

PEMBUATAN FOTO STEREO NORMAL MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON

Martinus Edwin Tjahjadi¹, Ahmad Gazali²

Dosen Teknik Geodesi Institut Teknologi Nasional Malang¹
Mahasiswa Teknik Geodesi Institut Teknologi Nasional Malang²
Malang, Indonesia
E-mail: gazaliahmad928@gmail.com

ABSTRAK

Bahasa pemrograman python merupakan bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah intruksi multiguna secara langsung (*interpretatif*) dengan metode orientasi objek. Bahasa pemrograman python ini telah banyak dipakai salah satunya untuk melakukan *image processing*, dengan semakin canggihnya teknologi bahasa pemrograman python ini pun semakin dikembangkan, sehingga dapat digunakan untuk melakukan *image processing*. Data yang digunakan dalam pembuatan foto stereo normal yaitu, dua buah pasang foto stereo, di mana masing-masing foto tersebut telah di tentukan parameter *interior* orientasi dan parameter *eksterior* orientasinya. *Normalized stereo pair* adalah salah satu teknik pengolahan gambar dalam fotogrametri digital, komplikasi foto yang dinormalisasikan dengan syarat bahwa parameter *interior* dan *eksterior* orientasinya diketahui pada sepasang gambar stereo. Sehingga dari parameter orientasi tersebut maka dilakukan tahap-tahap transformasi untuk mendapatkan hasil dari foto asli ke foto normal. Di dalam perhitungan tersebut terdapat parameter eksterior orientation yang berfungsi untuk mentransformasikan gambar asli ke posisi normal dengan mencari nilai d_{max} sebagai refrensi untuk mendapatkan resolusi dan pixel size baru dari gambar yang dinormalisasi. Kemudian untuk mengisi nilai pixel baru dari gambar normal, maka digunakan metode *bilinier interpolation* dan *persepective warping* untuk melakukan *resampling*.

Kata Kunci : exterior orientation, fotogrametri, Normalized image, python, resampling, transformasi

ABSTRACT

Python programming language is a high programming language that can execute a number of multipurpose instructions directly (interpretatively) with the object orientation method. Python programming language has been widely used one of them to do image processing, with the increasingly sophisticated python programming language technology is increasingly being developed, so it can be used to do image processing. The data used in making normal stereo photos are, two pairs of stereo photos, in which each photo has been determined interior orientation parameters and exterior orientation parameters. Normalized stereo pair is one of the image processing techniques in digital photogrammetry, photo complications are normalized on condition that the interior and exterior orientation parameters are known in a stereo pair. So from the orientation parameters, the transformation stages are carried out to get the results from the original photo to a normal photo. In the calculation there are exterior orientation parameters that function to transform the original image to its normal position by finding the d_{max} value as a reference to get the new resolution and pixel size of the normalized image. Then to fill the new pixel value from a normal image, then the binary interpolation and perspective warping methods are used to do resampling.

Keywords: exterior orientation, photogrammetry, Normalized image, python, resampling, transformation

PENDAHULUAN

Maraknya penggunaan komputer .di berbagai kehidupan manusia menuntut setiap orang untuk mengetahui dan mempelajari berbagai macam software pendukung yang dapat beroperasi pada komputer tersebut. Bahasa pemrograman atau sering di istilahkan juga dengan bahasa komputer adalah teknik komando/intruksi untuk memerintahkan komputer. Bahasa pemrograman python adalah bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah intruksi multiguna secara langsung dengan metode orientasi objek (Rahayu, 2009).

Python adalah bahasa pemrograman model skrip yang berorientasi objek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Dengan semakin canggihnya teknologi bahasa pemrograman python ini pun semakin di kembangkan, sehingga dapat dignakan untuk melakukan *image processing* (Triasanti, 2010)

Sehubungan dengan hal diatas, maka akan dibuat sebuah *coding* bahasa pemrograman menggunakan python dengan beberapa library pendukung untuk melakukan proses foto stereo

normal menggunakan metode *stereo image pair* dalam fotogrametri digital.

Pada pembuatan foto stereo normal, diperlukan sepasang foto stereo dengan parameter orientasi luarnya. Parameter tersebut di gunakan untuk melakukan transformasi terhadap foto asli ke foto stereo yang di normalisasi.

Artikel ini mencoba membahas tentang pembuatan foto stereo normal menggunakan metode *stereo image pair*. Pembuatan algoritma *stereo image pair* ini yaitu dengan menggunakan bahasa pemrograman python. Hasil dari foto stereo normal ini juga dapat digunakan untuk melakukan image matching.

Rumusan masalah dalam penelitian ini antara lain :

1. Bagaimana pembuatan algoritma *stereo image pair* dalam bahasa pemrograman python untuk mendapatkan hasil foto stereo normal ?
2. Bagaimana hasil *resampling* dari koordinat titik pojok pada foto asli ke foto yang dinormalisasi ?

Sementara tujuan dilakukannya penelitian ini antara lain :

1. Pembuatan algoritma *stereo image pair* menggunakan bahasa pemrograman python untuk foto stereo normal.
2. Untuk mendapatkan resolusi baru foto stereo normal dari hasil *resampling*.

METODE

Lokasi Penelitian

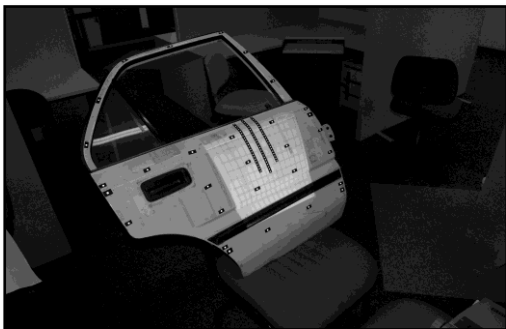
Lokasi penelitian dilaksanakan di laboratorium fotogrametri Institut Teknologi Nasional Malang, Jawa Timur.

Alat dan Bahan Penelitian

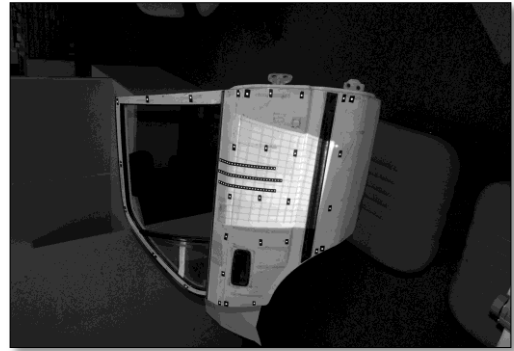
Bahan dan peralatan penelitian yang digunakan dalam pelaksanaan penelitian ini terdiri dari:

a. Foto Stereo

Adapun data dari foto stereo sebagai berikut :



Gambar 1. Original foto kiri



Gambar 2. Original foto kanan

- b. Parameter exterior orientasi foto stereo
 Adapun data parameter foto stereo sebagai berikut:

Station	Value	Standard Error
X	-208.7169	2.3875E-001
Y	-221.7774	1.8877E-001
Z	1251.5457	2.4117E-001
Omega	10.4875	
Phi	-28.8499	
Kappa	-24.1934	

Gambar 3. Parameter eksterior orientasi foto kiri

Station	Value	Standard Error
X	1049.7744	3.0665E-001
Y	-185.0397	2.5409E-001
Z	1325.4042	1.9967E-001
Omega	10.8649	
Phi	20.9323	
Kappa	-83.0798	

Gambar 4. Parameter eksterior orientasi foto kiri

Peralatan penelitian yang akan digunakan terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan antara lain:

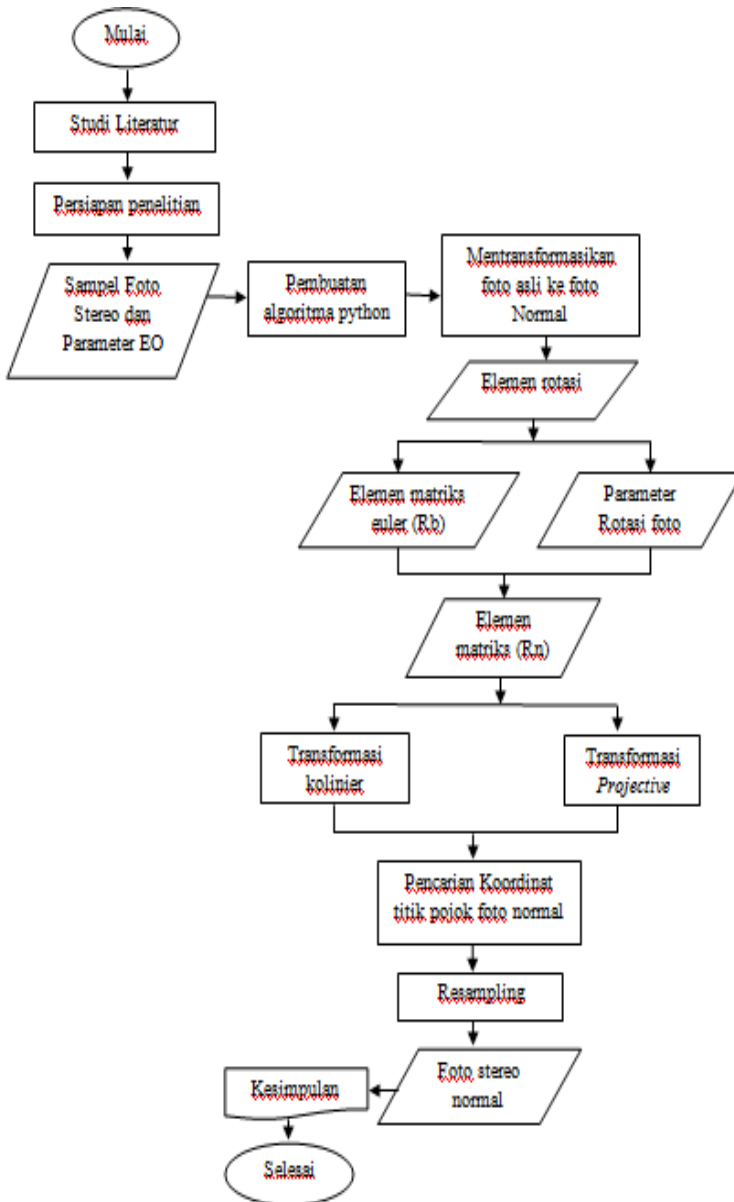
1. Laptop untuk melakukan *image processing* dan penulisan laporan.
2. Printer untuk mencetak hasil pengerjaan.
3. Kamera Nikon D200 untuk pengambilan gambar.

Sedangkan perangkat lunak (*software*) yang akan digunakan adalah:

1. *Microsoft office word 2007*
2. *Notepad*
3. *Anaconda* dengan *Python* versi 3.5.6 dimana dilengkapi *software* pendukung seperti:
 - a. *opencv version 3.1.0*
 - b. *Numpy version 1.15.2*
 - c. *Matplotlib version 3.0.0-py3.5.*

Tahapan Penelitian

Adapun tahapan-tahapan penting dalam penelitian dapat dilihat pada gambar diagram alir berikut ini:



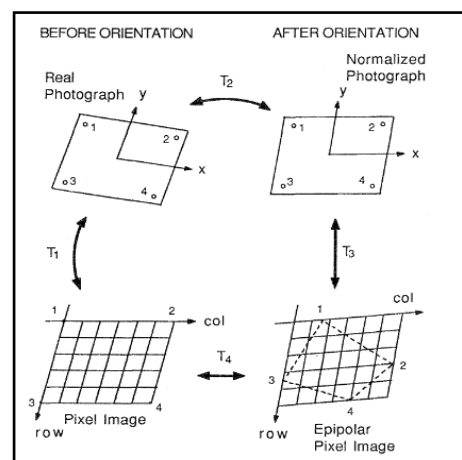
Gambar 5. Diagram alir penelitian

Foto Stereo Normal

Pasangan stereo yang dinormalisasi ditandai oleh kedua gambar yang dinormalisasi memiliki paralaks nol dalam arah sumbu y. Oleh karena itu konjugasikan lokasi piksel pada gambar yang dinormalisasi memiliki nilai baris yang sama (nilai koordinat y yang sama). Kondisi akan ini terpenuhi ketika dua kamera memancarkan gambar stereo pasangan sejajar satu sama lain dan tegak lurus ke dasar kamera. Oleh karena itu, titik konjugasi memiliki nol-paralaks dan pencarian entitas yang cocok terbatas pada piksel arah x saja.

Prosedur yang dibahas pada bagian sebelumnya menetapkan transformasi antara gambar piksel dan gambar normal. Parameter distorsi ditentukan selama kalibrasi kamera. Saat resampling nilai abu-abu untuk gambar yang dinormalisasi, juga menerapkan koreksi. Dengan demikian, perhitungan gambar yang dinormalisasi berlangsung dalam empat langkah

1. Transformasi antara gambar piksel dan foto asli (*diapositive*). Parameter transformasi ditentukan selama kalibrasi kamera. Referensi umum untuk ini parameter transformasi adalah tanda, dan fitur ground yang berbeda.
2. T2: Transformasi proyektif antara foto asli dan foto normal. Definisi sistem koordinat untuk gambar piksel dalam geometri epipolar (gambar yang dinormalisasi).
3. T3: Untuk meminimalkan penurunan resolusi (atau untuk mengoptimalkan ukuran), pertama empat sudut gambar piksel ((O, O), (O, N), (N, O), (N, N)) diubah menjadi foto nyata dan kemudian ke koordinat foto yang dinormalisasi melalui T1, T2. Prosedur berikut mendefinisikan sistem koordinat gambar yang dinormalisasi.
 - a. Tentukan maksimum y koordinat titik sudut di kedua gambar. Ini mendefinisikan baris 0 pada kedua gambar yang dinormalisasi.
 - b. Tentukan perbedaan titik sudut x dan y di kedua foto dan hitung jarak maksimum d_{max} , baik dalam arah x atau y (kedua foto). Ini menentukan ukuran gambar piksel epipolar dalam koordinat foto.
 - c. Ubah dari koordinat foto ke koordinat piksel dengan menggunakan hubungan $d_{max} = \text{resolusi gambar pixel}$.
4. T4: Transformasi dari gambar yang dinormalisasi ke gambar piksel untuk melakukan *resampling*. Ini dilakukan dengan menggunakan T3, T2 dan T1.



Gambar 6. Hubungan antara foto dan gambar *pixel*, (Cho dan Schenk, 1999)

Transformasi dari Foto Asli ke Foto Normal

Gambar diubah dari vertical yang sebenarnya ke posisi normal. Ini melibatkan memutar gambar vertikal yang benar tentang pangkalan udara, dilambangkan dengan R_b . Diperlukan dua sudut untuk menentukan arah spasial alas di ruang objek. Dua sudut adalah ϕ dan κ , Mereka dapat ditentukan dari komponen dasar [BX, BY, BZ] *transpose*. Pertama, yaitu memutar gambar vertikal yang benar tentang sumbu Y oleh ϕ diikuti oleh rotasi tentang sumbu Z (κ) (Cho dan Schenk, 1999).

$$BX = X''c - X'c \dots\dots\dots [1]$$

$$BY = Y''c - Y'c \dots\dots\dots [2]$$

$$BZ = Z''c - Z'c \dots\dots\dots [3]$$

$$\phi = \arctan\left(\frac{BZ}{BX}\right) \dots\dots\dots [4]$$

$$\kappa = \arctan\left(\frac{BY}{(BX^2 + BZ^2)^{1/2}}\right) \dots\dots\dots [5]$$

Pada titik ini kita masih dapat memutar gambar yang dinormalisasi tentang basis untuk memberikan solusi yang unik, rotasi ini harus diperbaiki. Pilihan yang masuk akal adalah mendefinisikan rotasi anegele sebagai berikut (Cho dan Schenk, 1999).

$$\Omega = \frac{\omega' + \omega''}{2} \dots\dots\dots [6]$$

Keterangan :

Xc', Yc', Zc' : koordinat X, Y, Z pada foto kiri.

Xc'', Yc'', Zc'' : koordinat X, Y, Z pada foto kanan.

Ω, ϕ, κ : sudut rotasi terhadap sumbu X, Y, dan Z.

Dengan $\omega' + \omega''$ sudut rotasi tentang sumbu-x dari orientasi eksterior, setiap kali kita memutar gambar, area bersih menjadi lebih besar jika kita lebih tertarik untuk mempertahankan ukuran piksel, maka resolusi yang lebih tinggi diperlukan, pilihan Ω meminimalkan peningkatan ukuran gambar dinormalisasi. mari kita gabungkan tiga rotasi dasar ke dalam rotasi dasar R_b (Cho dan Schenk, 1999).

$$R_b = R_\phi R_\kappa R_\Omega \dots\dots\dots [7]$$

$$R_\Omega = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos(\Omega) & \sin(\Omega) \\ 0 & -\sin(\Omega) & \cos(\Omega) \end{vmatrix} \dots\dots\dots [8]$$

$$R_\phi = \begin{vmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\Omega) \end{vmatrix} \dots\dots\dots [9]$$

$$R_\kappa = \begin{vmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{vmatrix} \dots\dots\dots [10]$$

Matriks rotasi dinormalisasi R_n adalah produk dari dua matriks, matriks rotasi dari gambar ke vertikal benar, dan matriks rotasi dasar. menggabungkan dua rotasi dan memperoleh :

$$R_n = R_b R^T \dots\dots\dots [11]$$

Keterangan :

R_Ω : matriks rotasi terhadap sumbu X.

R_ϕ : matriks rotasi terhadap sumbu Y.

R_κ : matriks rotasi terhadap sumbu Z.

R : elemen matriks rotasi x, y z.

Karena R transpose merefleksikan matriks rotasi orientasi luar, itu berbeda untuk setiap gambar dan R_n harus ditentukan untuk kedua gambar *stereopair*. Disini akan menggunakan transformasi *projective* atau transformasi menggunakan persamaan kolinier untuk mentransformasi dari gambar asli ke gambar normal.

Transformasi menggunakan kondisi kolinier

Hubungan antara gambar dapat dengan mudah diungkapkan oleh persamaan *collinearity*, transformasi diwakili oleh persamaan berikut (Cho dan Schenk, 1999).

$$X_n = -fn \frac{r_{11}x_o + r_{12}y_o - r_{13}f_o}{r_{31}x_o + r_{32}y_o - r_{31}f_o} \dots\dots\dots [12]$$

$$Y_n = -fn \frac{r_{21}x_o + r_{22}y_o - r_{23}f_o}{r_{31}x_o + r_{32}y_o - r_{31}f_o} \dots\dots\dots [13]$$

Keterangan :

X_n, Y_n : koordinat titik pada foto pada titik pojok *normalized*.

x_o, y_o : koordinat foto pada *original*

f_n, f_o : panjang fokus kamera.

r_{11}, \dots, r_{33} : elemen matriks rotasi

di mana x_o dan y_o adalah koordinat foto dari gambar asli, x_n, y_n dari gambar yang dinormalisasi, dan r_{11}, \dots, r_{33} adalah elemen dari R_n .

Projective Trasformation

Transformasi proyektif dapat diterapkan karena baik gambar asli dan gambar normal adalah planar

$$X_n = \frac{c_{11}x_o + c_{12}y_o - c_{13}}{c_{31}x_o + c_{32}y_o + 1} \dots\dots\dots [14]$$

$$y_n = \frac{c_{21}x_o + c_{22}y_o + c_{23}}{c_{31}x_o + c_{32}y_o + 1} \dots\dots\dots [15]$$

Dengan membandingkan koefisien dalam transformasi proyektif dengan yang ada di persamaan kolinier, di temukan identitas berikut:

$$c_{11} = \frac{fn r_{11}}{fo r_{33}} \quad c_{21} = \frac{fn r_{21}}{fo r_{33}}$$

$$c_{12} = \frac{fn r_{12}}{fo r_{33}} \quad c_{22} = \frac{fn r_{22}}{fo r_{33}}$$

$$c_{13} = -\frac{fn r_{13}}{r_{33}} \quad c_{31} = -\frac{r_{31}}{fo r_{33}}$$

$$c_{23} = -\frac{fn r_{23}}{r_{33}} \quad c_{32} = -\frac{fn r_{32}}{fo r_{33}}$$

Keterangan :

c_{11}, \dots, c_{32} : komponen transformasi proyeksi.

f_n, f_o : panjang fokus kamera.

r_{11}, \dots, r_{33} : elemen matriks rotasi.

Ketika melakukan transformasi dari gambar asli ke gambar yang dinormalisasi, proses kuadrat gambar asli akan menjadi terdistorsi. biasanya lebih mudah untuk memproyeksikan *tesselation* gambar yang dinormalisasi kembali ke gambar asli untuk menentukan tingkat abu-abu.

Ketika melakukan transformasi gambar pixel ke gambar yang dinormalisasi, relation tes kuadrat dari gambar pixel menghasilkan tes sel non kuadrat dari gambar normalisasi. itu gambar kembali ke gambar pixel, koefisien untuk proyeksi mundur diperoleh dengan cara yang sama oleh Rn *transpose* jika panjang fokus piksel dan gambar yang dinormalisasi sama (Cho dan Schenk, 1999).

$$\begin{aligned} C'_{11} &= C_{11} & C'_{21} &= C_{12} \dots\dots\dots [16] \\ C'_{12} &= C_{21} & C'_{22} &= C_{22} \\ C'_{13} &= C_{31}fpfN & C'_{31} &= C_{32}fpfN \\ C'_{23} &= C_{13} \frac{1}{fpfN} & C'_{32} &= C_{32} \frac{1}{fpfN} \end{aligned}$$

Resampling

Setelah menerapkan transformasi geometris dari gambar yang dinormalisasi ke gambar piksel, masalahnya sekarang adalah menentukan nilai abu-abu dari lokasi piksel baru dalam gambar yang dinormalisasi, karena posisi yang diproyeksikan dalam gambar piksel tidak identik dengan pusat piksel. Oleh karena itu, nilai abu-abu harus diinterpolasi. Prosedur ini biasanya disebut sebagai resampling. Beberapa metode interpolasi dapat digunakan.

1. interpolasi tanpa urutan: nilai abu-abu dari tetangga terdekat dipilih. Ini identik dengan membulatkan posisi yang diproyeksikan ke bilangan bulat, sesuai dengan *tessellation* sistem gambar piksel. Proses ini dapat menyebabkan efek kabur yang tidak dapat diterima.
2. bilinear interpolasi: nilai abu-abu dari empat piksel pembulatan berkontribusi pada nilai abu-abu dari titik yang diproyeksikan tergantung pada jarak antara proyeksi dan empat piksel tetangga.

HASIL DAN PEMBAHASAN

Proses transformasi dari foto asli ke foto yang dinormalisasi tersebut terbagi menjadi empat tahapan transformasi yang dihitung secara manual menggunakan bahasa pemrograman python dalam *library* opencv. Setelah tahapan transformasi, selanjutnya akan dilakukan proses *resampling* untuk mendapatkan hasil *reslut* dari foto normal tersebut.

Hasil Perhitungan Elemen Matriks Rotasi pada Foto Stereo

Perhitungan matriks rotasi pada foto *original* bertujuan untuk mendapatkan nilai matriks transformasi pertama. Nilai tersebut yaitu matriks elemen rotasi Rn yang diperoleh dari hasil perkalian antara matriks *euler angel* Rb dalam satuan *degree* pada parameter *ekterior* orientasinya, dengan matriks R yang ditransposekan dalam satuan *radians* pada

parameter *ekterior* orientasinya. Hasil akhir dari transformasi tersebut adalah matriks ordo 3x3 sebagai elemen matriks Rn. Elemen pada matriks ini akan menjadi parameter yang diketahui dalam transformasi, yang berfungsi untuk merubah *original image* (diapositif) ke *normalized* (diapositif). Biasanya transformasi di sebut dengan transformasi T1 ke T2.

a. Hasil perhitungan parameter *omega*, *phi*, dan *kappa* dalam *degree* dan *radian*

Hasil dari perhitungan ini adalah parameter *omega*, *phi*, dan *kappa* sebagai nilai input pada matrik transformasi, baik itu pada foto kiri dan foto kanan. Berikut hasil perhitungannya :

Tabel 1. Hasil perhitungan parameter *omega*, *phi*, dan *kappa* dalam *degree* dan *radian*

Parameter	Nilai parameter Eo dalam (degree)		Nilai parameter Eo dalam (radians)	
	Foto kiri	Foto kanan	Foto kiri	Foto kanan
Omega	15.919	15.919	0.183	0.189
Phi	0.9	-1.404	-0.503	0.365
Kappa	-0.108	0.173	-0.422	-1.45

b. Hasil Perhitungan matrik transformasi Rn pada foto kiri

Transformasi matriks Rn di dapatkan dari hasil perkalian antara transformasi matrik Rb dan transformasi matrik R *transpose* yang masing-masing adalah matrik ordo 3x3. Fungsi dari elemen matrik Rn ini yaitu sebagai nilai *input* untuk algoritma kondisi kolinier. Berikut hasil perhitungan matrik Rn pada foto kiri :

```

Python console
In[1]: import cv2
In[2]: from math import radians, degrees
In[3]: omega_deg = 15.919
In[4]: phi_deg = 0.9
In[5]: kappa_deg = -0.108
In[6]: omega_rad = radians(omega_deg)
In[7]: phi_rad = radians(phi_deg)
In[8]: kappa_rad = radians(kappa_deg)
In[9]: Rb = cv2.Rodrigues2([omega_rad, phi_rad, kappa_rad])
In[10]: R = cv2.Rodrigues2([omega_rad, phi_rad, kappa_rad])
In[11]: Rn = Rb.dot(R.T)
In[12]: print("Rb =")
[[ 0.61753872 -0.27620628 -0.73867296]
 [-0.00754739 -0.95389075  0.29245475]
 [-0.78363466 -0.1307067  -0.6073158 ]]
In[13]: print("R =")
[[ 0.7989553  0.32285782 -0.5073788 ]
 [-0.35895414  0.93292281  0.028407 ]
 [ 0.48251668  0.15942979  0.86125478 ]]
In[14]: print("Rn =")
[[ 1.00000000e+00 -2.06166817e-17  0.00000000e+00]
 [-1.21430643e-17  1.00000000e+00 -2.06166817e-17]
 [-5.55111512e-17 -6.93889390e-18  1.00000000e+00]]
In[15]: print("Rn foto kiri =")
[[ 0.7989553  -0.35895414  0.48251668 ]
 [ 0.32285782  0.93292281  0.15942979 ]
 [-0.5073788  0.028407  0.86125478 ]]
In[16]: print("Rn foto kanan =")
[[ 0.78093657 -0.49472756 -0.38129085 ]
 [-0.51032378 -0.85735226  0.0672867 ]
 [-0.36914958  0.14209762 -0.92200997 ]]
    
```

Gambar 7. Hasil perhitungan elemen rotasi Rn pada foto kiri

c. Hasil perhitungan matrik transformasi Rn pada foto kanan

Untuk transformasi matrik Rn pada foto kanan, perhitungannya sama dengan transformasi matrik Rn pada foto kiri, hanya saja perbedaannya adalah *input* dari parameter

exterior orientasinya. Berikut hasil Rn pada foto kanan :

```

IPython console
In [2]: runfile('E:/TES/Z projek normalized stereo pair/Coding normalized
         lengkap foto kanan.py', wdir='E:/TES/Z projek normalized stereo pair')
omega_in_radians 0.18962827789993192
phi_in_radians 0.3653375598152091
kappa_in_radians -1.4500160518983851
vektor normalisasi = 1.0
Omega_degrees = 15.91995
phi_degrees = -1.4049686004049624
kappa_degrees = 0.17304679060943615
Rn =
[[[ 0.16260341  0.37274702  0.91357528]
 [ 0.02842227 -0.92728282  0.37328106]
 [ 0.98628206 -0.03473089 -0.16137367]]]
Rn Foto Kanan =
[[[ 0.11253509  0.98303373 -0.1448466 ]
 [ -0.92719893  0.05147545 -0.37101538]
 [ -0.35726459  0.17605386  0.91726062]]]
identitas =
[[[ 1.00000000e+00 -6.93889390e-18  2.77555756e-17]
 [ 6.93889390e-18  1.00000000e+00  0.00000000e+00]
 [ -2.77555756e-17  0.00000000e+00  1.00000000e+00]]]
Rn Foto Kanan =
[[[ 0.25239322 -0.47052887  0.84551774]
 [ -0.96242028 -0.21257841  0.16899003]
 [ 0.10022413 -0.85639535 -0.50649988]]]
    
```

Gambar 8. Hasil perhitungan elemen rotasi Rn pada foto kanan

Hasil perhitungan Koordiat Titik Pojok Normalisasi pada Foto Kiri dan Foto Kanan

Perhitungan koordinat titik pojok pada normalisasi foto kiri dan foto kanan diperoleh dari perhitungan transformasi *collinearity*, dimana nilai dari perhitungan ini yaitu untuk mencari koordinat titik pojok baru dari foto asli ke foto *normalized*. Berikut *result* perhitungannya menggunakan bahasa pemrograman python.

a. Hasil perhitungan koordinat titik pojok pada foto kiri menggunakan transformasi *collinearity*.

Perhitungan transformasi kondisi kolinier pada foto kiri di dapatkan dari algoritma *collinearity condition* pada metode *stereo image pair*, di mana parameter *input* dari perhitungan ini adalah transformasi matrik Rn. Berikut hasil perhitungannya :

```

IPython console
In [4]: runfile('E:/TES/Z projek normalized stereo pair/Persamaan colinier foto
         kiri fix.py', wdir='E:/TES/Z projek normalized stereo pair')
P1 = -7.369387172106362 1.2989354264850588
P2 = -54.84490935886181 28.694313298300816
P3 = -29.75721507077623 45.612458566973906
P4 = 0.8597935973318502 13.891858005927796
    
```

Gambar 9. Hasil perhitungan *collinearity* pada foto kiri

b. Hasil perhitungan koordinat titik pojok pada foto kanan menggunakan transformasi *collinearity*.

Perhitungan transformasi kondisi kolinier pada foto kanan mempunyai algoritma yang sama dengan foto kiri, hanya saja parameter *inputnya* yang berbeda, yaitu parameter transformasi Rn. Berikut hasil perhitungannya :

```

IPython console
In [6]: runfile('E:/TES/Z projek normalized stereo pair/Persamaan colinier foto
         kanan fix.py', wdir='E:/TES/Z projek normalized stereo pair')
P1 = 29.747708288843025 5.94554777308733
P2 = 14.47823330535252 39.74356482736044
P3 = -149.5333083356976 -258.73818303077616
P4 = -92.92359254964892 -26.224809346266444
    
```

Gambar 10. Hasil perhitungan *collinearity* pada foto kanan

c. Hasil penentuan Dmax dan Resolusi baru *normalized image*

Hasil dari perhitungan koordinat *pixel* foto asli (dalam millimeter) yaitu resolusi dikalikan dengan *pixel size*, hasil perhitungannya ada di tabel dibawah ini :

Tabel 2. Hasil perhitungan koordinat titik pojok foto asli

No. titik pojok	Koordinat titik pojok foto asli			
	Foto kiri (mm)		Foto kanan (mm)	
	Arah kolom (x)	Arah baris (y)	Arah kolom (x)	Arah baris (y)
P1	0	0	0	0
P2	23.168	0	23.168	0
P3	23.168	15.552	23.168	15.552
P4	0	15.552	0	15.552

Tabel 3. Hasil perhitungan koordinat titik pojok pada foto normal

No. titik pojok	Koordinat titik pojok foto normal			
	Foto kiri (mm)		Foto kanan (mm)	
	Arah kolom (x)	Arah baris (y)	Arah kolom (x)	Arah baris (y)
P1	7.369	1.298	29.747	5.945
P2	54.844	28.694	14.478	39.743
P3	29.757	45.612	149.533	258.738
P4	0.859	13.891	92.923	26.224

Tabel diatas adalah koordinat titik pojok *normalized image*. Dari tahapan ini, maka bisa menentukan *dmax* untuk mencari resolusi dan *pixel size* dari foto *normalized*. Untuk hasil resolusi dan *pixel size* dari foto normal bisa dilihat pada tabel dibawah ini :

Tabel 4. Hasil perhitungan Dmax pada foto stereo

Nilai dmax Foto kiri	Nilai dmax Foto kanan		
	Arah kolom(x)	Arah baris(y)	Arah baris(y)
	54.8449	45.6124	258.7381

Tabel 5. Hasil resolusi *normalized* pada foto kiri

Resolusi original Foto kiri		Resolusi <i>normalized</i> Foto kiri	
Arah kolom	Arah baris	Arah kolom	Arah baris
2896	1944	3634	2728

Tabel 6. Hasil resolusi *normalized* pada foto kanan

Resolusi original Foto kiri		Resolusi <i>normalized</i> Foto kiri	
Arah kolom	Arah baris	Arah kolom	Arah baris
2896	1944	14953	12936

Dari hasil resolusi *normalized image*, maka akan mendapatkan nilai *pixel size* baru dari *normalized image* tersebut. Berikut nilai *pixel size* dari *normalized image* foto kiri dan foto kanan:

Tabel 7. Hasil *Pixel size normalized* foto kiri

Pixel size original foto kiri		Pixel size <i>normalized</i> foto kiri	
Arah kolom	Arah baris	Arah kolom	Arah baris
0.008	0.008	0.018	0.023

Tabel 8. Hasil *Pixel size normalized* foto kanan

Pixel size original foto kiri		Pixel size <i>normalized</i> foto kiri	
Arah kolom	Arah baris	Arah kolom	Arah baris
0.008	0.008	0.018	0.023

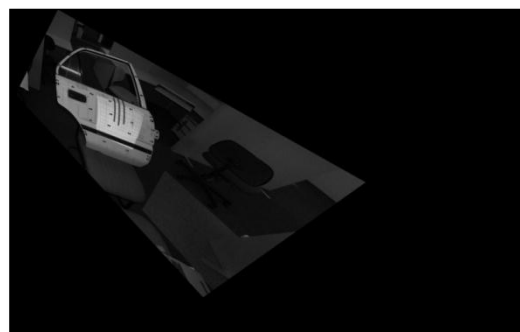
Dari hasil *Pixel size* baru ini, maka akan dilakukan perkalian kembali dengan nilai d_{max} pada *normalized image*. Dari hasil perkalian tersebut maka akan menghasilkan nilai resolusi baru pada *normalized stereo image*.

Hasil Resampling

Hasil *resampling* yaitu hasil transformasi *original pixel* ke *normalized pixel* foto dengan menggunakan metode *warp persepective* dan metode *bilinier interpolation* untuk mengisi nilai *pixel* setelah di *warping*, berikut hasilnya:

a. Hasil resampling foto kiri

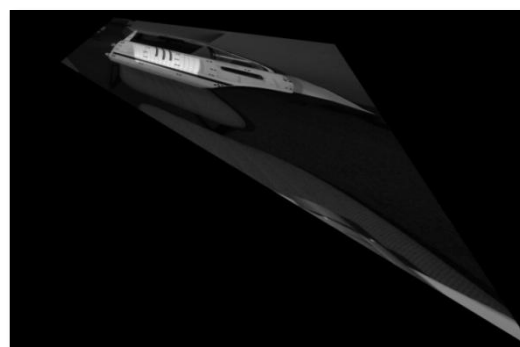
Setelah melakukan *resampling* pada foto kiri, didapatkan hasil resolusi foto sebesar 3634x2728 *pixel*. Resolusi dari normalisasi pada foto kiri ini lebih besar dari jumlah resolusi pada foto aslinya, dimana *pixel* dari foto aslinya sebesar 2896x1944 *pixel*.



Gambar 11. Hasil resampling foto kiri

b. Hasil resampling foto kanan

Setelah melakukan *resampling* pada foto kiri, didapatkan hasil resolusi foto sebesar 14953x12936 *pixel*. Resolusi dari normalisasi pada foto kiri ini lebih besar dari jumlah resolusi pada foto aslinya, dimana *pixel* dari foto aslinya sebesar 2896x1944 *pixel*.



Gambar 12. Hasil resampling foto kanan

KESIMPULAN

Berdasarkan hasil pembuatan algoritma *stereo image pair* menggunakan bahasa pemrograman python untuk foto stereo normal, *coding* fungsi dari *opencv* dan *numpy* sangat efektif, untuk penerapan metode *stereo image pair* ini, baik itu untuk melakukan proses matematika perhitungan matrik dan proses *resampling* pada foto stereo normal.

UCAPAN TERIMA KASIH

Dalam penelitian ini telah banyak pihak yang berperan. Ucapan terimakasih disampaikan kepada Keluarga, Dosen, Tenaga Kependidikan, dan teman-teman teknik Geodesi ITN Malang atas segala dukungannya

DAFTAR PUSTAKA

- Jan Erick Solem 2012. *Programing Computer Vision With Python*.
 This version of the work is a pre-production draft made available under the terms of the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License.
- Luhmann et al., 2006, *Epipolar Geometry*

M. E. Tjahjadi, and F. Handoko, 2017, " Precise Wide Baseline Stereo Image Matching for Compact Digital Cameras", Proc. EECSE 2017, Yogyakarta, Indonesia, 19-21 September 2017,

R. Hartley and A. Zisserman, 2000, Multiple View Geometry in Computer Vision, Cambridge University Press

Santoso, B. 2001. Pengantar Fotogrametri. Departemen Teknik Geodesi. Institut Teknologi Bandung.

Simmons, Gordon 1996. "Clarence G. Henning: The Man Behind the Macro". Stereo World

Suharsana, 1999, fotogrametri dasar, Teknik Geodesi Fakultas Teknik Universitas Gadjah Mada, Yogyakarta.

Triasanti, Dina. 2001. Konsep Dasar Phyton. Jakarta.
Woosug Cho and Toni Schenk. 1999, *DIGITAL IMAGERY TO EPIPOLAR GEOMETRY*, Department of Geodetic Science and Surveying.