

Penerapan Teknik *Fast Motion Estimation* Pada Algoritma *Frame Rate Up-Conversion* Video

Ni Putu Widya Yuniari, I Made Oka Widyantara

Magister Teknik Elektro

Universitas Udayana

Denpasar-Bali, Indonesia

widyayuniari2010@gmail.com, oka.widyantara@unud.ac.id

Abstrak— *Video surveillance* merupakan salah satu aplikasi yang sering dimanfaatkan pada sistem keamanan yakni dengan menempatkan satu atau beberapa kamera untuk memantau suatu ruangan kemudian ditransmisikan ke dekoder. Pada jaringan yang memiliki keterbatasan bandwidth transmisi, untuk tetap memperoleh video dengan resolusi tinggi terdapat suatu mekanisme yakni pada sisi enkoder harus menurunkan frame rate dengan mengirimkan sebagian dari urutan video asli. Sedangkan di dekoder harus mengupgrade kembali frame rate tersebut dengan suatu teknik yang dinamakan *Frame Rate Up-Conversion* (FRUC). FRUC merupakan suatu teknik menyisipkan sebuah frame yang baru (*intermediate*) ke dalam suatu urutan video asli untuk meningkatkan frame rate. Salah satu teknik FRUC adalah *Motion Compensation Interpolation* (MCI). Teknik ini menyandarkan pada suatu proses yang disebut *Motion Estimation* (ME). Proses ME untuk algoritma MCI pada FRUC yakni memprediksikan posisi frame berupa *Motion Vector* (MV) yang sekarang berdasarkan pada frame referensi. MV yang telah diperoleh disimpan di dekoder sebagai informasi yang digunakan untuk penyisipan frame *intermediate*. Terdapat suatu metode untuk mempercepat proses ME dengan mengurangi jumlah blok kandidat yang disebut *Fast Search*. Salah satu metode *Fast Search* adalah *Three Step Search* (TSS). Dengan mengadopsi pengkodean video berbasis blok, pada paper ini mengajukan penerapan teknik *Fast Search* TSS untuk ME pada algoritma FRUC. Sasarannya adalah menurunkan kompleksitas di enkoder dengan mengurangi jumlah kandidat blok pencarian dan untuk memberikan informasi MV frame *intermediate* yang akan disisipkan pada dekoder sehingga mampu meningkatkan frame rate.

Kata kunci: *Motion Compensation Interpolation*, *Motion Estimation*, *Fast Search*, *Three Step Search*.

I. Pendahuluan

Perkembangan teknologi berbasis digital video saat ini cukup pesat. Salah satu pemanfaatan teknologi digital video yang telah banyak diterapkan adalah pada sistem keamanan ruang. *Video surveillance* merupakan salah satu aplikasi yang sering dimanfaatkan pada sistem keamanan yakni dengan menempatkan satu atau beberapa kamera untuk memantau suatu ruangan kemudian ditransmisikan ke dekoder. Pada jaringan yang memiliki keterbatasan bandwidth transmisi, untuk tetap memperoleh video dengan resolusi tinggi maka terdapat suatu mekanisme yakni pada sisi enkoder harus menurunkan frame rate dengan mengirimkan sebagian dari

urutan video asli. Sedangkan di dekoder harus mengupgrade kembali frame rate tersebut dengan suatu teknik yang dinamakan *Frame Rate Up-Conversion* (FRUC).

FRUC merupakan suatu teknik menyisipkan sebuah frame yang baru (*intermediate*) ke dalam suatu urutan video asli untuk meningkatkan frame rate. Aplikasi ini digunakan dalam pengolahan sinyal video untuk mengatasi keterbatasan bandwidth transmisi untuk meningkatkan resolusi temporal. Secara umum algoritma FRUC terbagi menjadi dua kategori. Kategori pertama tidak mempertimbangkan informasi dari pergerakan objek yakni dengan metode *Frame Repetition* dan *Frame Averaging*. Proses dari metode ini mudah diimplementasikan namun memberikan kualitas visual yang rendah seperti kabur dan *artifact* untuk objek yang bergerak. Kategori kedua dengan mempertimbangkan informasi dari pergerakan video yakni dengan metode *Motion Compensation Interpolation* (MCI). Teknik MCI menyandarkan pada suatu proses yang disebut *Motion Estimation* (ME). Proses ME untuk algoritma MCI pada FRUC yakni memprediksikan posisi frame berupa *Motion Vector* (MV) yang sekarang berdasarkan pada frame referensi. MV yang telah diperoleh sebagai informasi yang digunakan untuk frame *intermediate*.

Terdapat algoritma ME yang diterapkan untuk FRUC [1-3]. Metode ME ini dilakukan hanya berdasarkan pada arah mundur yakni menggunakan frame sebelumnya sebagai frame referensi dengan menyalin MV dari frame yang sebelumnya menjadi MV untuk frame *intermediate*. Akan tetapi metode ini memberikan pergerakan video yang buruk. Untuk mengatasi hal ini maka terdapat suatu metode yang menerapkan teknik kombinasi linier yakni menggunakan frame referensi pada dua sisi yakni frame sebelum (arah mundur) dan selanjutnya (arah maju) untuk informasi frame *intermediate*. Choi et al [4] mengusulkan metode MCI menggunakan teknik *Bilateral* ME. Namun teknik ini memiliki kompleksitas yang tinggi. Untuk mengurangi kompleksitas, Zhang et al [5] mengusulkan metode ME *Unidirectional* untuk algoritma FRUC.

Algoritma [1-5] menggunakan metode pencarian ME *Full Search*. Dengan mengadopsi pengkodean video berbasis blok, metode *Full Search* mencari sebuah kandidat blok terbaik pada semua posisi blok di frame referensi dengan jendela pencarian yang ditetapkan [6]. Terdapat sebuah argumentasi bahwa ME dengan metode pencarian *Full Search* ini membutuhkan proses pencarian kandidat blok terbaik dengan

waktu yang lama sehingga menyebabkan kompleksitas enkoder video menjadi lebih tinggi. Suatu metode yang dapat digunakan untuk mengurangi jumlah kandidat blok untuk proses ME disebut *Fast Search*. Salah satu metode *Fast Search* adalah *Three Step Search* (TSS). TSS melakukan proses pencarian sebuah kandidat blok terbaik melalui tiga langkah. Pada paper ini mengajukan penerapan teknik TSS untuk ME pada algoritma FRUC. Sasarannya adalah menurunkan kompleksitas di enkoder dengan mengurangi jumlah kandidat blok pencarian dan untuk memberikan informasi MV frame *intermediate* yang akan disisipkan pada dekoder sehingga mampu meningkatkan frame rate. Untuk saat ini penelitian masih dalam progress.

II. Teknik *Motion Estimation* Fruc Dengan Mci

Metode FRUC dengan teknik MCI menggunakan 2 frame yang telah didekodekan sebelumnya sebagai frame referensi yakni frame sebelum dan sesudahnya. Terdapat beberapa teknik MCI yang digunakan pada algoritma FRUC yakni sebagai berikut:

A. *Unidirectional ME*

Dari 2 buah frame yang telah didekodekan yakni frame f_{t-1} dan f_{t+1} maka untuk mencari frame *intermediate* sebagai frame interpolasi f_t dapat dimodelkan sebagai berikut:

$$f_t(x,y) = \frac{1}{2} f_{t-1}(x - \frac{1}{2}x, y - \frac{1}{2}y) + f_{t+1}(x + \frac{1}{2}x, y + \frac{1}{2}y) \quad (1)$$

Dimana (x,y) dan (x, y) adalah posisi pixel dan unidirectional MV masing-masing dari arah x dan y. x dan y dicari berdasarkan metode pencarian *Full Search*.

MV terbaik untuk frame *intermediate* adalah nilai rata-rata terkecil dari pengurangan sisi maju dan mundur. Maka,

$$v = (x, y) = \arg \min_{\Delta x, \Delta y} (f_{t-1}, f_{t+1}; x, y) \quad (2)$$

Nilai adalah fungsi jarak yang tepat. Untuk estimasi gerak *unidirectional* nilai dari arah maju dapat didefinisikan sebagai berikut:

$$(f_{t-1}, f_{t+1}; x, y) = \sum_{x,y \in G} |f_{t-1}(x,y) - f_{t+1}(x+x, y+y)| \quad (3)$$

untuk arah mundur dapat didefinisikan sebagai berikut:

$$(f_{t-1}, f_{t+1}; x, y) = \sum_{x,y \in G} |f_{t+1}(x,y) - f_{t-1}(x+x, y+y)| \quad (4)$$

Nilai G adalah grup dari pixel, biasanya adalah ukuran blok dari sebuah gambar. Pada usulan ini menggunakan blok 8×8 .

Pada sisi tepi biasanya memiliki informasi yang akurat sehingga mudah diketahui oleh persepsi visual. Untuk meningkatkan kualitas ME dan mempertajam sisi tepi maka nilai didefinisikan sebagai berikut:

$$(f_{t-1}, f_{t+1}; x, y) = \sum_{x,y \in G} |f_{t-1}(x,y) - f_{t+1}(x-x, y-y)| + |D f_{t+1}(x,y) - (D f_{t-1})(x-x, y-y)| \quad (5)$$

D adalah sebuah high pass filter, $D f_{t-1}$ menunjukkan konten frekuensi tinggi dari f_{t-1} . adalah factor pembobotan yang dapat menyeimbangkan pengaruh dari dua perbedaan bentuk pada persamaan 5. [5] menggunakan nilai adalah 0.3.

B. *Bilateral ME*

Pada metode *Bilateral ME*, persamaan yang digunakan untuk mencari MV frame *intermediate* dapat dicari menggunakan persamaan 1 dan 2. x dan y pada metode *Bilateral ME* juga menggunakan metode pencarian *Full Search*. Untuk nilai sebagai fungsi jarak yang tepat pada metode *Bilateral ME* dari arah maju dapat didefinisikan sebagai berikut:

$$(f_{t-1}, f_{t+1}; x, y) = \sum_{x,y \in G} |f_{t-1}(x - x/2, y - y/2) - f_{t+1}(x + x/2, y + y/2)| \quad (6)$$

Untuk arah mundur dapat didefinisikan sebagai berikut:

$$(f_{t-1}, f_{t+1}; x, y) = \sum_{x,y \in G} |f_{t+1}(x - x/2, y - y/2) - f_{t-1}(x + x/2, y + y/2)| \quad (7)$$

Secara umum, *Bilateral ME* memiliki performansi lebih baik daripada metode *Unidirectional*, tetapi proses *Bilateral ME* memiliki kompleksitas yang tinggi dibandingkan *Unidirectional ME*.

III. Teknik *Fast Search Motion Estimation* Pada MCI

3.1 Teknik *Fast Search TSS*

Teknik MCI pada FRUC yang telah disebutkan di atas yakni *Unidirectional* dan *Bilateral* menggunakan metode pencarian x dan y dengan metode *Full Search*. Dengan menggunakan *Full Search* jumlah kandidat blok n untuk blok berukuran panjang b_w dan lebar b_h dan jendela pencarian dengan panjang W dan lebar H (dimana $W > b_w$ dan $H > b_h$) dapat dicari dengan persamaan berikut:

$$n = (W - b_w + 1) * (H - b_h + 1) \quad (8)$$

Misalkan untuk b_w dan $b_h = 16$ dan $W, H = 30$. Maka jumlah kandidat blok n adalah 225 blok. Mencari sebuah kandidat blok terbaik keseluruhan frame adalah hal yang mungkin namun secara praktis tidak layak terutama untuk resolusi tinggi, misalkan b_w dan $b_h = 16$ dan $W = 1920$ dan $H = 1080$. Maka total terdapat 2.028.825 kandidat blok. Kandidat blok terbaik (*best match*) di peroleh dari nilai *Mean Square Error* (MSE) terkecil yang dihitung untuk setiap kandidat blok [7] menggunakan persamaan berikut:

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (9)$$

Hal ini tentu membutuhkan waktu yang lama dan komputasi yang berat. Terdapat suatu metode untuk mempercepat proses ME dengan mengurangi jumlah kandidat blok untuk algoritma FRUC yakni *Fast Search*, salah satunya adalah algoritma TSS.

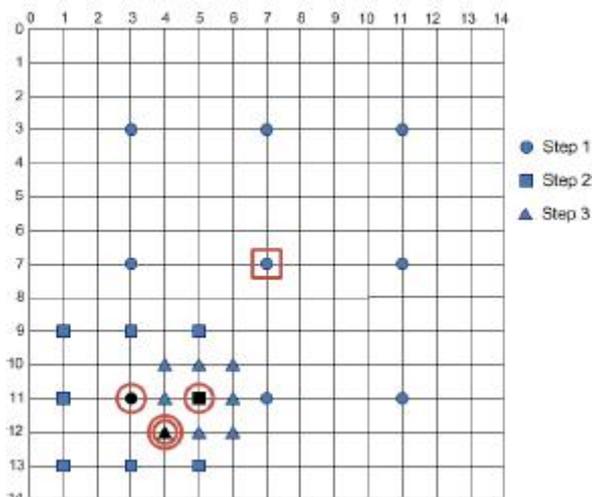
Metode TSS merupakan salah satu metode pertama untuk teknik pencocokan blok [8] untuk mengurangi proses dari *Full Search*. TSS melakukan pencarian untuk mendapatkan sebuah kandidat blok terbaik melalui tiga langkah secara iterative, yakni :

1. Langkah I: Cocokkan dengan semua posisi termasuk dengan posisi tengah pada jarak sejauh 4 pixel dari posisi tengah. Pada langkah ini melibatkan 9 blok pencarian, $s_1=9$.
2. Langkah II: Pusatkan wilayah pencarian sekitar kandidat blok terbaik dari langkah I. Cocokkan semua pixel pada jarak sejauh 2 pixel dari posisi tengah. Pada langkah ini melibatkan 8 blok pencarian, $s_2=8$.
3. Langkah III: Pusatkan wilayah pencarian sekitar kandidat blok terbaik dari langkah II. Cocokkan semua pixel pada jarak sejauh 1 pixel dari posisi tengah. Pada langkah ini melibatkan 8 blok pencarian, $s_3=8$. Kandidat terbaik ditemukan pada langkah ini.

Pada algoritma TSS hanya melibatkan pencarian terhadap 25 kandidat blok. Proses TSS dapat dilihat pada gambar 1.

$$n=s_1+s_2+s_3=25$$

Jika b_w dan b_h yang digunakan adalah 16 dan W,H adalah 30. Maka TSS menunjukkan kompleksitas 9x lebih rendah dibandingkan proses *Full Search*.



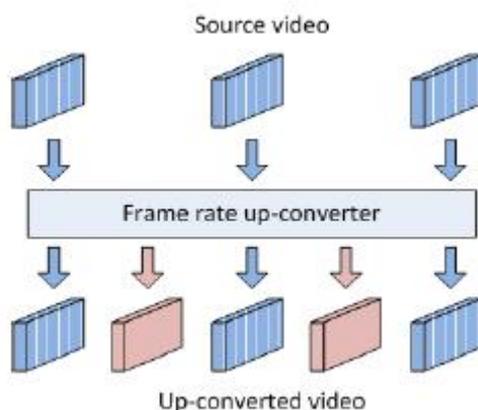
Gambar 1 Ilustrasi Algoritma TSS [8]

3.2 Penerapan Metode TSS Pada *Unidirectional* ME

Teknik pencarian MV yang diusulkan untuk algoritma FRUC pada penelitian ini adalah *Fast Search* yakni menggunakan metode TSS. MV frame *intermediate* dicari menggunakan langkah-langkah pada persamaan *Unidirectional* ME yang telah disebutkan pada persamaan (1)-(5) di atas. Namun untuk x dan y dicari menggunakan metode TSS yang telah jelaskan pada sub bab 3.1 di atas.

IV. Model Implementasi *Fast Search* ME Pada FRUC

Untuk analisis metode yang diusulkan dalam penelitian ini menerapkan model transmisi video pada aplikasi *video surveillance*. Dari gambar 2 dapat dilihat bahwa enkoder mengirimkan video dengan frame rate yang rendah dengan hanya mengirimkan sebagian dari urutan video asli dengan mendrop beberapa frame video (frame *intermediate*). Teknik *Fast Search* yakni TSS untuk proses ME diimplementasikan untuk memprediksikan MV dari frame *intermediate* menggunakan frame sebelum dan sesudahnya sebagai frame referensi. Pada dekoder harus melakukan proses FRUC dengan menyisipkan kembali frame yang di drop di enkoder ke dalam urutan video asli sehingga frame rate meningkat menggunakan informasi MV dari frame *intermediate* yang diperoleh dari teknik TSS



Gambar 2 Ilustrasi Algoritma FRUC [6]

Untuk evaluasi kualitas secara objective dari frame *intermediate* yang disisipkan di sisi dekoder digunakan pengukuran *Peak Signal To Noise Ratio* (PSNR) [8]. Asumsi yang digunakan adalah frame yang di drop disimpan di enkoder untuk kemudian dibandingkan dengan frame *intermediate* yang dihasilkan di dekoder. Persamaan PSNR dapat dilihat dibawah ini.

$$PSNR=10 \log_{10} \frac{(2^k - 1)^2}{MSE} \tag{10}$$

k adalah jumlah bit dalam 1 sampel pixel. Nilai PSNR ini digunakan untuk menganalisis kinerja *Fast MCI* pada model implementasi *video surveillance*.

V. KESIMPULAN

FRUC merupakan suatu teknik menyisipkan sebuah frame *intermediate* ke dalam suatu urutan video asli untuk meningkatkan frame rate. Aplikasi ini digunakan dalam pengolahan sinyal video untuk mengatasi keterbatasan

bandwidth transmisi untuk meningkatkan resolusi temporal. Salah satu teknik FRUC yakni dengan metode MCI. Teknik MCI mempertimbangkan informasi dari pergerakan video dengan menyandarkan pada suatu proses yang disebut *Motion Estimation* (ME). Proses ME untuk algoritma MCI pada FRUC yakni memprediksikan posisi frame berupa *Motion Vector* yang sekarang berdasarkan pada frame referensi. Frame referensi yang digunakan pada penelitian ini adalah dari dua sisi yakni frame sebelum (arah mundur) dan selanjutnya (arah maju). Suatu metode untuk mempercepat proses ME dengan mengurangi jumlah kandidat blok pencarian disebut *Fast Search*. Salah satu metode *Fast Search* adalah *Three Step Search* (TSS). TSS melakukan proses pencarian sebuah kandidat blok terbaik melalui tiga langkah. Pada penelitian ini mengajukan suatu teknik ME dengan metode *Fast Search* TSS pada algoritma FRUC. Sasarannya adalah menurunkan kompleksitas di enkoder dengan mengurangi jumlah kandidat blok pencarian dan untuk memberikan informasi MV frame *intermediate* yang akan disisipkan pada dekoder sehingga mampu meningkatkan frame rate. Untuk saat ini penelitian masih dalam progress.

REFERENSI

- [1] G.D. Haan, P.W.A.C. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368-379, Oct. 1993.
- [2] C.C. Cheng, W.L. Hwang, Z. Shen, and T. Xia, "Advanced motion compensation techniques for blocking artifacts reduction in 3-D video coding systems," in *Proc. ICIP*, Sep. 2005, vol. 3, pp. 89-92.
- [3] Y.K. Chen, A. Vetro, H. Sun, and S. Y. Kung, "Frame-rate up-conversion using transmitted true motion vectors," in *Proc. IEEE Workshop Multimedia Signal Process.*, Dec. 1998, vol. 2, pp. 622-627.
- [4] B.D. Choi, J.W. Han, C.S. Kim, and S. J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp.407-416, Apr. 2007.
- [5] L. Zhang, C. Wang, W. Zhang, and Y. P. Tan, "Frame Rate Up-Conversion With Edge-Weighted Motion Estimation and Trilateral Interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 886-893, June. 2010.
- [6] A. Isberg, J. Jostell, "Frame Rate Up-Conversion Using High Definition Remote Video Surveillance," Technical Report," Dept. Of Computer Science and Engineering, Chalmers University Of Technology Sweden, Apr. 2012.
- [7] A. Barjatya, Block Matching Algorithms For Motion Estimation, "Technical Report," ECE Department at Utah State University. 2004.
- [8] T. Koga, K. Linuma, A. Hirano, Y, and T. Ishigiro, "Motion Compensated Interframe Coding For Video Conferencing," in *Proc. National Telecommunication Conference*," pp 531-535, 1981.