

# **TUGAS AKHIR**

## **STUDI PENENTUAN TITIK KONJUGASI PADA FOTO YANG BERTAMPALAN (*IMAGE MATCHING*) MENGUNAKAN METODE *AREA-BASED MATCHING* PADA FOTO STEREO**



**MILIK  
PERPUSTAKAAN  
ITN MALANG**

**Disusun oleh :**

**MARTINCE NOVIANTI BANI**

**NIM. 05.25.005**

**JURUSAN TEKNIK GEODESI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
INSTITUT TEKNOLOGI NASIONAL  
MALANG  
2010**

3010

1970

INSTITUT TEKNOLOGI AGUNG  
KULONAGAR LAMPUNG SELATAN DAN BERKAWAN  
SIMPANG BERTUKAR BERGEMER

1970 02 20 0000

STASIUN KUALITAS BINA

000000 0000

111 111111  
DEKORASI  
M I G K

1970 02 20 0000

INSTITUT TEKNOLOGI AGUNG  
KULONAGAR LAMPUNG SELATAN DAN BERKAWAN  
SIMPANG BERTUKAR BERGEMER

1970 02 20 0000

## LEMBAR PENGESAHAN

**Studi Penentuan Titik Konjugasi Pada Foto Yang Bertampalan (*Image Matching*) Menggunakan Metode *Area-Based Matching* Pada Foto Stereo**

### TUGAS AKHIR

Dipertahankan di hadapan Panitia Penguji Skripsi Jurusan Teknik Geodesi,  
Fakultas Teknik Sipil dan Perencanaan, Institut Teknologi Nasional Malang.  
Dinyatakan Lulus dan diterima untuk memenuhi syarat guna memperoleh gelar  
Sarjana Strata Satu (S-1) Teknik Geodesi.

**Hari/tanggal : Sabtu, 21 Agustus 2010**

Disusun oleh

**Martince Novianti Bani**

**05.25.005**

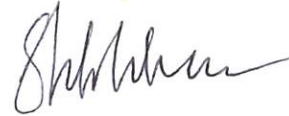
**Panitia Ujian Tugas Akhir**

**Ketua**



**Hery Purwanto, ST, M.Sc**

**Sekretaris**



**Silvester Sari Sai, ST, MT**

**Anggota Penguji**

**Penguji I**



**Ir. Agus Darpono, MT**

**Penguji II**



**Hery Purwanto, ST, M.Sc**

**Penguji III**



25/6 12011

**Dr. Edwin Tjahjadi, ST, M.Geom.Sc**

**JURUSAN TEKNIK GEODESI**

**FAKULTAS TEKNIK SIPIL DAN PERENCANAAN**

**INSTITUT TEKNOLOGI NASIONAL**

**MALANG**

**2010**

LEMBAR PENGESAHAN

Studi Penelitian Titik Kontrol Pada Foto Yang Bertampan (Luang  
Maling) Menggunakan Metode Ura-Buas Maling Pada Foto Stereo

TUGAS AKHIR

Dipersembahkan di hadapan Panitia Penguji Skripsi Jurusan Teknik Geodesi,  
Fakultas Teknik Sipil dan Perencanaan, Institut Teknologi Nasional Malang  
Dinyatakan lulus dan diterima untuk memenuhi syarat guna memperoleh gelar

Sarjana Teknik Sipil (S-1) Teknik Geodesi

Halvansyah : Sabtu, 21 Agustus 2010

Diancam oleh

Martine Noviani Bani

05.23.002

Pratita Ejiya Fugan Akbar

Segetaris

Ketua

Siswanto ST, MT

Hary Purwanto ST, MT

Anggota Penguji

Penguji III

Penguji II

Penguji I

Dr. Edwin Tjahjedi ST, MT, Geomat

Hary Purwanto ST, MT

Ir. Agus Darsono MT

JURUSAN TEKNIK GEODESI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

INSTITUT TEKNOLOGI NASIONAL

MALANG

2010



**LEMBAR PERSETUJUAN  
TUGAS AKHIR**

**Studi Penentuan Titik Konjugasi Pada Foto Yang Bertampalan (*Image Matching*) Menggunakan Metode *Area-Based Matching* Pada Foto Stereo**

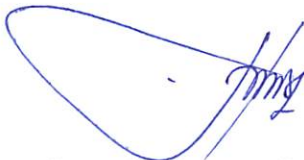
Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana Teknik Geodesi  
Strata Satu (S-1)  
Institut Teknologi Nasional Malang

Disusun Oleh :

**Martince Novianti Bani  
05.25.005**

Meyetujui,

**Dosen Pembimbing I**



**Hery Purwanto, ST, M.Sc**

**Dosen Pembimbing II**



**Dr. Edwin Tjahjadi, ST. M.Geom.Sc**

**Mengetahui**

**Ketua Jurusan Teknik Geodesi S-1**



**Hery Purwanto, ST, M.Sc**

**LEMBAR PERSETUJUAN  
TEGAS AKHIR**

Studi Penelitian Teknik Konjugasi Pada Foto Yang Berantakan (Image  
Matching) Menggunakan Metode Area-Based Matching Pada Foto Stereo

Dijadikan Sebagai Salah Satu Syarat Menyerah Gelar Sarjana Teknik Geodesi

Semester Satu (2-1)

Institut Teknologi Nasional Malang

Disusun Oleh :

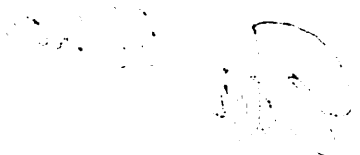
Martine Noviani Bani

0222002

Mengetahui,

Dosen Pembimbing II


Dosen Pembimbing I

  
Dr. Edwin Triandhi, ST, MT, Geom. Sc.

  
Heri Purwanto, ST, MT, Sc.

Mengetahui,

Ketua Jurusan Teknik Geodesi 2-1

  
Heri Purwanto, ST, MT, Sc.

## *Abstraksi*

Pencocokan citra (*image matching*) merupakan suatu pengukuran secara otomatis yang digunakan dalam fotogrametri. Salah satu metode yang digunakan dalam proses pencocokan citra yakni *area-based* yang mana teknik perhitungannya menggunakan teknik *Normalized Cross Correlation* (NCC) untuk mendapatkan nilai pendekatan, kemudian akan dilanjutkan dengan teknik perhitungan lain yaitu *Least Squares Matching* (LSM). Teknik LSM untuk menyempurnakan pencarian titik konjugasi hingga mencapai keakurasian sub-piksel.

Kata kunci: *image matching, cross correlation, least squares matching.*

Книга "История культуры" была опубликована в 1988 году.

Эта книга является частью серии "История культуры".

Второй выпуск "История культуры" (ИСК) был опубликован в 1988 году. В этом выпуске рассматриваются вопросы культуры в СССР. Книга посвящена истории культуры в СССР. В ней рассматриваются вопросы культуры в СССР. Книга посвящена истории культуры в СССР. В ней рассматриваются вопросы культуры в СССР.

История

## PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini

**Nama : Martince Novianti Bani**

**NIM : 0525005**

**Program Studi : Teknik Geodesi S-1**

**Fakultas : Fakultas Teknik Sipil Dan Perencanaan**

Menyatakan dengan sesungguhnya bahwa Skripsi saya dengan judul “**Studi Penentuan Titik Konjugasi Pada Foto Yang Bertampalan (*Image Matching*) Menggunakan Metode *Area-Based Matching* Pada Foto Stereo**” adalah hasil karya saya sendiri, bukan merupakan duplikat serta tidak mengutip atau menyadur dari hasil karya orang lain kecuali disebutkan sumbernya.

Malang, September 2010

Yang membuat pernyataan

**Martince Novianti Bani**

**0525005**

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini

Nama : Marlina Noriani Bani

NIM : 0525005

Program Studi : Teknik Geodesi 2-1

Fakultas : Fakultas Teknik Sipil dan Perencanaan

Menyatakan dengan sesungguhnya bahwa skripsi saya dengan judul "Studi Penelitian Teknik Konjugasi Pada Foto Yang Bertumpangtindih (Image Matching) Menggunakan Metode Area-Based Matching Pada Foto Stereo" adalah hasil karya saya sendiri, bukan merupakan duplikat serta tidak mengutip atau menyalin dari hasil karya orang lain kecuali disebutkan sumbernya.

Malang, September 2010

Yang membuat pernyataan

Marlina Noriani Bani

0525005



*As you put your faith in Him, to acknowledge in all you do, He'll open or  
close the doors with love and guide each step for you.  
Ask, and it will be given to you, seek and you will find; knock and the door  
will be opened to you.*

*Dengan kerendahan hati saya persembahkan karya ini untuk  
Tuhan dan Penyelamatku, Yesus Kristus atas segala berkat, pimpinan dan  
kekuatan untuk menyelesaikan penelitian ini.*

*Terima kasih sepenuhnya kepada Mama tersayang, Sartjie Maria Bani,  
atas segala kasih, kesabaran dan doanya untuk keberhasilan ini.  
My beloved sisters and brothers: kak Ani & kak Tinus, kak Leni & kak Lexy,  
Maya, Irna, dan Robby. Trima kasih untuk doa, dan dukungannya.  
My lovely cousin, Abe. Trima kasih selalu menemani dan menghiburku.  
My nephew; Willy, Cha n' Martheen and my lovely niece; Queen.*

*Teman-teman seperjuanganku di Geodesi 2005: Tansil, Enno, Chandra,  
Riri, Dody, Agus, Lia, Gede, Alben, Wenny, Ona. Trima kasih untuk  
persahabatan kita.*

*Kak Dessy, kak Yusak, kak Akbar, dan kak Roudger. Trima kasih untuk  
pertemanan kita sebelumnya dan selama di Lab SIG.*

*Teman-teman dalam suka dan dukaku: Nopy Angesti, buat segala  
persahabatan kita yang tak akan pernah aku lupakan☺; dr. Sandra, thanks  
ya bu dok buat segala nasehat dan doanya; Nina dan Ribut Purwaningrum  
yang selalu menjadi pendengar setiaku, trima kasih atas waktu-waktunya;  
dr. Yully, trima kasih buat obatnya saat aku sakit dan buat segala  
kegilaanmu yang membuat aku selalu tertawa; dan Try, trima kasih buat  
doa serta nasehatnya.*

*Untuk Papa tersayang, Marthen Bani. Papa terhebat yang pernah saya  
miliki. Trima kasih untuk setiap doa dan pengorbanan Papa. Trima kasih  
telah memberikan saya tempat untuk bersandar dan trima kasih karena  
saya terlahir sebagai anak Papa.  
Saya sayang Papa.*

As you put your faith in Him to acknowledge in all you do. He'll open or  
close the doors with love and guide each step for you.  
Ask, and it will be given to you; seek and you will find; knock and the door  
will be opened to you.

Dengan kerendahan hati saya persendahkan karya ini untuk  
Tribun dan Penyelamatku, Yesus Kristus atas segala berkat, pimpinan dan  
kekutan untuk menyelesaikan penelitian ini.

Terima kasih sepenuhnya kepada Maria tersayang, Sartje Maria Bani,  
atas segala kasih, kesabaran dan doanya untuk keberhasilan ini.  
My beloved sisters and brothers: kak Ani & kak Tims, kak Leni & kak Lexy,  
Maya, Irena, dan Robby. Terima kasih untuk doa dan dukungannya.  
My lovely cousin, Abe. Terima kasih selalu menemani dan menghibur.  
My nephews: Willy, Cha n' Marthen and my lovely niece: Queen.

Teman-teman seperjuanganku di Geodesi 2005: Tansil, Etno, Chandra,  
Riri, Doby, Agus, Lia, Gede, Alber, Wenny, Oca. Terima kasih untuk  
persahabatan kita.  
Kak Desy, kak Yusak, kak Akbar, dan kak Roudet. Terima kasih untuk  
pertemanan kita sebelumnya dan selama di Lab SIG.

Teman-teman dalam sukla dan dukukan: Nopy Anesti, buai segala  
persahabatan kita yang tak akan pernah aku lupakan @: dr. Sandra, thanks  
ya bu dok buat segala nasihat dan doanya; Nina dan Ribut Purnaningrum  
yang selalu menjadi pendengar setia, terima kasih atas waktu-waktunya;  
dr. Yully, terima kasih buat obatnya saat aku sakit dan buat segala  
kegiatanmu yang membuat aku selalu tertawa; dan Tj, terima kasih buat  
doa serta nasihatnya.

Untuk Papa tersayang, Marthen Bani. Papa terlewat yang pernah saya  
miliki. Terima kasih untuk setiap doa dan pengorbanan Papa. Terima kasih  
telah memberikan saya tempat untuk bersandar dan terima kasih karena  
saya terlahir sebagai anak Papa.  
Saya sayang Papa.

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat-Nya, penulis dapat menyelesaikan skripsi yang berjudul ” **Studi Penentuan Titik Konjugasi Pada Foto Yang Bertampalan (*Image Matching*) Menggunakan Metode *Area-Based Matching* Pada Foto Stereo**, di mana penulisan skripsi ini disusun sebagai salah satu syarat untuk meraih gelar Sarjana Teknik pada Jurusan Teknik Geodesi Fakultas Teknik Sipil Dan Perencanaan Institut Teknologi Nasional Malang.

Penulisan ini tidak akan dapat terselesaikan tanpa bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. A. Agus Santosa, MT selaku Dekan Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.
3. Bapak Heri Purwanto, ST., M.Sc. selaku Ketua Jurusan Teknik Geodesi Institut Teknologi Nasional Malang dan Dosen Pembimbing I, serta Dosen Penguji.
4. Bapak Dr. Edwin Tjahjadi, ST., MGeom.Sc. selaku Dosen Pembimbing II dan Dosen Penguji.
5. Bapak Ir. Agus Dapono, MT. selaku Dosen Penguji.
6. Segenap dosen, staff pengajar dan recording Jurusan Teknik Geodesi Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.



## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat-Nya, penulis dapat menyelesaikan skripsi yang berjudul "Studi Penelitian Teknik Konjugasi Pada Foto Yang Berstamping (Image Matching) Menggunakan Metode Area-Based Matching Pada Foto Stereo, di mana penelitian skripsi ini disusun sebagai salah satu syarat untuk meraih gelar Sarjana Teknik pada Jurusan Teknik Geodesi Fakultas Teknik Sipil Dan Perencanaan Institut Teknologi Nasional Malang.

Penelitian ini tidak akan dapat terselesaikan tanpa bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Eng. H. Abdillah Lomi, MSIE selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. A. Agus Samosa, MT selaku Dekan Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.
3. Bapak Hen Purwanto, ST, M.Sc. selaku Ketua Jurusan Teknik Geodesi Institut Teknologi Nasional Malang dan Dosen Pembimbing I serta Dosen Penguji.
4. Bapak Dr. Edwin Tjajadi, ST, MComm.Sc. selaku Dosen Pembimbing II dan Dosen Penguji.
5. Bapak Ir. Agus Dapono, MT selaku Dosen Penguji.
6. Segenap dosen staff pengajar dan recording Jurusan Teknik Geodesi Fakultas Teknik Sipil dan Perencanaan Institut Teknologi Nasional Malang.

7. Papa dan Mama yang selalu memberikan dukungan, semangat dan doa.
8. Team Rapid Mapping dan Deformasi yang selalu memberikan kerja sama dan dukungannya.

Semoga Tuhan Yang Maha Esa memberikan karunia atas budi baik dari semua pihak yang terlibat dalam penelitian ini. Penulis menyadari bahwa penulisan laporan ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Hasil penelitian ini dipersembahkan kepada almamater tercinta Jurusan Teknik Geodesi Institut Teknologi Nasional Malang.

Malang, September 2010

Penulis

7. Papa dan Mama yang selalu memberikan dukungan, semangat dan

doa.

8. Team Rapid Mapping dan Deformasi yang selalu memberikan kerja

sama dan dukungannya.

Semoga Tuhan Yang Maha Esa memberikan karunia mas hudi baik dari  
semua pihak yang terlibat dalam penelitian ini. Penulis menyadari bahwa  
penulisan laporan ini masih belum sempurna. Oleh karena itu kritik dan saran  
yang membangun sangat penulis harapkan. Hasil penelitian ini dipresentasikan  
kepada alumnus terata Jurusan Teknik Geodesi Institut Teknologi Nasional  
Malang.

Malang, September 2010

Penulis



## Daftar Gambar

Gambar 2.1. Citra hitam & putih dan citra berwarna serta matriks yang merepresentasikan kedua citra .....	7
Gambar 2.2 Pusat proyeksi .....	11
Gambar 2.3. Sistem koordinat piksel dan sistem koordinat foto .....	13
Gambar 2.4 Proses resampling .....	15
Gambar 2.5 Nearest neighbour dan bilinear interpolation.....	16
Gambar 2.6 <i>Epipolar plane</i> di proyeksikan dari pusat $O_1$ dan $O_2$ serta titik objek P	20
Gambar 2.7 Sepasang citra dalam bentuk visual .....	22
Gambar 2.8. Representatif dalam bentuk matriks.....	23
Gambar 2.9 Menghitung nilai koefisien korelasi antara template dan <i>search image</i>	24
Gambar 2.10 Posisi dari baris dan kolom untuk <i>least squares matching</i> .....	27
Gambar 3.1 Skema diagram alir penelitian.....	35
Gambar 3.2 Skema lanjutan diagram alir penelitian.....	36
Gambar 3.3 Skema diagram alir <i>normalized cross correlation</i> .....	37
Gambar 3.4 Skema diagram alir <i>least squares matching</i> .....	38
Gambar 4.1 Menambahkan pasangan foto kedalam project.....	46
Gambar 4.2 Cara memasukan file foto .....	46
Gambar 4.3 memasukan nama pair untuk ditampilkan di form utama .....	47
Gambar 4.4 Memulai proses <i>image matching</i> .....	48

## Daftar Gambar

Gambar 2.1. Citra binar & putih dan citra berwarna serta matriks yang mempresentasikan kedua citra .....	7
Gambar 2.2. Prasa proyeksi .....	11
Gambar 2.3. Sistem koordinat piksel dan sistem koordinat foto .....	13
Gambar 2.4. Proses resampling .....	13
Gambar 2.5. Nearst neighbor dan bilinear interpolation .....	16
Gambar 2.6. Ekuivalensi di proyeksi dari pusat $O_1$ dan $O_2$ serta titik objek $P$ dan Gambar 2.7. Sepasang citra dalam bentuk visual .....	20
Gambar 2.8. Representasi dalam bentuk matriks .....	23
Gambar 2.9. Menghitung nilai koefisien korelasi antara template dan search image .....	24
Gambar 2.10. Posisi dari basis dan kolom untuk $K_{\text{row}}$ $K_{\text{col}}$ $K_{\text{row}}$ $K_{\text{col}}$ .....	27
Gambar 3.1. Skema diagram alir penelitian .....	32
Gambar 3.2. Skema lanjutan diagram alir penelitian .....	36
Gambar 3.3. Skema diagram alir normalized cross correlation .....	37
Gambar 3.4. Skema diagram alir $K_{\text{row}}$ $K_{\text{col}}$ $K_{\text{row}}$ $K_{\text{col}}$ .....	38
Gambar 4.1. Menambahkan pasangan foto kedalam project .....	46
Gambar 4.2. Cara memasukkan file foto .....	46
Gambar 4.3. menasukkan nama bar untuk ditampilkan di form utama .....	47
Gambar 4.4. Menulai proses $K_{\text{row}}$ $K_{\text{col}}$ .....	48

Gambar 4.5 Menjalankan proses image matching .....	48
Gambar 4.6 Output proses <i>Cross Correlation</i> .....	49
Gambar 4.6 Output proses <i>Least Squares Matching</i> .....	51

## Daftar Tabel

Tabel 2.1 Metode <i>image matching</i> .....	18
Tabel 2.2 Desain matriks <i>least-squares</i> .....	29
Tabel 4.1 Tabel koordinat piksel .....	52
Tabel 4.2 Tabel Koordinat foto dan standar deviasi dari proses <i>intersection</i> .....	53

## DAFTAR ISI

<b>Lembar Pengesahan</b> .....	i
<b>Lembar Persetujuan</b> .....	ii
<b>Abstraksi</b> .....	iii
<b>Pernyataan Keaslian Skripsi</b> .....	iv
<b>Kata Pengantar</b> .....	v
<b>Daftar Gambar</b> .....	vii
<b>Daftar Tabel</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>BAB 1 PENDAHULUAN</b>	
I.1. Latar Belakang .....	1
I.2. Rumusan Masalah.....	2
I.3. Maksud dan Tujuan .....	2
I.3.1. Maksud Penelitian .....	2
I.3.2. Tujuan Penelitian.....	2
I.4. Batasan Masalah .....	3
I.5. Manfaat Penelitian .....	3
I.5. Tinjauan Pustaka.....	3
<b>BAB II DASAR TEORI</b>	
II.1. Peninjauan <i>Image Matching</i> .....	6
II.1.1. Citra Digital.....	7
II.1.1.1. Koreksi Radiometrik .....	8

**DAFTAR ISI**

i ..... Lembar Pengesahan

ii ..... Lembar Persewaan

iii ..... Abstraksi

iv ..... Pernyataan Keaslian Skripsi

v ..... Kata Pengantar

vii ..... Daftar Gambar

ix ..... Daftar Tabel

x ..... **DAFTAR ISI**

**BAB I PENDAHULUAN**

1 ..... 1.1. Latar Belakang

2 ..... 1.2. Rumusan Masalah

3 ..... 1.3. Maksud dan Tujuan

2 ..... 1.3.1. Maksud Penelitian

2 ..... 1.3.2. Tujuan Penelitian

3 ..... 1.4. Batasan Masalah

3 ..... 1.5. Manfaat Penelitian

3 ..... 1.6. Tinjauan Pustaka

**BAB II DASAR TEORI**

6 ..... II.1. Peminjaman konsep Aljabar

7 ..... II.1.1. Ciri Digital

8 ..... II.1.1.1. Koreksi Radionomik



II.1.1.2. Koreksi Geometrik .....	10
II.1.1.3. Orientasi dan Georeferensi .....	11
II.1.1.3.1 Konversi Sistem koordinat .....	13
II.1.1.3.1.1 Konversi koordinat piksel ke koordinat foto .....	13
II.1.1.3.1.1 Konversi koordinat foto ke koordinat piksel .....	14
II.1.1.4 Transformasi Geometrik dan <i>Resampling</i> .....	15
II.1.1.4.1 Proses <i>resampling</i> .....	16
II.1.1.4.2 Proses interpolasi .....	16
II.1.2 Pencocokan Citra ( <i>Image Matching</i> ) .....	17
II.1.2.1 Metode <i>Area-Based</i> .....	19
II.1.2.1.1 <i>Normalized Cross Correlation</i> (NCC).....	21
II.1.2.1.2 <i>Least Squares Matching</i> (LSM).....	25
II.1.2.1.2.1 Prosedur Perhitungan <i>Least Squares Matching</i> (LSM) .....	26
II.1.2.1.2.1.1 Desain Matriks <i>Least Squares</i> .....	29
II.1.2.1.2.1.2 Nilai Pendekatan yang Disarankan .....	32
 <b>BAB III METODOLOGI PENELITIAN</b>	
III.1 Peralatan dan Bahan Penelitian.....	34
III.1.1 Deskripsi Data Penelitian.....	34
III.1.2 <i>Hardware</i> dan <i>Software</i> .....	34
III.2 Diagram Alir Penelitian .....	35
III.2.1 Diagram Alir <i>Normalized Cross Correlation</i> .....	37
III.2.1 Diagram Alir <i>Least Squares Matching</i> .....	37
III.3 Pelaksanaan Penelitian.....	39

III.3	Penyajian dan Bentuk Penelitian	30
III.3.1	Diagram Alir Sesuai dengan Menerang	31
III.3.1	Diagram Alir Diagramized Cross Correlation	31
III.3	Diagram Alir Penelitian	32
III.3	Metode dan Definisi	34
III.3.1	Deskripsi Data Penelitian	34
III.3	Penyajian dan Bentuk Penelitian	34
<b>BAB III METODOLOGI PENELITIAN</b>		
III.3.1.3.1.3	Metode Penelitian yang Digunakan	35
III.3.1.3.1.1	Desain Matriks Sesuai dengan	30
III.3.1.3.1	Prosedur Penelitian Sesuai dengan Menerang (M&M)	30
III.3.1.3	Sesuai dengan Menerang (M&M)	32
III.3.1.1	Diagramized Cross Correlation (DCC)	31
III.3.1	Metode Penelitian	10
III.3	Penggunaan Data (juga Menerang)	11
III.3.1.3	Proses Interpretasi	10
III.3.1.1	Proses Menerang	10
III.3.1.4	Transformasi Geometrik dan Kesamaan	12
III.3.1.1.1	Konversi Koordinat Foto ke Koordinat Bujur	14
III.3.1.1.1	Konversi Koordinat Bujur ke Koordinat Foto	13
III.3.1.1	Konversi Sistem Koordinat	13
III.3.1.3	Orientasi dan Georeferensi	11
III.3.1.3	Koreksi Geometrik	10

## **BAB IV ANALISA DAN HASIL**

IV.1 Analisa Algoritma <i>Image Matching</i> dengan Metode <i>Area-Based</i> .....	45
IV.1.1 Menampilkan Sepasang Foto Stereo .....	45
IV.1.2 Teknik <i>Normalized Cross Correlation</i> .....	47
IV.1.2.1 Hasil perhitungan teknik <i>normalized cross correlation</i> .....	47
IV.1.2.2 Analisa teknik <i>normalized cross correlation</i> .....	49
IV.1.3 Teknik <i>Least Squares Matching</i> .....	50
IV.1.3.1 Hasil perhitungan teknik <i>least squares matching</i> .....	50
IV.1.3.2 Analisa teknik <i>least squares matching</i> .....	51
IV.2 Pengujian Standar Deviasi Proses <i>Intersection</i> .....	51

## **BAB V PENUTUP**

V.1 Kesimpulan .....	54
V.2 Saran .....	55

<b>DAFTAR PUSTAKA</b> .....	<b>xiii</b>
-----------------------------	-------------

## **LAMPIRAN A**

## **LAMPIRAN B**

**ГҶҰҰҰҰҰҰ В**

**ГҶҰҰҰҰҰҰ V**

**DUYULAK BULGUKU** ..... 211

**V'3 2slyu** ..... 22

**V I Kesimpulan** ..... 24

**BAB A. BEJALILAH**

**IA'3 Berghijian Standar Deviasi Proses Inversioan** ..... 21

**IA'1'3'3 Analisis teknik jeksa ukuran2 mancing** ..... 21

**IA'1'3'1 Hasil berhitung2an teknik jeksa ukuran2 mancing** ..... 20

**IA'1'3 Teknik jeksa ukuran2 mancing** ..... 20

**IA'1'3'3 Analisis teknik normalisasi cross correlation** ..... 40

**IA'1'3'1 Hasil berhitung2an teknik normalisasi cross correlation** ..... 43

**IA'1'3 Teknik Normalisasi Cross Correlation** ..... 43

**IA'1'1 Menamponik2an Zepasan2 Foto Stereo** ..... 42

**IA'1 Analisis Algoritma Invers2 Mancing2an dengan Metode First-Base2** ..... 24

**BAB IA ANALISA DUA PASIR**

# BAB I

## PENDAHULUAN

### I.1. Latar Belakang

Dasar utama dalam proses fotogrametri adalah mengidentifikasi dan mengukur titik konjugasi dalam dua atau lebih foto yang bertampalan. Dalam sistem fotogrametri konvensional, titik konjugasi diamati dengan sepenuhnya mengandalkan indera penglihatan manusia (operator). Namun dalam fotogrametri digital, proses tersebut dapat digantikan oleh komputer. Proses tersebut yang kita ketahui sebagai proses *image matching* (Schenk, 1999).

Titik konjugasi merupakan "titik yang sama" yang berada pada dua atau lebih citra yang saling bertampalan (Elaksher, 2008). Pencarian titik konjugasi berdasarkan pada *gray value*, sehingga pencarian tersebut harus mencapai sub-piksel agar lebih teliti dalam menentukan kesamaan nilai dari titik konjugasi yang dicari. Nilai keabuan (*gray value*) pada sebuah citra dapat dibandingkan kemiripannya dengan sekumpulan *gray value* dari citra di sebelahnya yang bertampalan. Tingkat kemiripan dari kumpulan data itulah yang kemudian ditentukan untuk merepresentasikan titik konjugasi yang dicari.

Schenk (1999) menguraikan dengan rinci ketiga metode yang sejauh ini banyak digunakan dalam proses pencocokan citra. Ketiga metode yang dimaksud adalah *area-based*, *feature-based*, dan *symbolic matching*. Pada penelitian ini, lokasi titik konjugasi akan diidentifikasi dengan menggunakan metode *area-based* serta teknik perhitungannya menggunakan teknik *Normalized Cross Correlation* (NCC)

# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Dasar utama dalam proses fotografi adalah mengidentifikasi dan mengukur titik konjugasi dalam dua atau lebih foto yang bertampalan. Dalam sistem fotografi konvensional, titik konjugasi diambil dengan sepenuhnya mengandalkan indera penglihatan manusia (operator). Namun dalam fotografi digital, proses tersebut dapat digantikan oleh komputer. Proses tersebut yang kita ketahui sebagai proses image mapping (Zelawy, 1999).

Titik konjugasi merupakan "titik yang sama" yang berada pada dua atau lebih citra yang saling bertampalan (Zelawy, 2008). Pencarian titik konjugasi berdasarkan pada garis wilayah sehingga pencarian tersebut harus mencapai sub-piksel agar lebih teliti dalam menentukan kesamaan nilai dari titik konjugasi yang dicari. Nilai keabuan (gray value) pada sebuah citra dapat dibandingkan kemiripannya dengan skemponen gray value dari citra di sekitarnya yang bertampalan. Tingkat kemiripan dari kumpulan data inilah yang kemudian ditentukan untuk mengoptimalkan titik konjugasi yang dicari.

Schenk (1999) menggunakan dengan tiga ketiga metode yang sudah ini banyak digunakan dalam proses pencocokan citra. Ketiga metode yang dimaksud adalah area-based, feature-based, dan window method. Pada penelitian ini, lokasi titik konjugasi akan diidentifikasi dengan menggunakan metode area-based serta teknik perhitungannya menggunakan teknik normalized cross correlation (NCC).



untuk mendapatkan nilai pendekatan, kemudian akan dilanjutkan dengan teknik perhitungan lain yaitu *Least Squares Matching* (LSM). Teknik LSM untuk menyempurnakan pencarian titik konjugasi hingga mencapai keakurasian sub-piksel (*Forstner, 1982*).

## **I.2. Rumusan Masalah**

Pencocokan citra (*image matching*) merupakan suatu pengukuran secara otomatis yang digunakan dalam fotogrametri. Beberapa metode seperti yang telah disampaikan diatas masing-masing memiliki keakurasian tersendiri. Untuk itu dalam penelitian ini diusung salah satu metode pencocokan citra yang telah disampaikan diatas yakni metode *area-based* untuk mencari lokasi titik konjugasi yang mana entitas utamanya yaitu berdasarkan pada nilai keabuan (*gray value*).

## **I.3. Maksud dan Tujuan**

### **I.3.1. Maksud Penelitian**

Penelitian ini dimaksudkan untuk mencari lokasi titik konjugasi dan nilai keakurasian dalam penentuan lokasi titik konjugasi dengan menggunakan metode *area-based*.

### **I.3.2. Tujuan Penelitian**

Tujuan dari penelitian ini ialah menyajikan algoritma pencocokan citra (*image matching*) dengan menggunakan metode *area-based* dengan teknik *normalized cross correlation* dan *least square matching* untuk menghasilkan titik konjugasi dengan keakurasian sub-piksel.

untuk mendapatkan nilai pendekatan, kemudian akan dilanjutkan dengan teknik  
berhitung lain yaitu Area Square Method (ASM). Teknik ASM untuk  
menyempurnakan pencarian titik konjugasi hingga mencapai keakuratan sub-piksel  
(Vossler, 1982).

### 1.2. Rumusan Masalah

Pencocokan citra (image matching) merupakan suatu pengujian secara  
otomatis yang digunakan dalam fotografi. Beberapa metode seperti yang telah  
disebutkan diatas masing-masing memiliki keakuratan tersendiri. Untuk itu dalam  
penelitian ini diusahakan salah satu metode pencocokan citra yang telah disebutkan  
didasarkan untuk metode wave-based untuk mencari lokasi titik konjugasi yang mana  
entitas mananya yaitu berdasarkan pada nilai keabuan (gray value).

### 1.3. Maksud dan Tujuan

#### 1.3.1. Maksud Penelitian

Penelitian ini dimaksudkan untuk mencari lokasi titik konjugasi dan  
nilai keakuratan dalam penentuan lokasi titik konjugasi dengan menggunakan  
metode wave-based.

#### 1.3.2. Tujuan Penelitian

Tujuan dari penelitian ini ialah menyajikan algoritma pencocokan citra  
(image matching) dengan menggunakan metode wave-based dengan teknik  
normalized cross correlation dan area square method untuk menghasilkan titik  
konjugasi dengan keakuratan sub-piksel.

#### **I.4. Batasan Masalah**

Penelitian ini dibatasi pada :

1. Mencari lokasi titik konjugasi pada foto stereo.
2. Menganalisa keakurasian dan ketepatan titik konjugasi berdasarkan *gray value* hingga mencapai ketelitian sub-piksel.

#### **I.5. Manfaat Penelitian**

Manfaat penelitian ini adalah :

- Dapat menemukan titik konjugasi dengan keakurasian sub-piksel yang bersesuaian pada foto yang bertampalan dengan menggunakan metode *area-based*.

#### **I.5. Tinjauan Pustaka**

Menemukan titik konjugasi dalam dua atau lebih foto yang saling bertampalan secara otomatis merupakan dasar dalam fotogrametri digital. Proses ini biasanya disebut dengan pencocokan citra atau yang lebih dikenal dengan *image matching* (Schenk, 1999; Aguiris et al, 2000 ). Gruen (1985) mengkaji bahwa, jika pertampalan dinilai cukup maka pada citra yang sama nilai tersebut dapat diuraikan dengan suatu transformasi antar objek dalam kedua citra. Dalam transformasi ini terdapat 8 parameter dan dapat dilakukan pendekatan dengan menggunakan transformasi *affine* (6 parameter). Pendekatan ini untuk menyamakan keakurasian hingga mencapai akurasi subpiksel. Beberapa teknik pencocokan citra (*image matching*) adalah *area-based matching*, *feature-based matching* dan *symbolic*

#### 1.4. Batasan Masalah

Penelitian ini dibatasi pada :

1. Meneliti lokasi titik konjugasi pada foto stereo.
2. Menganalisis keakuratan dan ketepatan titik konjugasi berdasarkan gambar yang dihasilkan dengan metode triangulasi.

#### 1.5. Maksud Penelitian

Maksud penelitian ini adalah :

- Dapat menentukan titik konjugasi dengan keakuratan sub-piksel yang disesuaikan pada foto yang berimpitan dengan menggunakan metode triangulasi.

#### 1.6. Tinjauan Pustaka

Menentukan titik konjugasi dalam dua atau lebih foto yang saling berimpitan secara otomatis merupakan dasar dalam fotogrametri digital. Proses ini biasanya disebut dengan pencocokan citra atau yang lebih dikenal dengan *image matching* (Schneek 1999; Elgawhry et al 2000). (Gruen 1982) menguji bahwa jika berimpitan dinilai cukup maka pada citra yang sama nilai tersebut dapat diabaikan dengan suatu transformasi antar objek dalam kedua citra. Dalam transformasi ini terdapat 8 parameter dan dapat dilakukan pendekatan dengan menggunakan transformasi affine (6 parameter). Pendekatan ini untuk menentukan keakuratan biasanya mencapai akurasi sub-piksel. Beberapa teknik pencocokan citra (*image matching*) adalah *area-based matching*, *feature-based matching* dan *spatial*

*matching* (Schenk, 1999; Wolf dan Dewitt, 2000; Potuckova, 2004; Leica Geosystems, 2006).

Metode *area-based* membandingkan tingkat keabuan (*gray level*) antar (sebagian kecil) citra. Pada citra digital, akan sangat memungkinkan dan relatif mudah dilakukan proses penentuan letak titik dengan pemecahan matematis. Sekumpulan *gray value* pada sebuah citra dapat dibandingkan kemiripannya dengan sekumpulan *gray value* dari citra di sebelahnya (citra-2) yang bertampalan. Tingkat kemiripan kumpulan data tersebut ditentukan oleh variasi *gray value* yang merepresentasikan bentuk obyek. Tingkat kemiripannya dapat dihitung dengan mencari korelasi berdasarkan kwadrat terkecil (*least square*). Dengan berpedoman pada hasil hitungan nilai korelasi, maka selanjutnya dapat ditentukan tingkat “kesamaan” dua kumpulan data yang berasosiasi dengan citra tersebut

Pada penelitian ini akan dikaji tentang teknik pencocokan citra dengan metode *area-based* dan teknik perhitungannya menggunakan teknik *Normalized Cross Correlation* (NCC) dan *Least Squares Matching* (LSM).

Keunggulan teknik NCC adalah kecepatan komputasinya, tetapi hanya menghasilkan ketelitian 1 piksel (Mikhail, Bethel et al., 2001). Sebaliknya dengan metode LSM, karena menerapkan hitung kuadrat terkecil pada nilai keabuan, relatif lebih lambat untuk konvergen namun mampu menghasilkan ketelitian hingga 0,01 piksel (Luhmann, Robson et al. 2006). Disamping itu, kelemahan lain dari LSM adalah teknik LSM membutuhkan nilai pendekatan posisi yang cukup dekat terhadap nilai sebenarnya agar perhitungan iterasinya dapat konvergen (Gruen, 2001).

metode (Schuck, 1999; Wolf dan Brown, 2000; Fomkova, 2004; Arita (Gozarwan, 2006)).

Metode wave-based membandingkan tingkat keabuan (gray level) antar (sebagian kecil) citra. Pada citra digital, akan sangat memungkinkan dan relatif mudah dilakukan proses penentuan titik dengan pemecahan matematis. Selanjutnya gray wave pada sebuah citra dapat dibandingkan kemiripannya dengan sekumpulan gray wave dan citra di sebelahnya (citra-2) yang berurutan. Tingkat kemiripan data tersebut ditentukan oleh variasi gray wave yang direpresentasikan bentuk objek. Tingkat kemiripannya dapat dihitung dengan mencari korelasi berdasarkan kvadrat terkecil (least square). Dengan perbedaan pada hasil hitungan nilai korelasi, maka selanjutnya dapat ditentukan tingkat "kesamaan" dan kumpulan data yang berasosiasi dengan citra tersebut.

Pada penelitian ini akan dikaji tentang teknik pencocokan citra dengan metode wave-based dan teknik perhitungannya menggunakan teknik Wavelet Coorrelation (NCC) dan Area Spares Matching (ASM).

Kemampuan teknik NCC adalah kemampuan komputasinya tetapi hanya menghasilkan ketelitian 1 piksel (Wahid, Kibar, et al., 2001). Sebaliknya dengan metode LSM, karena menerapkan hitung kuadrat terkecil pada nilai keabuan relatif lebih lambat untuk konvergensi namun mampu menghasilkan ketelitian hingga 0,01 piksel (Luhman, Kobler, et al., 2006). Disamping itu ketelitian lain dari LSM adalah teknik LSM membutuhkan nilai pendekatan posisi yang cukup dekat terhadap nilai sebenarnya agar perhitungannya dapat konvergen (Cava, 2001).

Berdasarkan karakteristik masing-masing metode yang saling melengkapi, pada penelitian ini digunakan teknik NCC untuk mendapatkan nilai awal titik konjugasi yang dicari, lalu posisi dan ketelitian titik konjugasi ini dihitung ulang dengan teknik LSM untuk mendapatkan hasil yang lebih baik.

Adapun uraian tentang isi dari masing-masing bab dalam Laporan Tugas Akhir dengan judul **Studi Penentuan Titik Konjugasi Pada Foto Yang Bertampalan (*Image Matching*) Menggunakan Metode *Area-Based Matching* Pada Foto Stereo**, yaitu sebagai berikut: Bab I merupakan pendahuluan dari Laporan Tugas Akhir ini, Bab II berisi teori dasar penentuan titik konjugasi menggunakan metode *area-based* serta teknik yang digunakan yaitu NCC dan LSM yang digunakan dalam perencanaan pembuatan algoritma *image matching*. Pada Bab III berisi tentang metodologi penelitian yakni pembuatan algoritma *image matching*. Sedangkan pada Bab IV berisi hasil dan pembahasan dari algoritma yang telah dibuat, dan penutup dari Laporan Tugas Akhir ini yaitu Bab V yang memuat kesimpulan dan saran.

Berdasarkan karakteristik masing-masing metode yang saling melengkapi pada penelitian ini digunakan teknik NCC untuk mendapatkan nilai awal titik konjugasi yang dicari, lalu posisi dan ketelitian titik konjugasi ini dibantu dengan dengan teknik LSM untuk mendapatkan hasil yang lebih baik.

Adapun urutan tembang isi dari masing-masing bab dalam Laporan Tugas Akhir dengan judul **Studi Perencanaan Titik Konjugasi Pada Foto Yang Berambaran (Wide Viewing) Menggunakan Metode Area-Based Matching Pada Foto Stereo**, yaitu sebagai berikut: Bab I merupakan pendahuluan dari Laporan Tugas Akhir ini, Bab II berisi teori dasar perencanaan titik konjugasi menggunakan metode area-based serta teknik yang digunakan yaitu NCC dan LSM yang digunakan dalam perencanaan pemetaan algoritma wide viewing. Pada Bab III berisi tentang metodologi penelitian yakni pemetaan algoritma wide viewing. Sedangkan pada Bab IV berisi hasil dan pembahasan dari algoritma yang telah dibuat dan penutup dari Laporan Tugas Akhir ini yaitu Bab V yang memuat kesimpulan dan

saat ini.



## **BAB II**

### **DASAR TEORI**

Pencocokan citra (*image matching*) merupakan kunci utama dalam proses restitusi foto (Campbell dan Wu, 2008.). Dalam bab ini akan dijelaskan hal-hal yang menjadi dasar diperlukannya *image matching* dalam proses restitusi foto/citra dalam fotogrametri. Schenk (1999) menguraikan dengan rinci ketiga metode yang sejauh ini banyak digunakan dalam proses pencocokan citra. Ketiga metode yang dimaksud adalah *area-based*, *feature-based*, dan *symbolic matching*. Dalam penelitian ini akan dititikberatkan pada pembahasan mengenai penentuan titik konjugasi pada foto yang bertampalan (*image matching*) dengan menggunakan metode *area-based* serta teknik perhitungannya yaitu *Normalized Cross Correlation* (NCC) dan *Least Squares Matching* (LSM). Dalam bab ini akan ditinjau mengenai *image matching* serta metode dasar untuk pengukuran secara otomatis dalam fotogrametri terutama menggunakan metode *area-based*.

#### **II.1. Peninjauan *Image Matching***

Salah satu dasar dalam proses fotogrametri adalah mengidentifikasi dan mengukur titik konjugasi dalam dua atau lebih citra yang saling bertampalan. Dalam fotogrametri analog dan analitik, titik konjugasi diidentifikasi dan diukur secara manual sedangkan dalam fotogrametri digital tugas tersebut diharapkan dapat dikerjakan secara otomatis. Dalam tulisan ini akan dibahas tentang proses secara

## BAB II DASAR TEORI

Pencocokan citra (image matching) merupakan kunci utama dalam proses restitusi foto (Campbell dan Ha, 2008). Dalam bab ini akan dijelaskan hal-hal yang menjadi dasar dipertukarkannya image matching dalam proses restitusi foto-citra dalam fotogrametri. Schenk (1999) menggunakan dengan rinci ketiga metode yang adalah ini banyak digunakan dalam proses pencocokan citra. Ketiga metode yang dimaksud adalah area-based, feature-based, dan template matching. Dalam penelitian ini akan diteliti perbedaan pada pemahaman mengenai penemuan titik konjugasi pada foto yang bertumpangtindih (image matching) dengan menggunakan metode area-based serta teknik berikutnya yaitu Normalized Cross Correlation (NCC) dan Least Squares Matching (LSM). Dalam bab ini akan dijelaskan mengenai image matching serta metode dasar untuk penemuan secara otomatis dalam fotogrametri terutama menggunakan metode area-based.

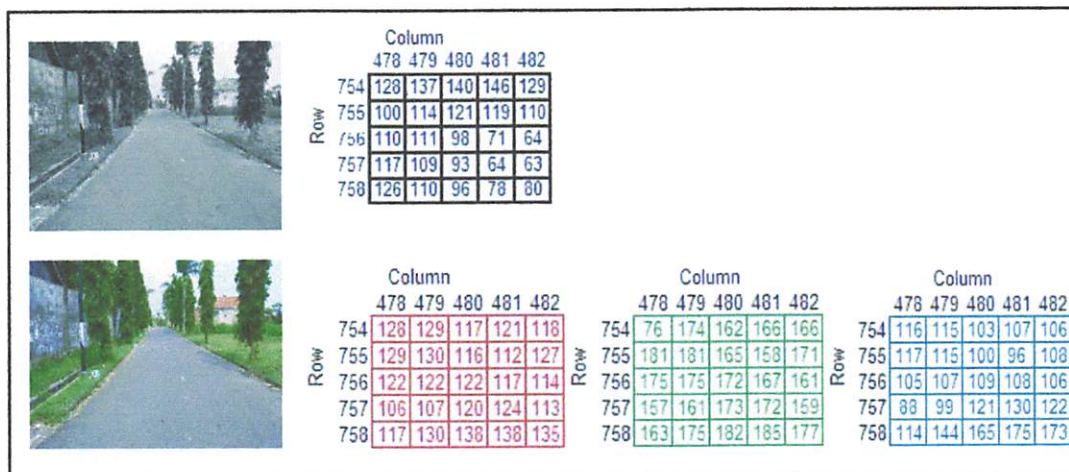
### 11.1. Peninjauan Image Matching

Salah satu dasar dalam proses fotogrametri adalah mengidentifikasi dan mengukur titik konjugasi dalam dua atau lebih citra yang saling bertumpangtindih. Dalam fotogrametri analog dan analitik, titik konjugasi diidentifikasi dan diukur secara manual sedangkan dalam fotogrametri digital tugas tersebut dibantu dengan dikalkulasikan secara otomatis. Dalam tulisan ini akan dibahas tentang proses secara

otomatis untuk menemukan hubungan antara beberapa citra yang disebut dengan pencocokan citra (*image matching*)(Schenk, 1999).

### II.1.1. Citra Digital

Citra merupakan fungsi kontinyu dari intensitas cahaya pada bidang dua dimensi (Mallat, 1999). Intensitas cahaya, merupakan hasil kali antara jumlah pancaran (illuminasi) cahaya yang diterima objek dengan derajat kemampuan obyek memantulkan cahaya. Citra digital umumnya direpresentasikan dalam bentuk matriks 2 dimensi dengan ukuran  $n \times m$  (Tsai dan Lin, 2002). Elemen terkecil dalam citra digital (elemen matriks) disebut “*picture element*” atau *pixel* (Potuckova, 2004). Setiap nilai piksel pada citra merepresentasikan nilai intensitas cahaya (Gambar 2.1).



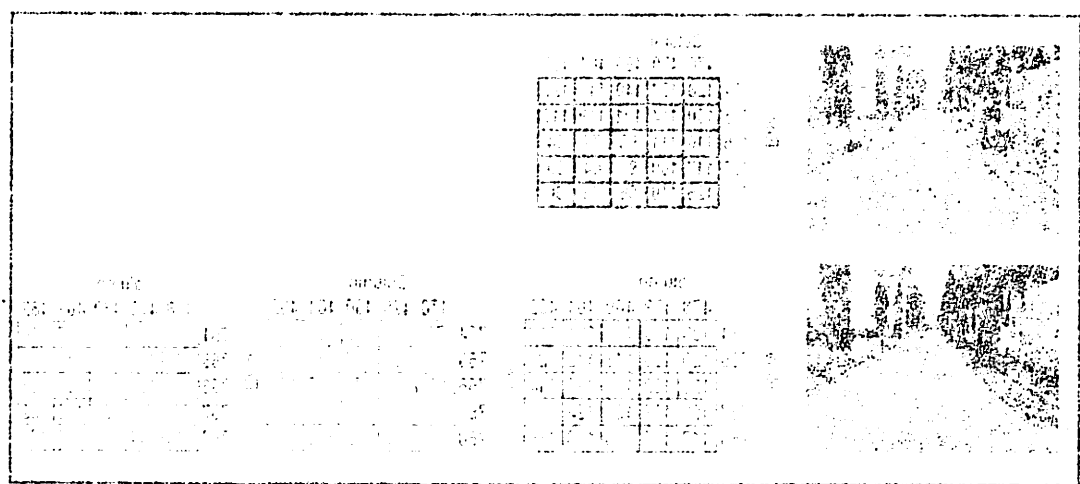
Gambar 2.1. Citra hitam & putih dan citra berwarna serta matriks yang merepresentasikan kedua citra.

otomatis untuk menentukan hubungan antara beberapa citra yang disebut dengan pencocokan citra (Waters dan Turk, 1999).

### 11.1.1 Citra Digital

Citra merupakan fungsi komposisi dari intensitas cahaya pada bidang dua dimensi (Waters, 1999). Intensitas cahaya merupakan hasil dari antara jumlah pancaran (illumination) cahaya yang diterima objek dengan derajat kemampuan objek memantulkannya. Citra digital umumnya direpresentasikan dalam bentuk matriks 2 dimensi dengan ukuran  $w \times h$  (Yick dan Liu, 2002). Elemen terkecil dalam citra digital (elemen matriks) disebut "pixel" (picture element) (Yamamoto, 2004). Setiap nilai pixel pada citra merepresentasikan nilai

intensitas cahaya (Gambar 11.1).



Gambar 11.1. Citra biner 2. Print-out citra biner yang telah direpresentasikan dalam bentuk matriks.

Citra digital dapat diperoleh dengan dua cara yaitu melalui sensor kamera atau dengan proses secara tidak langsung yakni dengan memindai (*scanning*) foto analog untuk menjadi citra digital. Informasi tentang obyek yang ditampilkan dalam citra kemudian dianalisa atau yang lebih dikenal dengan ‘proses data citra’. Kualitas citra digital tergantung pada parameter-parameter dari CCD (*Charge Coupled Device*), bagian-bagian *photosensitive scanner*, kamera digital dan sensornya. Ada dua proses yang terkait dengan akusisi data citra yang ketelitiannya berhubungan dengan parameter-parameter diatas yaitu *sampling* dan *quantizing*. **Sampling** berarti keseluruhan citra dibagi kedalam “*picture element*” atau *pixel*, dengan bentuk dan ukuran tergantung dari kapasitas bentuk dan ukuran CCD. Jadi, **sampling** untuk menentukan geometrik dari citra. **Quantizing** untuk menentukan kualitas nilai piksel dan menetapkan koreksi radiometrik pada citra (*Schenk, 1999; Potuckova, 2004*). Pada pokok bahasan selanjutnya, akan dibahas mengenai koreksi radiometrik dan koreksi geometrik pada citra digital. Kemudian dilanjutkan dengan orientasi citra dan transformasi geometrik.

#### **II.1.1.1. Koreksi Radiometrik**

Nilai intensitas antara minimal (hitam) dan maksimal (putih) dan jarak antara nilai-nilai tersebut merupakan parameter dalam proses kuantisasi (*Mikhail et al, 2001; Potuckova, 2004*). Syarat-syarat dari nilai keabuan (*gray value*), kepadatan citra atau nilai citra juga menggunakan hubungan tersebut (*Jahne, 1997*). Untuk alasan praktis, nilai 256 juga banyak digunakan (*Potuckova, 2004*). Sebuah citra yang memiliki karakteristik resolusi radiometrik 8-bit, nilai

Citra digital dapat diperoleh dengan dua cara yaitu melalui sensor kamera atau dengan proses secara tidak langsung yakni dengan memindai (scanning) foto analog untuk menjadi citra digital. Informasi tentang objek yang ditampilkan dalam citra kemudian dianalisa atau yang lebih dikenal dengan "proses data citra". Kualitas citra digital tergantung pada parameter-parameter dari CCD (Charge Coupled Device), bagian-bagian fotokamera, kamera digital dan sensornya. Ada dua proses yang terkait dengan akuisisi dari citra yang ketelitiannya berhubungan dengan parameter-parameter di atas yaitu sampling dan dithering. *Sampling* berarti keseluruhan citra dibagi kedalam "picture element" atau pixel, dengan bentuk dan ukuran tergantung dari kapasitas bentuk dan ukuran CCD. Jadi, sampling untuk menentukan geometrik dari citra. *Quantizing* untuk menentukan kualitas nilai piksel dan menetapkan kodeksi radiometrik pada citra (Zybek, 1999; Kowkora, 2004). Pada pokok bahasan selanjutnya akan dibahas mengenai kodeksi radiometrik dan kodeksi geometrik pada citra digital. Kemudian dilanjutkan dengan orientasi citra dan transformasi geometrik.

### 11.1.1. Kodeksi Radiometrik

Nilai intensitas antara minimal (hitam) dan maksimal (putih) dan jarak antara nilai-nilai tersebut merupakan parameter dalam proses kuantisasi (Akkabir et al, 2001; Kowkora, 2004). *Syarat-syarat* dari nilai keabuan (gray value) keabuan citra akan nilai citra juga menggunakan hubungan tersebut (Akkabir, 1997). Untuk alasan praktis, nilai 256 juga banyak digunakan (Kowkora, 2004). Sebuah citra yang memiliki karakteristik resolusi radiometrik 8-bit, nilai

intensitasnya di representasikan dengan 8-bit. Tingkat optimalnya tergantung dari aplikasinya. Citra dengan 1-bit cukup untuk menampilkan hasil pada pengoperasian seperti *edge detection* atau *image segmentation*. Citra dengan resolusi tinggi, misalnya 12-bit atau 16-bit dibutuhkan oleh aplikasi penginderaan jauh untuk membedakan corak termasuk nilai keabuan. Seperti yang telah dijelaskan terlebih dahulu, citra hitam dan putih digunakan dalam percobaan pada penelitian ini. Pengukuran secara otomatis dalam citra berwarna dapat digunakan dengan mengkonversikan nilai merah, hijau, dan biru kedalam intensitas *hue* dan *saturation* (Mikhail et al., 2001). Nilai rata-rata keabuan  $g_m$  dan standar deviasi  $\sigma_r$  (Persamaan 2.1) merupakan dua karakteristik statistik yang memberikan informasi tentang *brightness* dan *contrast* dari citra (Schenk, 1999).

$$g_m = \frac{1}{RC} \sum_{r=1}^R \sum_{c=1}^C g(r, c) \quad \sigma_r = \sqrt{\frac{\sum_{r=1}^R \sum_{c=1}^C (g(r, c) - g_m)^2}{RC - 1}} \quad (2.1)$$

Keterangan:

- $g_m$  : nilai rata-rata keabuan
- $\sigma_r$  : standart deviasi
- R,C : baris dan kolom dari citra
- $g(r, c)$  : nilai keabuan setiap piksel pada posisi  $r, c$

Frekuensi dari setiap nilai keabuan dalam citra dapat ditunjukkan dengan histogram.

intensitasnya di representasikan dengan 8-bit. Tingkat optimalnya terganggu dari aplikasinya. Citra dengan 1-bit cukup untuk menampilkan hasil pada pengoperasian seperti edge detection atau image segmentation. Citra dengan resolusi tinggi misalnya 12-bit atau 16-bit dibutuhkan oleh aplikasi penginderaan jauh untuk membedakan corak termasuk nilai keabuan. Seperti yang telah dijelaskan terlebih dahulu, citra hitam dan putih digunakan dalam percobaan pada penelitian ini. Pengukuran secara otomatis dalam citra berwarna dapat digunakan dengan mengkonversikan nilai merah, hijau, dan biru kedalam intensitas was dan kemudian (Mikolaj et al., 2001). Nilai rata-rata keabuan  $\bar{g}_m$  dan standar deviasi  $\sigma_g$  (Persamaan 2.1) merupakan dua karakteristik statistik yang memberikan informasi tentang brightness dan contrast dari citra (Zawak, 1999).

$$\bar{g}_m = \frac{1}{RC} \sum_{c=1}^R \sum_{r=1}^C g(r,c) \quad \sigma_g = \sqrt{\frac{\sum_{c=1}^R \sum_{r=1}^C (g(r,c) - \bar{g}_m)^2}{RC-1}} \quad (2.1)$$

Keterangan:

- $\bar{g}_m$  : nilai rata-rata keabuan
- $\sigma_g$  : standar deviasi
- $R, C$  : baris dan kolom dari citra
- $g(r,c)$  : nilai keabuan setiap piksel pada posisi  $r, c$

Frekuensi dari setiap nilai keabuan dalam citra dapat ditunjukkan dengan histogram.



### II.1.1.2. Koreksi Geometrik

Distorsi pada citra dapat disebabkan oleh sejumlah faktor diantaranya adalah kondisi optik dari sensor, pergerakan dari sistem *scanner*, pergerakan dari wahana, ketinggian terbang wahana, kecepatan wahana, kondisi relief dari bentang alam di bumi, dan pergerakan rotasi bumi (*Lillesand and Kiefer, 2000*). Koreksi geometrik bertujuan untuk memperbaiki suatu citra dari distorsi geometrik agar diperoleh citra dengan sistem proyeksi dan koordinat seperti yang ada pada peta.

Ukuran dan bentuk dari “*picture element*” atau *pixel* merupakan dasar dari koreksi geometrik pada citra digital. Ukuran piksel (resolusi geometrik) merupakan faktor yang mempengaruhi ketelitian pengukuran (*Potuckova, 2004*). Pertama, untuk mengenali objek yang mengalami efek acak (*noise*) pada citra, objek harus memuat 2-3 piksel. *Ground sample distance (gsd)* dihitung untuk memperoleh gambaran detail nyata dalam citra. Kedua, penelitian yang menunjukkan tentang akurasi antara  $1/2 - 1/3$  piksel dapat diterima untuk pengukuran secara manual (*Kraus, 1997*). Distorsi geometrik bersifat random maka koreksinya membutuhkan sejumlah titik kontrol (*ground control point (gcp)*) menghasilkan ketelitian  $1/5$  piksel (*Hahn, 1997*). Dalam aplikasi fotogrametri jarak dekat, akurasi dapat mencapai  $1/1000$  piksel, tergantung dari kualitas target dan teknik aplikasinya (*Luhmann, 2000*).

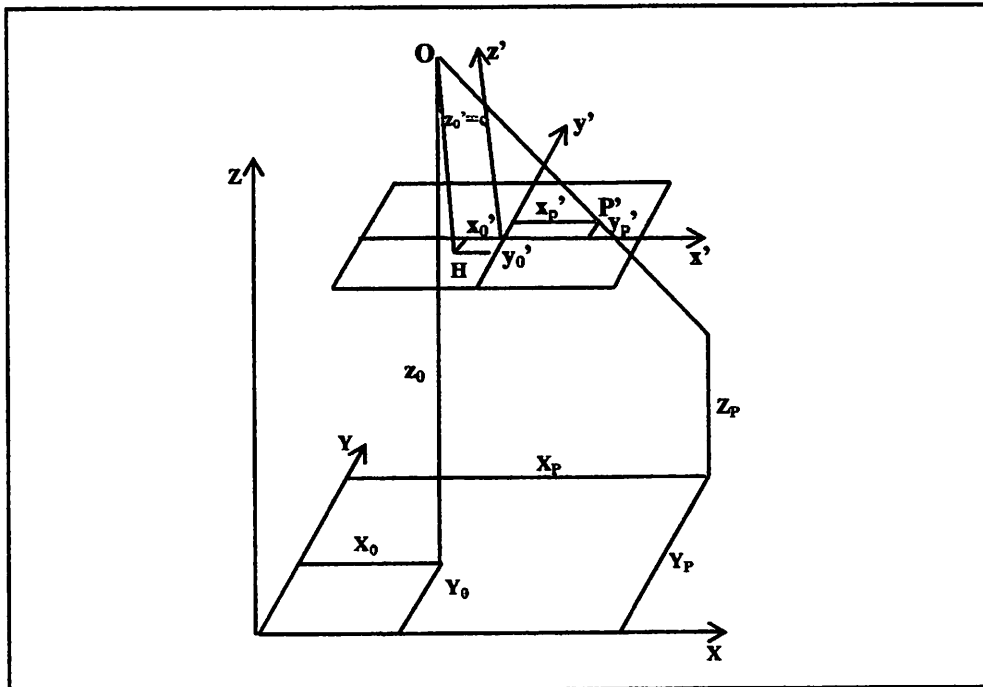
### 11.1.1.2. Koreksi Geometrik

Distorsi pada citra dapat disebabkan oleh sejumlah faktor diantaranya adalah kondisi optik dari sensor, pergerakan dari sistem kamera, pergerakan dari wahana, ketidngan terbang wahana, kecepatan wahana, kondisi relief dan bentuk alam di bumi, dan pergerakan rotasi bumi (Alfenz and Kiefer, 2000). Koreksi geometrik bertujuan untuk memperbaiki suatu citra dari distorsi geometrik agar diperoleh citra dengan sistem proyeksi dan koordinat seperti yang ada pada peta.

Ukuran dan bentuk dari "view frame" atau view merupakan dasar dari koreksi geometrik pada citra digital. Ukuran piksel (resolusi geometrik) merupakan faktor yang mempengaruhi ketelitian pengukuran (Tomaskovic, 2004). Pertama, untuk mengolah objek yang mengalami efek gerak (gerak) pada citra, objek harus memuat 2-3 piksel (view sample interval) diiringi untuk memperoleh gambaran dalam suatu dalam citra. Kedua, penelitian yang menunjukkan tentang akurasi antara 1/2 - 1/3 piksel dapat diterima untuk pengukuran secara manual (Kwon, 1997). Distorsi geometrik bersifat random maka koreksinya membutuhkan sejumlah trik kontrol ground control point (gcp) menghasilkan ketelitian 1/2 piksel (Wahr, 1997). Dalam aplikasi fotogrametri jarak dekat akurasi dapat mencapai 1/1000 piksel tergantung dari kualitas target dan teknik aplikasinya (Almami, 2000).

### II.1.1.3. Orientasi dan Georeferensi

Untuk menentukan koordinat objek pada titik-titik yang akan diukur pada citra/foto merupakan tugas pokok dalam fotogrametri. Hubungan geometris antara citra/foto dengan sistem koordinat objek (pusat proyeksi) ditunjukkan pada Gambar 2.2.



Gambar 2.2 Pusat proyeksi. Hubungan antara koordinat foto  $x'_p, y'_p$  dan koordinat objek  $X_p, Y_p, Z_p$ , pada titik P.  $x'$  dan  $y'$  dari sistem koordinat sama dengan image plane.  $Z'=0$  untuk semua titik diukur pada foto. Lihat persamaan 2.2 untuk memperjelas simbol yang lain

Persamaan kolinear :

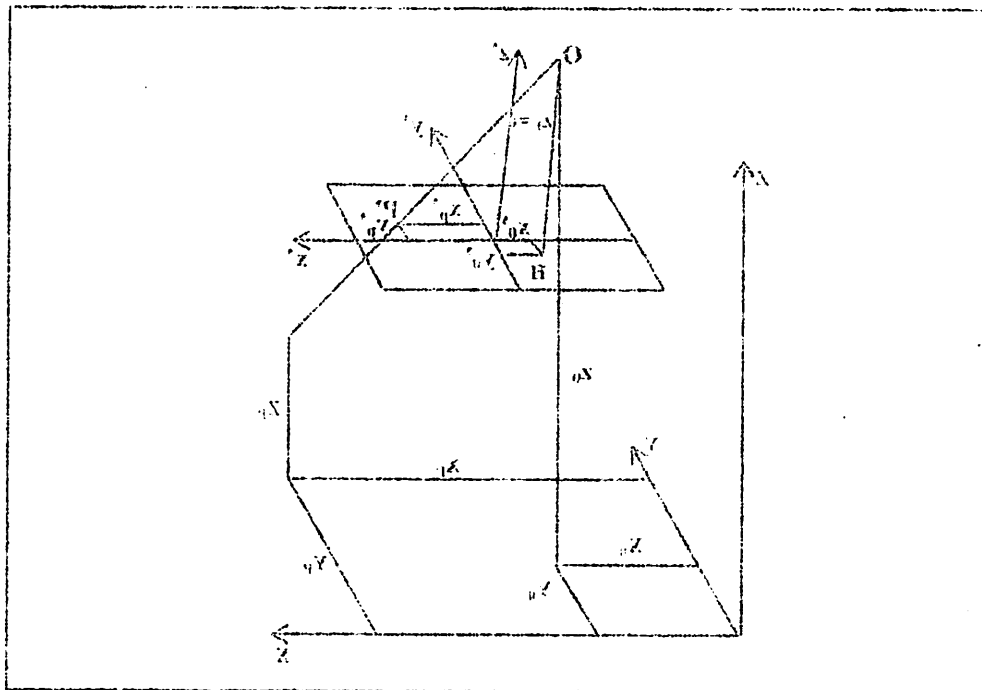
$$x'_p = x'_0 - c \frac{r_{11}(X_p - X_0) + r_{12}(Y_p - Y_0) + r_{13}(Z_p - Z_0)}{r_{31}(X_p - X_0) + r_{32}(Y_p - Y_0) + r_{33}(Z_p - Z_0)} = x'_0 - c \frac{U}{W} \quad (2.2a)$$

$$y'_p = y'_0 - c \frac{r_{21}(X_p - X_0) + r_{22}(Y_p - Y_0) + r_{23}(Z_p - Z_0)}{r_{31}(X_p - X_0) + r_{32}(Y_p - Y_0) + r_{33}(Z_p - Z_0)} = y'_0 - c \frac{V}{W} \quad (2.2b)$$

### 11.1.3. Orientasi dan Geometri

Untuk menentukan koordinat objek pada titik-titik yang akan diukur pada citra foto merupakan tugas pokok dalam fotogrametri. Hubungan geometris antara citra foto dengan sistem koordinat objek (pusat proyeksi) ditunjukkan pada

Gambar 2.2.



Gambar 2.2. Pusat proyeksi. Hubungan antara koordinat foto  $x', y', z'$  dan koordinat objek  $x, y, z$ . Pada titik  $P$ , dan  $P'$  dari sistem koordinat sama dengan imaji. Jarak  $X_0$  titik sumbu tak diukur pada foto. Lihat persamaan 2 untuk memperjelas simbol yang lain.

Persamaan kolinear :

$$(2.2a) \quad \frac{U}{W} = x'_0 = \frac{(x_0 - X_0) + f_1 \frac{z}{z_0} + f_2 \frac{y}{z_0}}{(x_0 - X_0) + f_2 \frac{z}{z_0} + f_3 \frac{y}{z_0}}$$

$$(2.2b) \quad \frac{V}{W} = y'_0 = \frac{(x_0 - X_0) + f_2 \frac{z}{z_0} + f_3 \frac{y}{z_0}}{(x_0 - X_0) + f_2 \frac{z}{z_0} + f_3 \frac{y}{z_0}}$$

Keterangan :

- $x'_p, y'_p$  : koordinat foto pada titik P
- $X_p, Y_p, Z_p$  : koordinat objek pada titik P
- $X_o, Y_o, Z_o$  : koordinat objek pada *perspective centre* O
- $r_{ij}$  : elemen matriks rotasi ( $\omega$ ,  $\phi$ , dan  $\kappa$ )
- $c$  : *principle distance* ( konstanta kamera)
- $x'_o, y'_o$  : koordinat foto pada *principal point* H

Parameter *interior orientation* (koordinat foto dari *principal point*, *principle distance* ( $f$ ) dan distorsi lensa) dan parameter *exterior orientation* (koordinat posisi kamera dan parameter rotasi) harus dipecahkan terlebih dahulu. Parameter *interior orientation* biasanya diambil dari informasi kalibrasi kamera. *Interior orientation* juga dapat ditentukan melalui proses perhitungan pada *exterior orientation* (*self-calibration*). Dalam kasus pemindahan foto (*scanned analogue photograph*), hubungan antar foto dan sistem koordinat piksel harus diselaraskan. Semua pengukuran dilakukan dalam sistem koordinat piksel. Dalam mengoreksi distorsi lensa dan kemungkinan menggunakan persamaan kolinear, maka diperlukan pengukuran posisi kedalam sistem koordinat foto. Hubungan tersebut akan dibahas pada pokok bahasan berikut ini.

- Keterangan :
- $x'_p, y'_p$  : koordinat foto pada titik P
  - $x_p, y_p, \Delta p$  : koordinat objek pada titik P
  - $x_0, y_0, \Delta_0$  : koordinat objek pada perspective center O
  - $r, \phi$  : elemen matriks rotasi ( $\phi, \phi'$  dan  $\kappa$ )
  - $c$  : principle distance (konstanta kamera)
  - $x'_0, y'_0$  : koordinat foto pada principal point H

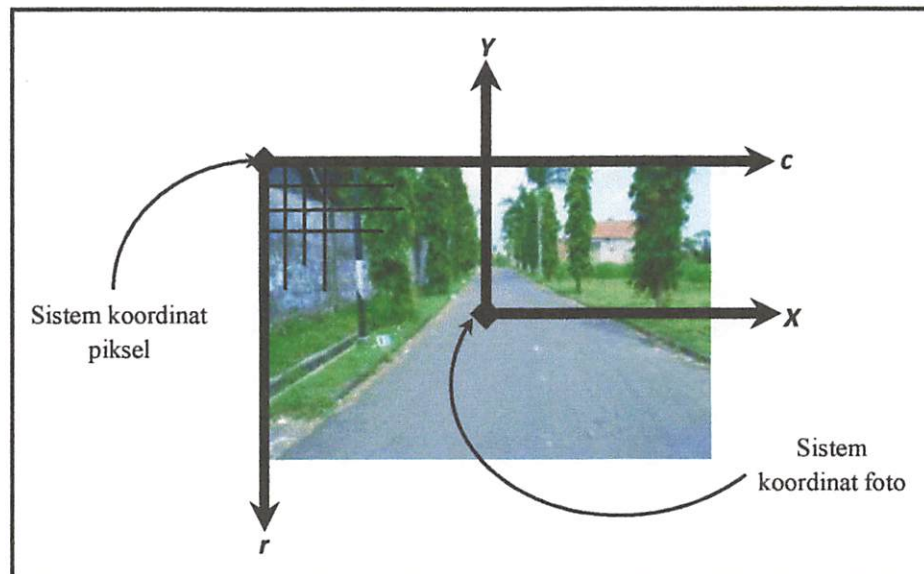
Parameter interior orientation (koordinat foto dari principal point, principle distance ( $c$ ) dan distorsi lensa) dan parameter exterior orientation (koordinat posisi kamera dan parameter rotasi) harus dipisahkan terlebih dahulu. Parameter interior orientation biasanya diambil dari informasi kalibrasi kamera. Interior orientation juga dapat ditentukan melalui proses perhitungan pada exterior orientation (self-calibration). Dalam kasus pemindaian foto (scanned analogues photographs), hubungan antar foto dan sistem koordinat piksel harus diselesaikan. Semua pengukuran dilakukan dalam sistem koordinat piksel. Dalam mengoreksi distorsi lensa dan kemiringan menggunakan persamaan kolinear maka diperlukan pengukuran posisi kedalaman sistem koordinat foto. Hubungan tersebut akan dibahas pada pokok bahasan berikut ini.

### II.1.1.3.1 Konversi Sistem koordinat

Koordinat titik  $(x,y)$  yang dihasilkan dari metode penentuan titik konjugasi masih berbentuk koordinat piksel maka perlu dikonversi kedalam koordinat foto.

#### II.1.1.3.1.1 Konversi koordinat piksel ke koordinat foto

Koordinat citra digital diperoleh dari sistem koordinat piksel. Sistem koordinat piksel biasanya merupakan sistem koordinat dengan pusat di pojok kiri atas kemudian  $x$  ke arah kanan dan  $y$  ke arah bawah, dan bagian dari piksel dinotasikan sebagai  $r$  dan  $c$  pada Gambar 2.3 (Leica Geosystems, 2006)



Gambar 2.3. Sistem koordinat piksel dan sistem koordinat foto

Gambaran sistem koordinat kartesian pada susunan CCD (*Charge Coupled Device*) diatas, maka persamaan untuk mengkonversi koordinat piksel ke koordinat foto adalah sebagai berikut:

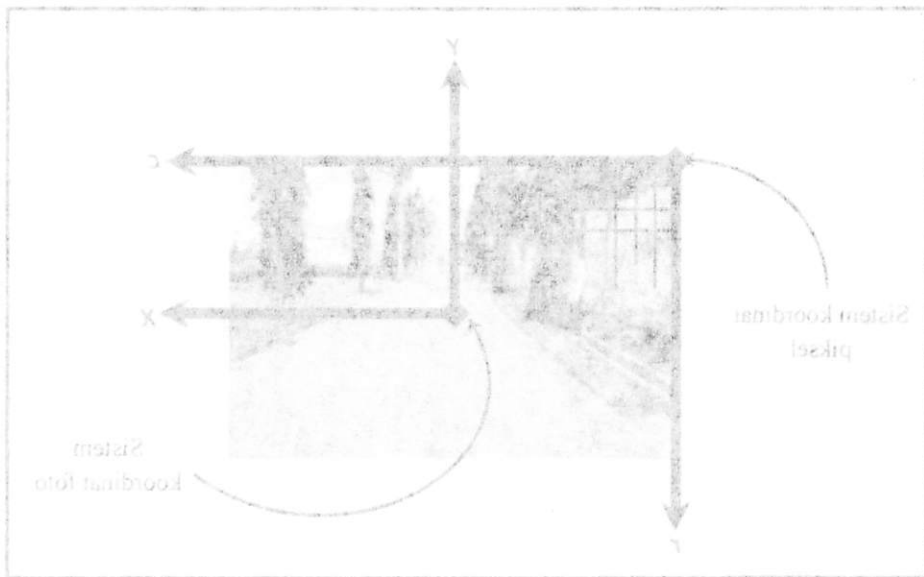
### 11.1.3.1 Konversi sistem koordinat

Koordinat titik  $(x, y)$  yang dihasilkan dari metode penentuan titik konjugasi masih berbentuk koordinat piksel maka perlu dikonversi ke dalam koordinat foto.

#### 11.1.3.1.1 Konversi koordinat piksel ke koordinat foto

Koordinat titik digital diperoleh dari sistem koordinat piksel. Sistem koordinat piksel biasanya merupakan sistem koordinat dengan pusat di pojok kiri atas kemudian  $x$  ke arah kanan dan  $y$  ke arah bawah, dan bagian dari piksel dinotasikan sebagai  $x$  dan  $y$  pada Gambar 2.3 (Aver

(Geogiswars, 2006)



Gambar 2.3. Sistem koordinat piksel dan sistem koordinat foto

Gambar sistem koordinat kartesian pada sensor CCD (Charge Coupled Device) data, maka persamaan untuk mengkonversi koordinat piksel ke koordinat foto adalah sebagai berikut.



$$X_{Foto} = (X - x'_c) * \text{ukuran piksel } x \quad (2.3a)$$

$$Y_{Foto} = (y'_c - Y) * \text{ukuran piksel } y \quad (2.3b)$$

$$x'_c = \left( \frac{nx'}{2} \right) - 0.5 \quad (2.4a)$$

$$y'_c = \left( \frac{ny'}{2} \right) - 0.5 \quad (2.4b)$$

Dari persamaan diatas,  $nx'$  adalah jumlah piksel pada kolom ( $c$ ) dan  $ny'$  adalah jumlah piksel pada baris ( $r$ ), dan  $X, Y$  adalah koordinat piksel, sedangkan  $x'_c$  dan  $y'_c$  dihitung menggunakan Persamaan (2.4a) dan (2.4b).

#### II.1.1.3.1 Konversi koordinat foto ke koordinat piksel

Persamaan untuk mengkonversi koordinat foto ke koordinat piksel adalah sebagai berikut:

$$X_{piksel} = \left( \frac{X_{foto}}{\text{ukuran piksel } x} \right) + x'_c \quad (2.5a)$$

$$Y_{piksel} = \left( \frac{Y_{foto}}{\text{ukuran piksel } y} \right) + y'_c \quad (2.5b)$$

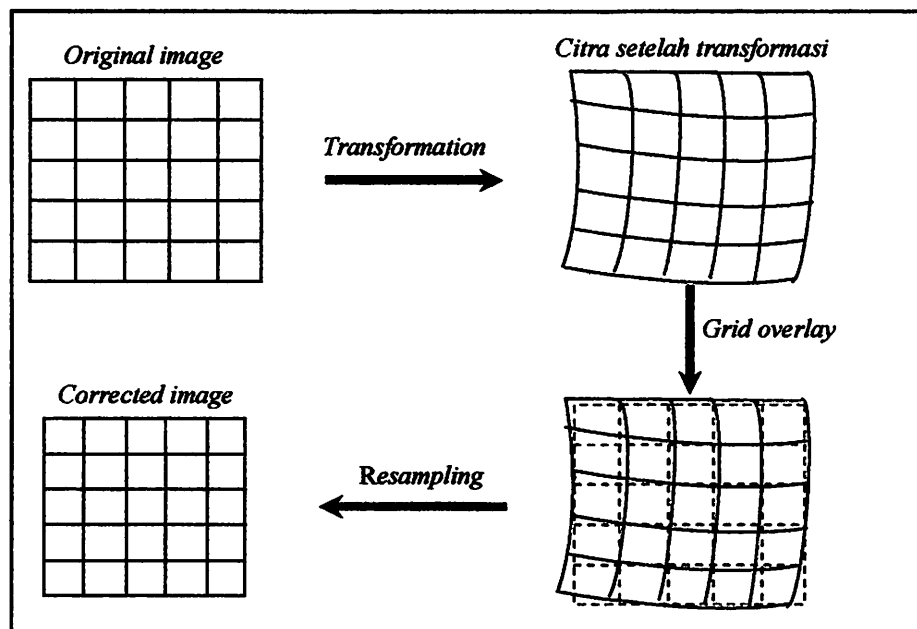
Dari persamaan diatas,  $x'_c$  dan  $y'_c$  dihitung menggunakan Persamaan (2.4a) dan (2.4b).

Proses untuk menemukan posisi pada sebuah citra dalam koordinat sistem referensi disebut dengan georeferensi (Potuckova, 2004).

#### II.1.1.4 Transformasi Geometrik dan *Resampling*

Untuk mengoreksi semua distorsi, diperlukan suatu sistem transformasi untuk mengoreksikan posisi antara citra satu dengan citra lainnya. Teknik transformasi yang umumnya digunakan adalah transformasi *affine* (Potuckova, 2004).

Setelah citra di-georeferensi, citra akan mempunyai koordinat untuk masing-masing pikselnya tetapi orientasi geometri belum sesuai dengan citra acuannya. Untuk membuat citra tersebut sesuai dan sama dengan acuannya maka perlu dilakukan proses *resampling*. *Resampling* adalah proses untuk menentukan *digital number* (DN) kedalam lokasi baru yang tepat dan hasilnya adalah citra *output* yang telah terkoreksi dari distorsi geometrik. Proses *resampling* disajikan pada gambar berikut (Thévenaz et al., 1999)



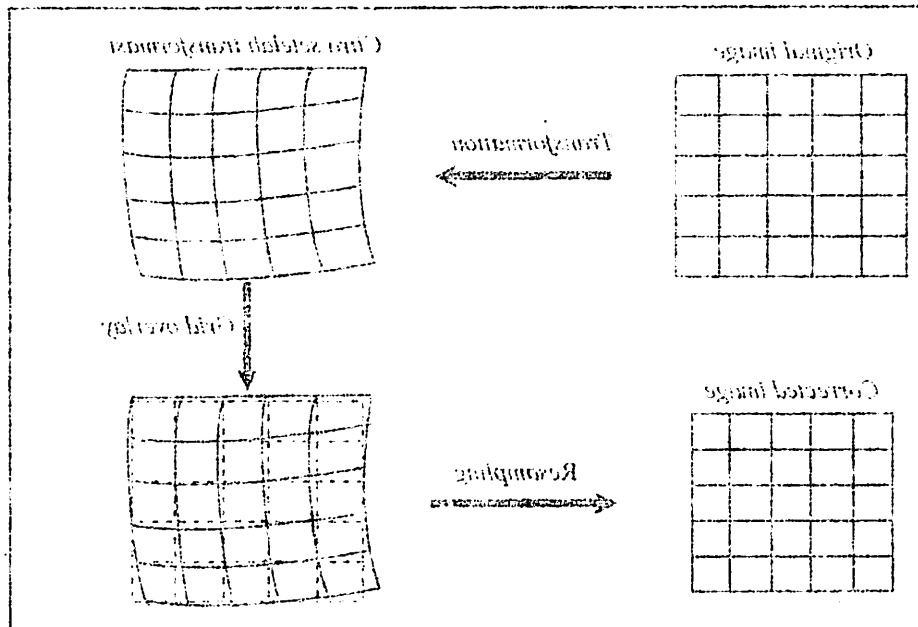
Gambar 2.4 Proses *resampling*

### 11.1.1.4 Transformasi Geometrik dan Resampling

Untuk mengoreksi semua distorsi diperoleh suatu sistem transformasi untuk mengoreksikan posisi antara citra satu dengan citra lainnya. Teknik transformasi yang umumnya digunakan adalah transformasi affine (Pottkover, 2004).

Setelah citra di-georeferensi, citra akan mempunyai koordinat untuk masing-masing pikselnya tetapi orientasi geometri belum sesuai dengan citra acuan. Untuk membuat citra tersebut sesuai dan sama dengan acuan maka perlu dilakukan proses resampling. Resampling adalah proses untuk menentukan digital number (DN) kedalaman lokasi baru yang tepat dan hasilnya adalah citra output yang telah terdistorsi dari distorsi geometrik. Proses resampling dilakukan

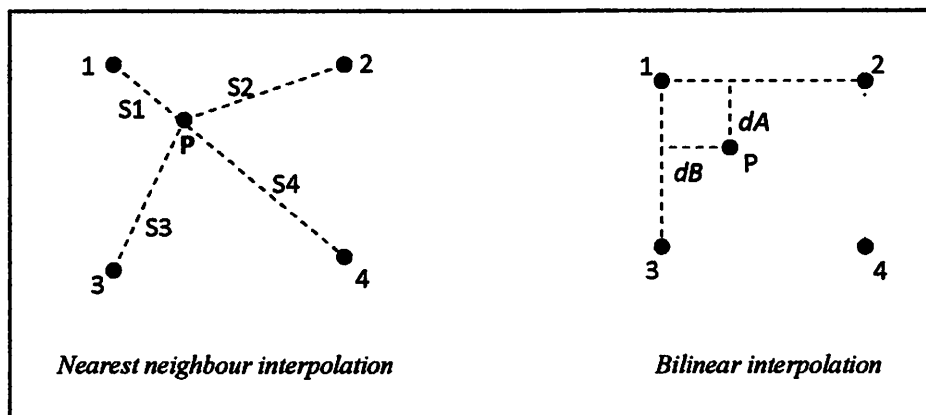
pada gambar berikut (Vibran et al., 1999)



Gambar 2.4 Proses resampling

#### II.1.1.4.1 Proses *resampling*

Proses *resampling* terdiri dari dua tahap yaitu tahap pertama, citra baru yang belum terkoreksi perlu dilakukan proses proyeksi dengan menggunakan teknik transformasi kedalam citra acuannya. Tahap kedua dilakukan relasi satu-satu antara *digital number* citra input dengan citra acuannya dengan menggunakan metode interpolasi (Thévenaz et al., 1999)



Gambar 2.5 Nearest neighbour dan bilinear interpolation (diadopsi dari potuckova, 2004)

#### II.1.1.4.2 Proses interpolasi

Interpolasi merupakan metode untuk merekonstruksi point data baru dari point data yang telah diketahui dengan fungsi tertentu. Nilai hasil interpolasi merupakan nilai estimasi. Pada umumnya interpolasi terbagi menjadi interpolasi *adaptive* dan *non-adaptive*. Pada interpolasi *non-adaptive*, proses komputasi dilakukan merata pada semua *piksel*, contoh metode ini di antaranya *nearest neighbour*, *bilinear interpolation*, dan *cubic convolution*. Sedangkan pada interpolasi *adaptive*, proses komputasi dilakukan berdasarkan kriteria konten tertentu, misalnya memberlakukan

proses yang berbeda pada frekuensi tinggi dan frekuensi rendah pada citra (*Thévenaz et al.*, 1999; *Sachs*, 2001)

Berikut ini merupakan penjelasan singkat mengenai proses interpolasi *non-adaptive* (*Thévenaz et al.*, 1999; *Sachs*, 2001)

1. *Nearest neighbour* : merupakan metode yang sederhana dan cepat, dimana nilai piksel dihitung berdasarkan nilai piksel yang terdekat dari citra aslinya.
2. *Bilinear interpolation* : nilai piksel dihitung berdasarkan rata-rata dari empat piksel terdekat dari citra aslinya.
3. *Cubic convolution* : nilai citra piksel *output* didasarkan pada 16 piksel disekitarnya.

### II.1.2 Pencocokan Citra (*Image Matching*)

Menemukan titik konjugasi dalam dua atau lebih foto yang saling bertampalan secara otomatis merupakan dasar dalam fotogrametri digital. Proses ini biasanya disebut dengan pencocokan citra atau yang lebih dikenal dengan *image matching* (*Schenk*, 1999; *Aguoris et al*, 2000 ). Gruen (1985) mengkaji bahwa, jika pertampalan dinilai cukup maka pada citra yang sama nilai tersebut dapat diuraikan dengan suatu transformasi antar objek dalam kedua citra. Dalam transformasi ini terdapat 8 parameter dan dapat dilakukan pendekatan dengan menggunakan transformasi *affine* (6 parameter). Pendekatan ini untuk menyamakan keakurasian hingga mencapai akurasi subpiksel. Beberapa teknik pencocokan citra (*image matching*) adalah *area-based matching*, *feature-based*

*matching* dan *symbolic matching* (Schenk, 1999; Wolf dan Dewitt, 2000; Potuckova, 2004; Leica Geosystems, 2006). Hubungan antara setiap metode dengan entitasnya diperlihatkan pada tabel berikut:

Tabel 2.1 Metode *image matching*

Metode <i>Image Matching</i>	Teknik Perhitungan Pencocokan Citra	Entitas
<i>Area-based</i>	<i>Normalized Cross Correlation, Least Squares Matching</i>	Derajat keabuan ( <i>grey level</i> )
<i>Feature-based</i>	Fungsi <i>cost</i>	Titik, tepi, daerah
<i>Symbolic matching</i>	Fungsi <i>cost</i>	Keterangan simbol

Penelitian ini dititikberatkan pada metode *area-based* dan teknik perhitungannya menggunakan teknik *Normalized Cross Correlation* (NCC) dan *Least Squares Matching* (LSM).

Keunggulan teknik NCC adalah kecepatan komputasinya, tetapi hanya menghasilkan ketelitian 1 piksel (Mikhail, Bethel et al., 2001). Sebaliknya dengan metode LSM, karena menerapkan hitung kuadrat terkecil pada nilai keabuan, relatif lebih lambat untuk konvergen namun mampu menghasilkan ketelitian hingga 0,01 piksel (Luhmann, Robson et al. 2006). Disamping itu, kelemahan lain dari LSM adalah teknik LSM membutuhkan nilai pendekatan posisi yang cukup

dekat terhadap nilai sebenarnya agar perhitungan iterasinya dapat konvergen (Gruen, 2001).

Berdasarkan karakteristik masing-masing metode yang saling melengkapi, pada penelitian ini digunakan teknik NCC untuk mendapatkan nilai awal titik konjugasi yang dicari, lalu posisi dan ketelitian titik konjugasi ini dihitung ulang dengan teknik LSM untuk mendapatkan hasil yang lebih baik.

### II.1.2.1 Metode *Area-Based*

Nilai keabuan (*gray value*) merupakan entitas dari metode *area-based*. *Image patch* diambil dari citra pertama yang kemudian disebut sebagai *template*, dan yang akan dicari pada citra kedua. *Template* biasanya berukuran  $m \times n$  piksel, atau  $m = n$ . Pusat *template* berada pada piksel tengah dari ukuran *template*, sehingga biasanya *template* berukuran ganjil. Nilai korelasi antara *template* dan *matching window* dihitung untuk memperoleh posisi objek pada foto kedua (*search window*). Untuk menghindari ketidakcocokan (*mismatch*), posisi pada *search window* harus ditentukan lebih teliti dalam metode ini. Ketika bekerja dengan foto stereo, dapat diterapkan juga prinsip *epipolar line* (Gambar 2.6). *Epipolar line* merupakan *intersection* dari *epipolar plane* dan *image plane*. *Epipolar plane* diperoleh dari proyeksi  $O_1$ ,  $O_2$  dan titik objek  $P$ . Oleh karena itu, titik konjugasi  $P'$  dan  $P''$  dimisalkan sebagai hubungan antara *epipolar line*  $e'$  dan  $e''$ . Agar *matching* sepanjang *epipolar line* lebih mudah maka citra dapat ditransformasikan terlebih dahulu atau yang disebut normalisasi citra (Potuckova, 2004).

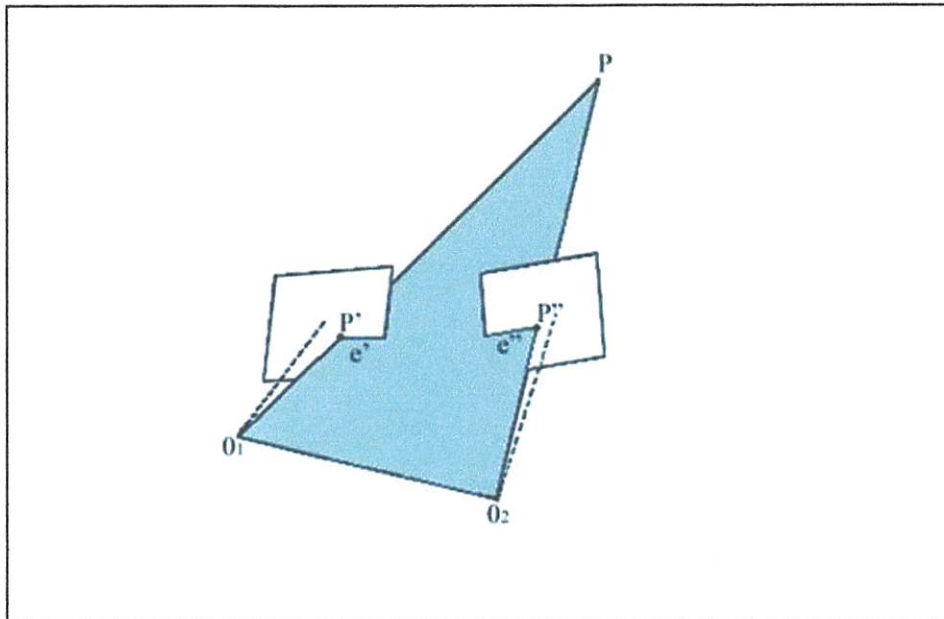
lekat terhadap nilai sebenarnya saat perhitungan tersebut dapat konversi  
(Graw, 2001).

Berdasarkan karakteristik masing-masing metode yang saling melengkapi pada penelitian ini digunakan teknik NCC untuk mendapatkan nilai awal titik konjugasi yang dicari, lalu posisi dan kedalaman titik konjugasi ini dihitung ulang dengan teknik LSM untuk mendapatkan hasil yang lebih baik.

### 11.2.1 Metode Area-Based

Nilai kedalaman (*gray value*) merupakan entitas dari metode *area-based* yang dapat diambil dari citra pertama yang kemudian disebut sebagai *template* dan yang akan dicari pada citra kedua. *Template* biasanya berukuran  $w \times w$  piksel, atau  $w - 1$  pada *template* berada pada piksel tengah dari ukuran *template*, sehingga biasanya *template* berukuran ganjil. Nilai korelasi antara *template* dan *search window* dihitung untuk memperoleh posisi objek pada foto kedua (*search window*). Untuk menghindari ketidakcocokan (*mismatch*), posisi pada *search window* harus ditentukan lebih teliti dalam metode ini. Ketika bekerja dengan foto stereo dapat ditradisikan juga prinsip *epipolar line* (Gamonar, 2004). *Epipolar line* merupakan interseksi dari *epipolar plane* dan *image plane*. *Epipolar plane* diperoleh dari proyeksi  $O_1, O_2$  dan titik objek  $P$ . Oleh karena itu titik konjugasi  $P_1$  dan  $P_2$  dimisalkan sebagai hubungan antara *epipolar line* dan  $e_1$ . Agar *search window* sepanjang *epipolar line* lebih mudah maka citra dapat ditransformasikan terlebih dahulu yang disebut *normalisasi citra* (Ponckova, 2004).





Gambar 2.6 Epipolar plane di proyeksikan dari pusat  $O_1$  dan  $O_2$  serta titik objek  $P$

Keterangan gambar :

$P$  : titik objek

$P'$ ,  $P''$  : titik objek pada foto

$e'$ ,  $e''$  : *intersection* antara *epipolar plane* dan *image plane*

$O_1, O_2$  : *epipolar plane*

Ukuran dari *search window* tergantung pada seberapa presisi lokasinya dan deformasi geometrik serta orientasi dari *image*.

Pada pokok bahasan berikut akan dikaji tentang teknik yang digunakan dalam pencocokan citra (*image matching*) menggunakan metode *area-based* yaitu *Normalized Cross Correlation* (NCC) dan *Least Squares Matching* (LSM).

### II.1.2.1.1 Normalized Cross Correlation (NCC)

*Normalized cross correlation* ( $r$ ) merupakan salah satu pengukuran titik konjugasi yang digunakan dalam fotogrametri. Prinsip *normalized cross correlation* yaitu mencari pasangan titik piksel antara citra pertama dengan citra pasangan/citra kedua. Pada citra pertama ditentukan jendela sasaran (*template*) yang memuat titik piksel yang akan dicari pasangannya pada citra kedua. Pada citra kedua ditentukan daerah selidik (*search window*) yang mempunyai ukuran lebih besar dari pada daerah sasaran (*template*). Pada daerah selidik (*search window*) dibentuk pula jendela/daerah sub selidik (*matching window*) dengan ukuran yang sama dengan jendela/daerah sasaran (*template*). *Matching window* ini bergerak (*moving window*) dengan *increment* 1 piksel sepanjang setiap baris dan kolom di daerah selidik (*search window*). Dihitung nilai korelasi ( $r$ ) antara *template* dan *matching window*. Nilai korelasi antara dua kelompok data *gray value* dihitung berdasarkan rumus matematis pada persamaan berikut (Mitchell dan Pilgrim, 1987; Schenk, 1999; Wolf dan Dewitt, 2000; Campbell et al., 2008) :

$$s_x^2 = \sum x_i^2 - \frac{(\sum x_i)^2}{n} \quad (2.6a)$$

$$s_y^2 = \sum y_i^2 - \frac{(\sum y_i)^2}{n} \quad (2.6b)$$

$$s_{xy} = \sum [x_i y_i] - \frac{(\sum x_i)(\sum y_i)}{n} \quad (2.6c)$$

$$\beta = \frac{S_{xy}}{S_x^2} \quad (2.6d)$$

$$\alpha = -\beta \frac{\sum x_i}{n} \quad (2.6e)$$

$$r = \frac{\sigma_{TS}}{\sigma_T \sigma_S} = \frac{\sum_{i=1}^R \sum_{j=1}^C (g_T(i,j) - \bar{g}_T)(g_S(i,j) - \bar{g}_S)}{\sqrt{\sum_{i=1}^R \sum_{j=1}^C (g_T(i,j) - \bar{g}_T)^2 \sum_{i=1}^R \sum_{j=1}^C (g_S(i,j) - \bar{g}_S)^2}} \quad (2.6f)$$

Keterangan :

$n$  : jumlah data (baris x kolom)

$r$  : koefisien *normalized cross correlation*

$\sigma_T, \sigma_S$  : standar deviasi pada *template* dan *search image*

$\sigma_{TS}$  : kovarian nilai *gray value* pada *template* dan *search image*

: nilai keabuan *template* dan *search image*

: nilai rata-rata dari *template* dan *search image*

R,C :baris dan kolom dari *image patches*



Gambar 2.7 Sepasang citra dalam bentuk visual

(2.64)

$$b = \frac{\sum xy}{\sum x^2}$$

(2.65)

$$a = -b \frac{\sum x}{n}$$

(2.67)

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Keterangan:

- $n$  : jumlah data (baris x kolom)
- $r$  : koefisien normalized cross correlation
- $\sigma_1, \sigma_2$  : standar deviasi pada template dan search image
- $\sigma_{12}$  : koefisien nilai gray value pada template dan search image
- $\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$  : nilai koefisien template dan search image
- $\sum_{i=1}^n (x_i - \bar{x})^2$  : nilai rata-rata dari template dan search image
- $\sum_{i=1}^n (y_i - \bar{y})^2$  : nilai rata-rata dari search image

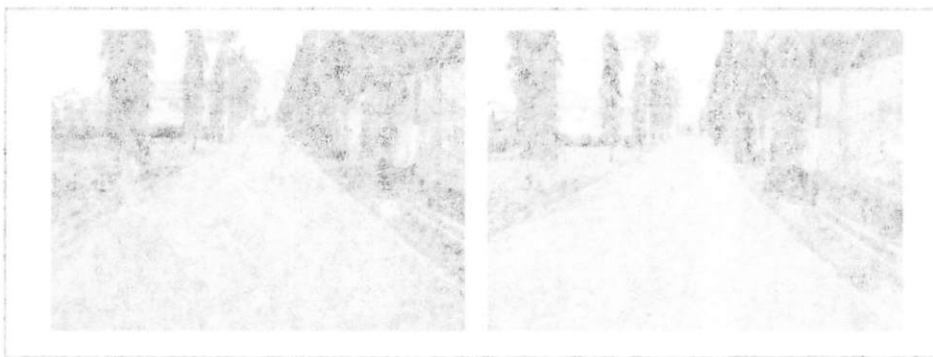


Figure 2.2. Template matching in grayscale images

128	137	140	146	129	128	137	140	146	129
100	114	121	119	110	100	114	121	119	110
110	111	98	71	64	110	111	98	71	64
117	109	93	64	63	117	109	93	64	63
126	110	96	78	80	126	110	96	78	80

Gambar 2.8. Representatif dalam bentuk matriks

Apabila pada foto kiri (Gambar 2.7) ditentukan sebuah objek sebagai titik acuan pencarian, maka mata manusia (operator) akan dengan mudah mengenal dan menemukan objek tersebut pada foto kanan. Tidak demikian halnya pada proses korelasi digital. Komputer harus menentukan objek tersebut pada foto kanan dengan mengamati sekumpulan nilai *gray value* seperti diilustrasikan pada Gambar 2.8. Variasi nilai piksel pada foto akan dipengaruhi oleh beberapa faktor antara lain kualitas dan kuantitas nilai-nilai piksel yang membentuk objek tersebut. Variasi nilai piksel juga akan dipengaruhi oleh tinggi terbang/pemotretan, jenis dan karakteristik permukaan bumi (terrain), serta sudut pemotretan objek oleh kamera.

Dalam domain digital, citra tersebut direpresentasikan sebagai variasi nilai piksel yang membentuk dimensi matriks  $m \times n$  (Gambar 2.8). Kemudian ditentukan sub-matriks berdimensi  $5 \times 5$  di sekeliling titik objek foto kiri (biasa disebut) sebagai *template*. *Template* akan berisi sekumpulan (25) nilai piksel dari piksel disekeliling titik acuan. Pada matriks kanan, ditentukan juga sub-matriks berdimensi sama dengan *template* dan dinamakan sebagai *matching window* yang terdapat dalam *search window*

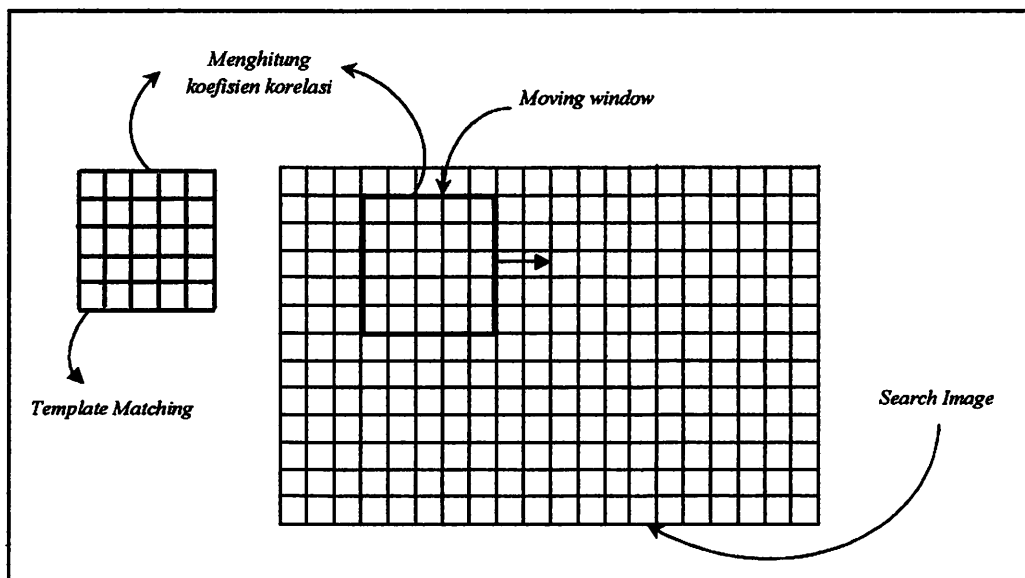
digunakan sebagai wara yang memiliki zona terdistribusi dalam warna. Untuk  
 menentukan jika ada masalah perhitungan zona dengan warna yang  
 (5.2) jika tidak ada tidak diperlihatkan oleh warna pada warna gambar  
 foto ini (pada gambar) sebagai masalah. Masalah akan pada gambar sebagai  
 kemudian akan akan ada masalah perhitungan 2 x 2 di diperlihatkan oleh objek  
 jika tidak ada menggunakan gambar warna m x n (gambar 5.8).  
 (jika gambar diperlihatkan oleh warna dengan menggunakan sebagai warna  
 kemudian pada (gambar) akan akan menggunakan objek oleh gambar  
 diperlihatkan oleh gambar dengan menggunakan warna dan menggunakan  
 jika tidak ada menggunakan objek dengan. Akan ada jika tidak ada akan  
 diperlihatkan oleh diperlihatkan gambar warna pada gambar dan kemudian akan  
 sebagai diperlihatkan pada (gambar 5.2). Akan ada jika tidak ada pada gambar  
 diperlihatkan pada foto gambar dengan menggunakan diperlihatkan oleh gambar  
 warna pada proses perhitungan gambar. Kemudian pada menggunakan objek  
 menggunakan dan menggunakan objek diperlihatkan pada gambar. Tidak diperlihatkan  
 yang akan menggunakan warna akan menggunakan (gambar) akan dengan menggunakan  
 /bentuk pada foto ini (gambar 5.7) diperlihatkan sebagai objek sebagai

Gambar 5.7. Warna dengan gambar warna

158	110	88	58	80	158	110	88	58	80
112	108	83	84	83	112	108	83	84	83
110	111	88	51	84	110	111	88	51	84
100	114	151	118	110	100	114	151	118	110
158	132	170	178	158	158	132	170	178	158

yang tentunya berukuran lebih besar dari *matching window*. Sampai tahap ini, akan diperoleh dua buah matriks (*template* dan *matching window*) dengan dimensi yang identik.

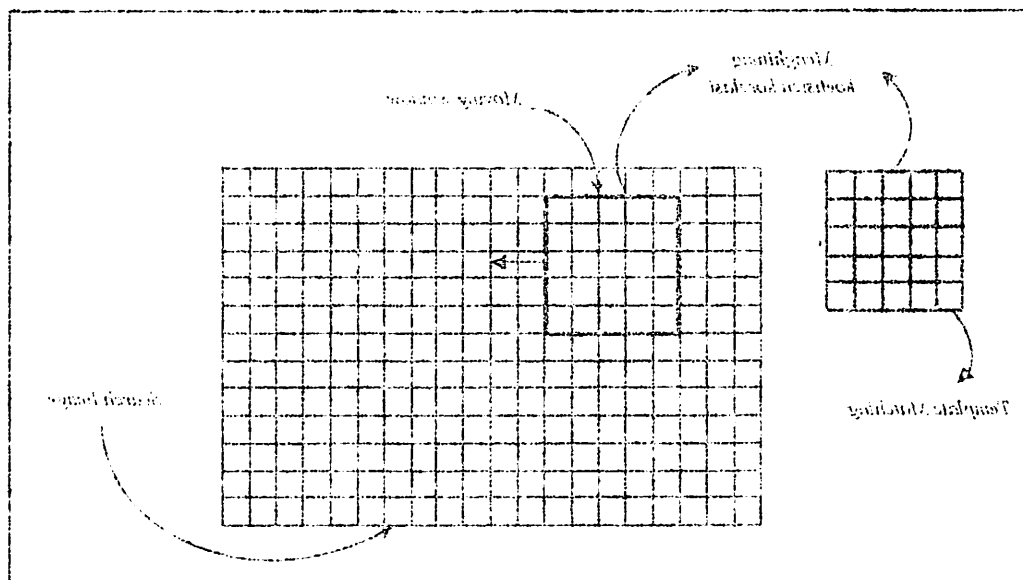
Penempatan *matching window* diawali dari posisi ujung kiri atas. Kemudian *matching window* digeser menelusuri citra kolom demi kolom ke arah kanan sampai mencapai ujung kanan. Setelah itu, *matching window* digeser kebawah sebanyak satu baris dan kembali menelusuri sepanjang baris tersebut ke arah kiri. Demikian seterusnya proses penelusuran (*searching*) dilakukan sampai ke seluruh citra. Untuk setiap tahap penelusuran nilai  $r$  dihitung dan dicatat oleh sistem komputer. Untuk kasus Gambar 2.9 akan diperoleh 25 ( $5 \times 5$ ) nilai  $r$ . Nilai  $r$  yang terbesar menunjukkan posisi *matching image* dengan bentuk objek yang paling mendekati bentuk objek di foto kiri.



Gambar 2.9 Menghitung nilai koefisien korelasi antara template dan search image

yang tentunya semakin lebih besar dari window. Sampai tahap ini, akan diperoleh dua buah matriks (template dan matching window) dengan dimensi yang identik.

Penempatan window window diawali dari posisi ujung kiri atas. Kemudian window window digeser menelusuri cara kolom demi kolom ke arah kanan sampai mencapai ujung kanan. Setelah itu, window window digeser kebawah sebanyak satu baris dan kembali menelusuri sepanjang baris tersebut ke arah kiri. Penelitian seterusnya proses penelusuran (searching) dilakukan sampai ke seluruh cara. Untuk setiap tahap penelusuran nilai  $v$  dihitung dan dicatat oleh sistem komputer. Untuk kasus Gambar 2.9 akan diperoleh 22 ( $2 \times 2$ ) nilai  $v$ . Nilai  $v$  yang terbesar menunjukkan posisi window yang dengan bentuk objek yang paling mendekati bentuk objek di foto kiri.



Gambar 2.9. Template matching with position correlation function and search range



Koefisien *normalized correlation* memiliki nilai antara  $-1 \leq r \leq 1$ . Nilai 1 apabila pencarian hanya jika *image patch*  $g_T$  dan  $g_S$  dihubungkan oleh hubungan linear  $g_T = r_s g_S + r_t, r_s > 0$ , dimana  $r_s$  merupakan faktor skala dan  $r_t$  adalah pergeseran antara nilai keabuan (*gray value*) dalam  $g_T$  dan  $g_S$ . Nilai yang mendekati 0 diindikasikan sebagai nilai yang tidak memiliki korelasi (*mismatch*) dan nilai -1 diperoleh ketika ada kecocokan dari citra positif dan negatif. Kemudian proses pencocokan citra dengan nilai korelasi positif yaitu mendekati 1 merupakan korelasi terbaik atau kedua objek yang dimaksud dapat dianggap sama (Potuckova, 2004, Bradski dan Kaehler, 2008). Nilai korelasi tertinggi itulah yang akan digunakan sebagai nilai pendekatan dalam proses perhitungan *least squares matching* (LSM)

#### II.1.2.1.2 *Least Squares Matching* (LSM)

Koefisien korelasi (*correlation coefficient*) tidak ideal apabila antara kedua *image patches* memiliki geometrik dan radiometrik yang berbeda. *Least Squares Matching* (LSM) merupakan metode untuk menyamakan geometrik dan radiometrik citra dari dua atau lebih *image patches* dari citra referensi (*template*) yang berhubungan dengan *image* kedua (*search image*) (Gruen, 1985; Haleva, 1988; Elaksher, 2008). Metode ini dikembangkan pada tahun 80-an (Bethmann dan Luhmann, 2010). Forstner (1982) membahas tentang pendekatan LSM untuk kasus

Koefisien korelasi (correlation coefficient) memiliki nilai antara  $-1 \leq r \leq 1$ . Nilai 1 apabila pergeseran hanya terjadi pada  $Q_1$  dan  $Q_2$  dihubungkan oleh hubungan linear  $Q_2 = r_1 Q_1 + r_2$ ,  $r_1 > 0$ , dimana  $r_2$  merupakan faktor skala dan  $r_1$  adalah pergeseran antara nilai keabuan (grey tone) dalam  $Q_1$  dan  $Q_2$ . Nilai yang mendekati 0 diindikasikan sebagai nilai yang tidak memiliki korelasi (unrelated) dan nilai -1 diperoleh ketika ada kecocokan dari ciri positif dan negatif. Kemudian proses pencocokan citra dengan nilai korelasi positif yaitu mendekati 1 merupakan korelasi terbalik atau kedua objek yang dimaksud dapat dianggap sama (homogeneous) (Wolcott dan Kwolek, 2008). Nilai korelasi tertinggi ialah yang akan digunakan sebagai nilai pendekatan dalam proses perhitungan level kesamaan matching (LSM).

### 1.1.3.1.3 Least Squares Matching (LSM)

Koefisien korelasi (correlation coefficient) tidak ideal apabila antara kedua gambar memiliki geometrik dan radiometrik yang berbeda. Least Squares Matching (LSM) merupakan metode untuk menyalurkan geometrik dan radiometrik citra dari dua atau lebih gambar (research image) (Günther, 1982; Holroyd, 1982; Kwolek, 2008). Metode ini dikembangkan pada tahun 80-an (Wahman dan Ahumada, 2010). Forner (1982) membahas tentang pendekatan LSM untuk kasus

1-Dimensi (diterapkan pada *image line*). Ackermann (1984), Pertl (1984) dan Gruen (1985) mengadopsi ide tersebut dan meningkatkannya dengan menambahkan parameter geometrik dan radiometrik ke dalam kasus 2-Dimensi dengan *squares* atau *rectangle patches*. Mereka juga menggunakan transformasi *affine* sebagai model linear geometriknya.

#### II.1.2.1.2.1 Prosedur perhitungan *Least Squares Matching* (LSM)

*Least-Square Matching* adalah suatu cara melakukan penyelesaian masalah yang tidak linier. Masalah tersebut dapat diselesaikan dengan model iterasi (Makarovic, 1984; Sarjakoski dan Lammi, 1996; Schenk, 1999). LSM juga memperkecil perbedaan nilai keabuan antara *template* dan *search window* dalam proses perataan dimana koreksi geometrik dan radiometrik pada *matching window* ditentukan (Schenk, 1999). Hubungan antara nilai keabuan dari dua *image patch* tersebut dapat dinyatakan dengan persamaan berikut (Wolf dan Dewitt, 2000)

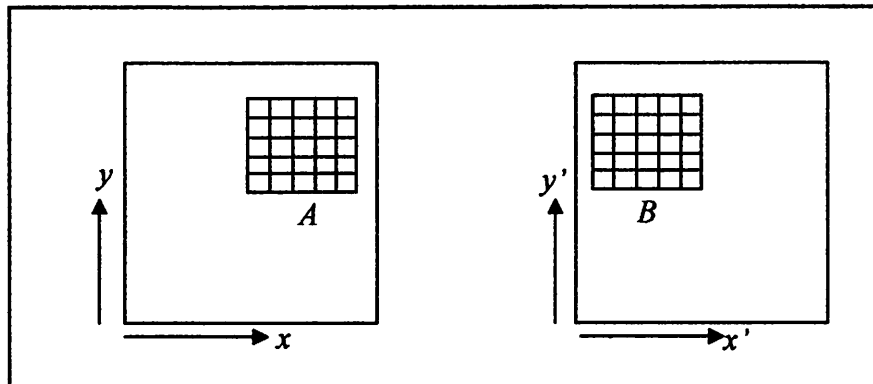
$$A(x, y) = h_0 + h_1 B(x', y') \quad (2.7)$$

$$x' = a_0 + a_1 x + a_2 y \quad (2.8)$$

$$y' = b_0 + b_1 x + b_2 y \quad (2.9)$$

Dimana,  $A(x, y)$  merupakan DN dari kandidat *subarray* dari foto kiri pada lokasi  $x, y$ ;  $B(x', y')$  merupakan DN dari *subarray* di foto kanan pada lokasi  $x', y'$ ;  $h_0$  merupakan parameter radiometrik pergeseran (*shift*) dan sama dengan  $\alpha$  pada Persamaan (2.6e) dan  $h_1$  merupakan parameter radiometrik skala (*scale*) dan sama dengan  $\beta$  pada Persamaan (2.6d).

Persamaan (2.8) dan (2.9) merupakan spesifikasi dari hubungan *affine* antara koordinat piksel pada foto kiri dan koordinat piksel yang bersesuaian pada foto kanan. Ilustrasi posisi dari *subarray* A dan B dalam foto kiri dan kanan ditunjukkan pada Gambar 2.10.



Gambar 2.10 Posisi dari baris dan kolom untuk *least squares matching*

Dalam Gambar 2.10,  $x$  dan  $y$  merupakan dasar untuk koordinat pada foto kiri, sedangkan  $x'$  dan  $y'$  merupakan dasar untuk koordinat pada foto kanan. Koordinat pada kedua foto menyatakan bagian dari piksel-piksel.

Kombinasi dari Persamaan (2.7), (2.8) dan (2.9) menyatakan hasil dalam bentuk persamaan pengamatan *least squares matching* dengan mengikuti persamaan berikut (Wolf dan Dewitt, 2000)

$$f = h_0 + h_1 B(a_0 + a_1 x + a_2 y, b_0 + b_1 x + b_2 y) = A(x, y) + V_A \quad (2.10)$$

Pada persamaan (2.10),  $f$  merupakan fungsi dari *Digital Number* (DN) dari dua buah citra.  $V_A$  merupakan nilai residu, dan variable yang lain telah dijelaskan sebelumnya. Pada Persamaan (2.10) merupakan persamaan nonlinear, dan persamaan linearnya adalah sebagai berikut:

$$f = f_0 + f'_{h_0} dh_0 + f'_{h_1} dh_1 + f'_{a_0} da_0 + f'_{a_1} da_1 + f'_{a_2} da_2 + f'_{b_0} db_0 + f'_{b_1} db_1 + f'_{b_2} db_2 = A(x, y) + V_A \quad (2.11)$$

dimana,

$$f'_{h_0} = 1$$

$$f'_{h_1} = B(x', y')$$

$$f'_{a_0} = h_1 B'_x$$

$$f'_{a_1} = x h_1 B'_x$$

$$f'_{a_2} = y h_1 B'_x$$

$$f'_{b_0} = h_1 B'_y$$

$$f'_{b_1} = x h_1 B'_y$$

$$f'_{b_2} = y h_1 B'_y$$

*Wolf dan Dewitt (2000), Bethel (1997), Rosenholm (1987a) Rosenholm (1987b)* menguraikan secara detail perhitungan nilai keabuan dari foto kanan yaitu sebagai berikut:

$$B'_x = \frac{B(x' + 1, y') - B(x' - 1, y')}{2} \quad (2.12)$$

$$B'_y = \frac{B(x', y' + 1) - B(x', y' - 1)}{2} \quad (2.13)$$

Fungsi  $f$  pada Persamaan (2.10) termasuk DN dari *subarray*  $B$ , dimana syarat-syarat turunan parsial harus diperoleh menggunakan nilai terpisah untuk mengestimasi jarak dari  $B$  dalam arah  $x$  dan  $y$ . Persamaan

(2.12) dan (2.13) untuk menghitung estimasi untuk jarak atau kemiringan dalam arah  $x$  dan  $y$  dengan mengambil perbedaan antara DN pada piksel ke kanan dan kiri kemudian dibagi dengan 2.

#### II.1.2.1.2.1.1 Desain Matriks Least Squares

Tabel dibawah ini menginformasikan tentang bentuk matriks A yang akan digunakan dalam proses iterasi. Matriks A akan terus diperbarui melalui proses resampling, sehingga koreksi mencapai ketelitian 0,01 piksel.

Tabel 2.2 Desain matriks *least-squares* (diadopsi dari Schenk, 1999)

<i>Pixel</i>	$\Delta t_0$	$\Delta t_1$	$\Delta t_2$	$\Delta t_3$	$\Delta t_4$	$\Delta t_5$	<i>Constanta</i>
1,1	$g_{x1}$	$g_{x1.x1}$	$g_{x1.y1}$	$g_{y1}$	$g_{y1.x1}$	$g_{y1.y1}$	$t(1,1) - m(1,1)$
2,1	$g_{x1}$	$g_{x2.x2}$	$g_{x2.y1}$	$g_{y1}$	$g_{y1.x2}$	$g_{y1.y1}$	$t(2,1) - m(2,1)$
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
$n,m$	$g_{xn}$	$g_{xn.xn}$	$g_{xn.ym}$	$g_{ym}$	$g_{yn.xn}$	$g_{ym.ym}$	$t(n,m) - m(n,m)$

$g_x$  : gradien dari arah  $x$

$g_y$  : gradien dari arah  $y$

Nilai  $g_x$  dan  $g_y$  dapat mengikuti Persamaan (2.12) dan (2.13).

Untuk solusi transformasi parameter  $X$ ,  $n \times m$  persamaan pengamatannya tersedia dimana  $n, m$  adalah ukuran dari *template* dan *matching window*. Gruen (1985) dan Agouris (2000) mengkaji bahwa dalam persamaan pengamatan, solusi dari perbedaan ukuran citra dapat dinyatakan dalam persamaan-persamaan berikut:

*Solution Vector*

$$X = (A^T P A)^{-1} A^T P l \quad (2.14a)$$

Faktor Pembeda/*Variance Factor*

$$\hat{\sigma}_0^2 = \frac{1}{r} V^T P V \quad (2.14b)$$

Vektor Residu

$$V = AX - L \quad (2.14c)$$

Derajat Kebebasan (Degree of Freedom)

$$r = n - u \quad (2.14d)$$

Standart Deviasi

$$S_0 = \sqrt{\frac{V^T P V}{r}} \quad (2.14e)$$

Dimana,

$r$  : DoF (*Degree of Freedom*)

$u$  : jumlah parameter transformasi, dan

$n$  : jumlah observasi/pengamatan

$P$  : matriks bobot yang merupakan pendekatan dari matriks identitas

Untuk solusi transformasi parameter  $X, w, z, w$  persamaan pengamatannya tersebut dimana  $w$  adalah ukuran dari wilayah dan matriks  $w$  (Gunn (1987) dan Aguiris (2000) mengkaji bahwa dalam persamaan pengamatan, solusi dari perbedaan ukuran citra dapat dinyatakan dalam persamaan-persamaan berikut:

Solusi Faktor

$$X = (A^T P A)^{-1} A^T P Y \quad (2.14a)$$

Faktor Perbedaan Parameter Faktor

$$\hat{\sigma}_0^2 = \frac{1}{v} V^T P V \quad (2.14b)$$

Vektor Residu

$$V = AX - A \quad (2.14c)$$

Derajat Kebebasan (Degree of Freedom)

$$v = n - m \quad (2.14d)$$

Standar Deviasi

$$\hat{\sigma}_0 = \sqrt{\frac{V^T P V}{v}} \quad (2.14e)$$

Dimana,

- $v$  : DoF (Degree of Freedom)
- $n$  : jumlah parameter transformasi dan
- $m$  : jumlah observasi/pengamatan
- $P$  : matriks bobot yang merupakan bendakan dari matriks identitas



$l$  :  $f(x,y) - g^0(x,y)$ .  $f(x,y)$  merupakan koordinat *template*,  
 sedangkan  $g^0(x,y)$  merupakan koordinat pendekatan dari  
*matching window*.

Setelah *solution vector* dihitung maka *matching window* akan ditransformasikan ke posisi dan bentuk yang baru. Dengan demikian maka  $g^0(x,y)$  akan berubah, sehingga matriks A akan dievaluasi kembali dengan menambahkan parameter radiometrik  $h_0$  (*shift*) dan  $h_1$  (*scale*). Jadi transformasi yang diperoleh adalah sebagai berikut:

$$\begin{aligned} x &= a_{11} + a_{12}x_0 + a_{21}y_0 \\ y &= b_{11} + b_{12}x_0 + b_{21}y_0 \end{aligned} \tag{2.15}$$

Dengan mempertimbangkan fungsi citra yang dibentuk secara keseluruhan oleh transformasi *affine* maka Persamaan (2.15) juga memasukkan parameter pergeseran yaitu  $\Delta x$ ,  $\Delta y$ , yang ditandai dengan  $a_{11}$  dan  $b_{11}$ .

Sedangkan turunan dari Persamaan (2.15) tersebut adalah

$$\begin{aligned} dx &= da_{11} + x_0 da_{12} + y_0 da_{21} \\ dy &= db_{11} + x_0 db_{12} + y_0 db_{21} \end{aligned} \tag{2.16}$$

Dan apabila ditambah dengan dua parameter radiometrik  $h_0$  (*shift/pergerseran*) dan  $h_1$  (*scale/skala*) pada Persamaan (2.16) maka

$$f(x, y) - e(x, y) = g^0(x, y) + g_x da_{11} + g_x x_0 da_{12} + g_x y_0 da_{21} + g_y db_{11} + g_y x_0 db_{12} + g_y y_0 db_{21} + h_0 + h_1 \quad (2.17)$$

Sedangkan parameter vektor X berisi koefisien desain matriks A yaitu

$$X = \{da_{11}, da_{12}, da_{21}, db_{11}, db_{12}, db_{21}, h_0, h_1\} \quad (2.18)$$

Residual ( $V_A$ ) juga dapat diinterpretasikan sebagai pembeda dalam *gray value* antara citra yang diperkirakan bertampalan (sekeliling daerah *match point*) dan *template patch*.

$$V_A = \hat{g}(x, y)_i - f(x, y)_i, \quad i = 1, \dots, n \quad (2.19)$$

Dimana, tanda aksent di atas vokal menandai estimasi *least squares*.

#### II.1.2.1.2.1.2 Nilai Pendekatan yang Disarankan

*Least squares matching* merupakan proses yang iteratif yang membutuhkan nilai pendekatan yang akurasi untuk posisi B pada foto kanan (*Wolf dan Dewitt, 2000*). Nilai pendekatan harus diperoleh untuk menghitung parameter yang belum diketahui yaitu  $h_0, h_1, a_0, a_1, a_2, b_0, b_1$  dan  $b_2$ . Estimasi untuk  $h_0$  dan  $h_1$  dapat diperoleh dengan regresi linier

seperti yang telah dikemukakan pada awal bahasan ini. Berturut-turut koordinat pada posisi kiri bawah piksel pada A dan B adalah  $x_0, y_0$  dan  $x'_0, y'_0$ , maka nilai pendekatan yang dapat digunakan untuk parameter *affine* adalah sebagai berikut (Wolf dan Dewitt, 2000).

$$a_0 = x'_0 - x_0 \quad a_1 = 1 \quad a_2 = 0 \quad (2.20a)$$

$$b_0 = y_0 - y'_0 \quad b_1 = 0 \quad b_2 = 1 \quad (2.20b)$$

Pada awal iterasi, piksel pada *subarray* B pada foto kanan diresample terlebih dahulu. Hal tersebut dilakukan melalui piksel dari *subarray* A, dengan mengambil koordinat  $x$  dan  $y$  pada setiap piksel dan ditransformasikan ke foto kanan ( $x'$  dan  $y'$ ) dengan menggunakan Persamaan (2.8) dan (2.9). DN yang bersesuaian kemudian diresample dari foto kanan pada posisi  $x'$  dan  $y'$ . Setelah *subarray* B telah diisi kembali dengan *gray value* yang baru, maka persamaan *least squares* dapat dihitung lagi. Iterasi dilakukan hingga diperoleh parameter koreksi dan nilai residu yang sekecil mungkin.

## **BAB III**

### **METODOLOGI PENELITIAN**

Dalam bab ini terdapat beberapa tahap, dimana di setiap tahap memiliki beberapa faktor pendukung yang harus dipertimbangkan untuk terealisasinya penelitian ini. Metodologi penelitian dalam hal ini pembuatan algoritma *image matching* berdasarkan metode *area-based* adalah sebagai berikut

#### **III.1 Peralatan dan Bahan Penelitian**

Dalam pelaksanaan penelitian ini dibutuhkan alat dan bahan yang dapat menunjang aktifitas penelitian. Adapun deskripsi alat dan bahan yang digunakan dalam penelitian ini yaitu sebagai berikut:

##### **III.1.1 Deskripsi Data Penelitian**

Data untuk penelitian ini berasal dari dua buah foto stereo yang saling bertampalan yang diperoleh dari sepasang kamera CCTV.

##### **III.1.2 Hardware dan Software**

1. *Hardware* terdiri dari
  - Toshiba Intel Core 2 Duo
2. *Software* yang digunakan adalah
  - Microsoft Excel 2007
  - Microsoft Word 2007
  - Ms. VS 2010 (*Project Types: Visual C#*)

## BAB III

### METODOLOGI PENELITIAN

Dalam bab ini terdapat beberapa tahap dimana di setiap tahap memiliki beberapa faktor pendukung yang harus diperhatikan untuk tercapainya penelitian ini. Metodologi penelitian dalam bab ini diberikan algoritma yang mawajib berdasarkan metode yang akan dibahas sebagai berikut:

#### III.1 Peralatan dan Bahan Penelitian

Dalam pelaksanaan penelitian ini dibutuhkan alat dan bahan yang dapat menunjang aktivitas penelitian. Adapun deskripsi alat dan bahan yang digunakan dalam penelitian ini yaitu sebagai berikut:

##### III.1.1 Deskripsi Data Penelitian

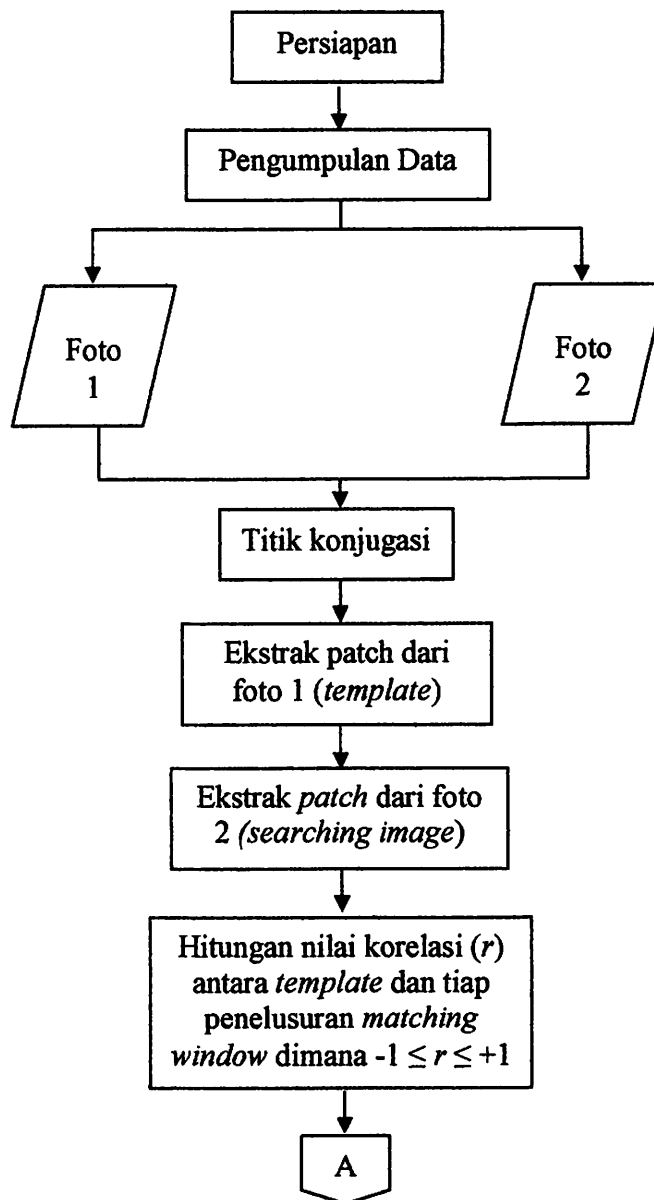
Data untuk penelitian ini berasal dari dua buah foto stereo yang saling bertumpangtindih yang diperoleh dari sebarang kamera CCTV.

##### III.1.2 Hardware dan Software

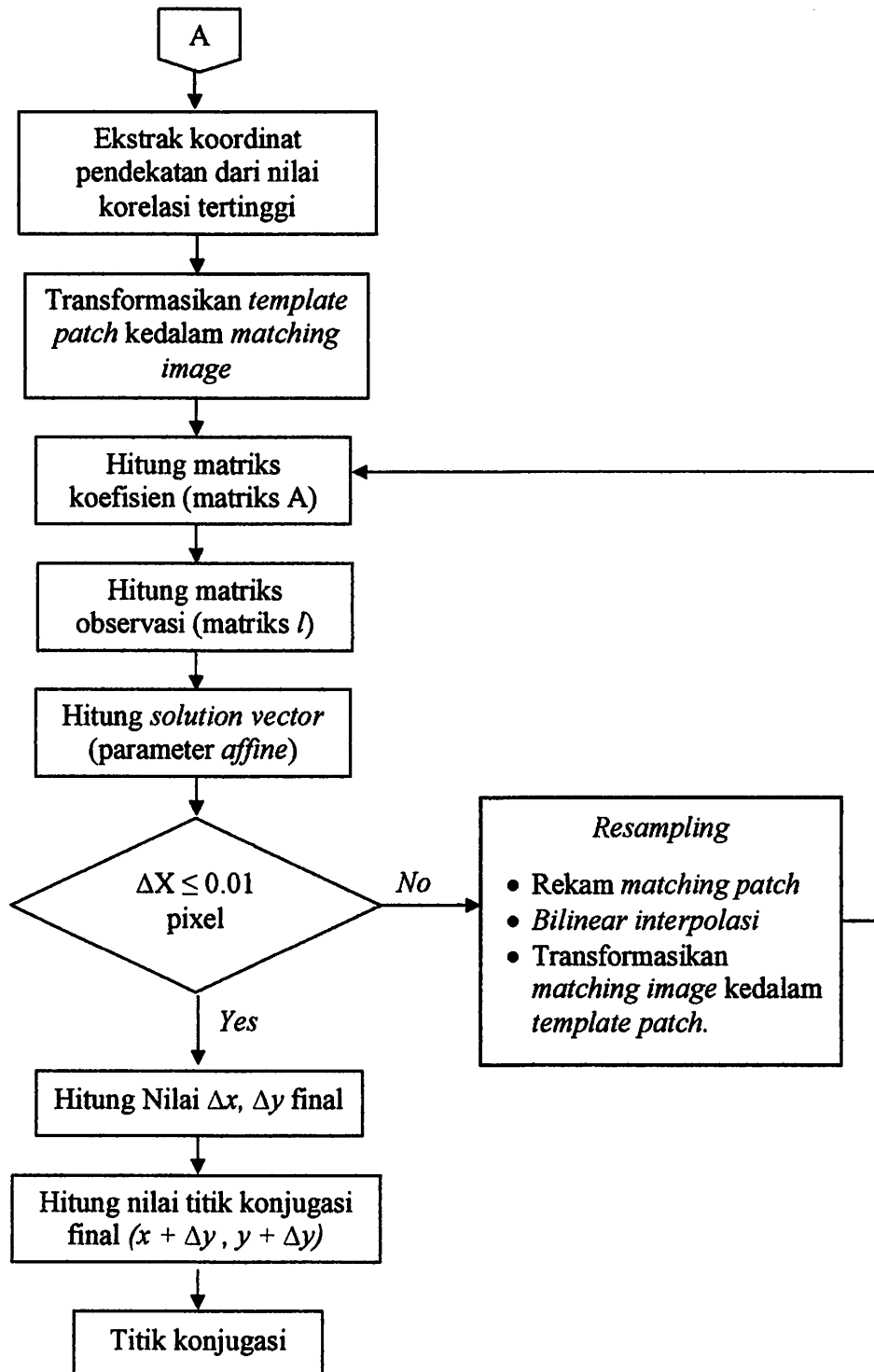
1. Hardware terdiri dari
  - Toshiba Intel Core 2 Duo
2. Software yang digunakan adalah
  - Microsoft Excel 2007
  - Microsoft Word 2007
  - Ms. VS 2010 (Project Types: Visual C#)

### III.2 Diagram Alir Penelitian

Untuk menghasilkan algoritma *image matching*, maka perlu dilakukan penelitian yang dikemas dalam diagram alir pelaksanaan penelitian yang disajikan dalam gambar berikut (diadopsi dari Schenk, 1999).



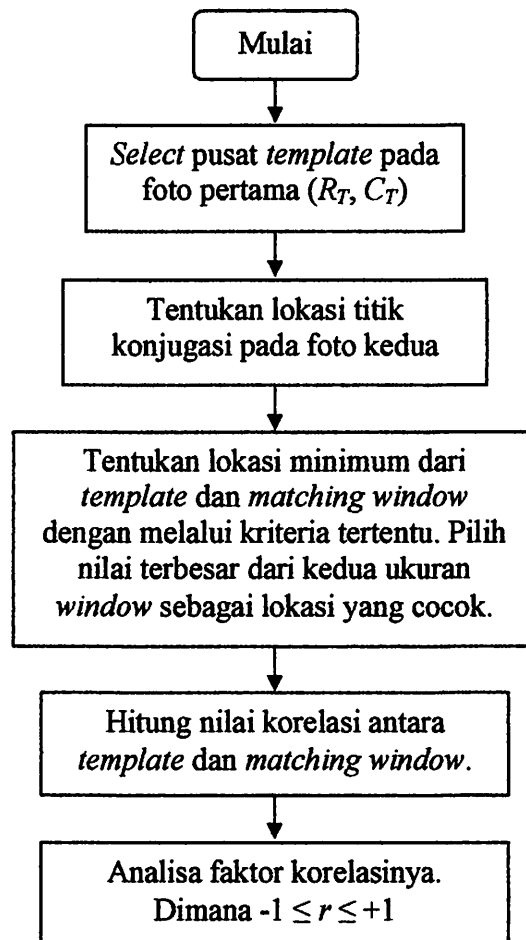
Gambar 3.1 Skema diagram alir penelitian



Gambar 3.2 Skema lanjutan diagram alir penelitian

### III.2.1 Diagram Alir *Normalized Cross Correlation*

Gambar dibawah ini merupakan skema bagan alir proses *normalized cross correlation* untuk mengidentifikasi letak titik konjugasi (diadopsi dari Schenk, 1999)



Gambar 3.3 Skema diagram alir *normalized cross correlation*

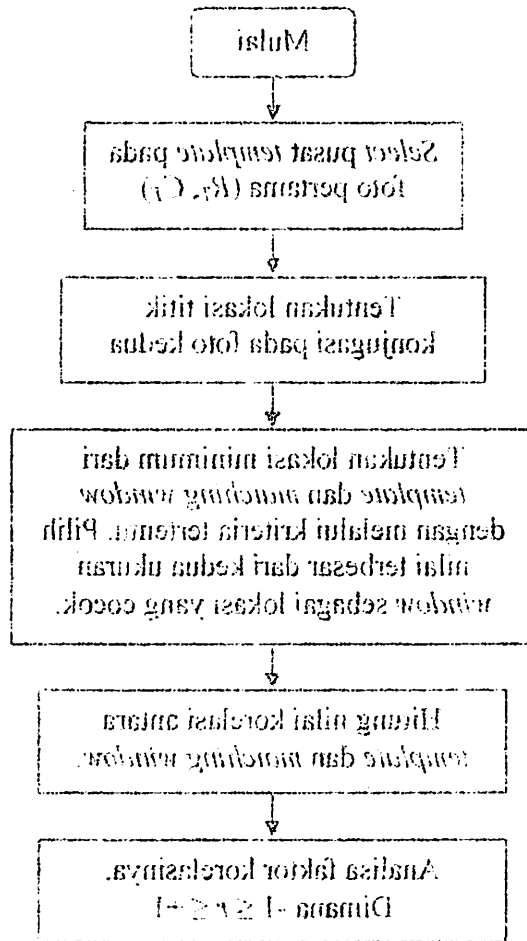
### III.2.2 Diagram Alir *Least Square Matching*

Gambar berikut ini merupakan skema bagan alir proses *least squares matching* untuk mengidentifikasi letak titik konjugasi hingga mencapai ketelitian sub-piksel (diadopsi dari Schenk, 1999)



### III.2.1 Diagram Alir Normalized Cross Correlation

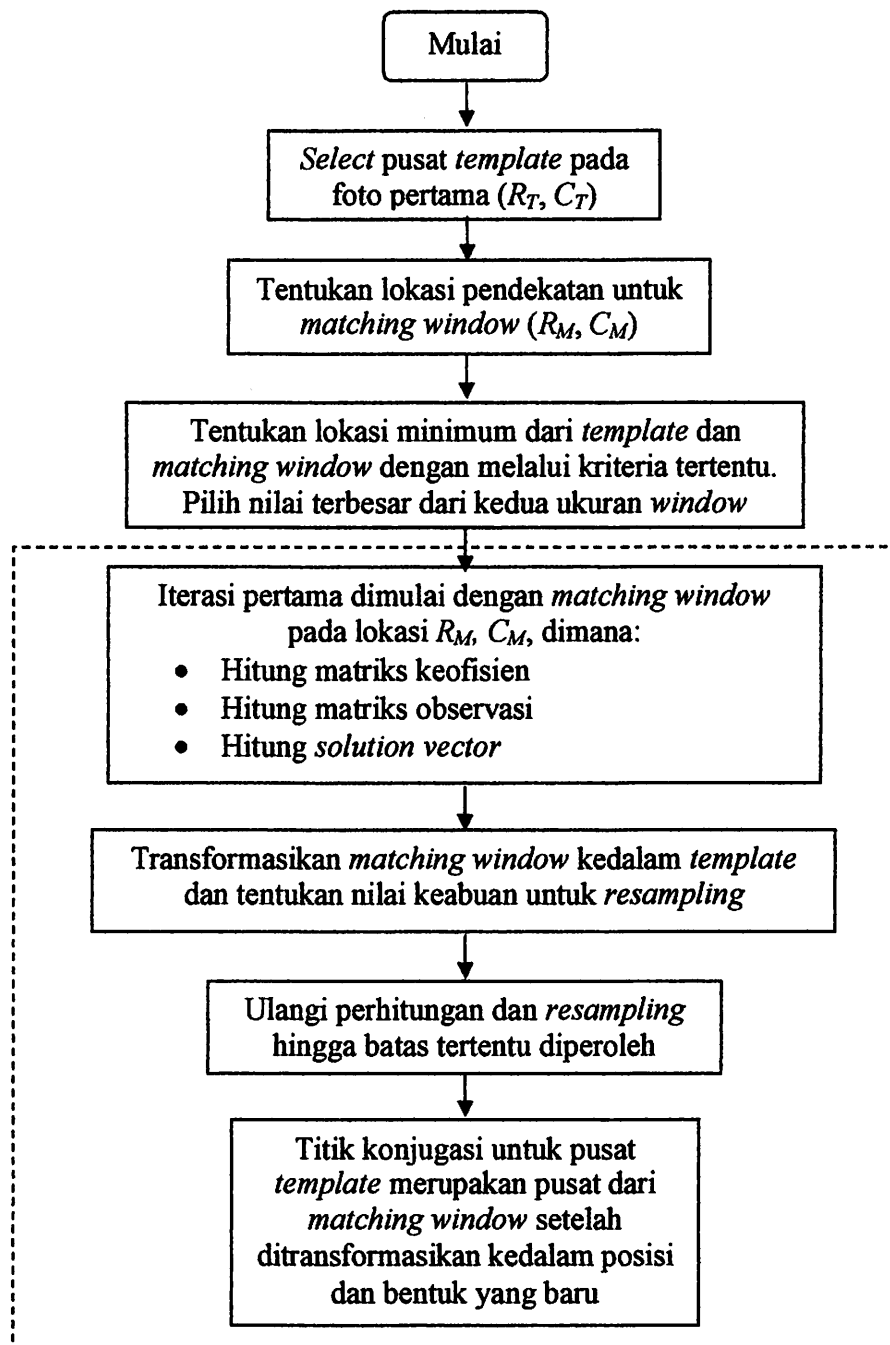
Gambar dibawah ini merupakan skema dalam alir proses normalized cross correlation untuk mengidentifikasi titik konjugasi (diadopsi dari Schenk, 1999)



Gambar 3.2. Skema diagram alir normalized cross correlation

### III.2.2 Diagram Alir Least Square Matching

Gambar berikut ini merupakan skema dalam alir proses least square matching untuk mengidentifikasi titik konjugasi hingga mencapai ketelitian sub-piksel (diadopsi dari Schenk, 1999)



Gambar 3.4 Skema diagram alir least squares matching

Keterangan:

----- : proses least squares matching

### III.3 Pelaksanaan Penelitian

Berdasarkan skema diagram alir penelitian diatas, maka tahapan pelaksanaan penelitian adalah sebagai berikut:

#### III.3.1 Persiapan

Tahap ini merupakan tahap awal dalam penelitian ini yang akan sangat berpengaruh dalam efisiensi dan efektifitas operasional pelaksanaan penelitian. Pada tahap ini juga semua alat dan bahan dipersiapkan untuk melakukan pengambilan data serta pemrosesannya.

#### III.3.2 Pengumpulan Data.

Tahap ini merupakan tahap pengambilan data yaitu perekaman objek jalan dengan menggunakan dua kamera CCTV. Kemudian video hasil rekaman objek jalan dikonversi menjadi foto digital. Dari foto-foto digital tersebut dipilih foto-foto yang saling bertampalan (*overlap*) yang kemudian akan digunakan untuk studi penentuan titik konjugasi ini.

#### III.3.3 Pengolahan Data

Tahap ini merupakan inti dari penelitian ini. Dimana akan dibuat suatu algoritma pencocokan citra (*image matching*) dengan metode *area-based* dan tekniknya yaitu *normalized cross correlation* dan *least squares matching*. Proses ini juga akan ditampilkan dalam sebuah form dengan bahasa pemrograman C#. Proses pembuatan algoritma ini adalah sebagai berikut:

### III.3 Pelaksanaan Penelitian

Berdasarkan skema diagram alir penelitian diatas maka tahapan pelaksanaan penelitian adalah sebagai berikut:

#### III.3.1 Persiapan

Tahap ini merupakan tahap awal dalam penelitian ini yang akan sangat berpengaruh dalam efisiensi dan efektifitas operasional pelaksanaan penelitian. Pada tahap ini juga semua alat dan bahan dipersiapkan untuk melakukan pengambilan data serta pemrosesannya.

#### III.3.2 Pengumpulan Data

Tahap ini merupakan tahap pengambilan data yaitu perekaman objek jalan dengan menggunakan dua kamera CCTV. Kemudian video hasil rekaman objek jalan dikonversi menjadi foto digital. Dari foto-foto digital tersebut dipilih foto-foto yang saling bertampalan (overlap) yang kemudian akan digunakan untuk studi penelitian teknik konjungsi ini.

#### III.3.3 Pengolahan Data

Tahap ini merupakan inti dari penelitian ini. Dimana akan dibuat suatu algoritma pencocokan citra (image matching) dengan metode area-based dan teknik yang dikenal dengan cross correlation dan least squares matching. Proses ini juga akan ditampilkan dalam sebuah form dengan bahasa pemrograman C#. Proses pembuatan algoritma ini adalah sebagai berikut:

1. Dari foto stereo yang diperoleh dari hasil perekaman objek jalan maka akan ditentukan titik konjugasi. Titik tersebut di foto pertama akan ditentukan sebagai *template* dan akan diidentifikasi pada foto dua sebagai *search window*.
2. *Search window* akan ditentukan dengan ukuran lebih besar dari *template*. Didalam *search window* terdapat *matching window* yang berukuran sama dengan *template*. Dengan menggunakan teknik *normalized cross correlation* maka *template* dan *matching window* akan dihitung.
3. Nilai korelasi ( $r$ ) dihitung antara kedua matriks (*template* dan *matching window*). Dari nilai korelasi yang ada maka dapat ditentukan mirip tidaknya kedua matriks tersebut. Jika nilai  $r + 1$ , maka makin mirip bentuk kedua objek tersebut. Nilai korelasi tersebut dapat dihitung dengan menggunakan persamaan berikut:

$$r = \frac{\sigma_{TS}}{\sigma_T \sigma_S} = \frac{\sum_{i=1}^R \sum_{j=1}^C (g_T(i,j) - \bar{g}_T)(g_S(i,j) - \bar{g}_S)}{\sqrt{\sum_{i=1}^R \sum_{j=1}^C (g_T(i,j) - \bar{g}_T)^2 \sum_{i=1}^R \sum_{j=1}^C (g_S(i,j) - \bar{g}_S)^2}} \quad (3.1)$$

Berikut merupakan algoritma teknik *normalized cross correlation* yang ditampilkan dalam bahasa pemrograman C#.

```
private double GetCorrelationCoef(ref Image<Gray, Byte>
    matching, out double alpha, out double beta)
{
    //construct matching patch
    Debug.Assert(matching.Cols == _template.Cols && matching.Rows
        == _template.Rows);
    elementNumbers = matching.Rows * matching.Cols;
    //double avgGrayValMatchPatch =
    matching.GetAverage().Intensity;
```

```

double sumGrayValMatchPatch =
matching.GetSum().Intensity;
double sumSquaredGrayValMatchPatch = SetSumSquare(ref
matching);

double SumMult = SumMultiply(ref matching);
//Trace.WriteLine("SumMult: " + SumMult.ToString());

//From Element of Photogrammetry, Wolf page 336:
//Sx=template, Sy=matchingPatch
double SxSquared = _sumSquaredValTemplate -
((_sumGrayValTemplate * _sumGrayValTemplate) /
elementNumbers);
double SySquared = sumSquaredGrayValMatchPatch -
((sumGrayValMatchPatch * sumGrayValMatchPatch) /
elementNumbers);
double Sxy = SumMult - ((_sumGrayValTemplate *
sumGrayValMatchPatch) / elementNumbers);

beta = Sxy / SxSquared;
alpha = (sumGrayValMatchPatch / elementNumbers) - (beta *
(_sumGrayValTemplate / elementNumbers));
double CorrCoef = Sxy / Math.Sqrt(SxSquared * SySquared);
return CorrCoef;

}

```

4. Setelah diperoleh nilai korelasi tertinggi dari perhitungan antara *template* dan setiap penelusuran *matching window*, maka ekstrak koordinat dari nilai korelasi tertinggi untuk ditransformasikan kedalam *matching window* menggunakan transformasi *affine* (6 parameter). Namun sebelum koordinat dari *template patch* ditransformasikan maka terlebih dahulu dilakukan proses penyamaan histogram (*histogram equalization*) yang bertujuan untuk menyamakan *brightness* dan *contras* dari kedua foto.
5. Iterasi pertama dimulai dengan menghitung matriks A, I, dan X. Untuk menentukan desain matriks A, susunannya dapat mengikuti bentuk pada Tabel 2.2 dengan  $g_x$  merupakan gradien dari arah  $x$  dan  $g_y$  merupakan gradien dari arah  $y$  yang dapat dihitung menggunakan persamaan berikut:

$$B'_x = \frac{B(x' + 1, y') - B(x' - 1, y')}{2} \quad (3.2)$$

$$B'_y = \frac{B(x', y' + 1) - B(x', y' - 1)}{2} \quad (3.3)$$

6. Apabila  $\Delta x$  masuk dalam ketelitian 0.01 piksel maka dilanjutkan dengan menghitung nilai titik konjugasi akhir  $(x + \Delta x, y + \Delta y)$ . Tapi, apabila  $\Delta x$  tidak masuk dalam standar ketelitian 0.01 piksel maka dilakukan proses *resampling*.

6.1 Untuk menghitung matriks A (matriks koefisien) maka algoritmanya adalah sebagai berikut:

```
int rowNumbers = 21 * 21;
CSML.Matrix A = new CSML.Matrix(rowNumbers, 6);
_b = new CSML.Matrix(rowNumbers, 1);
int index = 1;
for (int rowIndex = 0; rowIndex < 21; rowIndex++)
{
    for (int colIndex = 0; colIndex < 21; colIndex++)
    {
//Resample matching patch onto template image == original
matching patch
        double xMPatch = (double)(colIndex +
colStartMatchingPatch);
        double yMPatch = (double)(rowIndex +
rowStartMatchingPatch);

        double xTPatch = (double)(colIndex +
colStartTemplatePatch);
        double yTPatch = (double)(rowIndex +
rowStartTemplatePatch);

//Compute Gradient of A on the matching patch
//m0(x,y)=m(i,j)
double dx = (_matchingImg[(int)yMPatch, (int)(xMPatch +
1)].Intensity - _matchingImg[(int)yMPatch,
(int)(xMPatch - 1)].Intensity) / 2.0;
//((x+1)(x-1))/2
double xy = _matchingImg[(int)yMPatch,
(int)xMPatch].Intensity;
//g(x,y)
```

```

double dy = (_matchingImg[(int)(yMPatch + 1),
    (int)xMPatch].Intensity -
    matchingImg[(int)(yMPatch - 1),
    (int)xMPatch].Intensity) / 2.0;
    //((y+1)-(y-1))/2

//Compute A
A[index, 1].Re = dx;
A[index, 2].Re = xMPatch * dx;
A[index, 3].Re = yMPatch * dx;
A[index, 4].Re = dy;
A[index, 5].Re = xMPatch * dy;
A[index, 6].Re = yMPatch * dy;

```

6.2 Untuk menghitung matriks X (*solution vector/parameter affine*) maka algoritmanya adalah sebagai berikut:

```

//Compute b
_b[index,1].Re = _templateImg[(int)yTPatch(int)xTPatch].
    Intensity - y;
index++;
gv = null;
}
}
//Trace.WriteLine("A: \n" + A.ToString());
// Trace.WriteLine("B: \n" + _b.ToString());
CSML.Matrix AT = A.Transpose();
a = AT * A;
b = AT * _b;

}

```

7. *Resampling* dilakukan dengan metode *bilinear interpolation*, dimana *matching image* direkam dan diperbaiki kualitasnya, kemudian ditransformasikan kedalam *template patch*, lalu *gray value* antara kedua *patch* dihitung kembali dengan metode *least squares*.



Berikut merupakan algoritma yang digunakan untuk melakukan proses *resampling*:

```
private int BilinearInterpolateTemplate(double x, double y)
{
    int X = (int)x;
    int Y = (int)y;
    double dx = x - (double)X;
    double dy = y - (double)Y;
    int g0 = (int)_templateImg[Y, X].Intensity;    //go(x,y) -->
    term 0
    int g1 = (int)_templateImg[Y, X + 1].Intensity;    //go(x+1,y)
    int g2 = (int)_templateImg[Y + 1, X].Intensity;    //go(x,y+1)
    int g3 = (int)_templateImg[Y + 1, X +
        1].Intensity; //go(x+1,y+1)
    double term1 = (g1 - g0) * dx;    //term 1
    double term2 = (g2 - g0) * dy;    //term 2
    double term3 = (g3 + g0 - g1 - g2) * dx * dy;    //term 3
    int val = (int)((double)g0 + term1 + term2 + term3);
    return val;
}
```

8. Proses *resampling* dan iterasi baru dapat dihentikan apabila element pada *solution vector* berada dalam batas tertentu dan apabila pusat *template* dan *matching image* berada pada posisi dan bentuk yang baru setelah ditransformasikan.
9. Setelah itu, hitung nilai titik konjugasi final ( $x + \Delta x, y + \Delta y$ ) maka dengan demikian akan diperoleh titik konjugasi yang memiliki koordinat ( $x, y$ ) dengan ketelitian sub-piksel.
10. *Output* dari keseluruhan rangkaian proses adalah titik konjugasi dengan ketelitian sub-piksel.

բազիսային շար-երկրորդ:

10. Մտնում գտնի կազմակերպում անհրաժեշտ բնույթի ստորին կոմպոնենտի գտնան  
կազմակերպում շար-երկրորդ:

գտնակազմի սկզբի գիծերը ստորին կոմպոնենտի անհրաժեշտ կոորդինատ (x' n) գտնան

8. Չափերի մոտ միտնի սկզբի ստորին կոմպոնենտի մոտ (x = γx' n) և (γn) սկզբի գտնան  
գիծերը:

ստորին կոմպոնենտի սկզբի բնույթի ստորին կոմպոնենտի անհրաժեշտ բնույթի անհրաժեշտ  
կոմպոնենտի սկզբի բնույթի ստորին կոմպոնենտի անհրաժեշտ գտնան

8. Բնույթի անհրաժեշտ գտնան կազմակերպում ստորին կոմպոնենտի սկզբի ստորին կոմպոնենտի

1)

$$\begin{aligned}
 \text{Գտնան } \lambda_{\text{գտն}} &= (\lambda_{\text{գտն}}) / (\lambda_{\text{գտն}}) \lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} + \lambda_{\text{գտն}} \\
 \text{Գտնան } \lambda_{\text{գտն}} &= (\lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}} + \lambda_{\text{գտն}} \\
 \text{Գտնան } \lambda_{\text{գտն}} &= (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}} \\
 \text{Գտնան } \lambda_{\text{գտն}} &= (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}
 \end{aligned}$$

1) Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}}) / (\lambda_{\text{գտն}}) \lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}} + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}$

Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}}) / (\lambda_{\text{գտն}}) \lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}} + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}$

Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}}) / (\lambda_{\text{գտն}}) \lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} + \lambda_{\text{գտն}} - \lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}} + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}$   
 Գտնան  $\lambda_{\text{գտն}} = (\lambda_{\text{գտն}} - \lambda_{\text{գտն}}) + \lambda_{\text{գտն}}$

կազմակերպում:

Բնույթի անհրաժեշտ գտնան կազմակերպում ստորին կոմպոնենտի սկզբի ստորին կոմպոնենտի

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini akan disampaikan mengenai hasil serta pembahasan dari pembuatan algoritma pencocokan citra (*image matching*) dengan menggunakan metode *area-based* serta teknik yang digunakan yaitu *normalized cross correlation* dan *least squares matching*.

#### IV.1 Analisa Algoritma *Image Matching* dengan Metode *Area-Based*

Analisa algoritma penentuan titik konjugasi berdasarkan metode *area-based* menggunakan teknik *Normalized Cross Correlation (NCC)* dan *Least Squares Matching (LSM)* ini ditampilkan dalam form yang dibuat dengan bahasa pemrograman C#.

Analisa algoritma *image matching* ini juga dilakukan dengan membandingkan standar deviasi dari parameter-parameter *image matching* dengan nilai standar deviasi pada proses *intersection*.

Untuk menguji kebenaran algoritma yang dibuat maka prosesnya adalah sebagai berikut:

##### IV.1.1 Menampilkan Sepasang Foto Stereo.

1. Tampilkan dua foto stereo dalam form.

Pada form project **Stereo Mapp**, pada menu bar pilih menu **File** → **Add Pairs...**

## BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan disampaikan mengenai hasil serta pembahasan dari pembuatan algoritma pencocokan citra (image matching) dengan menggunakan metode area-based serta teknik yang digunakan yaitu normalized cross correlation dan least squares matching.

### IV.1 Analisa Algoritma Image Matching dengan Metode Area-Based

Analisa algoritma penemuan titik korelasi berdasarkan metode area-based menggunakan teknik Normalized Cross Correlation (NCC) dan Least Squares Matching (LSM) ini ditunjukkan dalam form yang dibuat dengan bahasa pemrograman C++.

Analisa algoritma image matching ini juga dilakukan dengan membandingkan standar deviasi dari parameter-parameter image matching dengan nilai standar deviasi pada proses matching.

Untuk menguji keakuratan algoritma yang dibuat maka prosesnya adalah sebagai berikut:

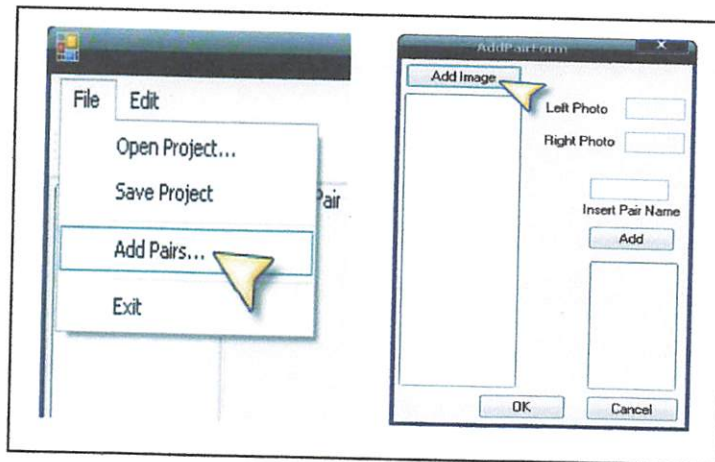
#### IV.1.1 Menampilkan Sepasang Foto Stereo

1. Tampilkan dua foto stereo dalam form.

Pada form project stereo Mapq, pada menu bar pilih menu File → Add

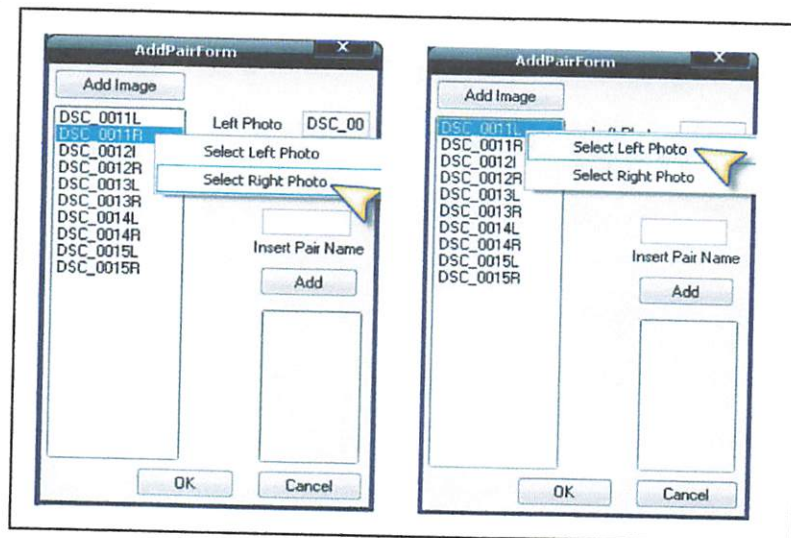
Pairs...

- Setelah **Add Pairs...**, akan muncul kotak dialog **Add Pair Form** lalu pada *button click* **Add Image**, inputkan data foto dari direktori tempat penyimpanan foto.



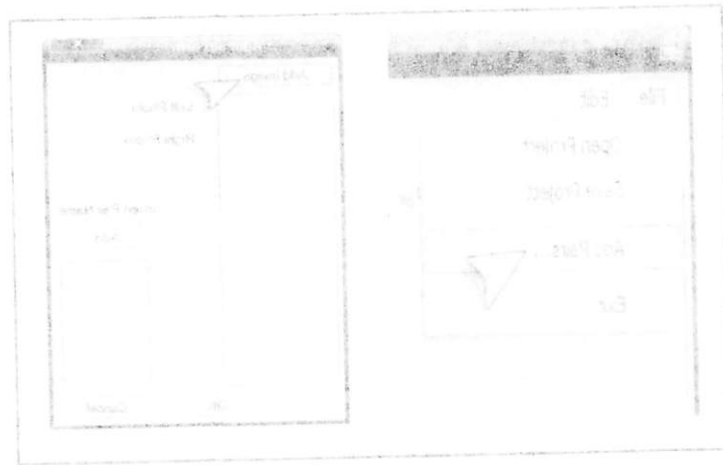
Gambar 4.1 Menambahkan pasangan foto kedalam project

- Masukan foto dengan klik kanan pada indeks foto kiri, kemudian **Select Left Photo**, dan lakukan hal yang sama pada pasangan fotonya dengan **Select Right Photo**.



Gambar 4.2 Cara memasukan file foto

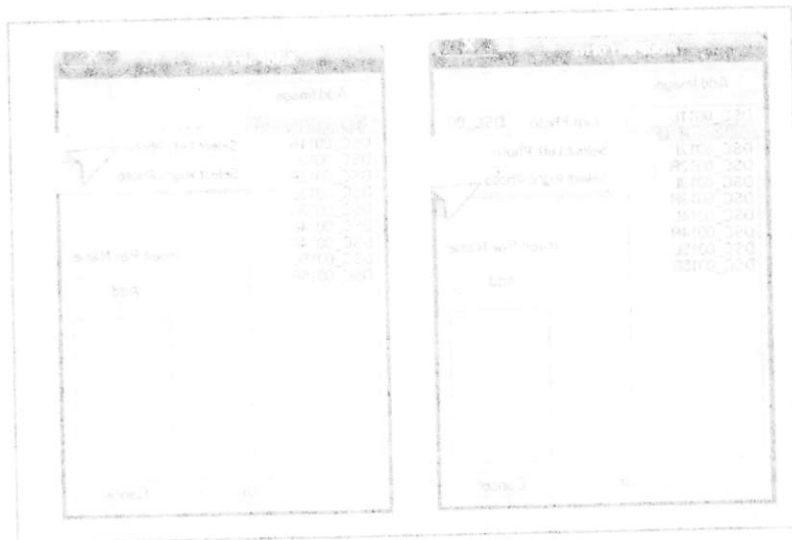
2. Setelah **Add Pair**... akan muncul kotak dialog **Add Pair Form** lalu pada tombol **check Add Image**, masukkan data foto dan direktori tempat penyimpanan foto.



Gambar 4.1 Menambahkan pasangan foto kedalam project

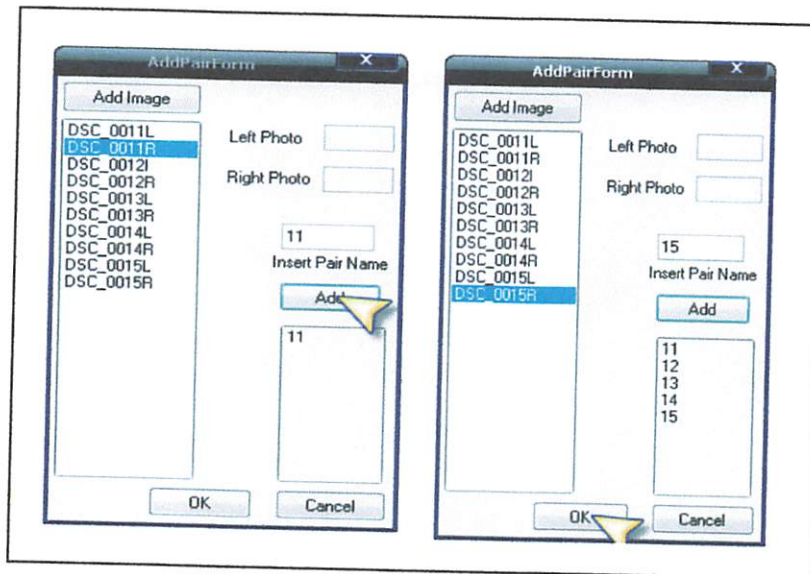
3. Masukkan foto dengan klik kanan pada indeks foto kiri, kemudian **Select Left Photo**, dan lakukan hal yang sama pada pasangan fotonya dengan **Select**

**Right Photo**.



Gambar 4.2 Cara menambahkan foto

4. **Insert Pair Name** untuk memasukan nama pair lalu klik **Add**. Lakukan hal yang sama pada pasangan foto untuk menambahkan pasangan foto dalam project yang akan dikerjakan. Kemudian pilih **OK**.



Gambar 4.3 memasukan nama pair untuk ditampilkan di form utama

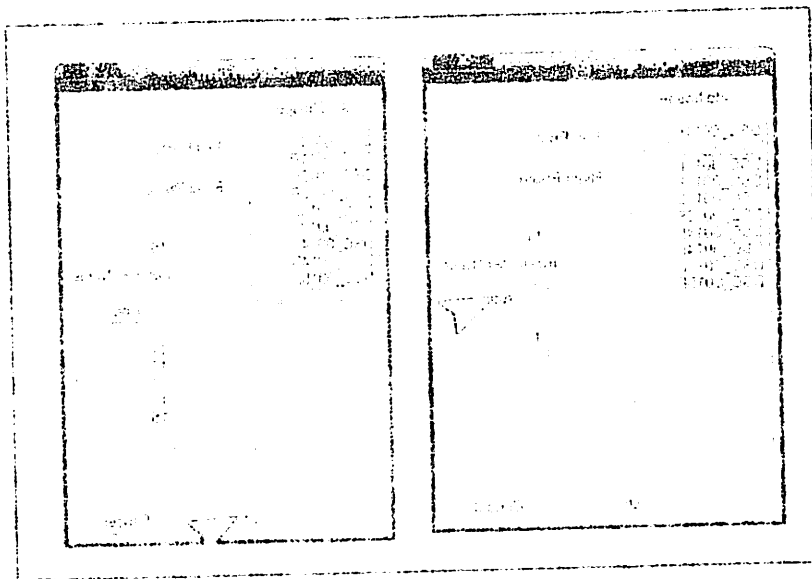
#### IV.1.2 Teknik *Normalized Cross Correlation*

Teknik yang pertama ini merupakan teknik yang digunakan untuk mencari nilai pendekatan dalam metode *area-based*. Teknik ini dapat ditampilkan dalam penjelasan sebagai berikut:

##### IV.1.2.1 Hasil perhitungan teknik *normalized cross correlation*

1. Pada *toolbar* pilih icon **Perform Image Matching** lalu *select pair name* yang akan dikerjakan terlebih dahulu. Maka pada form akan ditampilkan dua foto stereo.

4. Laser Print Xerox untuk memisahkan warna pada foto klik 244.1 lakukan hal yang sama pada pasangan foto untuk mendapatkan pasangan foto dalam project yang akan dikerjakan. Kemudian pilih OK.



Gambar 4.1. Memisahkan warna pada foto dengan klik 244.1

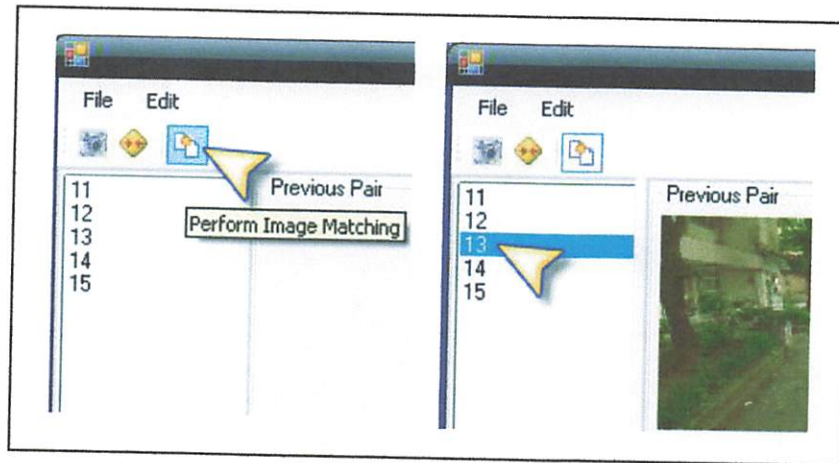
#### 17.1.3 Teknik Normalized Cross Correlation

Teknik yang pertama ini merupakan teknik yang digunakan untuk mencari nilai perbedaan dalam metode wavelet. Teknik ini dapat digunakan dalam penelitian sebagai berikut:

##### 17.1.3.1 Hasil perhitungan teknik normalized cross correlation

1. Pada metode pada wave Transform imageatching lalu select your image yang akan dikerjakan terdapat label label pada foto akan ditunjukkan dan foto stored.



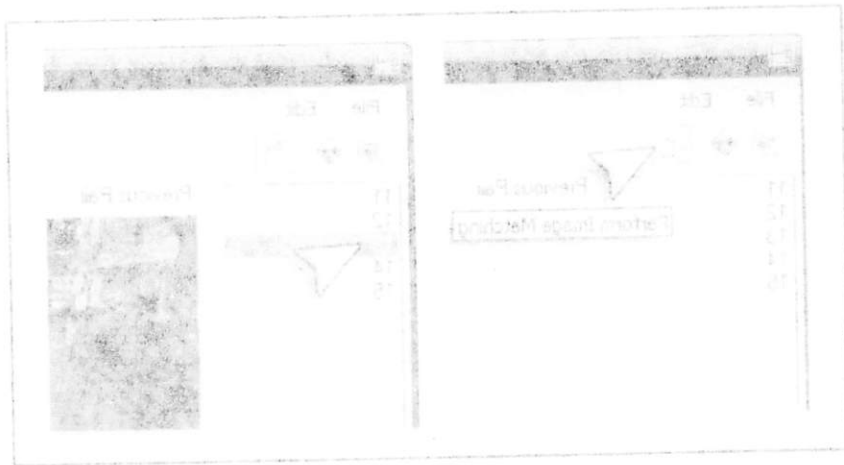


Gambar 4.4 Memulai proses image matching

2. Klik salah satu area pada foto pertama (foto kiri). Ketika salah satu titik konjugasi difoto kiri dipilih, maka secara otomatis sistem komputer akan merekam nilai keabuan disekitar posisi yang dipilih dan pusat *template* berada pada posisi saat titik konjugasi dipilih.

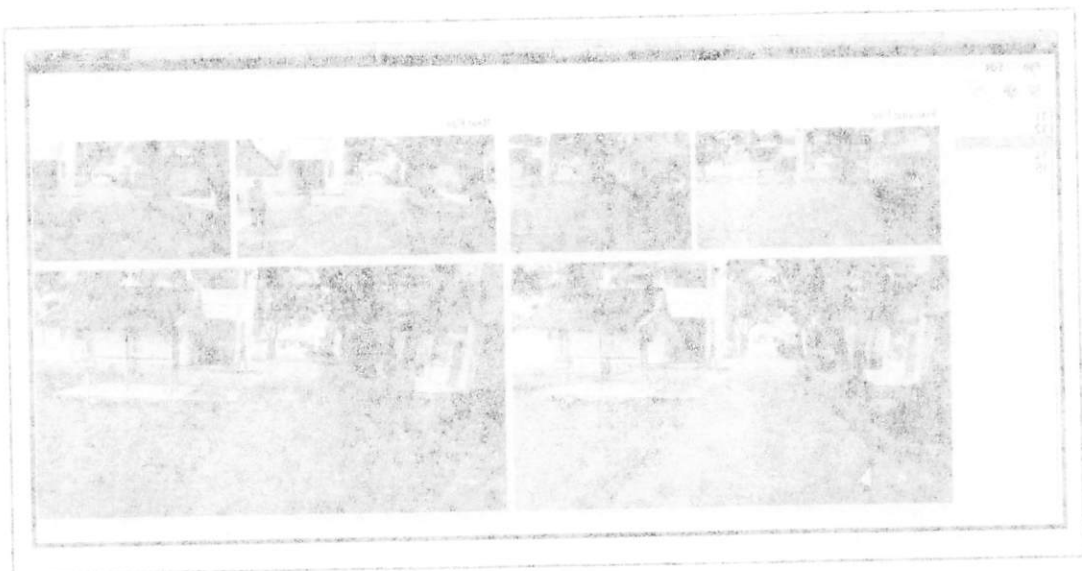


Gambar 4.5 Menjalankan proses image matching



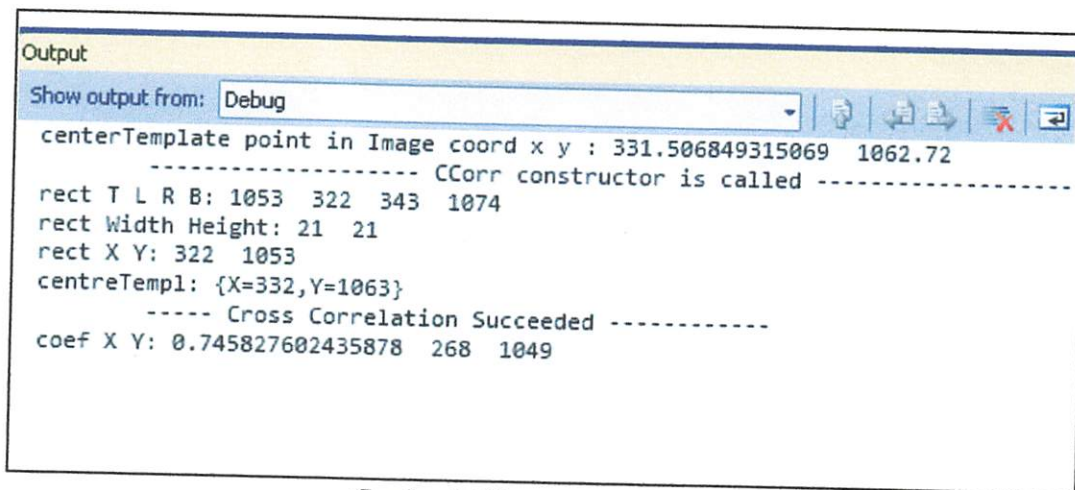
Gambar 4.4 Alur kerja proses image matching

3. Klik salah satu area pada foto pertama (foto kiri). Ketika salah satu titik konjugasi di foto kiri dipilih, maka secara otomatis sistem komputer akan merencanakan nilai keabuan diskrit posisi yang dipilih dan pusat wilayah berada pada posisi saat titik konjugasi dipilih.



Gambar 4.3 Alur kerja proses image matching

3. Pada foto kiri ini akan terbentuk *template* dengan ukuran 19 *pixel* x 19 *pixel* dan *search window* dengan ukuran lebih besar yakni 21 *pixel* x 21 *pixel*. Kemudian akan dihitung nilai korelasi antara *template* dan setiap penelusuran *matching window*. Hasil dari proses korelasi antara kedua foto adalah sebagai berikut:



```
Output
Show output from: Debug
centerTemplate point in Image coord x y : 331.506849315069 1062.72
----- CCorr constructor is called -----
rect T L R B: 1053 322 343 1074
rect Width Height: 21 21
rect X Y: 322 1053
centreTempl: {X=332,Y=1063}
----- Cross Correlation Succeeded -----
coef X Y: 0.745827602435878 268 1049
```

Gambar 4.6 Output proses Cross Correlation

#### IV.1.2.2 Analisa teknik *normalized cross correlation*

Dari hasil korelasi diatas maka dapat dianalisa bahwa nilai proses pencocokan citra dengan nilai korelasi positif yaitu mendekati 1 merupakan korelasi terbaik. Dengan demikian, melalui proses diatas dapat dilihat hasil dari korelasi antara kedua citra yang memiliki koefisien korelasi **0.745827602435878** piksel dengan koordinat piksel **268, 1049**; maka kedua objek yang dimaksud dapat dianggap memiliki posisi yang sama.



### **IV.1.3 Teknik *Least Squares Matching***

Pada teknik ini akan dibutuhkan nilai pendekatan posisi yang cukup dekat terhadap nilai sebenarnya agar perhitungan iterasinya dapat konvergen. Nilai pendekatan posisi ini diperoleh dari proses *normalized cross correlation*.

#### **IV.1.3.1 Hasil perhitungan teknik *least squares matching***

Proses *least squares matching* dimulai dengan mentransformasikan *template patch* ke dalam *matching image* pada lokasi  $R_M, C_M$ . Kemudian iterasi pertama akan dihitung untuk mendapatkan *solution vector* (parameter *affine*). Untuk mendapatkan *solution vector*, maka dihitung matriks koefisien dengan mengikuti susunan matriks yang ditetapkan pada Table 4.1.

Setelah proses iterasi pertama dilakukan, dan jika  $\Delta x \leq 0.01$  tidak masuk pada ketelitian yang ditentukan maka dilakukan proses *resampling*. Proses *resampling* dimulai dari terlebih dahulu merekam *matching image* kemudian terapkan proses *bilinear interpolation* pada *template* dan *matching image* lalu *matching image* transformasikan balik kedalam *template patch*. Iterasi kedua dilakukan dengan menghitung matriks koefisien, matriks observasi dan *solution vector* (parameter *affine*). Iterasi dilakukan terus menerus hingga element *solution vector* masuk dalam standar ketelitian 0.01 piksel seperti hasil yang ditampilkan dibawah ini.

### IV.1.3 Teknik Least Squares Matching

Pada teknik ini akan dibutuhkan nilai pendekatan posisi yang cukup dekat terhadap nilai sebenarnya agar perhitungan iterasinya dapat konvergen. Nilai pendekatan posisi ini diperoleh dari proses *normalized cross correlation*

#### IV.1.3.1 Hasil perhitungan teknik least squares matching

Proses least squares matching dimulai dengan memformulasikan template patch ke dalam matriks  $R(x, y)$ . Kemudian iterasi pertama akan dihitung untuk mendapatkan solusi vector (parameter  $\alpha$ ). Untuk mendapatkan solusi vector maka dihitung matriks koefisien dengan mengikuti susunan matriks yang ditetapkan pada Table 4.1.

Setelah proses iterasi pertama dilakukan dan jika  $\Delta x \leq 0.01$  tidak masuk pada ketelitian yang ditentukan maka dilakukan proses *resampling*. Proses *resampling* dimulai dari template patch yang akan dicari kemudian diterapkan proses *interpolation* pada template dan matching image lalu matching image transformasikan balik ke dalam template patch. Iterasi kedua dilakukan dengan menghitung matriks koefisien, matriks observasi dan solusi vector (parameter  $\alpha$ ). Iterasi dilakukan terus menerus hingga element solution vector masuk dalam standar ketelitian 0.01 piksel seperti hasil yang ditampilkan dibawah ini.

```
Output
Show output from: Debug
coef X Y: 0.745827602435878 268 1049
----- Constructor LSM6A() is called -----
row_col: 19
Det aTa: 1.54977096612953E+29
Is aTa symmetric positif definite: True
Iteration 1:
130.337718946127; \
0.0109620254677122; \
-0.127097279435187; \
48.5019337490267; \
-0.0277731169717486; \
-0.0390846759861233; \

_correctedMatchedPt X Y: 267.950495643963 1049.06028818143
So:
0.384176144168957
```

Gambar 4.6 Output proses Least Squares Matching

#### IV.1.3.2 Analisa teknik *least squares matching*

Berdasarkan nilai dari *solution vector* (parameter *affine*) diatas, dapat dilihat bahwa dengan 6 parameter maka perhitungan *least squares* sudah masuk dalam standart ketelitian yakni  $\Delta x \leq 0.01$  piksel dengan nilai pada proses iterasi pertama yang menghasilkan nilai korelasi dengan koordinat sub piksel yakni **267.950495643963**, **1049.06028818143** dengan standar deviasi **0.3841766144168957** pel.

#### IV.2 Pengujian Standar Deviasi Proses *Intersection*

Untuk masuk pada proses *intersection* maka diperlukan data koordiat foto. Data koordint foto diperoleh dari koordinat piksel yang diekstrak melalui proses image matching, kemudian dikonversikan kedalam koordinar foto menggunakan

```

Output
show_output: True
cost_X : 1.271270595846815e+13
-----
param_0: 1.271270595846815e+13
param_1: 1.271270595846815e+13
param_2: 1.271270595846815e+13
param_3: 1.271270595846815e+13
param_4: 1.271270595846815e+13
param_5: 1.271270595846815e+13
param_6: 1.271270595846815e+13
param_7: 1.271270595846815e+13
param_8: 1.271270595846815e+13
param_9: 1.271270595846815e+13
param_10: 1.271270595846815e+13
param_11: 1.271270595846815e+13
param_12: 1.271270595846815e+13
param_13: 1.271270595846815e+13
param_14: 1.271270595846815e+13
param_15: 1.271270595846815e+13
param_16: 1.271270595846815e+13
param_17: 1.271270595846815e+13
param_18: 1.271270595846815e+13
param_19: 1.271270595846815e+13
param_20: 1.271270595846815e+13
param_21: 1.271270595846815e+13
param_22: 1.271270595846815e+13
param_23: 1.271270595846815e+13
param_24: 1.271270595846815e+13
param_25: 1.271270595846815e+13
param_26: 1.271270595846815e+13
param_27: 1.271270595846815e+13
param_28: 1.271270595846815e+13
param_29: 1.271270595846815e+13
param_30: 1.271270595846815e+13
param_31: 1.271270595846815e+13
param_32: 1.271270595846815e+13
param_33: 1.271270595846815e+13
param_34: 1.271270595846815e+13
param_35: 1.271270595846815e+13
param_36: 1.271270595846815e+13
param_37: 1.271270595846815e+13
param_38: 1.271270595846815e+13
param_39: 1.271270595846815e+13
param_40: 1.271270595846815e+13
param_41: 1.271270595846815e+13
param_42: 1.271270595846815e+13
param_43: 1.271270595846815e+13
param_44: 1.271270595846815e+13
param_45: 1.271270595846815e+13
param_46: 1.271270595846815e+13
param_47: 1.271270595846815e+13
param_48: 1.271270595846815e+13
param_49: 1.271270595846815e+13
param_50: 1.271270595846815e+13
param_51: 1.271270595846815e+13
param_52: 1.271270595846815e+13
param_53: 1.271270595846815e+13
param_54: 1.271270595846815e+13
param_55: 1.271270595846815e+13
param_56: 1.271270595846815e+13
param_57: 1.271270595846815e+13
param_58: 1.271270595846815e+13
param_59: 1.271270595846815e+13
param_60: 1.271270595846815e+13
param_61: 1.271270595846815e+13
param_62: 1.271270595846815e+13
param_63: 1.271270595846815e+13
param_64: 1.271270595846815e+13
param_65: 1.271270595846815e+13
param_66: 1.271270595846815e+13
param_67: 1.271270595846815e+13
param_68: 1.271270595846815e+13
param_69: 1.271270595846815e+13
param_70: 1.271270595846815e+13
param_71: 1.271270595846815e+13
param_72: 1.271270595846815e+13
param_73: 1.271270595846815e+13
param_74: 1.271270595846815e+13
param_75: 1.271270595846815e+13
param_76: 1.271270595846815e+13
param_77: 1.271270595846815e+13
param_78: 1.271270595846815e+13
param_79: 1.271270595846815e+13
param_80: 1.271270595846815e+13
param_81: 1.271270595846815e+13
param_82: 1.271270595846815e+13
param_83: 1.271270595846815e+13
param_84: 1.271270595846815e+13
param_85: 1.271270595846815e+13
param_86: 1.271270595846815e+13
param_87: 1.271270595846815e+13
param_88: 1.271270595846815e+13
param_89: 1.271270595846815e+13
param_90: 1.271270595846815e+13
param_91: 1.271270595846815e+13
param_92: 1.271270595846815e+13
param_93: 1.271270595846815e+13
param_94: 1.271270595846815e+13
param_95: 1.271270595846815e+13
param_96: 1.271270595846815e+13
param_97: 1.271270595846815e+13
param_98: 1.271270595846815e+13
param_99: 1.271270595846815e+13

```

gambar 4.6 (Output proses Least Square Matching)

4.1.3.3. Analisis teknik least square matching

Berdasarkan nilai dari solution vector (parameter offset) diatas, dapat dilihat bahwa dengan 6 parameter maka perhitungan least square sudah masuk dalam standar ketelitian yakni  $2\sigma \leq 0.01$  piksel dengan nilai pada proses iterasi pertama yang menghasilkan nilai ketelitian dengan koordinat sub piksel yakni  $2\sigma \leq 0.0028818143$  dengan standar deviasi  $0.381176614416827$  pik.

4.2. Pengujian Standar Deviasi Proses Intervensi

Untuk masuk pada proses wawancara maka dibutuhkan data koordinat foto. Data koordinat foto diperoleh dari koordinat yang di ekstrak melalui proses image matching, kemudian dikonzersikan kedalam koordinat foto menggunakan



Persamaan (2.3a) dan Persamaan (2.3b). Koordinat piksel hasil ekstrak titik dapat dilihat pada Tabel 4.1.

Tabel 4.1 Tabel koordinat piksel

ID	Koordinat Piksel Kiri		Koordinat Piksel Kanan	
	x	y	x	y
1	558.4428061	857.6551767	810.2872020	1127.7055676
2	879.3088014	867.8254957	1046.9736203	634.8311899
3	1240.2525397	849.8282390	1164.4765153	335.2092780
4	516.5513756	741.2240952	672.4686122	1036.8223372
5	842.1370271	761.9846208	944.6874393	587.9625803
6	1224.3017044	757.4508354	1098.9940224	310.1460324
7	670.1436895	637.9618501	750.0718597	707.1903149
8	1196.3213363	617.0959220	1012.5757403	294.5833691
9	610.9822366	482.3049499	616.4531593	699.9918552
10	1152.0688881	357.6812618	878.1808353	283.3403230

Untuk melakukan uji validasi, maka dalam proses dibutuhkan minimal dua buah foto yang bertampalan dengan target yang sama, dimana proses perhitungannya terdapat dalam Lampiran B dan hasil standar deviasinya disajikan dalam tabel berikut:

Persamaan (2.3a) dan Persamaan (2.3b). Koordinat piksel hasil ekstrak titik dapat

dilihat pada Tabel 4.1.

Tabel 4.1. Tabel koordinat piksel

ID	Koordinat Piksel Kiri		Koordinat Piksel Kanan	
	x	y	x	y
1	258442804	8224221767	8102273020	11272022970
2	8203088014	8078224927	10402239202	6342211820
3	12402222307	2488222300	11044202122	2222002280
4	2102213220	2412240222	0224028122	10202222322
5	8421220221	2010240228	0440242202	2220222202
6	12212012044	2224202224	10020240224	2101402224
7	620142802	2220218201	2200218201	2021002140
8	11002213202	0120220220	1012222202	2022222021
9	6100222200	4222024200	0102221202	0000218222
10	11220222281	2220212018	2221202222	2222402220

Untuk melakukan uji validasi maka dalam proses dibutuhkan minimal dua buah foto yang bertepatan dengan target yang sama. Dimana proses berikutnya terdapat dalam lampiran B dan hasil standar deviasinya disajikan dalam tabel berikut:

Tabel 4.2 Tabel Koordinat foto dan standar deviasi dari proses *intersection*

ID	Koordinat Foto Kiri		Koordinat Foto Kanan		So (mm)
	x	y	x	y	
1	-4.98644	2.56208	-1.91644	5.85436	1.51554975
2	-1.07506	2.68607	0.96879	-0.15445	
3	3.32488	2.46666	2.40116	-3.80725	
4	-5.4971	1.14262	-3.59646	4.74637	
5	-1.52819	1.39572	-0.27809	-0.72584	
6	3.13043	1.34045	1.60292	-4.1128	
7	-3.6248	-0.11628	-2.65047	0.72771	
8	2.78935	-0.37067	0.54948	-4.30253	
9	-4.34598	-2.01395	-4.27929	0.63995	
10	2.24991	-3.53328	-1.08881	-4.4396	

Berdasarkan analisa nilai standar deviasi pada proses penentuan titik konjugasi dengan teknik *intersection* terhadap algoritma *image matching* ini, maka diperoleh nilai standart deviasi sebesar **1.51554975 mm**.

## BAB V

### KESIMPULAN

#### V.1 Kesimpulan

Berdasarkan analisa hasil penelitian dengan judul : “Studi Penentuan Titik Konjugasi Pada Foto Yang Bertampalan (*Image Matching*) Menggunakan Metode *Area-Based Matching* Pada Foto Stereo”, maka dapat ditarik kesimpulan mengenai penelitian ini yaitu:

1. *image matching* merupakan salah satu proses yang sangat penting dalam proses restitusi foto.
2. Metode *area-based matching* merupakan metode untuk menentukan keakurasian titik konjugasi. Metode ini bekerja menggunakan teknik yang saling melengkapi dengan keunggulan masing-masing yaitu teknik NCC memiliki kecepatan komputasi, tetapi hanya menghasilkan ketelitian 1 piksel. Sebaliknya metode LSM, dengan menerapkan hitung kuadrat terkecil pada nilai keabuan, prosesnya relatif lebih lambat untuk konvergen namun mampu menghasilkan ketelitian hingga 0,01 piksel.
3. Evaluasi metode penentuan titik konjugasi dilakukan untuk menghasilkan koordinat piksel dengan keakurasian sub piksel. Pada kasus ini, hasil dari *normalized cross correlation* (antara kedua foto) sudah sangat identik (hasil *matching*). Namun dibutuhkan *least squares matching* untuk menghasilkan ketelitian yang lebih akurat.

## BAB V KESIMPULAN

### V.1 Kesimpulan

Berdasarkan analisis hasil penelitian dengan judul "Studi Penelitian Teknik Konjugasi Pada Foto Yang Berambaran (Image Matching) Menggunakan Metode Area-Based Matching Pada Foto Stereo", maka dapat ditarik kesimpulan

mengenai penelitian ini yaitu:

1. Metode matching merupakan salah satu proses yang sangat penting dalam

proses restitusi foto.

2. Metode area-based matching merupakan metode untuk menentukan

kekonsisian titik konjugasi. Metode ini bekerja menggunakan teknik yang

salah satunya dengan menggunakan masing-masing yaitu teknik NCC

memiliki kecepatan komputasi tetapi hanya menghasilkan ketelitian 1 piksel.

Sebaliknya metode LSM dengan menggunakan fungsi kuadrat terkecil pada

nilai keabuan, prosesnya relatif lebih lambat untuk konfigurasi ukuran gambar

menghasilkan ketelitian hingga 0,01 piksel.

3. Evaluasi metode penentuan titik konjugasi dilakukan untuk menghasilkan

koordinat piksel dengan kekonsisian sub piksel. Pada kasus ini, hasil dari

normalized cross correlation (antara kedua foto) sudah sangat identik (hasil

matching). Namun dibutuhkan *key point* matching untuk menghasilkan

ketelitian yang lebih akurat.

4. Untuk kasus pada penelitian ini juga dilakukan uji validasi untuk beberapa titik dengan menggunakan teknik *intersection* sehingga diperoleh standar deviasi yaitu **1.51554975 mm**.

## **V.2 Saran**

1. Perlunya diberikan ukuran *template* dan *search window* yang berbeda-beda pada metode *area based matching* untuk mengetahui tingkat keakurasian apabila ukuran *window* diubah-ubah.
2. Untuk memperoleh ketelitian maksimal maka hendaknya bagi peneliti yang ingin melanjutkan penelitian ini, maka tambahkan koreksi radiometrik untuk membandingkan ketelitian yang diperoleh sekarang.

## DAFTAR PUSTAKA

- Ackermann, F., 1984. "Digital Image Correlation: Performance and Potential Application in Photogrammetry". *Photogrammetric Record*, 11, pp.429-439.
- Agouris, P., and Schenk, T., 2000. "Multiple Image Matching". Department of Geodetic Science and Surveying. The Ohio University, Columbus, Ohio, USA.
- Bethel, J., 1997. "Least Squares Image Matching for CE604".
- Bethmann, F., and Luhmann T., 2010. "Least Squares Matching with Advanced Geometric Transformation Models". *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII, Part 5, Newcastle upon Tyne, UK. 2010.
- Bradski, G., and Kaehler, A., 2008. "Learning OpenCV". O'Reilly Media. USA.
- Campbell, N.A., and Wu, X., 2008. "Gradient Cross Correlation For Sub-Pixel Matching". *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B7. Beijing 2008
- Elaksher A.F., and Alharthy A.L., 2008. "Matching Conjugate Points Between Multi Resolution Satellite Images Using Geometric and Radiometric Properties". *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B3b. Beijing 2008.
- Elaksher, A., 2008. "A Multi-Photo Least Squares Matching Algorithm For Urban Area DEM Refinement Using Breaklines" *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B3a. Beijing 2008.
- Forstner, W., 1982. *On the Geometric Precision of Digital Correlation*. *International Archives of Photogrammetry and Remote Sensing, Symposium Helsinki Commission III, 24-Part 3*, 176-189.
- Grüen, A.W., 1985. "Adaptive Least Squares Correlation: A Powerful Image Matching Technique". *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14, pp.175-187.
- Grüen, A.W., 1996. "Least Squares Matching: A Fundamental Measurement Algorithm" in *Close Range Photogrammetry and Machine Vision*, K. B. Atkinson (ed.) Whittles Publishing, U.K., 217-255.

BAITARPUSTAKA

Abkerman, E. 1984. "Digital Image Correlation: Performance and Potential Application in Photogrammetry." *Photogrammetric Record*, 11, pp.430-439.

Agouri, P. and Schenk, T. 2000. "Multiple Image Matching: Description of Geodesic Science and Surveying, The Ohio University, Columbus, Ohio, USA.

Baibel, J. 1997. "Least Squares Image Matching for CCD's."

Beinhorn, E. and Lehmann, E. 2010. "Least Squares Matching with Advanced Geometric Transformation Models." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIII, Part 5, Warsaw upon June, UK, 2010.

Brackski, G., and Koehler, A. 2008. "Learning Open T.M." *CV Kelly Media*, USA.

Campbell, M.A., and Wu, Y. 2008. "Evaluation of Cross Correlation for Sub-Pixel Matching." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIII, Part B7, Beijing 2008.

Flückner, A.E., and Alparz, A.L. 2008. "Matching Conjugate Points between Multi Resolution Satellite Images Using Geometric and Radiometric Properties." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIII, Part B3B, Beijing 2008.

Flückner, A. 2008. "A Multi-Phase Least Squares Matching Algorithm for Urban Area DEM Adjustment Using Breaklines." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIII, Part B3A, Beijing 2008.

Foster, H. 1982. "On the Geometric Precision of Digital Correlation, International Archives of Photogrammetry and Remote Sensing, Symposium Helsinki Commission III, 24-27 June, 176-189.

Grain, A.H. 1982. "Adaptive Least Squares Correlation: A Forward Image Matching Technique." *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14, pp.173-187.

Grain, A.H. 1996. "Least Squares Matching: A Fundamental Assessment Algorithm." in *Close Range Photogrammetry and Machine Vision*, K. B. Atkinson (ed), White: Lubitztag, UK, 217-225.



- Hahn, M., 1997. "Automatic Control Point Measurement". *Photogrammetric Week '97*, Fritsch, D., Hobbie, D. (Eds.), pp. 115-126, Wichmann Verlag, Heidelberg.
- Helava, U.V., 1988. "Object Space Least Squares Correlation". *Photogrammetric Engineering & Remote Sensing*, Vol. 54(6), pp. 711-714.
- Jähne, B., 1997, "Practical Handbook on Image Processing for Scientific Applications". CRC Press LLC.
- Kraus, K., 1997. "Photogrammetry", Vol.2., 4<sup>th</sup> edition, Fer. Dümmers Verlag, Bonn.
- Leica Geosystem,. 2006. "Stereo Analyst User's Guide". United States of America.
- Lillesand, T.M, and Kiefer, R.W., 2000. "Remote Sensing and Image Interpretation". John Wiley and Sons Inc., New York, NY.
- Lühmann, T., Robson, S., Kyle, S., and Harley, I., 2006. "Close Range Photogrammetry, Principles, Techniques and Applications". Whittes Publishing, Scotland, UK., 5100 pp.
- Lühmann, T., 2000, "Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen". Herbert Wichmann Verlag, Heidelberg. In Potuckova, 2004. Ph.D Thesis.
- Makarovic, B., 1984. "Considerations On Image Matching-An Engineering Perspective". International Institute for Aerospace Survey and Earth Sciences (ITC). Netherlands.
- Mallat S., 1999. "A Wavelet Tour a Signal Processing". 2nd edition. Academic Press.
- Mikhail, E., Bethel, J., and McGlone, J., 2001. "Introduction to Modern Photogrammetry". John Wiley & Sons Inc., New York.
- Mitchell, H.L., and Pilgrim, L.J., 1987. "Selection of an Image Matching Algorithm". Department of Civil Engineering and Surveying. University of Newcastle.
- Pertl, A., 1984. "Digital Image Correlation with the Analytical Plotter Planicomp C-100". *International Archives of Photogrammetry and Remote Sensing* 25 (3B), S. 874-882.

Edin, M., 1997. "Automatic Control Point Measurement". Photogrammetric Week '97, Fensch, D., Hobbs, D. (Eds.), pp. 115-120, Wichmann Verlag Heidelberg.

Edin, M., 1988. "Object Space Least Squares Correlation". Photogrammetric Engineering & Remote Sensing, Vol. 54(6), pp. 711-714.

Edin, M., 1997. "Practical Handbook on Image Processing for Scientific Applications". CRC Press LLC.

Kraus, K., 1997. "Photogrammetry", 4<sup>th</sup> edition, Per. Düncker Verlag Bonn.

Leica Geosystem, 2000. "Stereo Analyst User's Guide". United States of America.

Lilienthal, T.M. and Kiefer, R.H., 2000. "Remote Sensing and Image Interpretation". John Wiley and Sons Inc., New York, NY.

Lilienthal, T., Kobson, S., Kyle, S., and Burdy, A., 2000. "Close Range Photogrammetry: Practical Techniques and Applications". Hines Publishing, Scotland, UK, 2100 pp.

Lilienthal, T., 2000. "Neuereichtphotogrammetrie. Grundlagen, Methoden und Anwendungen". Aachen Wichmann Verlag Heidelberg, in Fortschritt, 2004. PhD Thesis.

Makris, R., 1984. "Considerations On Image Matching-In Engineering Perspective". International Institute for Aerospace Survey and Earth Sciences (ITC), Netherlands.

Mallat, S., 1999. "A Wavelet Tour of Signal Processing", 2nd edition, Academic Press.

Mikhail, E., Bejbal, A. and Mcdonell, J., 2001. "Introduction to Modern Photogrammetry". John Wiley & Sons Inc., New York.

Mitchell, H.L. and Pflug, L.J., 1987. "Selection of an Image Matching Algorithm". Department of Civil Engineering and Surveying, University of Newcastle.

Pert, A., 1984. "Digital Image Correlation with the Analytical Point-Plane Correlation". International Archives of Photogrammetry and Remote Sensing 22 (3B), 287-292.

- Potuckova, M., 2004. "Image Matching and Its Application in Photogrammetry".  
Department of Development Planning Aalborg University, Aalborg, Denmark.
- Rosenholm, D., 1987a. "Empirical Investigation of Optimal Window Size Using The Least Squares Image Matching Method" *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Scien*, Vol 42 (1987) pp. 133-125.
- Rosenholm, D., 1987b. "Least Squares Matching Method: Same Experimental Results". *Photogrammetric Record*, 12, pp.493-512.
- Thévenaz P., Blu T., and Unser M., 1999. "Image Interpolation and Resampling".  
Swiss Federal Institute of Technology-Lausanne.
- Tsai, D.M., and Lin, C.T., 2002. "Fast Normalized Cross Correlation for Defect Detection". Department of Industrial Engineering and Management Yuan-Ze University, Chung-Li, Taiwan, R.O.C.
- Sachs, J., 2001. "Image Resampling". *Digital Light & Color*.
- Sarjakoski, T., and Lammi, J., 1996. "Least Squares Matching By Search". *The International Archives of the Photogrammetry, Remote Sensing. Vol. XXXI Part B3. Vienna 1996*.
- Schenk, T., 1999. "Digital Photogrammetry", TerraScience, Ohio, USA.
- Wolf, P.R., and Dewitt, B. A., 2000. "Element of Photogrammetry with Application in GIS 3<sup>rd</sup>", McGraw-Hill Higher Education. pp 334-341.
- Zhaltov, S., Y. and Sibiryakov, A., V., 1997. "Adaptive Subpixel Cross-correlation in a Point Correspondence Problem". State Research Institute of Aviation Systems, Moscow, Russia.

Ponckov, A., 2004. "Image Matching and its Application in Photogrammetry".  
Department of Development Planning Aalborg University, Aalborg, Denmark.

Rosenholm, D., 1987a. "Empirical Investigation of Optical Window Size Using The  
Least Squares Image Matching Method". The International Archives of the  
Photogrammetry, Remote Sensing and Spatial Information Science, Vol. 43  
(1987) pp. 133-135.

Rosenholm, D., 1987b. "Least Squares Matching Method: Some Experimental  
Results". Photogrammetric Record, 12, pp.493-513.

Thieryn, F., Bin, T., and Lauer, M., 1999. "Image Interpolation and Resampling".  
Swiss Federal Institute of Technology-Lausanne.

Yan, D.M., and Lin, C.T., 2002. "Fast Normalized Cross Correlation for Object  
Detection". Department of Industrial Engineering and Management Yum-Su  
University, Chung-Li, Taiwan, R.O.C.

Zachs, J., 2001. "Image Resampling". Digital Light & Color.

Zepkoski, T., and Lamm, J., 1986. "Least Squares Matching By Search". The  
International Archives of the Photogrammetry, Remote Sensing, Vol. XXVI  
Part B3, Vienna 1986.

Zhang, T., 1999. "Digital Photogrammetry". TerraScience, Ohio, USA.

Hoff, P.R., and Dearth, B.A., 2000. "Element of Photogrammetry with Application  
in GIS". McGraw-Hill Higher Education, pp.334-341.

Zhdanov, S. Y., and Sibiryakov, A. N., 1997. "Adaptive Digital Cross-correlation in  
a Point Correspondence Problem". State Research Institute of Aviation  
Systems, Moscow, Russia.

## LAMPIRAN

### NORMALIZED CROSS CORRELATION

```
namespace StereoMap
{
    //Obsolete...
    public class CrossCorrelation
    {
        #region Fields
        Image<Gray, Byte> _template;
        double _sumGrayValTemplate;
        double _sumSquaredValTemplate;
        Point _maxLoc; //approximate value: input for
                       //photogrammetric crosscorrelation --> on
                       //the matching wnd in pxl
        Point _matchedPos; //result from ph. cross corr
        double _corrCoefficient;

        //Least Square Matching
        LSM _lsm; //Photogrammetric Least Square Image Matching
        #endregion

        #region Constructor
        public CrossCorrelation(ref Image<Gray, Byte> source, ref
            Image<Gray, Byte> matchingImg, PointF centerTemplate)
        {
            Trace.WriteLine("CrossCorrelation constructor is
                called");
            //construct template window
            Rectangle rectTemplate = new
                Rectangle((int)((centerTemplate.X + 0.5) - 10),
                    (int)((centerTemplate.Y + 0.5) - 10), 21, 21);
            Trace.WriteLine("rect T L R B: " +
                rectTemplate.Top.ToString() + " " +
                rectTemplate.Left.ToString() + " " +
                rectTemplate.Right.ToString() + " " +
                rectTemplate.Bottom.ToString());
            Trace.WriteLine("rect Width Height: " +
                rectTemplate.Width.ToString() + " " +
                rectTemplate.Height.ToString());
            Trace.WriteLine("rect X Y: " + rectTemplate.X.ToString()
                + " " + rectTemplate.Y.ToString());

            Point centreTempl = new Point((int)(centerTemplate.X +
                0.5), (int)(centerTemplate.Y + 0.5));
            Trace.WriteLine("centreTempl: " +
                centreTempl.ToString());
            double centreX = (double)rectTemplate.X +
                ((double)rectTemplate.Width) / 2.0;
            double centreY = (double)rectTemplate.Y +
                ((double)rectTemplate.Height) / 2.0;
        }
    }
}
```

```

Trace.WriteLine("centreX centreY: " + centreX.ToString()
+ " " + centreY.ToString());
_template = source.Copy(rectTemplate);
_template.ROI.Offset(rectTemplate.X, rectTemplate.Y);

//Trace.WriteLine("_template ROI Upper L T: " +
_template.ROI.X.ToString() + " " +
_template.ROI.Y.ToString());
//Trace.WriteLine("_template ROI T L R B: " +
_template.ROI.Top.ToString() + " " +
_template.ROI.Left.ToString() + " " +
//
_template.ROI.Right.ToString() + "
" + _template.ROI.Bottom.ToString());
//Trace.WriteLine("_template ROI width height: " +
_template.ROI.Width.ToString() + " " +
_template.ROI.Height.ToString());

_sumGrayValTemplate = _template.GetSum().Intensity;
_sumSquaredValTemplate = SetSumSquare(ref _template);

//CrossCorrelation using Emgu
Image<Gray, float> temp =
    matchingImg.MatchTemplate(_template,
        TM_TYPE.CV_TM_CCOEFF_NORMED);
double minVal = 0.0;
double maxVal = 0.0;
Point minLoc = new Point();
Point maxLoc = new Point();
CvInvoke.cvMinMaxLoc(temp, ref minVal, ref maxVal, ref
    minLoc, ref maxLoc, IntPtr.Zero);
//_maxLoc = maxLoc;
_maxLoc.X = maxLoc.X + 10;
_maxLoc.Y = maxLoc.Y + 10;
//Trace.WriteLine("Min Max Value: " + minVal.ToString()
+ " " + maxVal.ToString());
//Trace.WriteLine("Min Max Loc: " + minLoc.ToString() +
" " + _maxLoc.ToString());
_template.Save(@"D:\Skripsi Via\Program Saya\Program
Rapid Mapping-Mine02062010\Image Matching\Edit
disini2\StereoMapping\StereoMap\bin\Debug\Template.jpg
");

//Photogrammetric CrossCorrelation
double CorrCoef = 0.0;
double alpha = 0.0;
double beta = 0.0;
Point matchedPoint = PhNormCrossCorrelation(ref
    matchingImg, out CorrCoef, out alpha, out beta);
Point2D conjugatePoint;
bool success;
if (CorrCoef > 0.695 && matchedPoint.X != 0 &&
    matchedPoint.Y != 0)
{
    _corrCoefficient = CorrCoef;
}

```

```

        _matchedPos = matchedPoint;
        //construct Matching Patch window
        Rectangle rectMatchingPatch = new
            Rectangle((int)((_matchedPos.X + 0.5) - 10),
                (int)((_matchedPos.Y + 0.5) - 10), 21, 21);
        Trace.WriteLine("Matched Point from EMGU: " +
            _maxLoc.ToString());
        Trace.WriteLine("Matched Point from Ph Cross Corr: "
            + matchedPoint.ToString() + " with corr coef: " +
            CorrCoef.ToString());
        Trace.WriteLine("ho h1: " + alpha.ToString() + " "
            + beta.ToString());
        //Trigger LSM
        _lsm = new LSM(ref rectTemplate, ref centreTempl,
            ref _matchedPos, ref source,
            ref matchingImg, alpha, beta, rectMatchingPatch);
        conjugatePoint = _lsm.FindConjugatePoint(out
            success);
    }
}
#endregion

#region Properties
//A location where a maximum coeff value occurs
public Point MaxLocation
{
    get { return _maxLoc; }
}

public Point MatchedPointCorr
{
    get { return _matchedPos; }
}

public double CorrelationCoefficient
{
    get { return _corrCoefficient; }
}
}
#endregion

#region Methods
private Point PhNormCrossCorrelation(ref Image<Gray, Byte>
    matchingImg, out double corrCoef,
    out double alpha, out double beta)
{
    //Trace.WriteLine("template average value: " +
        _template.GetAverage().ToString());
    //Trace.WriteLine("template sum value: " +
        _template.GetSum().ToString());
    //Trace.WriteLine("template row col : " +
        _template.Rows.ToString() + " " +
        _template.Cols.ToString());

    //create a searching window which size of 15x15

```



```

Point matchedPt = new Point();
double maxCorrVal = 0.7;
double alph = 0.0;
double bet = 0.0;

for (int row = _maxLoc.Y - 15; row <= _maxLoc.Y + 15;
row++)
{
    for (int col = _maxLoc.X - 15; col <= _maxLoc.X +
15; col++)
    {
        Image<Gray, Byte> matchingPatch =
SetMatchingWindow(ref matchingImg, col, row);
//create matching patch

//PrintPatch(ref matchingPatch);
double temp = GetCorrelationCoef(ref
matchingPatch, out alph, out bet);
if (temp > maxCorrVal)
{
    matchedPt.X = col;
    matchedPt.Y = row;
    maxCorrVal = temp;
}
}
}
corrCoef = maxCorrVal;
alpha = alph;
beta = bet;
//Rectangle rect = new Rectangle((int)((_maxLoc.X + 0.5)
- 11), (int)((_maxLoc.Y + 0.5) - 11), 21, 21);
//Image<Gray, Byte> matchingWnd =
matchingImg.Copy(rect);
//matchingWnd.Save //matchingWnd.Save(@"D:\Skripsi
Via\Program Saya\Program Rapid Mapping-
Mine02062010\Image Matching\Edit
disini2\StereoMapping\StereoMap\bin\Debug\Search
Window.jpg");
Trace.WriteLine("Max Corr Coef: " +
maxCorrVal.ToString());
return matchedPt;
}

private double GetCorrelationCoef(ref Image<Gray, Byte>
matching, out double alpha, out double beta)
{
    //construct matching patch
    Debug.Assert(matching.Cols == _template.Cols &&
matching.Rows == _template.Rows);
    int elementNumbers = matching.Rows * matching.Cols;
    //double avgGrayValMatchPatch =
    matching.GetAverage().Intensity;
    double sumGrayValMatchPatch =
    matching.GetSum().Intensity;

```



```

double sumSquaredGrayValMatchPatch = SetSumSquare(ref
    matching);

double SumMult = SumMultiply(ref matching);
//Trace.WriteLine("SumMult: " + SumMult.ToString());

//From Element of Photogrammetry, Wolf page 336:
//Sx=template, Sy=matchingPatch
double SxSquared = _sumSquaredValTemplate -
    ((_sumGrayValTemplate * _sumGrayValTemplate) /
    elementNumbers);
double SySquared = sumSquaredGrayValMatchPatch -
    ((sumGrayValMatchPatch * sumGrayValMatchPatch) /
    elementNumbers);
double Sxy = SumMult - ((_sumGrayValTemplate *
    sumGrayValMatchPatch) / elementNumbers);

beta = Sxy / SxSquared;
alpha = (sumGrayValMatchPatch / elementNumbers) - (beta
    * (_sumGrayValTemplate / elementNumbers));
double CorrCoef = Sxy / Math.Sqrt(SxSquared *
    SySquared);
return CorrCoef;
}

private double SumMultiply(ref Image<Gray, Byte> img)
{
    double sum = 0.0;
    Debug.Assert(img.Cols == _template.Cols && img.Rows ==
        img.Rows);
    for (int y = 0; y < img.Rows; y++)
    {
        for (int x = 0; x < img.Cols; x++)
        {
            sum += _template[y, x].Intensity * img[y,
                x].Intensity;
        }
    }
    return sum;
}

private double SetSumSquare(ref Image<Gray, Byte> img)
{
    int row = img.Rows;
    int col = img.Cols;
    double sumSquared = 0;
    for (int y = 0; y < row; y++)
    {
        for (int x = 0; x < col; x++)
        {
            Emgu.CV.Structure.Gray g = img[y, x];
            sumSquared += (g.Intensity * g.Intensity);
        }
    }
}

```

```

    }

    return sumSquared;
}

private Image<Gray, Byte> SetMatchingWindow(ref Image<Gray,
    Byte> matchingImg, int x, int y)
{
    Rectangle rect = new Rectangle(x - 10, y - 10, 21, 21);
    Image<Gray, Byte> matchingPatch =
        matchingImg.Copy(rect);
    return matchingPatch;
}

private void PrintPatch(ref Image<Gray, Byte> img)
{
    Trace.WriteLine("Patch gray value:");
    int row = img.Rows;
    int col = img.Cols;
    for (int y = 0; y < row; y++)
    {
        for (int x = 0; x < col; x++)
        {
            Trace.Write(img[y, x].Intensity.ToString() + "
                ");
        }
        Trace.WriteLine("");
    }
}
#endregion
}
}

```

**LAMPIRAN**  
**LEAST SQUARE MATCHING**  
**6 Parameter**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.IO;
using System.Drawing;

using Emgu.CV;
using Emgu.Util;
using Emgu.CV.UI;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;

namespace StereoMap
{
    /// <summary>
    /// LSM Implementation according to Toni Schenk
    /// Using 6 affine Parameters: shift: dx dy; Scale: x y. Parameters:
    /// a0,a1,a2; b0, b1,b2
    ///  $x = a_0 + a_1 * i + a_2 * j$  ; i = col in pixel
    ///  $y = b_0 + b_1 * i + b_2 * j$  ; j = row in pixel
    /// Steps:
    /// 1. Select the centre of one template
    /// 2. Determine approximate location for the matching patch
    /// 3. Start first iteration with matching patch at the approximate
    /// location
    /// 4. Transform matching patch and determine the gray value for the
    /// tessellation (resampling)
    /// 5. Repeat Adjustment and resampling sequence until it is converged
    /// </summary>
    public class LSM6A
    {
        #region Fields
        /// <summary>
        /// Template Image
        /// </summary>
        Image<Gray, Byte> _templateImg;
        public CSML.Matrix A; // Koefisien Matrix
        public CSML.Matrix _b; // Weight Matrix
        public double r; // Degree of Freedom

        /// <summary>
        /// Matching Image
        /// </summary>
        Image<Gray, Byte> _matchingImg;
    }
}
```

# LAMPIKAN

## LEAST SQUARE MATCHING

Parameter

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.IO;
using System.Drawing;

using Emgu.CV;
using Emgu.CV.Util;
using Emgu.CV.UI;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;

namespace StereoMap
{
    // ...
    // 1. Select the centre of one template
    // 2. Determine approximate location for the matching patch
    // 3. Start first iteration with matching patch at the approximate
    //     location
    // 4. Transform matching patch and determine the gray value for the
    //     resampling
    // 5. Repeat adjustment and resampling sequence until it is converged
    // ...
}

region Fields
{
    // ...
    // Template Image
    // ...
    public double r; // Degree of Freedom
    public double[] b; // Weight Matrix
    public double[] A; // Koefisien Matrix
}

// ...
// Matching Image
// ...
// ...
}

```

```

/// <summary>
/// Coordinates of the centre of a selected template patch, in pixel
/// </summary>
Point2D _centreTemplate;

/// <summary>
/// Coordinate of the centre of the matching patch, in pixel. It is
/// resulted from CrossCorrelation
/// </summary>
Point2D _matchedPoint;          //in pixel

/// <summary>
/// Coordinate of the final matching patch, in pixel. It is from the LSM
/// </summary>
Point2D _correctedMatchedPt;   //final result

/// <summary>
/// Affine Parameters: Shift x
/// </summary>
double _a0;

/// <summary>
/// Affine Parameters: Shift y
/// </summary>
double _b0;

/// <summary>
/// Affine Parameters: Scale x
/// </summary>
double _a1;

/// <summary>
/// Affine Parameters: Scale y
/// </summary>
double _b2;

/// <summary>
/// Affine Parameters: Shear y
/// </summary>
double _a2;

/// <summary>
/// Affine Parameters: Shear x
/// </summary>
double _b1;

double _XRightPhoto;//Tambahan
double _YRightPhoto;
double _XLeftPhoto;
double _YLeftPhoto;

/// <summary>
/// constant: gray value different  $T(i,j) - m(i,j)$ 
/// </summary>

```

```

// Affine Parameters: Shear x
double _d1;

// Affine Parameters: Shear y
double _d2;

// Affine Parameters: Scale x
double _a1;

// Affine Parameters: Scale y
double _b2;

// Affine Parameters: Shift x
double _b0;

// Affine Parameters: Shift y
double _a0;

// Coordinate of the final matching patch, in pixel. It is from the LSM
// final result
int finalMatchedPoint;

// Affine Parameters: MatchedPoint;
int inPixel;

// Coordinate of the centre of the matching patch, in pixel. It is
// result from crossCorrelation
// Affine Parameters: MatchedPoint;
int inPixel;

// Coordinate of the centre of a selected template patch; in pixel
// Affine Parameters: MatchedPoint;
int inPixel;

```

```

//CSML.Matrix _b;

/// <summary>
/// Indicator whether the LSM iteration can converge
/// </summary>
bool _lsmConverge;

/// <summary>
/// Transformed grid.
/// m[row, 1] = grid for x direction (col) on Matching Img
/// m[row, 2] = grid for y direction (row) on Matching Img
/// m[row, 3] = grid for x direction (col) on Template Img
/// m[row, 4] = grid for y direction (row) on Template Img
/// </summary>
CSML.Matrix _gridsXY;

Point2D _previousMatched;
#endregion Fields

#region Constructors
public LSM6A()
{
    _centreTemplate = new Point2D();
    _matchedPoint = new Point2D();
    _correctedMatchedPt = new Point2D();

    _a0 = 0.0;
    _a1 = 0.0;
    _a2 = 0.0;
    _b0 = 0.0;
    _b1 = 0.0;
    _b2 = 0.0;
    _lsmConverge = false;

    _gridsXY = new CSML.Matrix(19, 4);
    _previousMatched = new Point2D();
}

public LSM6A(ref Image<Gray, Byte> source, ref Image<Gray, Byte>
    matchingImg, Point2D centerTemplate, Point centreMatchingPatch)
    : this()
{
    Trace.WriteLine("\t\t----- Constructor LSM6A() is
        called -----");
    _templateImg = source;
    _matchingImg = matchingImg;
    _centreTemplate.X = (double)((int)(centerTemplate.X + 0.5));
    _centreTemplate.Y = (double)((int)(centerTemplate.Y + 0.5));
    _matchedPoint.X = (double)centreMatchingPatch.X;
    _matchedPoint.Y = (double)centreMatchingPatch.Y;
    _correctedMatchedPt.X = (double)centreMatchingPatch.X;
    _correctedMatchedPt.Y = (double)centreMatchingPatch.Y;

    _previousMatched.X = (double)centreMatchingPatch.X;
    _previousMatched.Y = (double)centreMatchingPatch.Y;
}

```



```

        SetInitialGridsXY();
    }
#endregion Constructors

#region Properties
#endregion Properties

#region Private Methods
private Point2D Affine6Params(double x, double y)
{
    Point2D pt = new Point2D();
    pt.X = _a0 + _a1 * x + _a2 * y;
    pt.Y = _b0 + _b1 * x + _b2 * y;
    return pt;
}

private void SetInitialGridsXY()
{
    //Matching Patch 17x17 ==> add two more to account for border -->
    19x19
    int rowStartMatchingPatch = (int)_matchedPoint.Y - 9;
    int rowEndMatchingPatch = (int)_matchedPoint.Y + 9;
    int colStartMatchingPatch = (int)_matchedPoint.X - 9;
    int colEndMatchingPatch = (int)_matchedPoint.X + 9;

    //Template Patch 17x17 ==> add two more to account for border -->
    19x19
    int rowStartTemplatePatch = (int)_centreTemplate.Y - 9;
    int rowEndTemplatePatch = (int)_centreTemplate.Y + 9;
    int colStartTemplatePatch = (int)_centreTemplate.X - 9;
    int colEndTemplatePatch = (int)_centreTemplate.X + 9;

    //int rowNumbers = 17 * 17; //Tambahan
    //A = new CSML.Matrix(rowNumbers, 6); //Tambahan
    //_b = new CSML.Matrix(rowNumbers, 1); //Tambahan
    int index = 1;
    for (int counter = 0; counter < 19; counter++)
    {
        int xMPatch = counter + colStartMatchingPatch;
        int yMPatch = counter + rowStartMatchingPatch;
        int xTPatch = counter + colStartTemplatePatch;
        int yTPatch = counter + rowStartTemplatePatch;

        _gridsXY[index, 1].Re = (double)xMPatch;
        _gridsXY[index, 2].Re = (double)yMPatch;
        _gridsXY[index, 3].Re = (double)xTPatch;
        _gridsXY[index, 4].Re = (double)yTPatch;
        index++;
    }
    //Trace.WriteLine("_gridsXY: \n" + _gridsXY.ToString());
}

private void RearrangeGridsXY(CSML.Matrix solution)

```



```

        SetInitialGridsXY();
    }
    //region Constructors
    //region Properties
    //region Private Methods
    private void AffineParams(double x, double y)
    {
        double pt = new Point(x, y);
        pt.X = _a0 + _a1 * x + _a2 * y;
        pt.Y = _b0 + _b1 * x + _b2 * y;
        return pt;
    }
    private void SetInitialGridsXY()
    {
        //Matching Patch IXIV ==> add two more to account for border -->
        IXIV
        int rowStartMatchPatch = (int)_matchedPoint.Y - 2;
        int rowEndMatchPatch = (int)_matchedPoint.Y + 2;
        int colStartMatchPatch = (int)_matchedPoint.X - 2;
        int colEndMatchPatch = (int)_matchedPoint.X + 2;

        //Match Patch IXIV ==> add two more to account for border -->
        IXIV
        int rowStartMatchPatch = (int)_centerPoint.Y - 2;
        int rowEndMatchPatch = (int)_centerPoint.Y + 2;
        int colStartMatchPatch = (int)_centerPoint.X - 2;
        int colEndMatchPatch = (int)_centerPoint.X + 2;

        //int rowNumbers = IX + 1; //IXIV
        //A = new CSM.Matrix(rowNumbers, 6); //IXIV
        //B = new CSM.Matrix(rowNumbers, 1); //IXIV
        int index = 1;
        for (int counter = 0; counter < 12; counter++)
        {
            int xPatch = counter + rowStartMatchPatch;
            int yPatch = counter + rowEndMatchPatch;
            int xTPatch = counter + colStartMatchPatch;
            int yTPatch = counter + colStartMatchPatch;

            _GridsXY[index, 1].Re = (double)xPatch;
            _GridsXY[index, 2].Re = (double)yPatch;
            _GridsXY[index, 3].Re = (double)xTPatch;
            _GridsXY[index, 4].Re = (double)yTPatch;
            index++;
        }
        //Trace.WriteLine("GridsXY: /n" + _GridsXY.ToString());
    }
    private void RearrangeGridsXY(CSM csm, double solution)

```

```

    {
        int rows = _gridsXY.RowCount;
        //Trace.WriteLine("row counts: " + row.ToString());
        for (int index = 1; index <= rows; index++)
        {
            _gridsXY[index, 1].Re += solution[1, 1].Re + //shift x
                solution[2, 1].Re * _gridsXY[index, 1].Re + //scale x
                solution[3, 1].Re * _gridsXY[index, 2].Re; //shear x
            _gridsXY[index, 2].Re += solution[4, 1].Re + //shift y
                solution[5, 1].Re * _gridsXY[index, 1].Re + //scale y
                solution[6, 1].Re * _gridsXY[index, 2].Re; //shear y
        }
        //Trace.WriteLine("Rearrange GridXY: \n" + _gridsXY.ToString());
    }
}

private void ComposeA_b6Params1stIteration(ref CSML.Matrix a, ref
CSML.Matrix b)
{
    int rowNumbers = 17 * 17;
    //CSML.Matrix A = new CSML.Matrix(rowNumbers, 6);
    A = new CSML.Matrix(rowNumbers, 6);
    _b = new CSML.Matrix(rowNumbers, 1);
    int index = 1;
    int row_col = _gridsXY.RowCount;
    Trace.WriteLine("row_col: " + row_col.ToString());

    for (int rowIndex = 2; rowIndex <= row_col - 1; rowIndex++)
    {
        for (int colIndex = 2; colIndex <= row_col - 1; colIndex++)
        {
            //Resample matching patch onto template image == original
            matching patch
            //Compute Gradient of A on the matching patch
            //m0(x,y)=m(i,j)
            double x1 = _gridsXY[colIndex - 1, 1].Re; //x-1
            double x = _gridsXY[colIndex, 1].Re; //x
            double x2 = _gridsXY[colIndex + 1, 1].Re; //x+1

            double y1 = _gridsXY[rowIndex - 1, 2].Re; //y-1
            double y = _gridsXY[rowIndex, 2].Re; //y
            double y2 = _gridsXY[rowIndex + 1, 2].Re; //y+1

            double dx1 = _matchingImg[(int)y, (int)x1].Intensity;
                //g(x-1)
            double dx2 = _matchingImg[(int)y, (int)x2].Intensity;
                //g(x+1)
            double dx = (dx2 - dx1) / 2.0; //((g(x+1)-g(x-1))/2

            double dy1 = _matchingImg[(int)y1, (int)x].Intensity;
                //g(y-1)
            double dy2 = _matchingImg[(int)y2, (int)x].Intensity;
                //g(y+1)
            double dy = (dy2 - dy1) / 2.0; //((g(y+1)-g(y-1))/2

```

```

int rows = _gridXY.RowCount;
\\Trace.WriteLine("row counts: " + row.ToString());
for (int index = 1; index <= rows; index++)
{
    _gridXY[index, 1].Re += solution[1, 1].Re +
    solution[2, 1].Re * _gridXY[index, 1].Re +
    solution[3, 1].Re * _gridXY[index, 2].Re;
    _gridXY[index, 2].Re += solution[4, 1].Re +
    solution[5, 1].Re * _gridXY[index, 1].Re +
    solution[6, 1].Re * _gridXY[index, 2].Re;
}
\\Trace.WriteLine("Rearrange GridXY: /n" + _gridXY.ToString());
}
}

```

generate void ComposeA\_B\_Parameters(Iteration of CSM, row numbers, col

CSM, col numbers)

```

int rowNumbers = 17 * 17;
\\CSM.Matrix A = new CSM.Matrix(rowNumbers, 6);
A = new CSM.Matrix(rowNumbers, 6);
B = new CSM.Matrix(rowNumbers, 1);
int index = 1;
int row_col = _gridXY.RowCount;
\\Trace.WriteLine("row col: " + row_col.ToString());
for (int rowindex = 1; rowindex <= row_col - 1; rowindex++)
{
    for (int colindex = 1; colindex <= row_col - 1; colindex++)
    {
        \\Resample matching patch onto Equate image == original
        matching patch
        \\Compute gradient of A on the matching patch
        \\m(x,y)=m(i,j)
        double x1 = _gridXY[colindex - 1, 1].Re;
        double x = _gridXY[colindex, 1].Re;
        double x2 = _gridXY[colindex + 1, 1].Re;
        \\x-1
        double y1 = _gridXY[rowindex - 1, 2].Re;
        double y = _gridXY[rowindex, 2].Re;
        double y2 = _gridXY[rowindex + 1, 2].Re;
        \\y-1
        double dx1 = _matchingImg[(int)y1, (int)x1].Intensity;
        \\g(x-1)
        double dx2 = _matchingImg[(int)y, (int)x2].Intensity;
        \\g(x+1)
        double dx = (dx2 - dx1) * 2.0;
        \\(g(x+1)-g(x-1))/2
        double dy1 = _matchingImg[(int)y1, (int)x].Intensity;
        \\g(y-1)
        double dy2 = _matchingImg[(int)y2, (int)x].Intensity;
        \\g(y+1)
        double dy = (dy2 - dy1) * 2.0;
        \\(g(y+1)-g(y-1))/2
    }
}
}
}

```

```

double dn = _matchingImg[(int)y, (int)x].Intensity;
           //g(x,y)

//Compute A
A[index, 1].Re = dx;      //a0: dx
A[index, 2].Re = dx * x; //a1: dx * x
A[index, 3].Re = dx * y; //a2: dx * y
A[index, 4].Re = dy;     //b0: dy
A[index, 5].Re = dy * x; //b1: dy * x
A[index, 6].Re = dy * y; //b2: dy * y

//Compute b
double xT = _gridsXY[colIndex, 3].Re; //x on template
double yT = _gridsXY[rowIndex, 4].Re; //y on template
double dnT = _templateImg[(int)yT, (int)xT].Intensity;
           //t(x,y)
_b[index, 1].Re = dnT - dn;

index++;
}
}
//Trace.WriteLine("A: \n" + A.ToString());//tambahan
//Trace.WriteLine("B: \n" + _b.ToString());//tambahan==> cek

CSML.Matrix AT = A.Transpose();
a = AT * A;
b = AT * _b;
}

private void ComposeA_b6Params(ref CSML.Matrix a, ref CSML.Matrix b)
{
//looping: 19
int rowNumbers = 17 * 17;
//CSML.Matrix A = new CSML.Matrix(rowNumbers, 6);
A = new CSML.Matrix(rowNumbers, 6);
int index = 1;
int row_col = _gridsXY.RowCount;
Trace.WriteLine("row_col: " + row_col.ToString());

for (int rowIndex = 2; rowIndex <= row_col - 1; rowIndex++)
{
for (int colIndex = 2; colIndex <= row_col - 1; colIndex++)
{
//Resample matching patch onto template image == original
matching patch
//Compute Gradient of A on the matching patch
//m(x,y) != m(i,j) ==> for matching image
double x1 = _gridsXY[colIndex - 1, 1].Re; //x-1
double x = _gridsXY[colIndex, 1].Re; //x
double x2 = _gridsXY[colIndex + 1, 1].Re; //x+1

double y1 = _gridsXY[rowIndex - 1, 2].Re; //y-1
double y = _gridsXY[rowIndex, 2].Re; //y
double y2 = _gridsXY[rowIndex + 1, 2].Re; //y+1
}
}
}
}

```

```

double db = _matching[m](x, y, (int)x, Intensity);
    \A(x,y)

    \Compute A
    A[index, 1].Re = dx; \A0: dx
    A[index, 2].Re = dx * x; \A1: dx * x
    A[index, 3].Re = dx * y; \A2: dx * y
    A[index, 4].Re = dy; \A0: dy
    A[index, 5].Re = dy * x; \A1: dy * x
    A[index, 6].Re = dy * y; \A2: dy * y

    \Compute b
    double xt = _gridXY[colIndex, 3].Re; \x on template
    double yt = _gridXY[rowIndex, 4].Re; \y on template
    double bnt = _template[m](xt, yt, (int)xt, Intensity);
    \A(x,y)
    _b[index, 1].Re = bnt - db;

    index++;
}
}
\Trace.WriteLine("A: /n" + A.ToString()); \tabspan
\Trace.WriteLine("B: /n" + _b.ToString()); \tabspan=> csk

CSML.Matrix A = A.Transpose();
a = AT * A;
b = AT * _b;
}

private void Compose_Parameters_of_CSML_Matrix_a_and_b(CSML.Matrix a, CSML.Matrix b)
{
    \looping: 10
    int rowNumbers = 17 * 17;
    \CSML.Matrix A = new CSML.Matrix(rowNumbers, 6);
    A = new CSML.Matrix(rowNumbers, 6);
    int index = 1;
    int row_col = _gridXY.RowCount;
    \Trace.WriteLine(row_col + row_col.ToString());
    for (int rowindex = 2; rowindex <= row_col - 1; rowindex++)
    {
        for (int colindex = 2; colindex <= row_col - 1; colindex++)
        {
            \Reassemble matching patch onto template image == original
            matching patch
            \Compute Gradient of A on the matching patch
            \m(x,y) != m(i,j) ==> for matching image
            double x1 = _gridXY[colIndex - 1, 1].Re; \x-1
            double x = _gridXY[colIndex, 1].Re; \x
            double x2 = _gridXY[colIndex + 1, 1].Re; \x+1
            double y1 = _gridXY[rowIndex - 1, 2].Re; \y-1
            double y = _gridXY[rowIndex, 2].Re; \y
            double y2 = _gridXY[rowIndex + 1, 2].Re; \y+1
        }
    }
}

```

```

//Bilinear Interpolation
int dx1 = BilinearInterpolateMatching(x1, y); //g(x-1)
int dx2 = BilinearInterpolateMatching(x2, y); //g(x+1)
int dgx = (dx2 - dx1) / 2; //x gradient

int dn = BilinearInterpolateMatching(x, y); //g(x,y)
int dy1 = BilinearInterpolateMatching(x, y //g(y-1)
int dy2 = BilinearInterpolateMatching(x, //g(y+1)
int dgy = (dy2 - dy1) / 2; //y gradient

//Compose A
A[index, 1].Re = dgx; //a0: dx
A[index, 2].Re = dgx * x; //a1: dx * x
A[index, 3].Re = dgx * y; //a2: dx * y
A[index, 4].Re = dgy; //b0: dy
A[index, 5].Re = dgy * x; //b1: dy * x
A[index, 6].Re = dgy * y; //b2: dy * y

//Compute b
//=> for template image
double xT = _gridsXY[colIndex, 3].Re; //x on template
double yT = _gridsXY[rowIndex, 4].Re; //y on template
double dnT = _templateImg[(int)yT, (int)xT].Intensity;
//t(x,y)
_b[index, 1].Re = dnT - dn;
index++;
}
}

//Trace.WriteLine("A: \n" + A.ToString());//tambahan//
//Trace.WriteLine("B: \n" + _b.ToString());//tambahan=> cek

CSML.Matrix AT = A.Transpose();
a = AT * A;
b = AT * _b;
}

/// <summary>
/// Bilinear interpolation on Matching Image
/// </summary>
/// <param name="x">on Matching Image Coord</param>
/// <param name="y">on Matching Image Coord</param>
/// <returns>Gray Value from the Matching Image</returns>
private int BilinearInterpolateMatching(double x, double y)
{
int X = (int)x;
int Y = (int)y;
double dx = x - (double)X;
double dy = y - (double)Y;
double g0 = _matchingImg[Y, X].Intensity; //go(x,y) --> term 0
double g1 = _matchingImg[Y, X + 1].Intensity; //go(x+1,y)
double g2 = _matchingImg[Y + 1, X].Intensity; //go(x,y+1)
double g3 = _matchingImg[Y + 1, X + 1].Intensity; //go(x+1,y+1)
double term1 = (g1 - g0) * dx; //term 1
double term2 = (g2 - g0) * dy; //term 2
}

```





```

        double term3 = (g3 + g0 - g1 - g2) * dx * dy; //term 3
        int val = (int)(g0 + term1 + term2 + term3 + 0.5);
        return val;
    }

#endregion Private Methods

#region Public Methods
public bool FindLSMConjugatePoint()
{
    CSML.Matrix aTa = new CSML.Matrix();
    CSML.Matrix aTb = new CSML.Matrix();
    CSML.Matrix solution = new CSML.Matrix();
    CSML.Matrix AX = new CSML.Matrix();//Tambahan
    double rCol;//Tambahan
    double rRow;//Tambahan

    ComposeA_b6Params1stIteration(ref aTa, ref aTb);
    bool aTaCond = aTa.IsSymmetricPositiveDefinite();
    double aTaDet = aTa.Determinant().Re;
    Trace.WriteLine("Det aTa: \t" + aTaDet.ToString());
    Trace.WriteLine("Is aTa symmetric positif definite: \t"
        + aTaCond.ToString());
    //Trace.WriteLine("aTa: \n" + aTa.ToString());
    //Trace.WriteLine("ATb : \n" + aTb.ToString());
    //Trace.WriteLine("aTa Inverse: \n" + aTa.Inverse().ToString());

    int iteration = 2;

    int Wdth = _matchingImg.Width;//Tambahan
    int Hght = _matchingImg.Height;//Tambahan

    int Wdh = _templateImg.Width;//Tambahan
    int Hgt = _templateImg.Height;//Tambahan

    const double DELTA = 0.01; //pixel
    if (aTaCond && aTaDet > 0.0)
    {
        solution = CSML.Matrix.Solve(aTa, aTb);
        RearrangeGridsXY(solution);
        _correctedMatchedPt.X += solution[1, 1].Re +
            solution[2, 1].Re * _correctedMatchedPt.X + solution[3,
            1].Re * _correctedMatchedPt.Y;
        _correctedMatchedPt.Y += solution[4, 1].Re +
            solution[5, 1].Re * _correctedMatchedPt.X + solution[6,
            1].Re * _correctedMatchedPt.Y;
        _lsmConverge = Threshold(solution, DELTA);
        // _lsmConverge = Threshold2();
        //if (!_lsmConverge)
        //{
        //    _previousMatched.X = _correctedMatchedPt.X;
        //    _previousMatched.Y = _correctedMatchedPt.Y;
        //}
    }
}

```



```

        double term3 = (g3 + g0 - g1 - g2) * dx * dy; // term 3
        int val = (int)(g0 + term1 + term2 + term3 + 0.5);
        return val;
    }

    Region Private Methods

    Region Public Methods
    public void FindSMConjugatePoint()
    {
        CSM.Solve(ata = new CSM.Solve(ata, atb));
        CSM.Solve(atb = new CSM.Solve(ata, atb));
        CSM.Solve(solution = new CSM.Solve(ata, atb));
        CSM.Solve(AX = new CSM.Solve(ata, atb)); // Tamaban
        double rCoj; // Tamaban
        double rRow; // Tamaban

        ComposeSeparation(ata, atb, atb);
        bool atCond = ata.IsSymmetricPositiveDefinite();
        double atDet = ata.Determinant().Re;
        Console.WriteLine("Det ata: /n" + atDet.ToString());
        Console.WriteLine("Is ata symmetric positive definite: /n"
            + atCond.ToString());
        // Trace.WriteLine("ata: /n" + ata.ToString());
        // Trace.WriteLine("atb: /n" + atb.ToString());
        // Trace.WriteLine("ata Inverse: /n" + ata.Inverse().ToString());

        int iteration = 5;

        int Wdth = _matching.Wdth; // Tamaban
        int Hght = _matching.Hght; // Tamaban

        int Wdh = _tempMatch.Wdth; // Tamaban
        int Hgt = _tempMatch.Hght; // Tamaban

        const double DELTA = 0.01; // pixel
        if (atCond && atDet > 0.0)
        {
            solution = CSM.Solve.Solve(ata, atb);
            RearrangeXY(solution);
            _correctMatchedPt.X += solution[1].Re +
                solution[2].Re * _correctMatchedPt.X + solution[3].
                Re * _correctMatchedPt.Y;
            _correctMatchedPt.Y += solution[4].Re +
                solution[5].Re * _correctMatchedPt.X + solution[6].
                Re * _correctMatchedPt.Y;
            _IsmConverge = Threshold(solution, DELTA);
            // _IsmConverge = Threshold();
            // if (!_IsmConverge)
            // {
            //     _previousMatched.X = _correctMatchedPt.X;
            //     _previousMatched.Y = _correctMatchedPt.Y;
            // }
        }
    }
}

```

```

_XRightPhoto = (_correctedMatchedPt.X - ((Wdth / 2) - 0.5))
*0.01219; //Tambahan
_YRightPhoto = (((Hght / 2) - 0.5) - _correctedMatchedPt.Y) *
0.01219; //Tambahan

_XLeftPhoto = (_centreTemplate.X - ((Wdh / 2) - 0.5)) *
0.01219; //Tambahan
_YLeftPhoto = (((Hgt / 2) - 0.5) - _centreTemplate.Y) *
0.01219; //Tambahan

Trace.WriteLine("Iteration 1: " + "\n" + solution.ToString());
Trace.WriteLine("_correctedMatchedPt X Y: " +
    _correctedMatchedPt.X.ToString() + " " +
    _correctedMatchedPt.Y.ToString());

//Trace.WriteLine("Left Photo Coordinate XPhoto YPhoto: " +
    _XLeftPhoto.ToString() + " " +
    _YLeftPhoto.ToString());//tambahan
//Trace.WriteLine("Right Photo Coordinate XPhoto YPhoto: " +
    _XRightPhoto.ToString() + " " +
    _YRightPhoto.ToString());//tambahan

AX = A * solution;
CSML.Matrix AxB = new CSML.Matrix();
AxB = AX - _b;
CSML.Matrix VT = new CSML.Matrix();
VT = AxB.Transpose();
CSML.Matrix VtV = new CSML.Matrix();
VtV = VT * AxB;
rCol = A.ColumnCount;
rRow = A.RowCount;
r = rRow - rCol;
//double tes = VtV[1].Re;
double VtVCo = VtV[1].Re;

//VtVCo = VtV[0];
// double VtVDo = VtVCo.Re;
double So;
So = Math.Sqrt(VtVCo) / (r);

//Trace.WriteLine("AX: " + "\n" + AX.ToString());
//Trace.WriteLine("V: " + "\n" + AxB.ToString());
//Trace.WriteLine("VT: " + "\n" + VtV.ToString());
//Trace.WriteLine("r: " + "\n" + r.ToString());
Trace.WriteLine("So: " + "\n" + So.ToString());

while (_lsmConverge && iteration <= 75)
{
    ComposeA_b6Params(ref aTa, ref aTb);
    solution = CSML.Matrix.Solve(aTa, aTb);
    RearrangeGridsXY(solution);
    _correctedMatchedPt.X += solution[1, 1].Re +
        solution[2, 1].Re * _correctedMatchedPt.X +
        solution[3, 1].Re * _correctedMatchedPt.Y;
}

```

```

_XrightPhoto = (_correctedMatchedPr.X - ((Width \ 2) - 0.5)) * 0.0125; \Tampahan
_YrightPhoto = (((Height \ 2) - 0.5) - _correctedMatchedPr.Y) * 0.0125; \Tampahan

_XleftPhoto = (_centerTemplate.X - ((Width \ 2) - 0.5)) * 0.0125; \Tampahan
_YleftPhoto = (((Hgt \ 2) - 0.5) - _centerTemplate.Y) * 0.0125; \Tampahan

Trace.WriteLine("Iteration 1: " + "\n" + solution.ToString());
Trace.WriteLine("CorrectedMatchedPr.X: " +
    _correctedMatchedPr.X.ToString() + " " +
    _correctedMatchedPr.Y.ToString());

Trace.WriteLine("Left Photo Coordinate X:Photo Y:Photo: " +
    _XleftPhoto.ToString() + " " +
    _YleftPhoto.ToString()); \Tampahan
Trace.WriteLine("Right Photo Coordinate X:Photo Y:Photo: " +
    _XrightPhoto.ToString() + " " +
    _YrightPhoto.ToString()); \Tampahan

AX = A * solution;
CSML.Solve(AxB = new CSML.Solve());
AxB = AX - _b;
CSML.Solve(VT = new CSML.Solve());
VT = AxB.Transpose();
CSML.Solve(VTV = new CSML.Solve());
VTV = VT * AxB;
rcol = A.ColumnCount;
rrow = A.RowCount;
n = rrow - rcol;
double tes = VTV[1].Re;
double VVCO = VTV[1].Re;

VVVCO = VTV[0];
double VVDO = VVCO.Re;
double 20;
So = Math.Sqrt(VVCO) \ (r);

Trace.WriteLine("AX: " + "\n" + AX.ToString());
Trace.WriteLine("V: " + "\n" + AxB.ToString());
Trace.WriteLine("VT: " + "\n" + VTV.ToString());
Trace.WriteLine("r: " + "\n" + r.ToString());
Trace.WriteLine("So: " + "\n" + So.ToString());

while (!_isConverge && Iteration <= 25)
{
    Compose_Params(eta, eta, eta);
    solution = CSML.Solve(eta, eta);
    RearrangeGridXY(solution);
    _correctedMatchedPr.X += solution[1].Re +
        solution[2].Re * _correctedMatchedPr.X +
        solution[3].Re * _correctedMatchedPr.Y;
}

```

```

_correctedMatchedPt.Y += solution[4, 1].Re +
    solution[5, 1].Re * _correctedMatchedPt.X +
    solution[6, 1].Re * _correctedMatchedPt.Y;
_lsmConverge = Threshold(solution, DELTA);
//_lsmConverge = Threshold2();
//if (!_lsmConverge)
//{
//    _previousMatched.X = _correctedMatchedPt.X;
//    _previousMatched.Y = _correctedMatchedPt.Y;
//}

_XRightPhoto = (_correctedMatchedPt.X - ((Wdh / 2) - 0.5)) *
    0.01219; //Tambahan
_YRightPhoto = (((Hgt / 2) - 0.5) - _correctedMatchedPt.Y) *
    0.01219; //Tambahan

_XLeftPhoto = (_centreTemplate.X - ((Wdh / 2) - 0.5)) *
    0.01219; //Tambahan
_YLeftPhoto = (((Hgt / 2) - 0.5) - _centreTemplate.Y) *
    0.01219; //Tambahan

Trace.WriteLine("Iteration: " + iteration.ToString() + "\n"
    + solution.ToString());
Trace.WriteLine("_correctedMatchedPt X Y: " +
    _correctedMatchedPt.X.ToString() + " " +
    _correctedMatchedPt.Y.ToString());

//Trace.WriteLine("Left Photo Coordinate XPhoto YPhoto: " +
    _XLeftPhoto.ToString() + " " +
    _YLeftPhoto.ToString());//tambahan
//Trace.WriteLine("Right Photo Coordinate XPhoto YPhoto: "
    + _XRightPhoto.ToString() + " " +
    _YRightPhoto.ToString());//tambahan

AX = A * solution;
CSML.Matrix AxB6 = new CSML.Matrix();
AxB = AX - _b;
CSML.Matrix VT6 = new CSML.Matrix();
VT = AxB.Transpose();
CSML.Matrix VtV6 = new CSML.Matrix();
VtV = VT * AxB;
rCol = A.ColumnCount;
rRow = A.RowCount;
r = rRow - rCol;
//double tes = VtV[1].Re;
double VtVCo6 = VtV[1].Re;

//VtVCo = VtV[0];
// double VtVDo = VtVCo.Re;
double So6;
So6 = Math.Sqrt(VtVCo6) / (r);

//Trace.WriteLine("AX: " + "\n" + AX.ToString());
//Trace.WriteLine("V: " + "\n" + AxB.ToString());

```



```

        //Trace.WriteLine("VT: " + "\n" + VtV.ToString());
        //Trace.WriteLine("r: " + "\n" + r.ToString());
        Trace.WriteLine("So: " + "\n" + So6.ToString());

        iteration++;
    }
}
return _lsmConverge;
}

private bool Threshold(CSML.Matrix m, double threshold)
{
    if (/*(Math.Abs(m[1, 1].Re) <= 0.5 || Math.Abs(m[4, 1].Re) <= 0.5)
        && */ //shift
        (Math.Abs(m[2, 1].Re) <= 0.03 || Math.Abs(m[6, 1].Re) <= 0.03)
        && //scale
        (Math.Abs(m[3, 1].Re) <= 0.03 || Math.Abs(m[5, 1].Re) <= 0.03))
        //shear
        return false;

    //if (Math.Abs(m[1, 1].Re) <= threshold && Math.Abs(m[2, 1].Re) <=
        threshold &&
    //  Math.Abs(m[3, 1].Re) <= threshold && Math.Abs(m[4, 1].Re) <=
        threshold &&
    //  Math.Abs(m[5, 1].Re) <= threshold && Math.Abs(m[6, 1].Re) <=
        threshold)
    //  return false;
    else
        return true;
}

private bool Threshold2()
{
    if (Math.Abs(_correctedMatchedPt.X - _previousMatched.X) <= 0.01 &&
        Math.Abs(_correctedMatchedPt.Y - _previousMatched.Y) <= 0.01)
        return false;
    else
        return true;
}
#endregion Public Methods
}
}

```



# **LAMPIRAN B**



LAMPYRAN B

## LAMPIRAN

### Proses Validasi algoritma *image matching* dengan menggunakan metode *intersection*

Data foto

Berikut data foto stereo

a) DSC\_0013L.jpg



b) DSC\_0013R.jpg



# LAPORAN

Proses 7 alihazi algoritma image matching dengan menggunakan metode

intersection

Data foto

Berikut data foto stereo

a) DSC\_0013L.jpg



b) DSC\_0013R.jpg



### Data Koordinat (x,y) Foto

Berikut disajikan tabel koordinat foto untuk perhitungan *intersection*:

ID	Koordinat Foto Kiri		Koordinat Foto Kanan	
	x	y	x	y
1	-4.98644	2.56208	-1.91644	5.85436
2	-1.07506	2.68607	0.96879	-0.15445
3	3.32488	2.46666	2.40116	-3.80725
4	-5.4971	1.14262	-3.59646	4.74637
5	-1.52819	1.39572	-0.27809	-0.72584
6	3.13043	1.34045	1.60292	-4.1128
7	-3.6248	-0.11628	-2.65047	0.72771
8	2.78935	-0.37067	0.54948	-4.30253
9	-4.34598	-2.01395	-4.27929	0.63995
10	2.24991	-3.53328	-1.08881	-4.4396

### Persamaan yang digunakan dalam proses *intersection*

Wolf & Dewitt, (2000) mengkaji bahwa *intersection* mengacu pada penentuan posisi titik-titik pada *object space* dengan perpotongan garis dari dua atau lebih foto. Persamaan kesegarisan (*collinearity equations*) merupakan metode dasar dalam penentuan koordinat titik pada *object space*. Persamaan kesegarisan dinyatakan sebagai berikut (Wolf & Dewitt, 2000).

$$x_a = x_0 - f \frac{m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \quad (1a)$$

$$y_a = y_0 - f \frac{m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)}{m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L)} \quad (1b)$$

L<sub>a</sub> dan A berada pada satu garis lurus, dimana:

$x_a, y_a$  : koordinat foto titik a

$X_A, Y_A, Z_A$  : koordinat *object space* titik A

$X_L, Y_L, Z_L$  : koordinat *object space* stasiun pemotretan

Data Koordinat (x,y) Foto

Berikut disajikan tabel koordinat foto untuk perhitungan inverse geom...

ID	Koordinat Foto Kiri		Koordinat Foto Kanan	
	x	y	x	y
1	-4.08214	3.30208	-1.01041	2.82430
2	-1.07206	2.08007	0.09879	-0.12142
3	3.32488	2.46606	2.40116	-3.80722
4	-2.4071	1.14202	-2.86642	4.74627
5	-1.32810	1.02373	-0.17209	-0.72284
6	3.13013	1.24012	1.00202	-4.1128
7	-3.0248	-0.11028	-2.09047	0.72771
8	2.78922	-0.27067	0.21912	-4.30223
9	-4.21208	-3.01392	-4.22020	0.03992
10	2.24901	-2.22228	-1.08881	-1.4320

Persamaan yang digunakan dalam proses intersection

(Wolf & Dewar (2000) mengkaji bahwa intersection menjadi bagian penentuan posisi titik-titik pada objek space dengan perpotongan garis dari dua atau lebih foto. Persamaan kesamaan (collinearity equations) merupakan metode dasar dalam penentuan koordinat titik pada objek space. Persamaan kesamaan digunakan sebagai berikut (Wolf & Dewar, 2000).

$$(1a) \quad x_0 - x_0 = \frac{m_{11}(X_A - X_0) + m_{12}(Y_A - Y_0) + m_{13}(Z_A - Z_0) - (X_0 - X_0)}{m_{21}(X_A - X_0) + m_{22}(Y_A - Y_0) + m_{23}(Z_A - Z_0)}$$

$$(1b) \quad y_0 - y_0 = \frac{m_{21}(X_A - X_0) + m_{22}(Y_A - Y_0) + m_{23}(Z_A - Z_0) - (Y_0 - Y_0)}{m_{11}(X_A - X_0) + m_{12}(Y_A - Y_0) + m_{13}(Z_A - Z_0)}$$

dan A adalah pada satu garis lurus dimana:

$Z_0, Y_0$  : koordinat foto titik a

$X_A, Y_A, Z_A$  : koordinat objek space titik A

$X_0, Y_0, Z_0$  : koordinat objek space stasiun pemotretan

- $f$  : panjang fokus kamera  
 $x_o, y_o$  : koordinat *principle point* (dapat diketahui dari kalibrasi kamera)  
 $m$  : fungsi dari 3 sudut rotasi, terdiri dari  $\omega, \varphi, \kappa$ .

Persamaan diatas menunjukkan bahwa posisi sebuah titik di permukaan bumi, di foto dan pusat proyeksi kamera terletak pada satu garis lurus. Persamaan kesejarisan tersebut mengandung 9 parameter yang belum diketahui yaitu  $\omega, \varphi, \kappa$ , (omega, phi, kappa) yang merupakan parameter rotasi kamera, dan tiga parameter posisi kamera yakni  $X_L, Y_L, Z_L$  yang terdapat dalam Orientasi Luar (*Exterior Orientation*). Sedangkan tiga parameter lainnya merupakan koordinat titik objek ( $X_A, Y_A, Z_A$ ). Dimana  $f$  merupakan panjang fokus kamera. Ketiga parameter ini merupakan parameter dari Orientasi Dalam (*Interior Orientation*).

Dengan menggunakan tiga sudut rotasi, hubungan antara sistem koordinat foto ( $x, y, z$ ) dan sistem koordiant objek ( $X, Y, Z$ ) dapat ditentukan. Sebuah matriks dengan dimensi 3 x 3 dapat mendefinisikan hubungan antara dua sistem yang digunakan.

Adapaun matriks rotasi dapat didefinisikan sebagai berikut:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (2)$$

Dari persamaaan diatas, nilai masing-masing element matriks M dapat diperoleh dengan menggunakan persamaan berikut ini:

$$m_{11} = \cos \varphi \cos \kappa$$

$$m_{12} = \sin \omega \cos \kappa + \cos \omega \sin \kappa$$

- $\lambda$  : panjang fokus kamera
- $Z_c$  : koordinat vertikal pada (dapat diketahui dari kalibrasi kamera)
- $w$  : fungsi dari  $\gamma$  sudut rotasi terdapat  $\phi, \kappa$

Persamaan diatas menunjukkan bahwa posisi sebuah titik di permukaan bumi di foto dan pusat proyeksi kamera terletak pada satu garis lurus. Persamaan kesamaan tersebut mengandung 3 parameter yang belum diketahui yaitu  $\phi, \kappa, \omega$  (omega, phi, kappa) yang merupakan parameter rotasi kamera, dan tiga parameter posisi kamera yakni  $X_c, Y_c, Z_c$  yang terdapat dalam Orientasi Luar (Exterior Orientation). Sedangkan tiga parameter lainnya merupakan koordinat titik objek  $(X_o, Y_o, Z_o)$ . Dimana  $\lambda$  merupakan panjang fokus kamera. Ketiga parameter ini merupakan parameter dari Orientasi Dalam (Interior Orientation).

Dengan menggunakan tiga sudut rotasi hubungan antara sistem koordinat foto  $(x, y)$  dan sistem koordinat objek  $(X, Y, Z)$  dapat diturunkan. Sebuah matriks dengan dimensi  $3 \times 3$  dapat mendefinisikan hubungan antara dua sistem yang digunakan. Adapun matriks rotasi dapat didefinisikan sebagai berikut:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (2)$$

Dari persamaan diatas nilai masing-masing element matriks M dapat diperoleh dengan menggunakan persamaan berikut ini:

$$m_{11} = \cos \phi \cos \kappa$$

$$m_{12} = \sin \phi \cos \kappa + \cos \phi \sin \kappa$$

$$\begin{aligned}
m_{13} &= -\cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa \\
m_{21} &= -\cos \varphi \sin \kappa \\
m_{22} &= -\sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa \\
m_{23} &= \cos \omega \sin \varphi \sin \kappa + \sin \omega \cos \kappa \\
m_{31} &= \sin \varphi \\
m_{32} &= -\sin \omega \cos \varphi \\
m_{33} &= \cos \omega \cos \varphi
\end{aligned} \tag{3}$$

Sedangkan untuk menghitung nilai  $q, r, s$  maka digunakan persamaan dibawah ini.

$$\begin{aligned}
q &= m_{31}(X_A - X_L) + m_{32}(Y_A - Y_L) + m_{33}(Z_A - Z_L) \\
r &= m_{11}(X_A - X_L) + m_{12}(Y_A - Y_L) + m_{13}(Z_A - Z_L) \\
s &= m_{21}(X_A - X_L) + m_{22}(Y_A - Y_L) + m_{23}(Z_A - Z_L)
\end{aligned} \tag{4}$$

Dari persamaan diatas ( $X_A, Y_A$ ) adalah koordinat foto target dan  $Z_A$  adalah fokus kamera ( $c$ ). Kemudian matriks A dapat dihitung dengan menggunakan persamaan berikut

$$\begin{aligned}
b_{14} &= \frac{f}{q^2}(rm_{31} - qm_{11}) & b_{24} &= \frac{f}{q^2}(sm_{31} - qm_{21}) \\
b_{15} &= \frac{f}{q^2}(rm_{32} - qm_{12}) & b_{25} &= \frac{f}{q^2}(sm_{32} - qm_{22}) \\
b_{16} &= \frac{f}{q^2}(rm_{33} - qm_{13}) & b_{24} &= \frac{f}{q^2}(sm_{33} - qm_{23})
\end{aligned} \tag{5}$$

Matriks L merupakan matriks yang berisi nilai fungsi dimana nilai koordiant foto dikurangi nilai pendekatan koordinat object space. Matriks L dapat dihitung menggunakan persamaan berikut :



menyusun persamaan berikut :

diketahui nilai bendakan koordinat object space. Matriks  $L$  dapat dihitung Matriks  $L$  merupakan matriks yang berisi nilai fungsi dimana nilai koordinat foto

$$\begin{aligned}
 p_{1e} &= \frac{f}{d^2} (m_{11}x - d_{m11}) & p_{1z} &= \frac{f}{d^2} (m_{13}x - d_{m13}) \\
 p_{2e} &= \frac{f}{d^2} (m_{21}x - d_{m21}) & p_{2z} &= \frac{f}{d^2} (m_{23}x - d_{m23}) \\
 p_{3e} &= \frac{f}{d^2} (m_{31}x - d_{m31}) & p_{3z} &= \frac{f}{d^2} (m_{33}x - d_{m33})
 \end{aligned}$$

berikut

kamera (c). Kemudian matriks  $A$  dapat dihitung dengan menggunakan persamaan

Dari persamaan diatas ( $X, Y, Z$ ) adalah koordinat foto target dan  $X', Y', Z'$  adalah fokus

$$\begin{aligned}
 z &= m_{31}(X - X_L) + m_{32}(Y - Y_L) + m_{33}(Z - Z_L) \\
 y &= m_{11}(X - X_L) + m_{12}(Y - Y_L) + m_{13}(Z - Z_L) \\
 x &= m_{21}(X - X_L) + m_{22}(Y - Y_L) + m_{23}(Z - Z_L)
 \end{aligned}$$

Sebagian untuk menghitung nilai  $p, r, s$  maka digunakan persamaan dibawah ini.

$$\begin{aligned}
 m_{33} &= \cos \omega \cos \phi \\
 m_{32} &= -\sin \omega \cos \phi \\
 m_{31} &= \sin \phi
 \end{aligned}$$

$$m_{23} = \cos \omega \sin \phi \sin \kappa + \sin \omega \cos \kappa$$

$$m_{22} = -\sin \omega \sin \phi \sin \kappa + \cos \omega \cos \kappa$$

$$m_{21} = -\cos \phi \sin \kappa$$

$$m_{13} = -\cos \omega \sin \phi \cos \kappa + \sin \omega \sin \kappa$$

(3)

$$J = x_a - x_0 + f \frac{r}{q} \quad (5a)$$

$$K = y_a - y_0 + f \frac{s}{q} \quad (5b)$$

Sedangkan koreksi diperoleh dengan menghitung matriks X dengan menggunakan persamaan dibawah ini.

$$X = (A^T A)^{-1}(A^T L) \quad (6)$$

Untuk memperoleh nilai residu, maka matriks V dapat dihitung menggunakan persamaan berikut :

$$V = AX - L \quad (7)$$

Standart Deviasi ( $S_0$ ) dapat dihitung menggunakan persamaan berikut :

$$S_0 = \sqrt{\frac{V^T V}{r}} L \quad (8)$$

**Uji validasi metode *image matching* dengan menggunakan teknik *intersection*.**

Berikut akan dilakukan uji validasi metode *area based matching* dengan teknik *intersection*, dimana parameter-parameter dari teknik *least square matching* akan dihitung standar deviasinya.

$$F = 18.5558$$

$$X_0 = 0.00$$

$$Y_0 = 0.00$$

$$F_w = 1936$$

$$F_h = 1296$$

$$j = x_0 - x_0 + j \frac{1}{p}$$

(2a)

$$k = x_0 - x_0 + j \frac{2}{p}$$

(2b)

Sebanyak koreksi diperoleh dengan menghitung matriks  $Z$  dengan menggunakan persamaan dibawah ini.

$$X = (A^T A)^{-1} (A^T L)$$

(c)

Untuk memperoleh nilai residu, maka matriks  $V$  dapat dihitung menggunakan persamaan berikut :

$$V = AX - L$$

(7)

Standart Deviasi (2), dapat dihitung menggunakan persamaan berikut :

$$s_0 = \sqrt{\frac{V^T V}{r}}$$

(8)

Uji validasi metode image matching dengan menggunakan teknik intersection. Berikut akan dilakukan uji validasi metode area based matching dengan teknik intersection, dimana parameter-parameter dari teknik fast square matching akan dihitung standar deviasinya.

- F = 18.2238
- X<sub>0</sub> = 0.00
- Y<sub>0</sub> = 0.00
- F<sub>w</sub> = 1936
- F<sub>H</sub> = 1296

Ukuran CCD = 23.6 x 15.8

Tabel parameter orientasi luar tiap foto

Photo	$X_L$	$Y_L$	$Z_L$	$\omega$	$\phi$	$\kappa$
Left	-0.342	-1.8718	-0.0908	0.3346	0.6424	-0.0338
Right	-2.9792	0.1888	0.2852	-0.5369	-0.3954	1.0051

Orientasi luar terdiri dari 3 parameter posisi kamera ( $X_L, Y_L, Z_L$ ) dan 3 parameter rotasi yaitu omega, phi, kappa ( $\omega, \phi, \kappa$ ).

$$M_{\text{Left}} \begin{pmatrix} 0.99994 & -0.00052 & -0.01122 \\ 0.00059 & 0.99998 & 0.00583 \\ 0.01121 & -0.00584 & 0.99992 \end{pmatrix}$$

$$M_{\text{Right}} \begin{pmatrix} 0.99998 & 0.00480 & -0.00355 \\ -0.00498 & 0.99864 & -0.05196 \\ 0.00330 & 0.05197 & 0.99864 \end{pmatrix}$$

Setelah mendapatkan nilai m, selanjutnya dilakukan perhitungan untuk menentukan nilai q, r, dan s. Nilai q, r, s ini digunakan untuk menghitung nilai yang akan digunakan dalam menyusun matriks A. Perhitungan nilai q, r, dan s menggunakan Persamaan (4). Berikut adalah nilai q, r, s dari sepasang titik menggunakan dua buah foto:

ID	r_left	q_left	s_left	ID	r_right	q_right	s_right
1	0.89332	-4.41324	-0.60800	1	1.04802	-5.61846	-1.77453
2	-0.47900	-18.21375	0.62181	2	-0.42299	-19.31349	0.258177

Ukuran CCD = 53.0 x 17.8

Tabel parameter orientasi luar tiap foto

Photo	$Z_i$	$X_i$	$Y_i$	$\omega$	$\phi$	$K$
Left	-0.342	-1.8718	-0.0608	0.3316	0.0454	-0.0338
Right	-0.9792	0.1888	-0.3822	-0.2300	-0.3924	1.0021

Orientasi luar terdiri dari 3 parameter posisi kamera ( $Z_i$ ,  $X_i$ ,  $Y_i$ ) dan 3 parameter rotasi

yaitu omega ( $\omega$ ,  $\phi$ ,  $K$ )

$M_{Left}$	$Z_i$	$X_i$	$Y_i$
0.01121	-0.00284	0.00004	-0.01121
0.00029	0.00098	0.00029	0.00283
0.00001	-0.00022	0.00001	-0.01121

$M_{Right}$	$Z_i$	$X_i$	$Y_i$
0.00330	0.0197	0.00028	0.00330
-0.00492	0.00284	-0.00492	-0.02190
0.00222	-0.00420	0.00222	-0.00322

Persamaan (4) berikut adalah nilai  $d$  dan  $s$  dari sebarang titik menggunakan dua buah digunakan dalam mensuam marka A. Perhitungan nilai  $d$  dan  $s$  menggunakan nilai  $d$  dan  $s$  ini digunakan untuk menghitung nilai yang akan nilai  $d$  dan  $s$ . Nilai  $d$  dan  $s$  ini digunakan untuk menghitung nilai yang akan setelah mendapatkan nilai ini selanjutnya dilakukan perhitungan untuk menentukan

foto:

ID	$r_{left}$	$p_{left}$	$z_{left}$	ID	$r_{right}$	$p_{right}$	$z_{right}$
1	0.80332	-1.41324	-0.60800	1	1.04802	-2.01848	-1.77423
2	-0.47000	-1.82122	0.62181	2	-0.42500	-1.921349	0.228122

Selanjutnya untuk menyusun matriks A (matriks koefisien) menggunakan Persamaan

(5).

$$A = \begin{pmatrix} -4.2138 & 0.0072 & -0.8039 & 0 & 0 \\ 0.0040 & -4.2079 & 0.5547 & 0 & 0 \\ -3.3046 & -0.0479 & -0.6035 & 0 & 0 \\ 0.0199 & -3.2439 & 1.2133 & 0 & 0 \\ 0 & 0 & 0 & -1.0184 & 0.0004 \\ 0 & 0 & 0 & -0.0010 & -1.0186 \\ 0 & 0 & 0 & -0.9607 & -0.0035 \\ 0 & 0 & 0 & 0.0047 & -0.9601 \end{pmatrix}$$

$$A^T = \begin{pmatrix} -4.2138 & 0.0040 & -3.3046 & 0.0199 & 0 & 0 & 0 & 0 \\ 0.0072 & -4.2079 & -0.0479 & -3.2439 & 0 & 0 & 0 & 0 \\ -0.8039 & 0.5547 & -0.6035 & 1.2133 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.0184 & -0.0010 & -0.9607 & 0.0047 \\ 0 & 0 & 0 & 0 & 0.0004 & -1.0186 & -0.0035 & -0.9601 \\ 0 & 0 & 0 & 0 & 0.0382 & -0.0407 & 0.0244 & 0.0371 \end{pmatrix}$$

$$A^T * A = \begin{pmatrix} 28.6775 & 0.0466 & 5.4080 & 0 & 0 & 0 \\ 0.0466 & 28.2317 & -6.2467 & 0 & 0 & 0 \\ 5.4080 & -6.2467 & 2.7901 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.9601 & -0.0005 & -0.0622 \\ 0 & 0 & 0 & -0.0005 & 1.9593 & 0.0058 \\ 0 & 0 & 0 & -0.0622 & 0.0058 & 0.0051 \end{pmatrix}$$

$$(A^T * A)^{-1} = \begin{pmatrix} 0.1278 & -0.1090 & -0.4918 & 0 & 0 & 0 \\ -0.1090 & 0.1632 & 0.5767 & 0 & 0 & 0 \\ -0.4918 & 0.5767 & 2.6028 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8344 & -0.0300 & 10.2233 \\ 0 & 0 & 0 & -0.0300 & 0.5132 & -0.9500 \\ 0 & 0 & 0 & 10.2233 & -0.9500 & 322.3430 \end{pmatrix}$$

Sedangkan matriks L merupakan matriks yang berisikan nilai fungsi dimana nilai koordinat dikurangi nilai pendekatan koordinat *object space*. Penyusunan matriks L diawali dengan menghitung nilai J dan K, dimana formula yang digunakan terdapat dalam Persamaan (5a) dan Persamaan (5b).

$$L \begin{pmatrix} -1.23042 \\ 0.005691 \\ 1.544815 \\ -0.00629 \\ -1.56306 \\ 3.319557 \\ 0.56239 \\ 0.093598 \end{pmatrix}$$

Selanjutnya Matriks X dihitung menggunakan Persamaan (6), dan hasilnya adalah sebagai berikut:

$$X \begin{pmatrix} -0.00614 \\ 0.00740 \\ 0.04723 \\ -0.83747 \\ -1.64522 \end{pmatrix}$$

$$A * X \begin{pmatrix} -0.01204 \\ -0.00498 \\ -0.00857 \\ 0.033166 \\ -0.80095 \\ 3.438152 \\ -0.24624 \\ -0.02895 \end{pmatrix}$$

$$V \begin{pmatrix} 1.218381 \\ -0.010667 \\ -1.553381 \\ 0.039454 \\ 0.762102 \\ 0.118594 \\ -0.808628 \\ -0.122549 \end{pmatrix}$$

$$So \quad \parallel \quad 1.51554975 \text{ mm} \quad \parallel$$

Koordinat *Object Space* pendekatannya diperoleh dari penjumlahan transformasi koordinat 3D dengan delta X, delta Y dan delta Z. Hasilnya adalah sebagai berikut:

<b>ID</b>	<b>delta_X</b>	<b>delta_Y</b>	<b>delta_Z</b>
1	-0.00613994	0.007401998	0.047226725
2	-0.83746999	-1.645	-43.260

Dari hasil diatas maka dapat dihitung adjustment value dari masing-masing koordinat dan hasilnya adalah sebagai berikut:

<b>ID</b>	<b>X</b>	<b>Y</b>	<b>Z</b>
1	0.495283337	-2.44708643	-4.47002906
2	-1.86228309	-2.7886048	-61.5537661



ID	delta X	delta Y	delta Z
1	-0.0013001	0.007101908	0.047339732
2	-0.83246009	-1.012	-13.300

Dari hasil diatas maka dapat dihitung adjustment value dari masing-masing koordinat

dan hasilnya adalah sebagai berikut:

ID	Z	Y	X
1	0.402382331	-5.44208943	-473005009
2	-1.84358300	-7.7880048	-01.2237001