

Aplikasi Deteksi Tepi pada *Realtime Video* menggunakan Algoritma *Canny Detection*

Edy Winarno

Program Studi Teknik Informatika, Universitas Stikubank

email: edywinarno3@gmail.com

Abstract

Real-time video and image processing is used in a wide variety of applications from video surveillance and traffic management to medical imaging applications. This paper presents the implementation of an canny edge-detection using in realtime video from camera. The Canny algorithm uses an optimal edge detector based on a set of criteria which include finding the most edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and marking edges only once when a single edge exists for minimal response.

Keywords : *OpenCV, realtime video, Canny detection*

PENDAHULUAN

Pengolahan Citra merupakan sebuah proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup seluruh data dua dimensi.

Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan intensitas mutu, misalnya mengandung cacat atau derau (noise), warnanya terlalu kontras atau kabur tentu citra seperti ini akan sulit di representasikan sehingga informasi yang ada menjadi berkurang. Agar citra yang mengalami gangguan mudah direpresentasikan maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik.

1. Citra Digital

Citra digital merupakan citra yang dihasilkan dari pengolahan dengan menggunakan komputer, dengan cara merepresentasikan citra secara numerik dengan nilai-nilai diskret. Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi

ukurannya dinyatakan dalam tinggi kali lebar atau panjang kali lebar. Citra digital yang tingginya N , lebarnya M , dan memiliki nilai derajat keabuan L dapat dianggap sebagai fungsi:

$$f(x,y) = \begin{cases} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{cases} \dots\dots\dots(1)$$

Citra digital yang berukuran $N \times M$ biasanya dinyatakan dalam bentuk matrik yang berukuran N baris dan M kolom sebagai berikut:

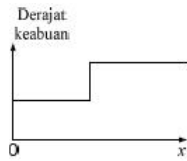
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \dots\dots\dots(2)$$

2. Deteksi Tepi

Tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Terdapat tiga macam tepi di dalam citra digital, yaitu:

a. Tepi Curam

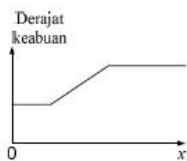
Tepi curam adalah tepi dengan perubahan intensitas yang tajam. Arah tepi berkisar 90°. Gambar 1 menunjukkan deteksi tepi tipe tepi curam.



Gambar 1. Tepi curam

b. Tepi Landai

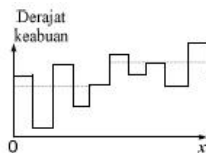
Tepi landai yaitu tepi dengan sudut arah yang kecil. Tepi landai dapat dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan. Gambar 2 menunjukkan deteksi tepi tipe tepi landai.



Gambar 2. Tepi landai

c. Tepi yang mengandung derau

Umumnya tepi yang terdapat pada aplikasi visi komputer mengandung derau. Operasi peningkatan kualitas citra dapat dilakukan terlebih dahulu sebelum pendektasian tepi. Gambar 3 menunjukkan deteksi tepi tipe tepi curam dengan derau.



Gambar 3. Tepi curam dengan derau

Deteksi tepi memiliki berbagai macam operator dalam mendeteksi tepi suatu citra, yaitu:

2.1. Operator Sobel

Pengaturan piksel di sekitar piksel (x,y):

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator sobel adalah magnitude dari gradien yang dihitung dengan:

$$M = \sqrt{s_x^2 + s_y^2} \dots\dots\dots(3)$$

Turunan parsial dihitung dengan

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

Dengan konstanta c adalah 2, dalam bentuk mask, sx dan sy dapat dinyatakan sebagai:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Arah tepi dihitung dengan persamaan

$$A(x, y) = \tan^{-1}(S_x / S_y) \dots\dots\dots(4)$$

2.2. Operator Prewitt

Persamaan gradien pada operator Prewitt sama seperti operator Sobel, tetapi menggunakan nilai konstanta c = 1

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

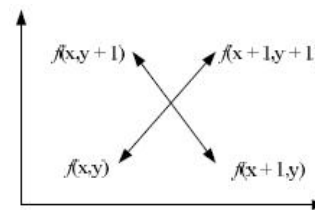
2.3. Operator Roberts

Operator Roberts sering disebut juga operator silang. Gradien Roberts dalam arah-x dan

arah-y dihitung dengan rumus:

$$R_+(x, y) = f(x+1, y+1) - f(x, y)$$

$$R_-(x, y) = f(x, y+1) - f(x+1, y)$$



Gambar 4. Operator Silang

Gambar 4 menunjukkan bahwa operator R+ adalah hampiran turunan berarah dalam arah 45°, sedangkan operator R- adalah hampiran turunan berarah dalam arah 135°. Dalam bentuk maks konvolusi, operator Roberts adalah:

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

2.4. Algoritma Canny

Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan metode Canny. Deteksi tepi Canny ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia. Ada beberapa kriteria pendeteksi tepian paling optimum yang dapat dipenuhi oleh algoritma Canny:

a. Mendeteksi dengan baik (criteria deteksi)

Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.

b. Melokalisasi dengan baik (criteria lokalisasi)

Dengan Canny dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

c. Respon yang jelas (kriteria respon)

Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya. Pemilihan parameter deteksi tepi Canny sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut antara lain :

1. Nilai Standart Deviasi Gaussian

2. Nilai Ambang

Pendekatan algoritma canny dilakukan dengan konvolusi fungsi gambar dengan operator gaussian dan turunan-turunannya. Turunan pertama dari fungsi citra yang dikonvolusikan dengan fungsi gaussian,

$$g(x,y) = D[\text{gauss}(x,y) * f(x,y)] \quad (5)$$

ekivalen dengan fungsi citra yang dikonvolusikan dengan turunan pertama dari fungsi gaussian,

$$g(x,y) = D[\text{gauss}(x,y)] * f(x,y) \quad (6)$$

Oleh karena itu, memungkinkan untuk mengkombinasikan tingkat kehalusan dan pendektasian tepi ke dalam suatu konvolusi dalam satu dimensi dengan dua arah yang berbeda (vertical dan horizontal).

METODE PENELITIAN

1. Perangkat Keras

Perangkat keras minimum yang digunakan dalam membuat aplikasi pada penelitian ini adalah computer 32/64 bit dengan spesifikasi Intel(R) Pentium 4 Core2Duo 2 GB of RAM, dan Harddisk 320 GB. Sebuah kamera / webcam yang digunakan untuk pengambilan gambar video.

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun aplikasi Deteksi tepi pada realtime video menggunakan algoritma Canny Detection adalah sebagai berikut:

1. Microsoft Visual Studio 2010 Ultimate

Sebuah compiler yang dibuat menggunakan bahasa pemrograman C++ yang akan digunakan untuk aplikasi sistem ini.

2. OpenCV 2.1.0

Sebuah library untuk fungsi-fungsi computer vision dan opencv yang mendukung pembuatan program ini.

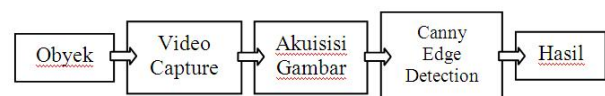
3. CMake 2.8.5

CMake digunakan untuk mengendalikan proses kompilasi perangkat lunak menggunakan platform sederhana dan file konfigurasi compiler independen. CMake menghasilkan makefiles asli dan ruang kerja yang dapat digunakan dalam lingkungan compiler yang digunakan dalam aplikasi ini.

4. TBB 4.0 (*Threading Building Block*)

Sebuah library yg memanfaatkan *fungsi* multithreading untuk mempermudah kerja processor

3. Perancangan Program



Gambar 5. Blok Diagram Sistem

Pada gambar 5, obyek yang merupakan data gambar bergerak seperti manusia dan lain-lain akan diambil secara realtime menggunakan Video Capture berupa kamera / webcam yang dipasang pada obyek. Oleh Video Capture, data

yang berupa gambar video realtime kemudian diolah menggunakan akuisisi data untuk menghasilkan gambar yang selanjutnya akan diproses menggunakan Canny Edge Detection untuk mendapatkan hasil berupa gambar video realtime yang sudah mengalami proses deteksi tepi.

Tahap awal dalam implementasi perancangan adalah dengan melakukan proses instalasi seluruh program mulai dari instalasi Visual Studio 2010 ultimate, instalasi openCV dan penggunaan CMake dan TBB untuk pendukung dari instalasi program ini.

Dalam proses capturing gambar dilakukan dengan menggunakan sebuah webcam atau kamera yang sudah tersedia di laptop, sehingga memudahkan aplikasi yang dijalankan. Bahasa pemrograman yang digunakan adalah Visual C++ yang diaplikasikan menggunakan library dari openCV.

Data video yang telah diakuisisi kemudian akan akan ditampilkan dalam bentuk gambar video realtime berwarna, kemudian data ini akan diproses menggunakan algoritma Canny Edge Detection untuk menghasilkan tampilan realtime video yang sudah mengalami proses pendeteksian tepi.

HASIL DAN PEMBAHASAN

Dari hasil aplikasi program yang telah diimplementasikan menggunakan Canny Detection dapat diperoleh hasil sebagai berikut:

1. Proses Akuisisi Gambar Video

Pada proses ini data obyek yang merupakan input data yang akan diolah, kemudian dilakukan proses akuisisi utk dapat menghasilkan gambar video realtime seperti pada gambar 6.



Gambar 6. Hasil Akuisisi data Video

Adapun proses pengambilan gambar di atas dapat dilakukan dengan program seperti di bawah ini :

```
int key = 0;
    CvCapture *capture = cvCaptureFromCAM
(0);
    //keluar pke keyboard 'q'
    while( key != 'q') {
        IplImage *frame =
cvQueryFrame(capture);
        if( !frame ) break;
```

Gambar 6. Listing program pengambilan gambar

Untuk menampilkan tampilan gambar obyek pada layar digunakan program seperti di bawah ini :

```
//.....program untuk menampilkan gambar
video pada layar
    IplImage* gray =
cvCreateImage(cvGetSize(frame),
IPL_DEPTH_8U, 0 );
    cvCvtColor( frame, gray,
CV_BGR2GRAY);
    cvShowImage( "Edywin_colour", frame );
//-----
```

Gambar 7. Listing program menampilkan gambar

2. Proses Deteksi Tepi dengan Canny Detection

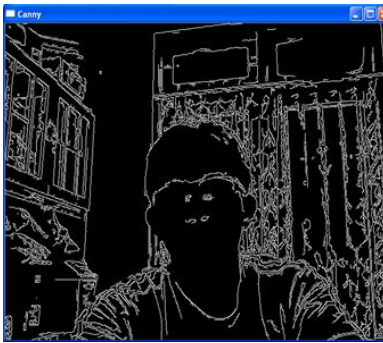
Setelah dilakukan proses pengambilan gambar, kemudian data diproses menggunakan algoritma Canny Edge Detection menggunakan program seperti di bawah ini :

```
//-----
    IplImage* canny = cvCreateImage
(cvGetSize( gray), IPL_DEPTH_8U, 0 );
    cvCanny( gray, canny, 70, 200, 3);
    cvShowImage( "Canny", canny);
//-----
```

Gambar 8. Listing program algoritma canny edge detection

IplImage canny* merupakan variabel gambar untuk menampilkan output sebuah gambar realtime video yang diproses menggunakan Canny Edge Detection. *cvCanny* digunakan sebagai pemanggilan fungsi canny pada openCV. Nilai 70 merupakan nilai

threshold minimum dari gambar video realtime yang akan ditampilkan, sedangkan nilai 200 merupakan nilai threshold maksimal yang akan ditampilkan. Nilai 3 merupakan nilai *aperture size*-nya, sedangkan fungsi *cvShowImage* digunakan sebagai pemuncil gambar canny-nya. Gambar 9 di bawah ini adalah hasil proses deteksi tepi menggunakan algoritma Canny Edge Detection pada tampilan layar computer. Gambar di bawah merupakan gambar realtime video dan akan bergerak sesuai capturing gambar obyek yang dideteksi.



Gambar 9. Hasil Deteksi Tepi dengan Canny

Pembahasan

Algoritma Canny Edge Detection yang digunakan pada proses pengolahan gambar video realtime ini memiliki prinsip yang hampir sama seperti pada proses pendeteksian dengan obyek gambar diam atau foto digital. Kelebihan pada aplikasi ini adalah dapat menerjemahkan atau mentranslasikan gambar video realtime menjadi tampilan gambar video yang sudah mendapatkan proses pendeteksian tepi menggunakan Canny Edge Detection, sehingga output dapat berjalan sesuai kondisi realtime obyek yang diambil, meskipun dalam kondisi bergerak.

Namun dalam proses perubahan data dari video realtime menjadi tampilan gambar deteksi tepi yang berjalan secara realtime memerlukan kemampuan memori yang sangat besar, karena proses translasi data yang kontinyu serta berjalan terus menerus mengakibatkan kebutuhan memory yang digunakan dalam komputer juga akan berpengaruh pada proses pendeteksian frame demi frame dari sinyal video yang digunakan.

Secara prinsip proses deteksi tepi menggunakan Canny Edge Detection dalam aplikasi ini mempunyai kemampuan hasil yang baik dan dapat diandalkan untuk proses pengolahan data selanjutnya.

Gambar-gambar di bawah ini menunjukkan proses deteksi tepi yang dilakukan pada gambar foto yang dibandingkan dengan gambar video realtime, semua deteksi dilakukan dengan menggunakan algoritma Canny Edge Detection.



Gambar 10. Gambar digital sebelum proses deteksi tepi



Gambar 11. Gambar digital setelah proses deteksi tepi



Gambar 12. Video realtime sebelum proses deteksi



Gambar 13. Video realtime setelah proses deteksi

KESIMPULAN

Dari hasil implementasi pendeteksi tepi pada realtime video menggunakan algoritma Canny Detection dapat diberikan beberapa kesimpulan diantaranya:

1. Algoritma Canny Edge Detection adalah salah satu dari beberapa algoritma deteksi tepi yang memiliki kelebihan dapat memberikan hasil deteksi tepi yang optimal dan mampu memberikan sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.
2. Canny Edge Detection dapat diterapkan dalam deteksi sebuah gambar / foto digital maupun gambar video realtime yang diambil dari webcam / kamera.
3. Dalam aplikasi menggunakan gambar video realtime, Canny Edge Detection membutuhkan kemampuan memori computer yang tinggi untuk dapat memproses seluruh data gambar video yang diambil agar tidak terjadi error dalam proses pengolahannya.

DAFTAR PUSTAKA

- Bovik A, C., *The Essential Guide to Video Processing*, Academic Press USA, 2009.
- Castleman, Kenneth R., 2004, *Digital Image Processing*, Vol. 1, Ed.2, Prentice Hall, New Jersey.
- Gonzalez. R. C, Woods. R. E., *Digital Image Processing* third Edition, Pearson Prentice Hall, New Jersey, 2008.

Jahne, B., Haubecker, H., Geibler, P., *Handbook Of Computer Vision And Applications*, Volume 2, Signal Processing and Pattern Recognition, 1999, Accademic Press.

Jain, A.K., *Fundamental of Digital Image Processing*, Prentice Hall, Inc., Singapore, 1989.

Low, A., *Introductory Computer Vision and Image Processing*, McGraw-Hill , UK, 1991.

Munir, R., *Pengolahan Citra Digital dengan Pendekatan Algoritmik* , Informatika Bandung , 2004..

Murni, A. dan S. Setiawan, *Pengantar Pengolahan Citra*, Elex Media Komputindo, Jakarta, 1992.

Nixon, M, S., Aguado, A, S., *Feature Extraction and Image Processing second edition.*, Academic Press, 2008.

Pitas, I., *Digital Image Processing Algorithms*, Prentice Hall, Singapore, 1993.

Robert Laganieri, *OpenCV 2 Computer Vision Application Programming Cook Book*, Packt Publishing, Birmingham, UK, 2011.