

# DETEKSI SUDUT PADA GAMBAR 2D BERURUTAN DENGAN MENGGUNAKAN METODE HARRIS/PLESSEY CORNER DETECTOR

Sany Aji Marsetio  
Jurusan Teknik Informatika  
Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember Surabaya  
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111  
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114  
Email: [snyhope@gmail.com](mailto:snyhope@gmail.com)

## Abstrak

Persamaan elemen pada beberapa citra 2D sangatlah mudah untuk dipastikan secara penglihatan mata, namun sangatlah sulit bila ditentukan berdasarkan perhitungan matematis. Oleh karena itu, ditemukanlah sebuah metode yang disebut corner detection yang bertujuan mendeteksi sudut-sudut pada suatu citra untuk mendapatkan korelasi dari beberapa citra yang memiliki objek yang sama. Proses penentuan titik-titik tersebut merupakan fundamental dalam penentuan koordinat sumbu Z sehingga dapat ditentukan kedalaman titik tersebut yang berguna dalam rekonstruksi 3D.

## 1. Latar Belakang

Pada dunia image processing, banyak sekali masalah menarik yang bisa dibahas. Salah satunya adalah pencarian persamaan elemen suatu citra dengan citra yang lainnya atau lebih dikenal dengan feature matching. Yang merupakan basic dari feature matching tersebut adalah Corner Matching, yaitu mencocokkan sudut-sudut yang ada pada suatu citra dengan citra lainnya untuk mendapatkan titik-titik yang berkorespondensi antar citra yang mempunyai objek sama. Corner Matching sendiri mempunyai suatu metode yaitu Corner Detector.

Deteksi Sudut atau Corner Detection merupakan sebuah metode sebagai tahap awal dalam mencari Sudut/Corner suatu objek pada citra 2D. Corner tersebut bermanfaat dalam proses Correlation, yang selanjutnya bisa digunakan untuk menentukan jarak titik tersebut sehingga bisa mendapatkan koordinat 3D yaitu  $[x, y, z]$  yang merupakan elemen dasar dalam proses rekonstruksi 3D.

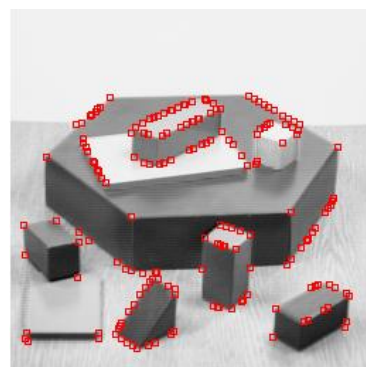
## 2. Dasar Teori

Teori-teori yang digunakan dalam penyelesaian proyek akhir akan dibahas dalam bab ini sesuai kaitannya dengan *deteksi sudut Harris/Plessey* sebagai metode yang digunakan dalam pendeteksian sudut. Serta membahas mengenai *Normalized Cross Correlation (NCS)* yang digunakan dalam menentukan sudut-sudut yang berkorelasi antar citra.

### 2.1 HARRIS PLESSEY CORNER DETECTOR

Corner Detection merupakan suatu pendekatan yang digunakan dalam sistem Computer Vision untuk mengekstraksi beberapa jenis fitur dan menyimpulkan isi dari suatu gambar. Corner Detection sering digunakan dalam deteksi gerakan, pencocokan gambar, pelacakan, mosaicing gambar, panorama stitching, pemodelan 3D dan pengenalan obyek.

Terdapat banyak metode yang digunakan dalam Corner Detection ini, salah satunya yang paling terkenal dan paling pertama adalah Moravec Operator yang dikembangkan oleh Hans P. Moravec pada tahun 1977.



Gambar 2.1: Implementasi Moravec Operator pada Blok Test Image

Untuk deteksi sudut pada proyek akhir ini digunakan metode yang dikembangkan oleh Chris Harris dan Mike Stephens pada tahun 1988 yang dikenal dengan Harris/Plessey Operator. Yang merupakan pengembangan lebih lanjut dari Moravec Operator, dimana Harris dan Stephens menggabungkan deteksi sudut dan tepi untuk mengatasi keterbatasan Moravec Operator. Berikut adalah langkah-langkahnya:

### 2.1.1 Konvolusi

Konvolusi adalah perkalian total dari dua buah fungsi f dan f yang didefinisikan dengan:

$$f * h = \int_0^T f(t)h(T-t)dt$$

Untuk fungsi f dan h yang berdimensi 2, maka konvolusi dua dimensi didefinisikan dengan:

$$f * h = \int_0^{T_x} \int_0^{T_y} f(x, y)h(T_x - x, T_y - y)dx dy$$

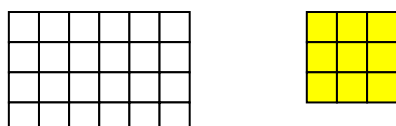
Konvolusi 2D inilah yang banyak digunakan pengolahn citra digital, sayangnya rumus diatas sangat sulit diimplementasikan menggunakan komputer, karena pada dasarnya komputer hanya bisa melakukan perhitungan pada data yang diskrit sehingga tidak dapat digunakan untuk menghitung intregral di atas.

Konvolusi pada fungsi diskrit f(n,m) dan h(n,m) didefinisikan dengan:

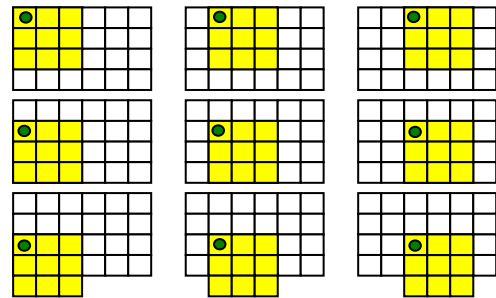
$$y(k1, k2) = \sum_{n=1}^{Tn} \sum_{m=1}^{Tm} f(k1+n, k2, m)h(n, m)$$

**Perhitungan konvolusi semacam ini dapat digambarkan dengan:**

h(x,y)



**Bila ingin dihitung  $y = f * h$ , maka proses perhitungannya dapat dilakukan dengan:**



Gambar 2.2. Perhitungan konvolusi secara grafis

Filter pada citra pada bidang spasial dapat dilakukan dengan menggunakan konvolusi dari citra (I) dan fungsi filternya (H), dan dituliskan dengan:

$$I' = H \otimes I$$

Dan dirumuskan dengan:

$$I'(x, y) = \sum_{i=-n}^n \sum_{j=-m}^m h(i, j)I(x+i, y+j)$$

dimana :

m,n adalah ukuran dari fungsi filter dalam matrik

### 2.1.2 Prewitt

Konvolusi pada citra untuk mendeteksi tepi secara horizontal dan vertical, dengan menggunakan Operator Prewitt.

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Turunan-x                      Turunan-y

Gambar 2.3: Operator Prewitt

Dalam Corner Detection, diperlukan 3 buah Image turunan yaitu Turunan X, Turunan Y, Turunan X\*Y.



Gambar 2.4: Konvolusi

### 2.1.3 Blur

Proses Blur merupakan pengolahan citra agar suatu citra terlihat kabur. Prosesnya sama dengan proses konvolusi, namun dengan operator konvolusi atau kernel yang berbeda. Untuk proses blur ini menggunakan operator Gaussian, dimana operator gaussian yang dianalogikan sebagai window 3x3 tersebut didapat dari perhitungan rumus:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Dengan  $\sigma$  merupakan sigma yang merupakan variable penting dalam penentuan tingkat blur citra. Dengan  $\sigma=1$  maka akan didapat sebuah operator Gaussian seperti ini:

$$\begin{matrix} G(-1,-1)=0.0585498 & G(0,-1)=0.0965324 & G(1,-1)=0.0585498 \\ G(-1,0)=0.0965324 & G(0,0)=0.159155 & G(1,0)=0.0965324 \\ G(-1,1)=0.0585498 & G(0,1)=0.0965324 & G(1,1)=0.0585498 \end{matrix}$$

Namun pada pendeteksian sudut kali ini digunakan  $\sigma$  bernilai 1.4.

Setelah operator Gaussian ditentukan, maka proses konvolusi dengan operator tersebut dapat dijalankan dengan menghitung nilai suatu pixel yang menjadi

titik tengah dari suatu window 3x3 dengan menggunakan perhitungan konvolusi sebagai berikut:

$$I'(x, y) = \sum_{i=-n}^n \sum_{j=-m}^m G(i, j)I(x+i, y+j)$$

Dimana I merupakan citra awal, G adalah fungsi Gaussian dan I' merupakan citra hasil.

Citra hasil deteksi tepi dari proses sebelumnya kemudian di-blur. Proses blur sendiri dimulai dari pengambilan citra degradasi dari langkah 1. Seandainya I adalah citra (*image*), maka  $\partial I/\partial x$  adalah turunan x dan  $\partial I/\partial y$  adalah turunan y. Dalam proses blur kita mengkalkulasikan ketiga variable dibawah ini:

$$A = \left(\frac{\partial I}{\partial x}\right)^2 \otimes w \quad B = \left(\frac{\partial I}{\partial y}\right)^2 \otimes w$$

$$C = \left(\frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y}\right) \otimes w$$

Dimana w adalah perkalian matriks Gaussian. Tiga citra hasil konvolusi dengan menggunakan operator Prewitt pada proses sebelumnya akan di-blur dengan operator Gaussian tersebut, sehingga akan tampak citra yang sedikit lebih buram.





Gambar 2.6: Proses Gaussian Blur

#### 2.1.4 Operator Plessey

Untuk setiap titik pada citra dibangun matriks  $2 \times 2$   $M$ , dan mengkalkulasikan operator Plessey.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Dimana A adalah citra Turunan X yang telah di-Blur, B citra Turunan Y yang telah di-Blur, dan C adalah citra Turunan  $X*Y$  yang telah di-Blur dari proses Gaussian Blur sebelumnya.

Setelah itu baru nilai Plessey dari masing-masing pixel bisa didapatkan dengan menghitung melalui rumus berikut:

$$R(x,y) = \det(M) - k*[\text{tr}(M)]^2$$

$$\text{dan } k = 0,04$$

$\det(M)$  merupakan Determinan dari matrix M yaitu  $(A*D)-(C*C)$  dan  $\text{tr}(M)$  merupakan Trace dari matrix M yaitu  $(A+B)$ .

#### 2.1.5 Non-max suppression

*Proses Non Maximum Suppression* yang mirip dengan proses *thinning* (perampingan) dilakukan untuk menentukan piksel tepi dengan posisi paling mendekati lokasi terjadinya perubahan nilai piksel diantara banyaknya piksel tepi yang terdeteksi. Dimana pada umumnya, perubahan nilai piksel berada pada pusat kumpulan piksel tepi [Nixon dan Aguado, 2002].

Bila nilai Plessey dari sebuah titik tertentu yang dihasilkan adalah nilai maksimum lokal dalam sebuah wilayah  $3 \times 3$ , maka dapat disimpulkan sementara bahwa titik ini adalah sebuah titik pojok.

#### 2.1.6 Thresholding

*Thresholding* merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama (Katz,2000). Dalam pelaksanaannya *Thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*.

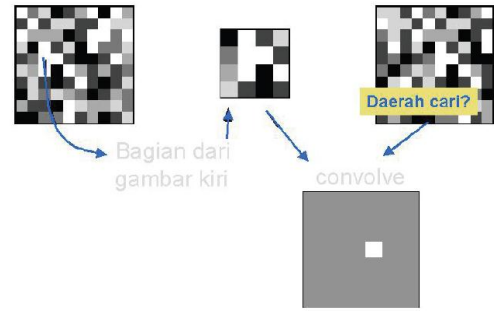
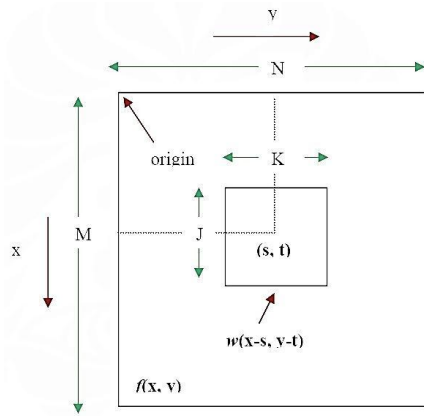
*Thresholding* digunakan untuk mempartisi citra dengan mengatur nilai intensitas semua piksel yang lebih besar dari nilai *threshold T* sebagai latar depan dan yang lebih kecil dari nilai *threshold T* sebagai latar belakang. Biasanya pengaturan nilai *threshold* dilakukan berdasarkan histogram *grayscale* (Gonzales dan Woods, 2002; Fisher, dkk, 2003; Xiaoyi dan Mojon, 2003).

Jumlah titik-titik yang akan dikumpulkan dari hasil langkah 4 akan membengkak, dan tidak semua titik merupakan titik-titik yang penting. Dari semua titik yang ditemukan jumlah ini akan diturunkan dengan cara mengambil sebagian dari titik-titik yang memiliki nilai Plessey terbesar

## 2.2 CORNER CORRELATION

Untuk mendapatkan titik-titik yang sama antar citra, maka dapat digunakan correlation-based. Dalam correlation-based kita memotong-motong citra yang pertama dengan ukuran tertentu pada bagian sudut kemudian per potongan kita kemudian membandingkan pixel per pixel dengan citra yang kedua, dimana citra yang kedua tetap utuh, untuk mencari letak persamaan dari kedua citra tersebut.

Titik-titik persamaan citra yang ditemukan pada proses ini akan disimpan pada sebuah temporary table (table sementara) yang dapat dilihat kembali (look-up table) yang kemudian akan digunakan untuk membangun ulang kedua citra.



Gambar 2.7: Proses Korelasi

F = citra kedua

w = potongan citra pertama

M = Panjang citra kedua

N = Lebar citra kedua

J = Panjang potongan citra pertama

K = Lebar potongan citra pertama

x, y = titik-titik koordinat yang akan dilalui oleh w

s,t = koordinat pusat w

$$\gamma(s,t) = \frac{\sum \sum [f(x,y) - \bar{f}(x,y)][w(x-s, y-t) - \bar{w}]}{(\sum \sum [f(x,y) - \bar{f}(x,y)]^2 \sum \sum [w(x-s, y-t) - \bar{w}]^2)^{1/2}}$$

Untuk rumus diatas kita mencari  $\gamma$  maksimum. Pada  $\gamma$  tertinggi, maka itu adalah lokasi persamaan citra yang telah dikorelasikan.

Karena metode korelasi yang digunakan adalah Normalized Cross Correlation maka perlu adanya Normalisasi terlebih dahulu pada w atau window citra yang akan dicari korelasinya.

$$W'(x, y) = \frac{W(x, y)}{\sum_{i=-n}^n \sum_{j=-m}^m W(i, j)^2}$$

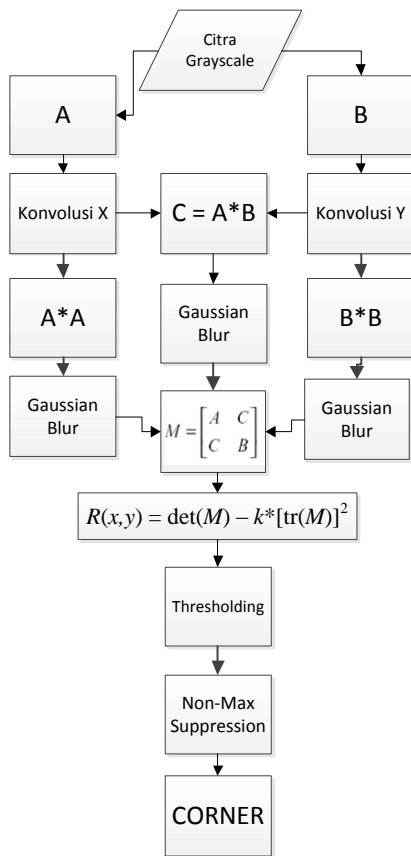
### 3. Perancangan dan Pembuatan Sistem

Sebelum pembuatan akan dilakukan perencanaan untuk alur pada sistem yang terdiri dari Deskripsi Umum, Deteksi Sudut, Korelasi, Rancangan Tampilan, Rancangan Proses.

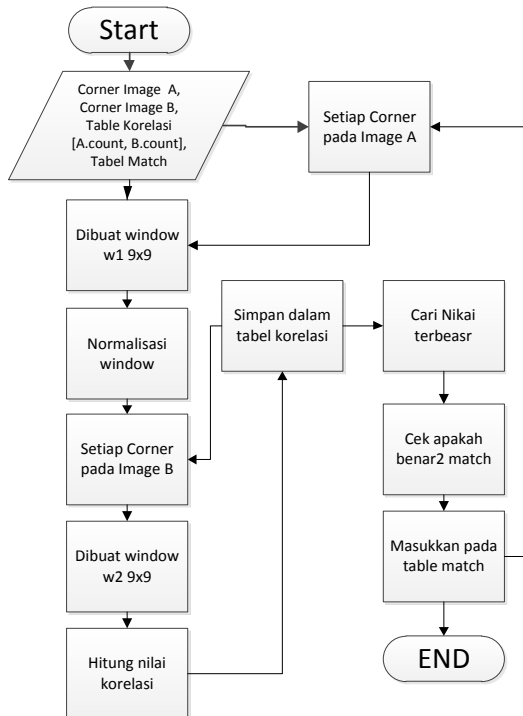
#### 3.1 Deskripsi Umum Sistem

Pada implementasi Deteksi Sudut ini diujicobakan tiga buah Citra dengan objek yang sama namun dengan angle yang berbeda. Hal ini dimaksudkan sebagai simulasi pendeteksian sudut pada kamera yang real-time. Ketiga citra tersebut akan dideteksi sudut-sudutnya dan dicari korelasinya.

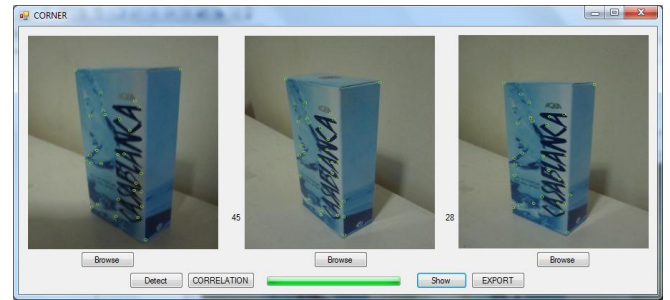
### 3.2 Proses Deteksi Sudut



### 3.3 Korelasi

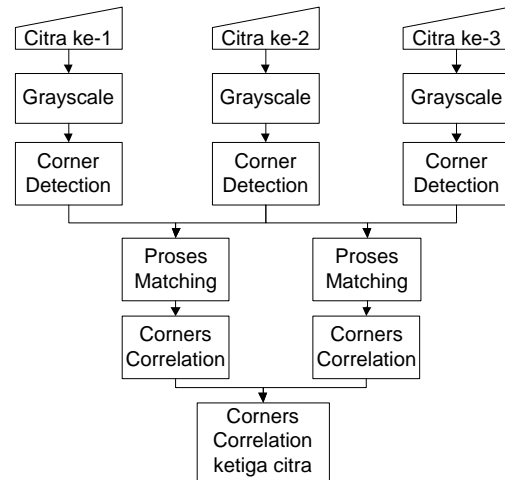


### 3.4 Rancangan Tampilan

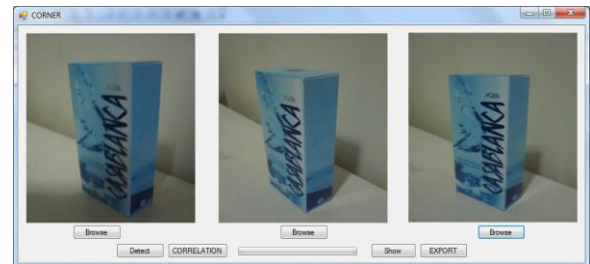


Gambar 3.1: Rancangan Aplikasi

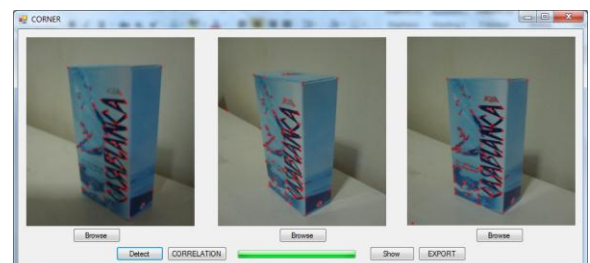
### 3.5 Rancangan Proses Sistem



### 4. Uji Coba dan Analisa

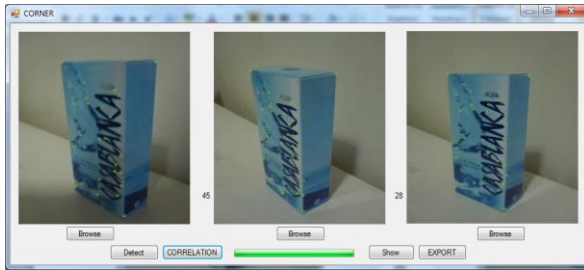


Gambar 4.1: Browse Image

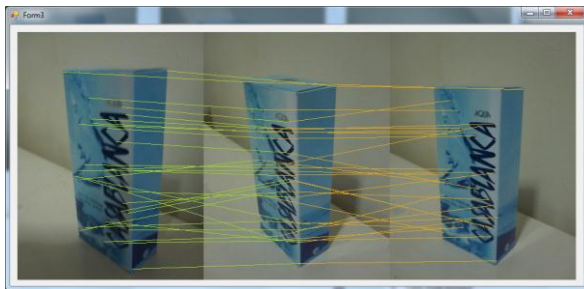


Gambar 4.2: Hasil Deteksi Sudut





Gambar 4.3: Hasil Korelasi



Gambar 4.4: Citra Hasil Korelasi

Inde x P1	Koordi nat P1	Inde x P2	Koordi nat P2	Inde x P3	Koordi nat P3
0	240, 62	4	213, 87	2	214, 92
1	74, 64	2	64, 80	0	71, 89
2	114, 106	8	95, 125	5	111, 126
3	168, 121	15	135, 145	3	98, 110
6	174, 133	20	140, 157	15	166, 148
10	118, 140	18	98, 155	20	113, 155
13	121, 147	22	101, 161	22	116, 160
14	98, 149	21	84, 160	23	93, 164
16	144, 155	27	119, 172	25	139, 167
21	160, 177	35	131, 194	69	142, 253
39	108, 214	48	93, 218	48	100, 221
41	151, 221	45	129, 213	16	143, 152
42	141, 222	54	118, 231	50	133, 226
44	97, 223	50	86, 223	53	88, 230
48	91, 232	89	156, 309	98	175, 299
51	99, 238	41	89, 206	61	89, 243
52	115, 239	101	120, 335	109	118, 315
62	178, 257	72	147, 266	52	168, 227

67	170, 276	65	138, 257	77	159, 270
70	167, 289	57	137, 237	108	156, 314
72	178, 292	61	146, 240	110	166, 316
73	142, 306	105	156, 339	100	131, 303
75	109, 315	86	98, 305	111	96, 317
77	179, 319	94	150, 319	86	168, 279
78	100, 320	69	86, 262	114	86, 323
84	138, 338	99	120, 329	73	88, 262
85	173, 340	102	146, 336	119	162, 330
88	185, 382	111	157, 373	134	172, 369

Tabel 4.1: Tabel Hasil Korelasi Ketiga Citra

## 5. Kesimpulan dan Saran

### 5.1 KESIMPULAN

Berdasarkan analisa dari beberapa pengujian yang diterangkan pada bab sebelumnya, kesimpulan yang didapatkan adalah :

1. Sigma, Threshold dan k, merupakan variable yang benar berpengaruh terhadap hasil pendeteksian sudut. Nilai default ketiga variable tersebut adalah Sigma = 1.4, Threshold = 1000 dan k = 0.04. Pemilihan nilai ketiga variable tersebut secara tepat akan mendapatkan deteksi sudut yang akurat.
2. Pada Citra yang memiliki warna atau tekstur akan lebih banyak terdeteksi sudut daripada citra yang polos.
3. Besar resolusi dari citra sangat mempengaruhi pendeteksian sudut, semakin besar resolusi akan semakin banyak sudut yang terdeteksi juga semakin lama proses pendeteksiannya. Contoh: pada uji coba tentang resolusi, sebuah citra beresolusi 300x400 terdeteksi 117 sudut, sedangkan pada citra beresolusi 600x800 terdeteksi 174 citra.
4. Perubahan sudut objek pada citra yang baik adalah kurang dari  $60^0$  atau sebaiknya dibawah  $20^0$  untuk mendapat hasil korelasi yang maksimal.

### 5.2 SARAN

1. Pengaksesan pixel dari Citra melalui operasi byte akan mempercepat pendeteksian sudut, daripada melalui SetPixel dan GetPixel.
2. Untuk kedepannya, agar lebih diperhatikan pergeseran sudut objek dalam pengambilan citra. Disarankan untuk

menghitung besar sudut perubahan dan membuat perubahan sudut tersebut secara konstan.

3. Proyek akhir ini diharapkan bisa berguna untuk perancangan sistem lain baik digunakan dalam aplikasi dengan menggunakan kamera secara real-time maupun dalam bidang robotika.

## 6. Daftar Pustaka

1. C.G. Harris and M.J. Stephens. "A combined corner and edge detector", Proceedings Fourth Alvey Vision Conference, Manchester. pp 147-151, 1988.
2. <http://kiwi.cs.dal.ca/~dparks/CornerDetection/index.htm>
3. Alison Noble, "Descriptions of Image Surfaces", PhD thesis, Department of Engineering Science, Oxford University 1989, p45.
4. P. D. Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing. School of Computer Science and Software Engineering, The University of Western Australia. Available in:
5. <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/Match/matchbycorrelation.m>
6. [http://en.wikipedia.org/wiki/Corner\\_detection](http://en.wikipedia.org/wiki/Corner_detection)
7. [http://id.wikipedia.org/wiki/Pengolahan\\_citra](http://id.wikipedia.org/wiki/Pengolahan_citra)
8. <http://crsouza.blogspot.com/2010/05/harris-corners-detector-in-c.html>
9. <http://lecturer.eepis-its.edu/~nana>
10. [http://codeproject.com/KB/recipes/automatic\\_panoramas.aspx](http://codeproject.com/KB/recipes/automatic_panoramas.aspx)
11. Jie Chen, Li-hui Zou, Juan Zhang, LiHua Dou. The Comparison and Application of Corner Detection Algorithms. Journal of Multimedia, 2009: 435-441
12. <http://siddhantahuja.wordpress.com/2010/04/11/correlation-based-similarity-measures-summary/>
13. <http://hasishibai.blogspot.com/2009/09/image-processing-c-tutorial-4-gaussian.html>
14. [http://en.wikipedia.org/wiki/Gaussian\\_blur](http://en.wikipedia.org/wiki/Gaussian_blur)
15. [http://en.wikipedia.org/wiki/Prewitt\\_operator](http://en.wikipedia.org/wiki/Prewitt_operator)
16. B. Senthilkumar, G. Umamaheswari. "A Novel Edge Detection Algorithm for the Detection of Breast Cancer", European Journal of Scientific Research ISSN 1450-216X Vol.53 No.1 (2011), pp.51-55, 2011.
17. Rajat Hphull, Pradip Mainali, Qiong Yang, Patrice Rondao Alface, Henk Sips. "Low Complexity Corner Detector Using CUDA for Multimedi Applications", MMEDIA 2011: The Third International Conferences on Advances in Multimedi, 2011.
18. Rupin Dalvial, Ilker Hacihaliloglua, Rafeef Abugharbieha. "3D Ultrasound Volume Stitching Using Phase Symmetry and Harris Corner Detection for Orthopaedic Applications", In B. M. Dawant and D. R. Haynor, editors, Medical Imaging 2010: Image Processing, volume 7623, page 762330. SPIE, 2010.
19. Simone Fintrop. "Visual Robot Localization and Mapping based on Attentional Landmarks", German Conference on Artificial Intelligence(KI), Osnarbruck, Germany, Sept. 2007
20. Jiong-Min Kim, Myung-A Kang, "Mosaic Based Real Time CCTV System Using Improved Harris Corner," fksd, vol. 6, pp.535-539, 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009.
21. Jie Zhao, Sha Chang, Guo-Zun Men: 3D Reconstruction based on the matching method of GA. ICMLC 2010: 1798-1801