

Analisis Perbandingan Algoritma Optical Flow dan Background Estimation untuk Pendeteksian Objek pada Video

¹Ety Sutanty, ²Ari Rosemala

^{1,2}Universitas Gunadarma, Jurusan Sistem Informasi, Depok, Indonesia

E-mail: ^{1,2}{ety_s, ari_rosemala }@staff.gunadarma.ac.id

Abstrak

Pada penelitian ini penulis akan melakukan analisis dari metode *optical Flow* dan *Background Estimation* dalam sebuah video *viptraffic.avi* berdurasi 8 detik dalam melakukan pendeteksian gerakan dan frame yang dihasilkan oleh kedua algoritma. Hasil Ujicoba pendeteksian objek dengan video input *viptraffic.avi* untuk dengan algoritma Optical Flow memberikan hasil yang lebih baik, dimana algoritma dapat menghitung secara akurat ketika arus kendaraan dalam keadaan padat atau lancar sedangkan *Background Estimation* bekerja kurang akurat. Proses menangkap objek bergerak lebih cepat dilakukan oleh *Optical Flow* dengan jumlah citra frame yang dihasilkan 14 Citra Frame/Detik sedangkan *Background Estimation* menghasilkan 25 Citra Frame/Detik. Hasil dari analisis kedua metode ini diharapkan dapat meminimalisir beberapa kendala dalam melakukan metode perhitungan kepadatan lalu lintas dengan data yang lebih akurat, sehingga memudahkan pihak terkait dalam melakukan pengaturan lalu lintas

Kata Kunci : Citra Frame, Objek, *Optical Flow*, *Background Estimation*

Pendahuluan

Salah satu bentuk manajemen lalulintas yang dilakukan oleh pemerintah adalah pemasangan *Closed Circuit Television* (CCTV) di beberapa jalan protokol maupun jalan tol. Penggunaan CCTV selama ini bukan hanya digunakan untuk pengamatan visual secara manual oleh petugas, tetapi juga digunakan untuk manajemen lalulintas khususnya dalam melakukan pemantauan data kepadatan lalu lintas di tiap ruas jalan. Data ini diperlukan untuk pengaturan yang lebih efektif di beberapa ruas jalan, pengaturan durasi lampu merah serta penempatan petugas polisi lalulintas untuk melakukan pengaturan di beberapa ruas jalan, terutama beberapa ruas jalan dengan tingkat kepadatan kendaraan yang tinggi. Selama ini untuk mendapatkan data tersebut dilakukan dengan cara manual yaitu dengan menerjunkan petugas langsung ke lapangan dengan tingkat akurasi data yang rendah. Pemerintah melalui dinas Dinas Pekerjaan Umum di Jalan Soekarno-Hatta, Bandung-Jawa Barat juga mencoba perangkat lunak Penghitung Lalu Lintas Otomatis (PLATO) untuk mendapatkan

data lalu lintas dengan akurasi tinggi, akan tetapi karena alat ini merupakan gabungan antara penggunaan sensor elektrik dan mekanik, maka sering terkendala pemasangan dan implementasinya di ruas jalan [1].

Salah satu hal penting dalam pengolahan video adalah pendeteksian gerakan (*Motion Detection*). Sistem pendeteksian gerakan ini mampu menentukan arah dan menghitung jumlah objek yang bergerak pada suatu video, salah satunya adalah pendeteksi gerakan pada video pemantau lalu lintas (*traffic*). Beberapa sistem yang digunakan untuk pemantau lalu lintas adalah *Optical Flow* dan *Background Estimation*. *Optical Flow* mampu menghitung kecepatan objek yang bergerak dengan mengikuti pola *brightness* setiap piksel dalam urutan suatu video dimana *brightness* tiap citra konstan dari waktu ke waktu, piksel bergerak tidak terlalu jauh antar citra, dan piksel yang bertetangga bergerak ke arah yang sama[2]. *Optical Flow* juga memiliki kemampuan langsung berkomunikasi dengan manusia dengan menggunakan webcam sebagai *interface*.

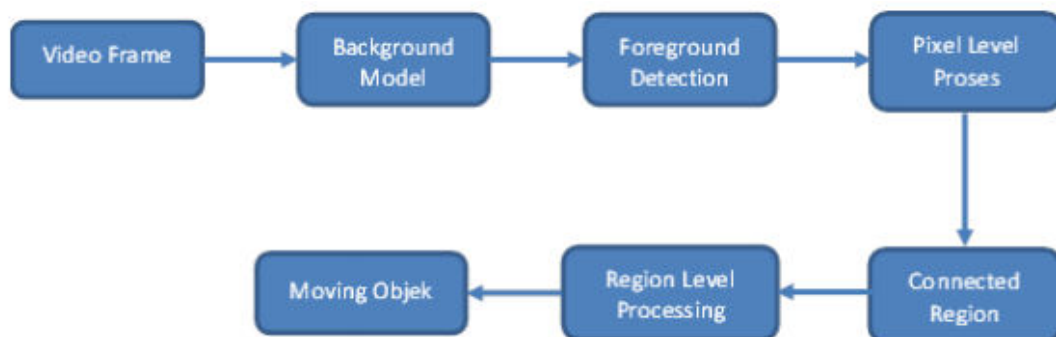
Beberapa penelitian terkait implementasi metode untuk pendeteksian dan *tracking* menggunakan *Optical Flow* dan *Background Estimation* dilakukan peneliti terdahulu. Penelitian dilakukan untuk menemukan jalur pergerakan objek yang bergerak di *frame* yang berbeda. Beberapa tahapan dilakukan untuk melakukan pendeteksian terhadap objek antara lain : akuisisi gambar, merubah gambar dari yang berwarna menjadi hitam-putih, pengenalan objek, dan memperkirakan pergerakan dari objek. Hasil ujicoba menunjukkan metode yang diterapkan peneliti menghasilkan ketepatan hingga 92%. *Tracking* dilakukan dengan kecepatan *frame* 25 *frame* per sec [3]. Penelitian lain untuk mendeteksi objek yang bergerak dilakukan dengan dua algoritma. Algoritma pertama melakukan deteksi terhadap objek berdasarkan tingkat perubahan keabu-abuan diantara dua *frame* yang berdekatan. Hasil ujicoba dari algoritma pertama ini terdapat informasi dari warna yang terbangun. Pada algoritma kedua kekurangan dari metode pengurangan latar dan metode warna menjadi hitam-putih dapat dikurangi, dimana kombinasi algoritma kedua pendeteksian dilakukan terhadap perubahan piksel yang terjadi pada sumbu x dan y. Hasilnya dengan menggunakan dengan menggunakan algoritma kedua *tracking* dapat dilakukan lebih baik dan memakan waktu pemrosesan yang lebih sedikit[4]. Penelitian pendeteksian objek bergerak juga dilakukan dengan *Background Substraction* yang dikombinasikan dengan algoritma neural

network yaitu *SelfOrganizingMap*. Algoritma SOM yang diterapkan memisahkan piksel *background* dan *foreground* pada citra. Beberapa permasalahan umum dari *Background Substraction* juga dapat diatasi. Hasil deteksi dengan nilai Pada penelitian selanjutny menerapkan *Background Substraction* berbasis SOM untuk deteksi objek pada kamera stationer [5].

Pada penelitian ini penulis akan melakukan analisis dari metode *optical Flow* dan *Background Estimation* dalam sebuah video *viptraffic.avi* berdurasi 8 detik dalam melakukan pendeteksian gerakan dan *frame* yang dihasilkan oleh kedua algoritma. Hasil dari analisis kedua metode ini diharapkan dapat meminimalisir beberapa kendala dalam melakukan metode perhitungan kepadatan lalu lintas dengan data yang lebih akurat, sehingga memudahkan pihak terkait dalam melakukan pengaturan lalu lintas

Metode Penelitian

Pada penelitian ini untuk melakukan pendeteksian terhadap objek bergerak dalam video dilakukan dengan metode *Optical Flow* dan *Background Estimation*. Penelitian menggunakan video *viptraffic .avi* yang berdurasi 8 detik, dimana kedua algoritma mengekstraksi data berbentuk citra *frame .png*.

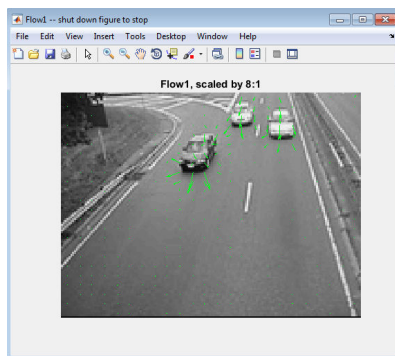


Gambar 1. Tahapan umum pendeteksian objek bergerak [6]

Pendekatan secara umum yang dilakukan untuk pendeteksian objek adalah menggunakan informasi dari setiap frame, dimana beberapa metode pendeteksian objek menggunakan informasi temporal yang didapat dari perhitungan urutan frame dengan tujuan mengurangi kesalahan deteksi (deteksi palsu). Informasi temporal biasanya diperoleh dari perbedaan frame, dimana memilih dan menandai area yang berubah secara dinamis. Proses tracking dijalankan untuk melacak korespondensi objek dari satu frame ke frame berikutnya yang menghasilkan jalur. Tahapan pendeteksian objek bergerak secara umum dapat dilihat pada Gambar 1[6].

Optical Flow

Pendeteksian objek bergerak dengan *Optical Flow* dilakukan dengan merepresentasikan gerakan kedalam aliran vektor, kemudian menghitung nilai koordinat centroid untuk menghitung kecepatan kendaraan yang bergerak. Pemilihan nilai densitas vektor dan selisih frame akan mempengaruhi citra frame yang dihasilkan. Gambar 2. menunjukkan algoritma optical flow bekerja dalam pendeteksian objek bergerak.



Gambar 2. Optical Flow dengan Viptraffi.avi

Optical flow bekerja untuk menemukan sebuah *template* gambar $T(x)$ didalam sebuah gambar $I(x)$ dimana $x = (x,y)^z$ adalah vektor dari koordinat piksel pada citra. *Optical flow* digunakan untuk melacak gambar dari $t=1$ sampai $t=2$, *template* $T(x)$ merupakan subbagian yang di ekstrak (misal: ukuran gambar 479 x 277 piksel) dari gambar $t=1$ dan $I(x)$

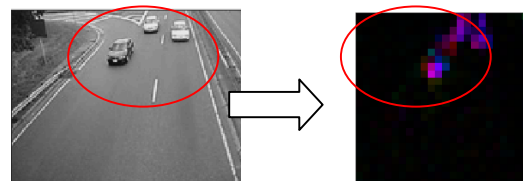
adalah gambar di $t=2$. Hasil algoritma disimpan kedalam vektor u dan v sejumlah 8 bit.

Berikut pseudocode untuk proses dengan *optical flow*

```

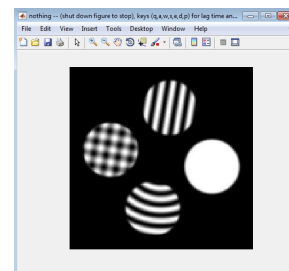
tInt = in.tIntegration;
[dx, dy, dt] =
grad3Drec(imNew,imPrev,sqrt(tInt),dx, dy, dt);
TC = single(5^2);
m = 550;
gam = 0.2;
nor = (dx.^2+dy.^2 + dt.^2 + m*(6*gam)^2+eps)/m;
    
```

Proses menganalisis vektor gerakan dengan *optical flow* pada viptraffi.avi dilakukan dengan menentukan bentuk ouput dari vektor gerakan *optical flow* kemudian menentukan delay frame referensi menjadi n frame sebelumnya sampai pada memplot pada video vektor gerakan. Setelah vektor gerakan berhasil diplot pada viptraffi.avi, maka dilakukan threshold sehingga menghasilkan Gambar 3



Gambar 3. Contoh objek terdeteksi dengan *Optical Flow*

Setelah hasil rekaman video melewati proses threshold, maka rekaman akan dianalisa melalui gumpalan-gumpalan (blob) untuk menentukan objek kendaraan yang lewat untuk dihitung jumlahnya



Gambar 4. Contoh proses rekaman dengan blob

Background Estimation

Sebuah *background* adalah objek statis dimana didalamnya terdapat objek-objek yang tidak bergerak. Citra dengan objek yang bergerak disebut *foreground*. Pada beberapa implementasi pemisahan antara *background* dan *foreground* dibutuhkan beberapa tahapan pemrosesan (*preprocessing*) untuk menghindari kesalahan dalam menghasilkan citra *foreground*. Beberapa perubahan yang mungkin terjadi pada *background* adalah perubahan intensitas cahaya dari siang hari ke sore hari, perubahan bayangan benda yang terdapat pada *background* yang diakibatkan oleh perubahan posisi matahari, perubahan posisi benda pada *background*, penambahan benda dalam *background*, dan sebagainya.

Tabel 1.
Hasil Nilai Rata-rata Keabuan Citra Frame RGB

Citra Ke -	Mean Gray Level	MeanRed Level	MeanGreen Level	Mean Blue Level
21	131.665	127.2494	134.239	129.936
22	131.697	127.276	134.279	129.955
23	131.695	127.273	134.273	129.970
24	131.756	127.352	134.325	130.028
25	131.845	127.457	134.412	130.061
26	131.819	127.416	134.406	129.997
27	131.971	127.613	134.528	130.152
28	131.977	127.615	134.538	130.147
29	132.107	127.688	134.691	130.316
30	126.737	122.129	129.377	125.185

Untuk mengurangi kesalahan yang terjadi karena perubahan *background* maka diperlukan suatu metode untuk menyesuaikan perubahan-perubahan yang terjadi pada

background, salah satunya adalah *background estimation*. Algoritma *Background Estimation* menghasilkan 198 citra frame yang menghasilkan nilai rata-rata level keabuan untuk citra frame RGB seperti dapat dilihat pada Tabel 1. Nilai rata-rata RGB Level 10 citra frame dari total 198 citra frame yang dihasilkan metode *Background Estimation*.



Gambar 5. Contoh Citra Frame

Metode *background estimation* dilakukan dengan merata-rata gabungan nilai piksel dari semua input yang berurutan, namun metode ini tidak dapat berjalan dengan baik jika diberi input yang memiliki banyak objek bergerak. Citraframe hasil ekstraksi mempunyai ruang warna RGB. Setiap *citra frame* akan masuk proses inialisasi. Pada proses inialisasi citra frame masukan diekstrak menjadi 3 bagian citra antara lain citra *Red*, Citra *Green* dan Citra *Blue*. Sebelum proses inisialisasi dilakukan, harus ditentukan terlebih dahulu citra yang akan model *background*. Citra frame pertama akan digunakan sebagai model *background* awal. Tahap pra-proses ini dilakukan sebagai persiapan dalam pendeteksian *foreground* menggunakan *Background Estimation*. Berikut pseudocode untuk proses inialisasi citra frame

```
meanRedLevels(frame) = mean(mean(thisFrame(:, :, 1)));
meanGreenLevels(frame) = mean(mean(thisFrame(:, :, 2)));
meanBlueLevels(frame) = mean(mean(thisFrame(:, :, 3)));
```

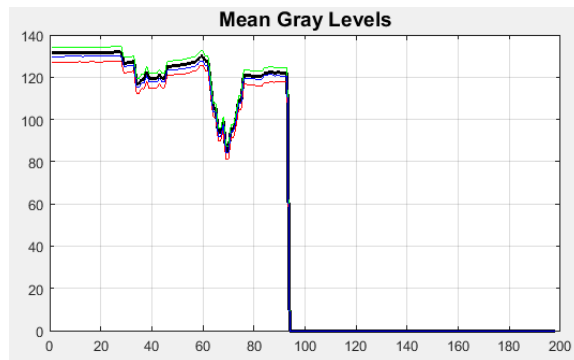
Nilai piksel pada citra hasil pengurangan akan dibandingkan dengan nilai threshold (t) yang telah ditetapkan. Piksel akan masuk kategori pencocokan yang terbaik jika nilainya lebih kecil atau sama dengan nilai threshold dan akan ditetapkan sebagai background dan diberi nilai 0, sedang kondisi lainnya akan masuk sebagai foreground diberi nilai 1. Berikut adalah

psudocode untuk perubahan background pada setiap frame

```

alpha = 0.5;
if frame == 1
    Background = thisFrame;
else
    Background = (1-alpha)* thisFrame + alpha * Background;
    
```

Gambar 6. menunjukkan Rata-rata level keabuan untuk citra frame yang diekstraksi menjadi 3 bagian citra antara lain citra *Red*, Citra *Green* dan Citra *Blue*, Frame Rate=25, bit/Piksel = 24.



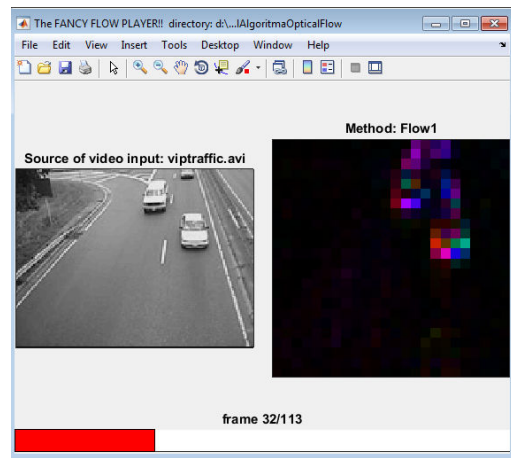
Gambar 6. Mean Gray Level dengan *Background Estimation*

Hasil dan Pembahasan

Hasil Pendeteksian Objek dengan *Optical Flow*

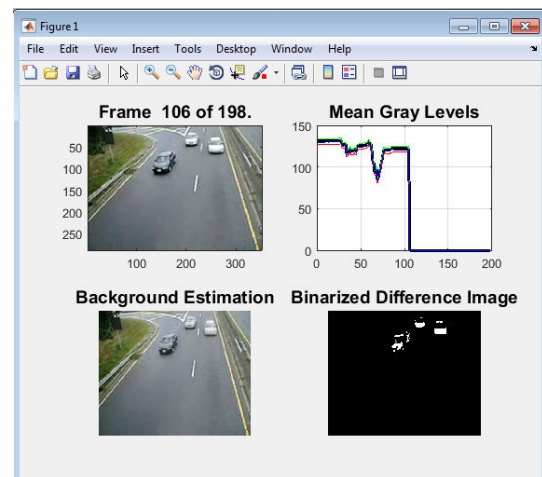
Hasil pendeteksian objek dengan algoritma *Optical flow* menghasilkan 113 citra frame seperti dapat dilihat pada Gambar 7. Algoritma *Background Estimation* menggunakan Σ - Δ filter yang merupakan operator non linear rekursif, dimana operator ini memiliki kemampuan mengestimasi dua orde dari temporal statistik pada setiap piksel dalam suatu frame, seperti dapat dilihat pada Gambar 8.

Data keluarannya akan menghasilkan 198 citra frame yang mempunyai deteksi perubahan piksel dengan level yang efisien dan sederhana. Secara umum algoritma ini berjalan baik pada *viptraffic* dimana video didapatkan dari hasil kamera. Keuntungan yang didapat dari algoritma ini adalah tingkat komputasi yang sederhana.

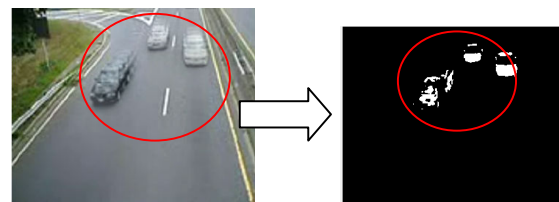


Gambar 7. Hasil algoritma *Optical Flow*

Hasil Pendeteksian Objek dengan *Background Estimation*



Gambar 8. Hasil Output *Background Estimation*



Gambar 9. Contoh objek terdeteksi dengan Nilai Threshold level 0.00784

Tabel 2 Perbandingan Algoritma *Optical Flow* dan *Background Estimation*

Perbedaan	Algoritma	
	Optical Flow	Background Estimation
Proses Komputasi	Kompleks. Proses terdiri dari beberapa parameter dan nilai parameter berubah-ubah tergantung pada vektornya	Penggunaan Σ - Δ filter yang merupakan operator non linear rekursif merupakan komputasi sederhana
Kecepatan menangkap object	Cepat. dimana algoritma melakukan analisis satu persatu pada arus lancar sehingga tidak membebani sistem sehingga dapat diproses lebih cepat	Lambat
Citra Frame yang dihasilkan	113 Citra Frame	198 Citra Frame
Waktu yang dibutuhkan untuk menghasilkan citra Frame/Detik	14 Citra Frame/Detik	25 Citra Frame/Detik
Delay Frame	Semakin besar delay frame, bentuk kendaraan yang tersegmentasi memiliki gumpalan yang lebih tebal sesuai dengan besarnya frame sehingga hasil citra frame lebih bagus	Perubahan pencahayaan akan mempengaruhi proses pendeteksian objek
Tingkat Akurasi	Akurasi baik. Algoritma dapat menghitung secara akurat ketika arus kendaraan dalam keadaan padat atau lancar	Metode ini tidak dapat berjalan dengan baik jika diberi input yang memiliki banyak obyek bergerak. Pada saat sebuah kendaraan dengan bayangannya bergerak, maka piksel yang bukan termasuk dalam <i>background</i> dapat dikategorikan sebagai bagian dari kendaraan dan bayangannya, jika tidak maka bagian <i>foreground</i> tersebut dapat dikatakan salah.

Dari hasil ujicoba Gambar 9 terlihat bahwa perlu adanya pengaturan nilai *Threshold*

untuk mendapatkan hasil yang maksimal. Nilai parameter tersebut tidak boleh terlalu tinggi

atau terlalu rendah, tergantung pada kondisi dari objek yang ingin direpresentasikan. Secara umum perbandingan kedua metode dalam melakukan pendeteksian objek bergerak pada *viptraffic.avi* dapat dilihat pada Tabel 2.

Penutup

Pada penelitian analisis dengan dua algoritma Optical Flow dan Background Estimation untuk pendeteksian objek pada *viptraffic.avi* untuk tingkat akurasi, Optical Flow memberikan hasil yang lebih baik, dimana algoritma dapat menghitung secara akurat ketika arus kendaraan dalam keadaan padat atau lancar sedangkan *Background Estimation* bekerja kurang akurat terutama pada saat sebuah kendaraan dengan bayangannya bergerak, maka piksel yang bukan termasuk dalam *background* dapat dikategorikan sebagai bagian dari kendaraan dan bayangannya, jika tidak maka bagian *foreground* tersebut dapat dikatakan salah. Proses menangkap objek bergerak lebih cepat dilakukan oleh *Optical Flow* dengan jumlah citra frame yang dihasilkan 14 Citra Frame/Detik sedangkan *Background Estimation* menghasilkan 25 Citra Frame/Detik. Semakin besar delay frame, bentuk kendaraan yang tersegmentasi memiliki gumpalan yang lebih tebal sesuai dengan besarnya frame sehingga hasil citra frame lebih bagus pada *Optical Flow*, sedangkan perubahan pencahayaan akan mempengaruhi proses pendeteksian objek dengan metode *Background Estimation*

Daftar Pustaka

- [1] Badan Penelitian dan Pengembangan Kementrian Pekerjaan Umum. PLATO: Penghitung Lalu Lintas Otomatis. <http://balitbang.pu.go.id/saritek/saritek%20jatan/6.plato.pdf>
- [2] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.
- [3] Soma Datta, Debotosh Bhattacharjee and Prमित Ghosh, Path Detection of a Moving Object. *International Journal of Recent Trends in Engineering*, Vol 2, No. 3, 2009
- [4] Gagandeep Kaur. Detection of Moving Objects in Colour based and Graph's axis Change method. *International Journal of Computing & Business Research*. ISSN (Online): 2229-6166. 2012
- [5] L. Maddalena and A. Petrosino, "A selforganizing approach to background subtraction for visual surveillance applications.," *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 17, no. 7, pp. 1168–77, Jul. 2008.
- [6]. K. Joshi and D. Thakore, "A Survey on Moving Object Detection and Tracking in Video Surveillance System," *International Journal of Soft Computing*, no. 3, pp. 44–48, 2012.

-

Halaman ini sengaja dikosongkan