.2) yang membandingkan antara *Grd.Truth* (hasil segmentasi manual) dengan *Ext.Obj* (hasil ekstraksi sistem yang diusulkan). Hasil ekstraksi dengan konstrain otomatis yang diperoleh dari *background subtraction* menunjukkan hasil yang kurang memuaskan, karena *matte* yang terekstrak terkena pengaruh efek morfologi (erosi dan dilasi) dan juga tidak memiliki distribusi warna

yang merata, sehingga ketika dikorelasikan dengan piksel dari input image terlihat kasar dan tidak natural.

$$MSE = \frac{\left( \sum_{i=1} \sum_{j=1} [Grd.Truth_{(i,j)} - Ext.Obj_{(i,j)}]^2 \right)}{MN} \quad (6.2)$$



Gambar 6.5. Grafik MSE evaluasi ekstraksi objek dengan *background subtraction*

#### 6.4. Kesimpulan

Ujicoba ini mengevaluasi standard video test yang diperoleh dari UCF Sport Action yang terdiri dari: *riding horse*, *lifting*, *skate boarding* dan *foreman* masing-masing 30 frame (disajikan dalam Lampiran C). Dalam percobaan ini frame awal dikonstruksi sebagai *keyframe* dengan konstrain didefinisikan oleh *user*. Konstrain

didefinisikan dengan *scribble* warna putih yang mewakili *definite foreground* ( $\alpha = 1$ ), dan *scribble* warna hitam mewakili *definite background* ( $\alpha = 0$ ).

Sedangkan frame-frame berikutnya konstan dikalkulasi dengan *background subtraction*. Setelah dievaluasi dengan membandingkan antara *matte* hasil ekstraksi sistem dengan segmentasi manual dengan MSE diperoleh rata-rata error setiap frame sebesar 28.291, artinya dengan frame berukuran 720 x 404 tingkat error piksel setiap frame berkisar 20.04%.

## BAB 7

# *MOTION VECTOR PREDICTION UNTUK ESTIMASI TEMPORAL CONSTRAINT*

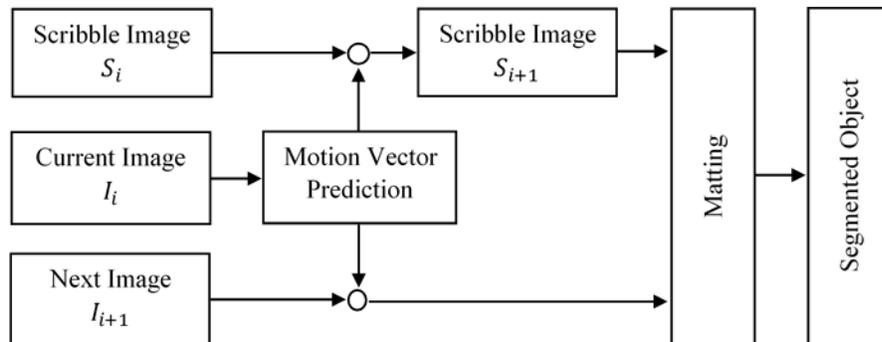
Ekstraksi *matte* untuk pemisahan objek pada ekstraksi objek video tidak hanya dilakukan dalam frame tunggal, tetapi juga dilakukan di seluruh frame dalam *video sequences*. Oleh karena itu, *trimap (user constraint)* sebagai inisialisasi *matting* tidak memungkinkan didefinisikan secara manual di seluruh frame.

Di lain sisi penempatan *constraint* yang secara langsung ditempatkan pada frame selanjutnya tanpa mempertimbangkan gerakan objek mengakibatkan *misclassification* dalam penempatan posisi *constraint*, dampaknya kualitas ekstraksi *matte* menurun dramatis. Diasumsikan bahwa pergerakan frame sekarang ke frame berikutnya memiliki koherensi secara temporal (*temporal coherence*), oleh karena itu pergerakan *scene* dari frame sekarang ke frame berikutnya merupakan jarak vektor, sehingga *constraint* pada frame berikutnya dapat diciptakan secara otomatis dengan memprediksi *motion vector*.

### **7.1. Blok Diagram Estimasi *Temporal Constraint***

Sistem ekstraksi objek seperti blok diagram pada Gambar 7.1, tahapan inisialisasi dilakukan dengan mendefinisikan *user constraint* untuk menentukan

area image yang mewakili daerah objek *foreground* dan *background*. *User constraint* direpresentasikan oleh *scribble image* ( $S_i$ ) berupa goresan manual dengan warna putih untuk mendefinisikan daerah objek *foreground* dan warna hitam untuk daerah *background*.



Gambar 7.1 Blok diagram ekstraksi objek video semi otomatis

**a. *Current image***

Data video pada dasarnya kumpulan beberapa image, oleh karena itu sebagai data input dalam sistem *video matting* data video dikonversi menjadi image. Image awal yang dijadikan data input dalam *video matting* disebut dengan *current image* ( $I_i$ ), yaitu image pada urutan ke- $i$  yang dijadikan referensi dalam image berikutnya.

**b. *Scribble image***

Salah satu permasalahan dalam ekstraksi *matte* adalah permasalahan *ill-posed*, sehingga proses ekstraksi memerlukan *constraint* sebagai inisialisasi untuk membedakan *region* yang mewakili daerah objek *foreground* dan *background*. *Scribble image* ( $S_i$ ) adalah coretan yang didefinisikan oleh user

dengan mempertimbangkan semantik dari image. Coretan diposisikan pada daerah yang merepresentasikan dan berkorelasi dengan piksel *foreground* dan *background*. *Scribble image* direpresentasikan dalam bentuk coretan manual warna putih untuk mendefinisikan daerah objek *foreground* dan warna hitam untuk daerah *background*. Pada *next image* ( $I_{i+1}$ ), *scribble image* ( $S_{i+1}$ ) adalah *constraint* yang didefinisikan secara otomatis dengan memprediksi jarak vektor.

**c. *Next image***

Image yang diestimasi agar dapat diperoleh jarak antara dua piksel dalam yang berkorelasi melibatkan dua image yang berurutan, yaitu *current image* ( $I_i$ ) dan *next image* ( $I_{i+1}$ ). *Next image* adalah *image sequence* yang posisinya berada persis di belakang *current image* ( $I_i$ ).

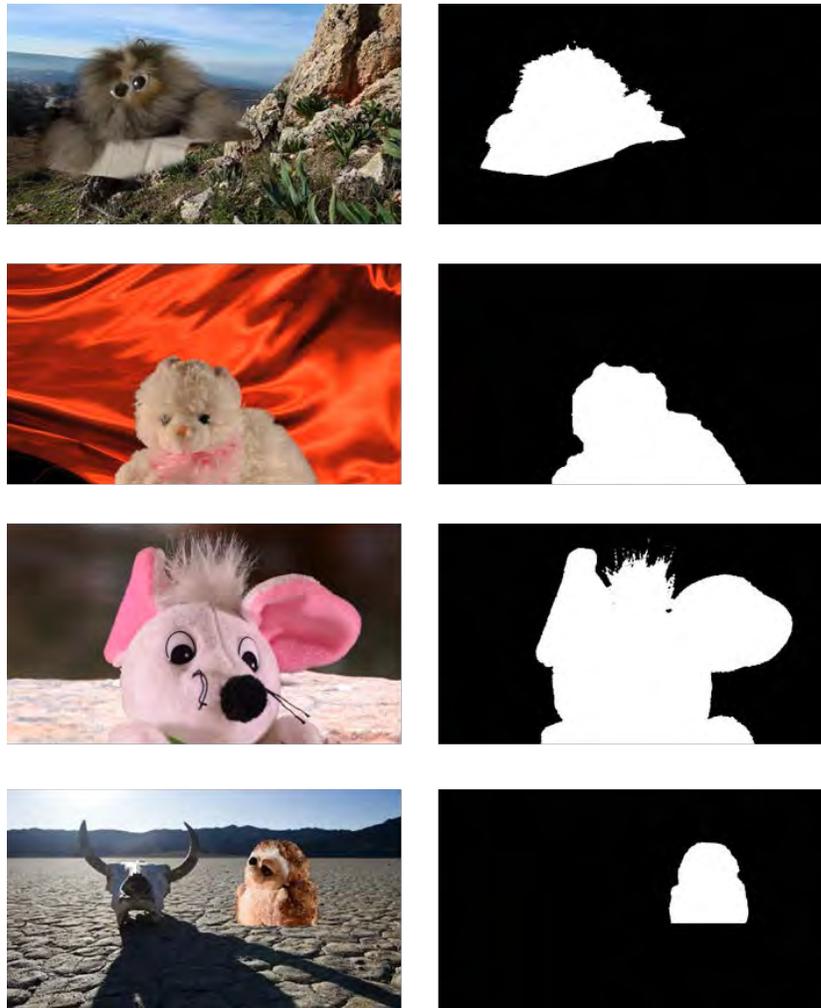
**d. *Motion vector prediction***

Hasil ekstraksi objek pada *next image* ( $I_{i+1}$ ) sangat tergantung pada bagaimana *constraint* (*scribble*) ditempatkan pada *next image*. Posisi *scribble* yang berada di luar daerah yang diwakili mengakibatkan akurasi ekstraksi *matte* akan menurun secara dramatis. Sementara itu jika perpindahan *constraint* dari ( $I_i$ ) ke ( $I_{i+1}$ ) ditempatkan pada koordinat yang sama kemungkinan besar akan mengalami *misclassification*, karena posisi *constraint* tetap sedangkan scene mengalami pergerakan. Pendefinisian *scribble* secara manual tidak memungkinkan untuk dilakukan di seluruh image, sehingga diperlukan *scribble* otomatis pada *next image*. Untuk

mendapatkan *scribble* otomatis dapat diperoleh dengan menggerakkan posisi piksel *scribble* dari *current image* ( $I_i$ ) ke *next image* ( $I_{i+1}$ ). Karena *scribble* sangat berkorelasi dengan piksel image, maka diasumsikan bahwa pergerakan *scribble* mengikuti pergerakan piksel image. Oleh karena itu pergerakan *scribble* dapat dikalkulasi menggunakan *motion vector* menggunakan algoritma *block matching*.

## 7.2. Pengujian *Temporal Constraint*

Uji coba untuk estimasi temporal *constraint* berbasis algoritma block matching dilakukan menggunakan *video matting* dataset **artem**, **alex**, **dmitriy** dan **vitaly** masing-masing 100 frame dengan visualisasi seperti Gambar 7.2.

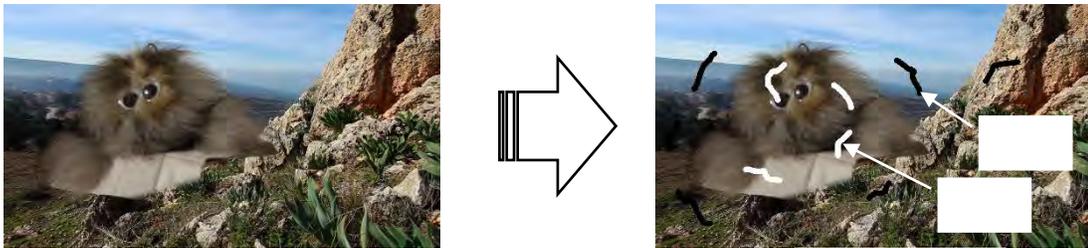


Gambar 7.2 Dataset *video matting*  
Gambar diambil dari (*video matting dataset*)

Image berwarna merupakan *raw image* yang ditangkap oleh kamera dan diperlakukan sebagai data yang akan diproses dalam *video matting*. Sedangkan *image biner* (hitam putih) merupakan *ground truth* atau hasil pemisahan objek secara manual berdasarkan persepsi atau penglihatan manusia yang mempertimbangkan *video context*. *Ground truth* juga diperoleh dari dataset *video matting*.

### 7.2.1. *Manual constraint*

Untuk mengatasi permasalahan *ill-posed*, proses ekstraksi diawali dengan mendefinisikan *constraint* secara manual pada *current image* ( $I_i$ ). Pembuatan *constraint* berupa *scribble* diilustrasikan pada dataset “**artem**” seperti ditunjukkan dalam Gambar 7.3.



Gambar 7.3 Pendefinisian konstrain manual

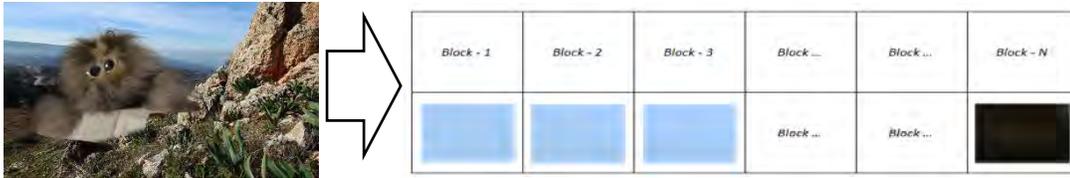
Gambar diambil dari (*video matting dataset*)

Daerah *foreground* direpresentasikan dalam coretan warna putih dengan  $constraint_{\alpha} = 1$  berkorelasi dengan piksel *foreground* dan coretan warna hitam merepresentasikan region *background* dengan  $constraint_{\alpha} = 0$ .

### 7.2.2. Estimasi *Temporal Constraint* dengan *Exhaustive Search*

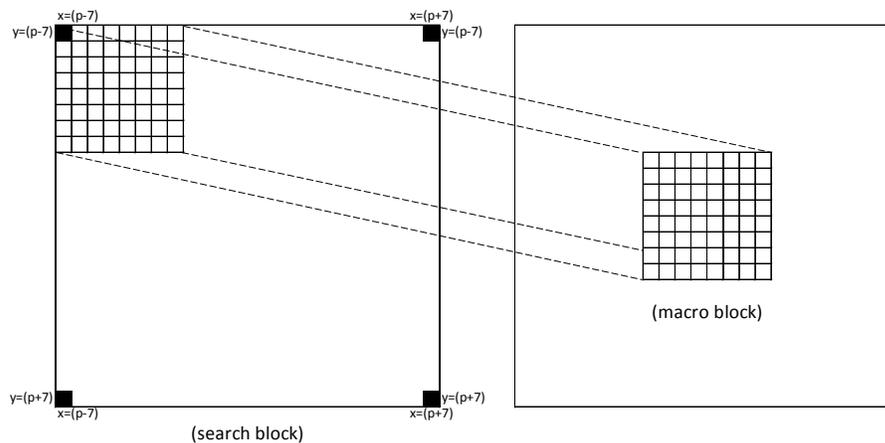
Tahapan awal untuk menggerakkan *scribble* dari *current image* ke *next image* dilakukan dengan memprediksi *motion vector*. Perhitungan dilakukan dengan menciptakan *macro block* pada *next image* dan *search block* pada *current image*. Penciptaan *macro block* dilakukan dengan membagi image dalam bentuk blok bujur sangkar berukuran 8 x 8 piksel [13] di seluruh image. Dalam membentuk *macro block*, iterasi dimulai dari kiri ke kanan, dari atas ke bawah dan setiap delapan kali iterasi

membentuk satu macro blok hingga iterasi berakhir ketika berada pada akhir posisi piksel (seperti dalam Gambar 7.4).



Gambar 7.4. *Macro block pada next image*

Untuk melakukan pencocokan, setiap *macro block* yang dihasilkan dibandingkan dengan blok yang berukuran sama pada *current image* (*search block*). *Macro block* dibandingkan dengan beberapa *search block* dengan batasan ‘ $x=p-7$ ,  $y=p-7$ ’ hingga ‘ $x=p+7$ ,  $y=p+7$ ’ dengan iterasi dimulai dari kiri ke kanan dan dari atas ke bawah seperti dalam Gambar 7.5.



Gambar 7.5 Proses pencocokan blok

Penentuan blok yang dipertimbangkan sebagai arah vektor dilakukan dengan menghitung *Peak Signal Noise to Ratio (PSNR)* seperti persamaan (7.1).

$$PSNR = 10 \log_{10} \left[ \frac{(P)^2}{MSE} \right] \quad (7.1)$$

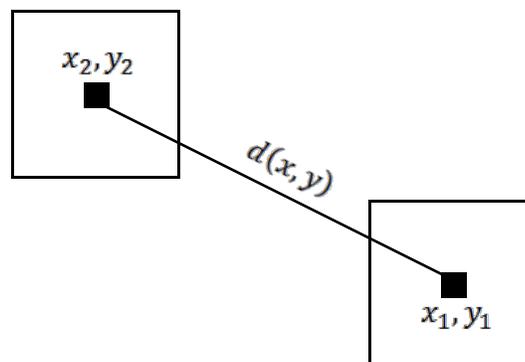
dimana  $P$  merupakan nilai tertinggi dari intensitas piksel dengan

$$MSE = \frac{1}{B^2} \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} (NI_{ij} - CI_{ij})^2 \quad (7.2)$$

Dengan  $B$  adalah ukuran blok pada  $NI_{ij}$  (*next image*) dan  $CI_{ij}$  (*current*). Hasil pencocokan dengan nilai PSNR yang tertinggi merupakan blok yang memiliki kemiripan tertinggi, sehingga pusat koordinat *search block* dengan pusat koordinat pada *macro block* seperti Gambar 7.6 dapat digunakan untuk menghitung jarak vektor  $d(x, y)$  seperti dinotasikan sebagai berikut :

$$d(x, y) = (x_1 - x_2), (y_1 - y_2) \quad (7.3)$$

Dengan  $(x_1, y_1)$  merupakan pusat koordinat *macro block* sedangkan  $(x_2, y_2)$  adalah pusat koordinat *search block* yang memiliki nilai PSNR yang tertinggi.



Gambar 7.6 Titik pusat perhitungan jarak

Cara kerja algoritma block matching untuk menggerakkan *constraint* dilakukan dengan langkah-langkah sebagai berikut :

1. Membagi *next image* menjadi blok-blok (*macro block*) berukuran  $N \times N$  (bujur sangkar) dalam hal ini  $8 \times 8$  piksel. Jika dalam eksperimen ini menggunakan video yang disetiap frame berukuran  $1080 \times 1920$  piksel, maka jumlah *macro block* ( $MB$ ) yang dihasilkan sebesar:

$$MB = br \times bc$$

dengan  $br$  adalah jumlah blok pada baris, sedangkan  $bc$  adalah jumlah blok pada kolom yang diperoleh dengan :

$$br = row / N, \text{ dan}$$

$$bc = col / N$$

dimana  $row$  adalah ukuran baris (dalam piksel),  $col$  adalah ukuran kolom, dan  $N$  adalah ukuran setiap *macro block*. Sehingga jika image berukuran  $1080 \times 1920$  piksel dan setiap macro blok  $N \times N$  berukuran  $8 \times 8$  piksel akan diperoleh jumlah *macro block* sebanyak:

$$br = row / N \Rightarrow br = 1080 / 8 \Rightarrow br = 135$$

$$bc = col / N \Rightarrow bc = 1920 / 8 \Rightarrow bc = 240$$

$$MB = br \times bc \Rightarrow MB = 135 \times 240 \Rightarrow MB = 32400$$

2. Mencocokkan *macro block* dengan *search block* mulai dari 'x=p-7, y=p-7' hingga 'x=p+7, y=p+7' dan setiap proses iterasi dikalkulasi dengan persamaan (MSE) dan (PSNR). Nilai PSNR yang terbesar merupakan block yang dianggap paling cocok sehingga jarak kedua blok itu dianggap sebagai *motion*

*vector*. Nilai perhitungan *motion vector* video artem dari *current image* ke *next image* direpresentasikan dalam Tabel 7.1.

Tabel 7.1 Hasil *motion vector* pada dua *frame sequence*

Blok	<i>Motion vector</i> sumbu X ( $MV_x$ )	<i>Motion vector</i> sumbu Y ( $MV_y$ )
1	1	1
2	0	-1
3	0	5
4	0	2
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
32400	-7	0

3. Nilai *motion vector* pada block yang berkorelasi dengan *constraintscribble* ditambahkan pada posisi koordinat piksel *scribble* dengan

$$NF(i + J_{mx}, j + J_{my}) = CF(i, j, D),$$

dengan

$$J_{mx} = (0 \dots MV_x) \text{ dan } J_{my} = (0 \dots MV_y) \quad (7.4)$$

dimana  $(i, j)$  adalah posisi koordinat piksel pada image,  $(D)$  adalah ruang warna,  $(NF)$  adalah frame berikutnya,  $(MV_x, MV_y)$  adalah nilai *motion vector*.

### 7.3. Permasalahan dalam *Exhaustive Search*

Dari Tabel 7.1 terlihat jelas bahwa dalam blok yang berdekatan memiliki nilai *motion vector* yang berbeda, sehingga apabila *constraint* yang didefinisikan saling berpotongan diantara dua blok atau lebih dan dipropagasi dengan nilai *motion vector* tersebut mengakibatkan piksel *constraint* pada *next image* akan menyebar dan saling menjauh, sehingga kualitas ekstraksi *matte* akan menurun secara dramatis.

### 7.4. Perbaikan Algoritma *Block Matching*

Untuk mengatasi permasalahan ini *block matching* hanya diestimasi pada *region* (dalam bentuk persegi panjang) yang berkorelasi dengan *constraint*, karena blok yang mempengaruhi pergerakan *constraint* secara temporal (dari *current image* ke *next image*) hanya blok yang berkorelasi erat dengan piksel *constraint*, sehingga estimasi *block matching* untuk menghasilkan *motion vector* hanya dilakukan pada blok yang berkorelasi erat dengan *constraint pixel*.

Blok dibuat persegi panjang dengan ukuran yang adaptif (antara  $1 - \infty$ ) menyesuaikan dengan ukuran *constraint*. Proses pencocokan blok pada *current image* dan *next image* dimulai dari  $'x=p-7, y=p-7'$  hingga  $'x=p+7, y=p+7'$  dengan *cost function*

$$f(u, v) = \sum_{w, xy} (CF(x+u, y+x) - PF(x+u, y+x)) \quad (7.5)$$

dengan  $u, v$  adalah nilai *motion vector* dengan  $w$  adalah blok yang adaptif dengan bobot berukuran 1 sampai  $\infty$  dan  $x, y$  adalah posisi koordinat piksel.

### 7.5. Ruang Warna HSV

Awalnya, perbaikan *block matching* dilakukan dalam ruang warna RGB. Jarak dua koordinat piksel antara *current image* dan *next image* dihitung berdasarkan asumsi bahwa piksel yang bergerak adalah piksel yang memiliki nilai intensitas yang sama atau yang mendekati sama. Sementara nilai intensitas piksel pada objek yang bergerak dari *current image* ke *next image* sangat dimungkinkan mengalami perubahan dramatis yang diakibatkan karena pengaruh *illuminance*.

Untuk mengatasi hal ini operasi pencocokan matrik *current image* dan *next image* diaplikasikan pada ruang warna *Hue, Saturation, Value* (HSV). *Motion vector*  $d(p_1, p_2)$  antara *current image* dan *next image* digunakan untuk menggerakkan *constraint* yang dihitung dengan *Euclidian Distance*. Asumsinya perbedaan terkecil sudut *hue* ( $h$ ) antara *current image* dan *next image* dipertimbangkan sebagai jarak vektor yang diaplikasikan pada model warna HSV[14].

Nilai perbedaan ini dihitung dari blok matrik pada *current image* yang berkorelasi dengan *scribble* dan blok matri yang menempati koordinat yang sama dengan *current image* pada *next image*.

$$d(p_1, p_2) = ((v_1 s_1 x_1 - v_2 s_2 x_2)^2 + (v_1 s_1 y_1 - v_2 s_2 y_2)^2 + (v_1 - v_2)^2)^{1/2} \quad (7.6)$$

dengan  $s, v$  nilai intensitas saturation dan value dari model warna HSV, sedangkan  $x$  adalah  $\cos H$  dan  $y$  adalah  $\sin H$  dengan

$$H = 2\pi h \quad (7.7)$$

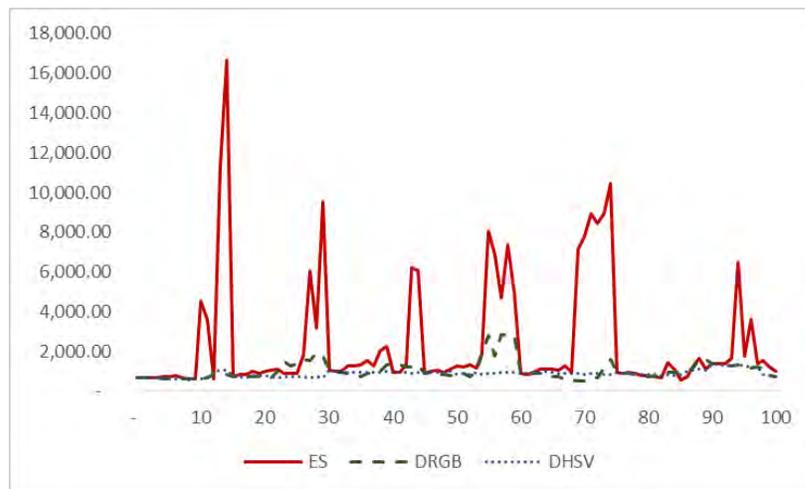
dimana  $h$  adalah nilai hue pada model warna HSV.

### 7.6. Evaluasi Estimasi *Temporal Constraint*

Eksperimen dilakukan pada empat dataset *videomattng*, yaitu: **artem**, **alex**, **dmitriy**, dan **vitaliy** masing-masing 100 frame. Ekstraksi *matte* dilakukan dengan *image matting*[11], [52] di setiap frame seperti yang dilakukan dalam Bab 4 dan 5. Hasil ekstraksi *matte* dievaluasi dengan menghitung perbedaan antara *matte* yang terekstraksi di setiap image dengan *ground truth*[14] dengan formulasi seperti dalam persamaan (7.8).

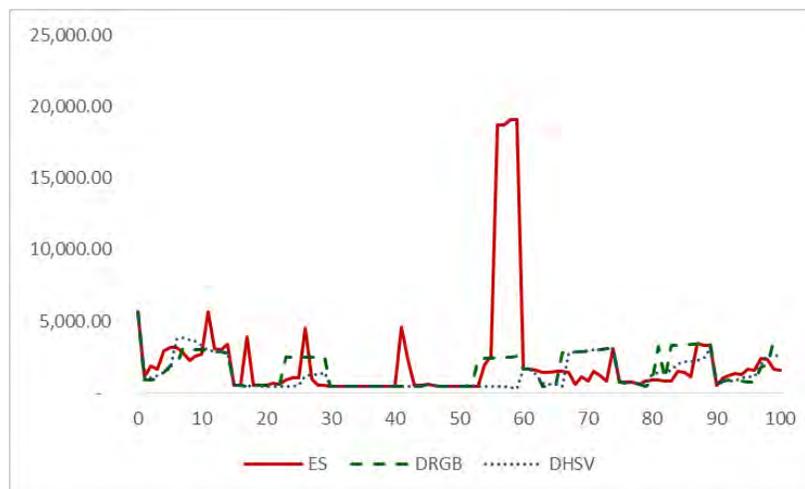
$$error = \frac{abs(GrdImg - MatteExt)^2}{M \times N} \quad (7.8)$$

*GrdImg* merupakan groundtruth sebagai *matte* referensi pembanding dari *MatteExt* yaitu *matte* yang diekstraksi dengan algoritma yang diusulkan. Sedangkan  $M \times N$  adalah ukuran dari image yang diuji.

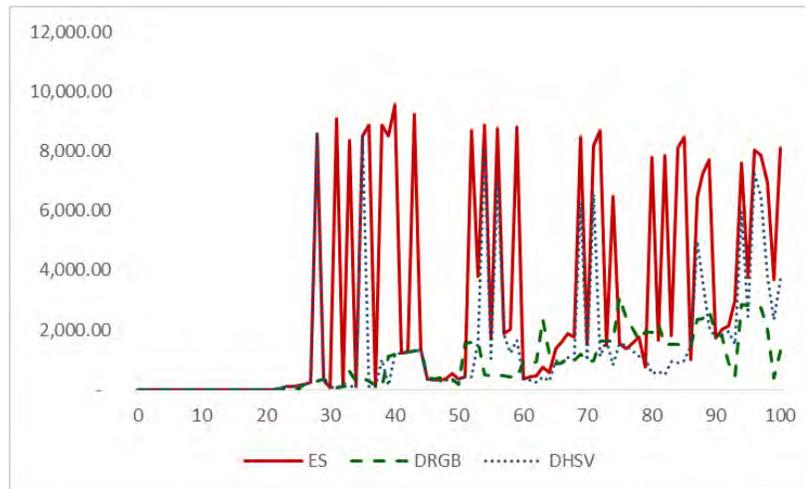


Gambar 7.7 Evaluasi ekstraksi objek dataset **artem**

Pada eksperimen awal, untuk estimasi *constraint* dilakukan menggunakan *block matching* dengan algoritma *exhaustive search*[13]. Evaluasi yang dilakukan disajikan dalam perbandingan kuantitatif dan perbandingan visual. Perbandingan kuantitatif disajikan dalam bentuk angka yang disajikan secara detail dalam lampiran 7.2 dan grafik yang diilustrasikan dalam Gambar 7.7 hingga Gambar 7.10.

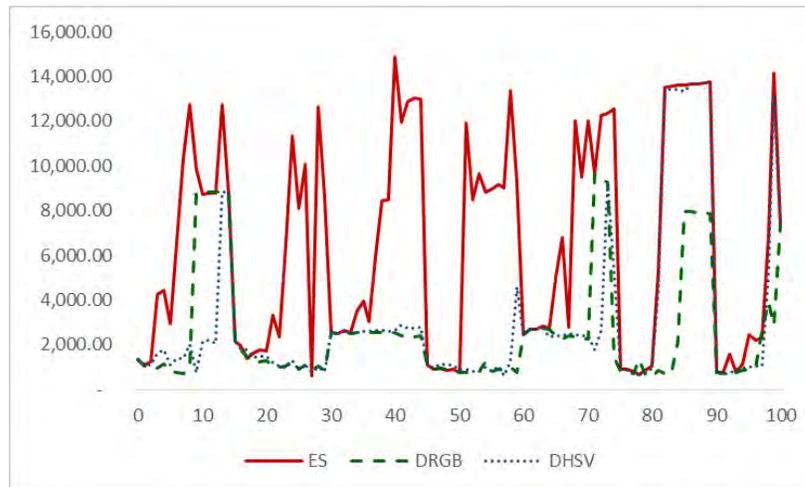


Gambar 7.8 Evaluasi ekstraksi objek dataset **alex**



Gambar 7.9 Evaluasi ekstraksi objek dataset **dmitriy**

Evaluasi dilakukan dengan membandingkan tiga algoritma yaitu *exhaustive search* (ES) pada ruang warna RGB, *sum of absolute difference* (SAD) dalam ruang warna (DRGB) dan SAD dalam ruang warna HSV (DHSV). Dari grafik terlihat bahwa dengan menerapkan blok adaptif dengan estimasi *cost function* SAD untuk menemukan *best finding* dalam algoritma *block matching* dapat memperbaiki algoritma *exhaustive search*. Peningkatan perbaikan dibuktikan dengan penurunan error di setiap frame disajikan dalam Tabel 7.2.



Gambar 7.10 Evaluasi ekstraksi objek dataset **vitaliy**

Tabel 7.2 Rata-rata perbedaan piksel dataset

Dataset	Rata-rata error		
	ES	DRGB	DHSV
<i>Artem</i>	2,433.60	1,050.40	885.82
<i>Alex</i>	2,148.53	1,644.22	1,334.38
<i>Vitaliy</i>	6,422.57	2,565.87	3,080.28
<i>Dmitriy</i>	3,029.50	872.94	1,350.83

Perbandingan visual disajikan dalam perbandingan kualitas *constraint* yang ditunjukkan dalam Gambar 7.11 perbandingan kualitas *matte* dengan groundtruth diilustrasikan dalam Gambar 7.12, dan perbandingan kualitas hasil segmentasi digambarkan dalam Gambar 7.13.

## 7.7. Kesimpulan

*Video matting* memiliki tantangan pada dua aspek, yang pertama adalah bagaimana objek dalam domain spasial dapat disegmentasi dengan intervensi user yang efisien. Aspek kedua adalah bagaimana mempropagasi *constraint* untuk

segmentasi objek pada frame selanjutnya. Awalnya proses ekstraksi dalam domain spasial dilakukan dengan *image matting*, algoritma ini melibatkan intervensi user untuk menyediakan trimap sebagai *constraint* untuk inisialisasi dalam ekstraksi *matte* yang selanjutnya digunakan untuk menarik region yang memiliki piksel yang berkorelasi. Hasilnya cukup ekspresif, namun tidak memungkinkan jika diaplikasikan dalam beberapa frame berurutan.

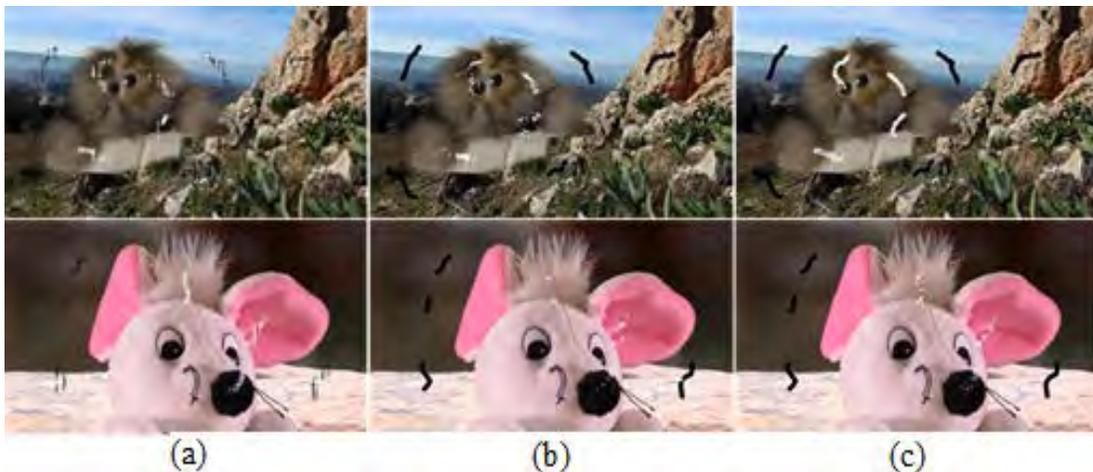
Untuk mengatasi permasalahan ini, algoritma *block matching* diusulkan untuk estimasi gerakan. Estimasi dilakukan untuk menentukan jarak vektor di antara dua blok yang berkorelasi, sehingga hasilnya dapat digunakan untuk propagasi *constraint*. *Constraint* yang dipropagasi dari frame sekarang ke frame selanjutnya untuk digunakan sebagai inisialisasi proses ekstraksi *matte* seperti yang dilakukan dalam domain spasial. Untuk menentukan dua blok yang cocok di antara dua frame yang berurutan awalnya digunakan algoritma *exhaustive search*, namun karena estimasi dalam *exhaustive search* dilakukan diseluruh blok, maka propagasi *constraint* menjadi tidak akurat.

Hal ini disebabkan hasil estimasi di antara blok yang berdekatan memiliki nilai *motion vector* yang berbeda, sehingga ketika digunakan untuk propagasi, piksel *constraint* saling menjauh dan dampaknya akurasi ekstraksi menurun secara dramatis. Perbaikan algoritma dilakukan dengan menciptakan blok yang memiliki ukuran dinamis terhadap area *scribble*. Asumsinya hanya pergerakan piksel yang berkorelasi dengan area *scribble* yang mempengaruhi propagasi *constraint*.

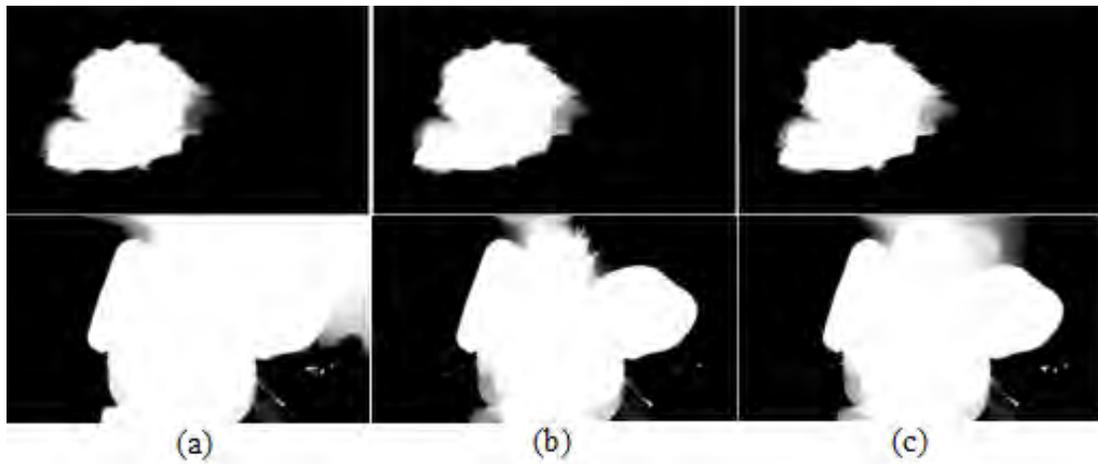
*Sum of absolute difference* (SAD) digunakan untuk menentukan blok matrik yang memiliki tingkat kecocokan tertinggi. Nilai SAD yang terkecil diperlakukan

sebagai blok yang paling cocok dan arah pergerakannya dihitung sebagai *motion vector* dan digunakan untuk propagasi *constraint*. Operasi ini dilakukan dalam ruang warna RGB dan HSV.

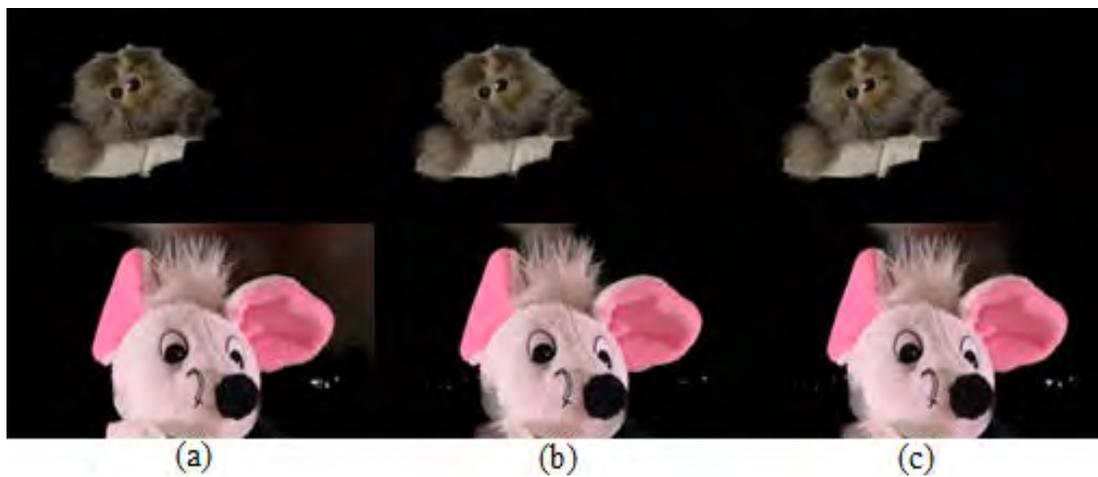
Hasil perbandingan dengan persamaan (7.8) menunjukkan bahwa dengan blok dinamis yang diestimasi dengan SAD pada ruang warna RGB mengalami peningkatan akurasi sebesar 56.30%, sedangkan apabila diaplikasikan dalam ruang warna HSV akan meningkatkan akurasi sebesar 52.61%. Ini membuktikan bahwa dengan SAD diestimasi pada ruang warna RGB maupun HSV lebih akurat dibandingkan dengan algoritma *exhaustive search*.



Gambar 7.11. Hasil propagasi konstrain dengan *motion vector* menggunakan (a). ES, (b). DRGB, (c). DHSV  
Gambar diambil dari (*video matting dataset*)



Gambar 7.12. Hasil ekstraksi *matte* dengan konstrain menggunakan (a). ES, (b). DRGB, (c). DHSV



Gambar 7.13 Hasil ekstraksi objek dengan konstrain menggunakan (a). ES, (b). DRGB, (c). DHSV

# BAB 8

## KESIMPULAN DAN RENCANA SELANJUTNYA

### 8.1. Kesimpulan

Penelitian ekstraksi objek video semi-otomatis menggunakan *alpha Matting* berbasis *motion estimation* telah dilakukan. Serangkaian uji coba dilakukan baik frame tunggal maupun frame berurutan. Pada frame tunggal proses ekstraksi diawali dengan inisialisasi *scribble* (coretan manual) yang mendefinisikan daerah yang merepresentasikan *definite foreground* dan *definite background*. Sedangkan *unknown region* merupakan campuran piksel *foreground* dan *background* merupakan daerah yang diestimasi.

Secara keseluruhan, tahapan penelitian terdiri dari estimasi *adaptive threshold* untuk *alpha* dalam *image matting*, perbaikan kualitas *matting* dalam image terkompresi serta estimasi *temporal constraint* untuk propagasi *scribble* dengan memprediksi *motion vector*. Framework yang diusulkan dengan estimasi *adaptive threshold* menggunakan FCM memberikan kontribusi peningkatan akurasi sebesar 11.05%. Perbaikan kualitas *matting* pada image terkompresi menggunakan DCT-2D untuk mereduksi *noise* memberikan kontribusi peningkatan akurasi sebesar 31.55%. Sedangkan estimasi *temporal constraint* kami prediksi dengan *motion vector* meningkatkan akurasi sebesar 56.30% pada data dalam ruang warna RGB, dan 52.61% dalam ruang warna HSV. Framework ini selanjutnya didefinisikan sebagai **KiMoHar**.

Peningkatan akurasi diestimasi dalam tiga tahapan. Pertama, *alpha* yang bernilai antara 0 dan 1 yang nilainya diestimasi dengan *adaptive threshold*. Estimasi dilakukan dengan mengaplikasikan algoritma FCM dan Otsu. Evaluasi dilakukan dengan menghitung perbedaan piksel antara *ground truth* (hasil yang direferensikan dalam dataset) dengan *matte* hasil ekstraksi. Dengan menggunakan MSE, FCM dapat menurunkan perbedaan piksel di setiap frame dari 30.325,10 menjadi 26.999,33, sedangkan dengan Otsu menjadi 28.921,70.

Kedua, pengujian juga dilakukan dengan input image terkompresi. Kualitas *matte* menurun akibat dari perubahan intensitas di setiap piksel. Perbaikan dilakukan dengan mereduksi noise di sekitar batas tepi menggunakan DCT-2D. Evaluasi dilakukan dengan estimasi RMSE dan hasilnya menurunkan RMSE dari 16.68 menjadi 11.44.

Ketiga, ekstraksi objek pada frame berikutnya dilakukan dengan estimasi *temporal constraint* untuk propagasi *scribble*. Tujuannya agar ekstraksi objek yang dilakukan menggunakan *image matting* pada frame selanjutnya diinisialisasi dengan konstrain otomatis. Estimasi *temporal constraint* dilakukan dengan memprediksi nilai *motion vector* menggunakan algoritma *exhaustive search* (ES) pada *block matching*. Secara visual propagasi *scribble* dalam beberapa pergerakan frame tidak akurat sehingga hasil kualitas ekstraksi *matte* menurun secara dramatis.

Dalam estimasi *temporal constraint* kami, kualitas ekstraksi *matte* diperbaiki dengan mendefinisikan blok matrik yang berukuran dinamis terhadap ukuran *scribble*, dan *motion vector* ditentukan dengan estimasi SAD antara frame sekarang dan frame berikutnya. Pengujian dataset *video matting* dilakukan dalam ruang warna

RGB dan HSV. Dari pengujian yang dilakukan terhadap empat video yang masing-masing terdiri dari 100 frame menunjukkan bahwa dalam ruang warna RGB, rata-rata perbedaan piksel di setiap frame menurun dari 3.058,55 menjadi 1.533,35, sedangkan dalam ruang waktu HSV menjadi 1.662,83.

## **8.2. Rencana Selanjutnya**

Sebagai *future work* terkait dengan penelitian ekstraksi objek video berbasis *motion estimation* dapat dikembangkan pada video yang memiliki fitur lebih kompleks seperti video dengan *multiple object*, pergerakan dengan oklusi maupun video yang memiliki homogenitas warna antara objek dan *background*.

## DAFTAR PUSTAKA

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H . 264 / AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.
- [2] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [3] H. Li and K. Ngan, "Automatic Video Segmentation and Tracking for Content-Based Applications," *Adv. Vis. Content Anal. Adapt. Multimed. Commun.*, pp. 27–33, 2007.
- [4] Y. Tsaig and A. Averbuch, "Automatic Segmentation of Moving Objects in Video Sequences: A Region Labeling Approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 597–612, 2002.
- [5] T. Meier and K. Ngan, "Automatic Segmentation of Moving Objects for Video Object Plane Generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 525–538, 1998.
- [6] A. Bovik, *The Hand Book of Image and Video Processing*. Academic Press, 1998.
- [7] M. Hariadi, H. Loy, and T. AOKI, "Semi-Automatic Video Object Segmentation using LVQ with Color and Spatial Features," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 7, pp. 1553–1560, 2005.
- [8] R. S. Basuki, M. A. Soeleman, M. Hariadi, M. . Purnomo, R. Pramunendar, and A. F. Yogananti, "Spectral-Based Video Object Segmentation using Alpha Matting and Background Subtraction," in *IIEEJ the 4th International Workshop on Image Electronics and Visual Computing, 7-10 October, 2014*.
- [9] C. Toklu, A. Tekalp, and A. . Erdem, "Semi-Automatic Video Object Segmentation in the Presence of Occlusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 4, pp. 624–629, 2000.
- [10] T. Porter and T. Duff, "Compositing digital images," *Comput. Graph. (ACM)*, vol. 18, no. 3, pp. 253–259, 1984.

- [11] A. Levin, D. Lischinski, and Y. Weiss, “A Closed-Form Solution to Natural Image Matting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 1–15, 2008.
- [12] A. Levin, A. Rav-acha, and D. Lischinski, “Spectral Matting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1699–1712, 2008.
- [13] A. Barjatya, “Block Matching Algorithms For Motion Estimation,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, 2004.
- [14] R. S. Basuki, M. Hariadi, E. M. Yuniarno, and M. H. Purnomo, “Spectral-Based Temporal-Constraint Estimation for Semi-Automatic Video Object Segmentation,” *Int. Rev. Comput. Softw.*, vol. 10, no. September, pp. 959–965, 2015.
- [15] L. Shuhua and G. Gaizhi, “The Application of Improved HSV Color Space Model in Image Processing,” in *International Conference on Future Computer and Communication*, 2010, pp. 10–13.
- [16] B. Kiger, “‘Videography’ What does it all mean?,” *American Cinematographer*, pp. 1114–1174, 1972.
- [17] L. I-Jong and S. . Kung, *Video Object Extraction and Representation: Theory and Application*. New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publisher, 2002.
- [18] H. Jiang, G. Zhang, H. Wang, and H. Bao, “Spatio-Temporal Video Segmentation of Static Scenes and Its Applications,” *IEEE Trans. Multimed.*, vol. 17, no. 1, pp. 3–15, 2015.
- [19] J. Wang and M. F. Cohen, “Image and Video Matting: A Survey,” *J. Found. Trends Comput. Graph. Vis.*, vol. 3, no. 2, pp. 97–175, 2007.
- [20] Y. Chuang, “New Models and Methods for Matting and Compositing,” University of Washington, 2004.
- [21] N. M. David Biedny, Bert Monroy, *Photoshop Channel CHOPS*, 1st ed. New Riders Publishing, 1998.
- [22] M. Ruzon and C. Tomasi, “Alpha estimation in natural images,” in

- Proceedings of IEEE CVPR*, 2000, pp. 18–25.
- [23] G. Wyszecki and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd ed. New York: John Wiley and Sons, 2000.
- [24] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, “A Bayesian Approach to Digital Matting,” in *Proceedings of IEEE CVPR*, 2001, pp. 264–271.
- [25] J. Wang and M. Cohen, “An Iterative Optimization Approach for Unified Image Segmentation and Matting,” in *Proceedings of ICCV*, 2005, pp. 936–943.
- [26] X. Bai and G. Sapiro, “A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting,” in *IEEE 11th International Conference on Computer Vision, 14-21 October 2007*, 2007, pp. 1–8.
- [27] C. Yang, R. Duraiswami, N. Duraiswami, and L. Davis, “Improved Fast Gauss Transform and Efficient Kernel Density Estimation,” in *Proceedings. Ninth IEEE International Conference on Computer Vision*, 2003, pp. 664–671.
- [28] Y. Mishima, “Soft Edge Chroma-key Ceneration Based Upon Hexoctahedral Color Space,” 5,355,174, 1993.
- [29] “Knockout user guide,” 2002.
- [30] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng, “Easy matting,” in *Proc. of Eurographics*, 2006.
- [31] J. Wang and M. Cohen, “Optimized Color Sampling for Robust Matting,” in *Proc. of IEEE CVPR*, 2007.
- [32] J. Sun, J. Jia, T. C.-K, and H.-Y. Shum, “Poisson Matting,” in *Proceedings of ACM SIGGRAPH*, 2004, pp. 315–321.
- [33] C. Rother, R. Kolmogorov, and A. Blake, “Grabcut - interactive foreground extraction using iterated graph cut,” in *Proceedings of ACM SIGGRAPH*, 2004, pp. 309–314.
- [34] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, “Random walks for interactive alpha-matting,” *Proc. VIIP*, vol. 2005, pp. 423–429, 2005.
- [35] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition

- using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.
- [36] J. S. . Li and H. Shum, “Video object cut and paste,” in *Proc. ACM SIGGRAPH*, 2005, pp. 595–600.
- [37] P. Villegas and X. Marichal, “Perceptually-weighted Evaluation Criteria for Segmentation Masks in Video Sequences,” *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1092–1103, 2004.
- [38] Y. Chuang, A. Agarwala, B. Curless, D. Salesin, and R. Szeliski, “Video Matting,” in *Proceedings of ACM SIGGRAPH*, 2002, pp. 243–248.
- [39] M. J. B. and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Comput. Vis. Image Underst.*, vol. 63, no. 1, pp. 75–104, 1996.
- [40] A. Agarwala, A. Hertzmann, D. . Salesin, and S. . Seitz, “Keyframe-based Tracking for Rotoscoping and Animationo,” in *Proceedings of ACM SIGGRAPH*, 2004, pp. 584–591.
- [41] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient n-d image segmentation,” *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, 2006.
- [42] Y. Li, J. Sun, C. Tang, and H. Shum, “Lazy Snapping †,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.
- [43] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. Cohen, “Interactive Video Cutout,” in *Proc. of ACM SIGGRAPH*, 2005.
- [44] L. Vincent and P. Soille, “Watersheds in Digital Spaces: an EfficientAlgorithm Based on Immersion Simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 583–598, 1991.
- [45] H. Shum, J. Sun, S. Yamazaki, Y. Li, and C. Tang, “Pop-up Light Field: An Interactive Image-based Mmodeling and Rendering System,” *ACM Trans. Graph.*, vol. 23, no. 2, pp. 143–162, 2004.
- [46] J. A. Bondy and U. S. . Murty, *Graph Theory*. Springer, 2008.
- [47] K. Lang, “Fixing Two Weakness of the Spectral Method,” in *Proc. Advances in Neural Information Processing System*, 2005.

- [48] S. . Yu and J. Shi, “Multiclass spectral clustering,” in *Proc. in International Conference Computer Vision*, 2003.
- [49] L. Zelnik-Manor and P. Perona, “Self-Tuning Spectral Clustering,” in *Proc. Advances in Neural Information Processing Systems*, 2005.
- [50] A. Y. Ng, I. M. Jordan, and Y. Weiss, “On Spectral Clustering: Analysis and an Algorithm,” in *Proc. Advances in Neural Information Processing Systems*, 2001.
- [51] D. Tolliver and G. Miller, “Graph Partitioning by Spectral Rounding: Applications in Image Segmentation and Clustering,” in *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition*, 2006.
- [52] R. S. Basuki, M. Hariadi, and R. A. Pramunendar, “Fuzzy C-Means Algorithm for Adaptive Threshold on Alpha Matting,” in *International Conference on Information Technology and Electrical Engineering (CITEE)*.
- [53] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Pearson Prentice Hall, 2007.
- [54] L. Bonian, *Fuzzy Mathematics and Its Applications*. Hefei University of Technology Press, 2007.
- [55] W. Pennebaker and J. Mitchell, *JPEG – Still Image Data Compression Standard*. Newyork: International Thomsan Publishing, 1993.
- [56] H. Mochamad, H. C. Loy, and T. Aoki, “Semi-Automatic Video Object Segmentation Using LVQ with Color,” no. 7, pp. 1553–1560, 2005.
- [57] S. Sun, D. Haynor, and Y. Kim, “Semi-Automatic Video Object Segmentation using Vsnakes,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 75–82, 2003.

## LAMPIRAN A

### Video matting dataset: *scribble image, danground truth*

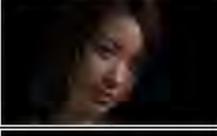
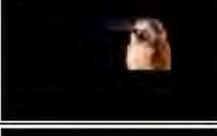
<i>File</i>	<i>Image input</i>	<i>Scribble image</i>	<i>Ground truth</i>
<i>Alex.bmp</i> (1)			
<i>Artem.bmp</i> (2)			
<i>City.bmp</i> (3)			
<i>Dmitriy.bmp</i> (4)			
<i>Football.bmp</i> (5)			
<i>Horse.bmp</i> (6)			
<i>Rain.bmp</i> (7)			
<i>Slava.bmp</i> (8)			
<i>Snow.bmp</i> (9)			
<i>Vitaliy.bmp</i> (10)			

Keteranganteknisvideo matting dataset:

- Resolusisetiap frame : 1920 x 1080 piksel
- Jumlah frame yang diuji : 100 frame

## LAMPIRAN B

### Perbandingan hasil ekstraksi *matte* secara visual

Image ke -	Threshold 0.12		Threshold FCM	
	Matte	Object	Matte	Object
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

## LAMPIRAN C

### *UCF Sport Action Dataset*

<i>File</i>	<i>Input image</i>	Resolusi (piksel)	Jumlah Frame
<i>Horse Riding</i>		720 x 404	100
<i>Playing Golf</i>		720 x 404	100
<i>Lifting</i>		720 x 404	100

<p><i>Skate Boarding</i></p>	 <p>708-75</p>	<p>480 x 360</p>	<p>100</p>
<p><i>Walk Front</i></p>	 <p>RF1-10578</p>	<p>720 x 488</p>	<p>100</p>
<p><i>Swing Bench</i></p>		<p>720 x 404</p>	<p>100</p>

## BIODATA



Nama : RuriSukoBasuki  
Tempat / TanggalLahir : Wonogiri, 17 Pebruari 1978  
Agama : Islam  
Pekerjaan : DosenFak. IlmuKomputer  
Alamat Kantor : Jl. Imam Bonjol 205 – 207, Semarang  
No. Telp. Kantor : 024 – 3517261  
AlamatRumah : Jl. Pari Raya, Blok I. No. 3, PerumahanGenuk  
Indah Semarang  
No. HP : 081390009716  
E-Mail : [ruri.basuki10@mhs.ee.its.ac.id](mailto:ruri.basuki10@mhs.ee.its.ac.id)  
[rurisb@research.dinus.ac.id](mailto:rurisb@research.dinus.ac.id)  
[risuba@gmail.com](mailto:risuba@gmail.com)  
Nama Istri : Agustina FitriAnulia  
Nama Anak : GayatriGhazia Gilda

### **RiwayatPendidikan**

SD Negeri 1 Ngrejo, Tirtomoyo, Wonogiri, Lulus tahun 1990  
SMP Negeri 1 Tirtomoyo, Wonogiri, Lulus tahun1993  
SMA Negeri 1 Wonogiri, Lulus tahun 1996  
S1 SistemInformasi, STMIK Dian Nuswantoro Semarang, Lulus tahun 2001  
S2 TeknikInformatikaUniversitas Dian Nuswantoro, Lulus tahun 2004  
S3 Jurusan T. Elektro, InstitutTeknologiSepuluhNopember, masuktahunajaran  
2010/2011

### **RiwayatPekerjaan**

Dosen Fak. Ilmu Komputer Universitas Dian Nuswantoro Semarang mulai tahun  
2001 – sekarang.