

# PENJEJAK OBJEK DENGAN BEBERAPA KAMERA MENGUNAKAN ALGORITMA *THREE STEP SEARCH*

Aghus Sofwan \*)

## Abstract

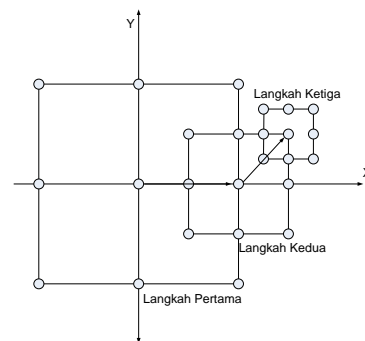
In this paper we address the development of an object tracking system with multiple fixed cameras using motion estimation. Three step search algorithm is used to estimate object movement that across camera. This algorithm will process two frame  $f(t-1)$  and  $f(t)$ , where  $f(t-1)$  represents reference image, and  $f(t)$  represents current image at one camera to produce motion vectors value. Sequence of images that arrive from cameras will be analyzed, then motion vectors and number of moving blocks are obtained. The motion vectors imply that object is moving to right, left, up, or down. Multiple cameras are placed as nodes in a map area having cardinal coordinate. View direction of each camera according cardinal coordinate is whether south, east, north, or west. Based on motion vectors value and number of blocks that are moving from each camera, our algorithm will respond by moving the view from one camera to another, tracking objects that cross the area covered by cameras.

*Key words* : object tracking, three step algorithm

## Pendahuluan

Seiring perkembangan teknologi informasi khususnya dalam bidang pengolahan citra, penelitian tentang algoritma perkiraan gerakan banyak dilakukan[1][2][3]. Pada tulisan ini memberikan hasil penelitian penjejakan objek yang menerapkan algoritma dalam bidang tersebut. Perkiraan gerakan adalah sebuah proses untuk menentukan *motion vector* yang menggambarkan transformasi sebuah bentuk citra 2 dimensi ke citra yang lain. Banyak algoritma perkiraan gerak yang diusulkan oleh para peneliti dalam bidang ini. Algoritma-algoritma tersebut didasarkan pada pembagian frame menjadi blok-blok, dalam hal ini disebut blok referensi. Setelah itu blok-blok tersebut akan dibandingkan dengan blok dengan ukuran yang identik, dalam hal ini disebut blok kandidat pada frame berikutnya. Dari perbandingan blok-blok dalam 2 frame tersebut akan diperoleh nilai-nilai *motion vector*. Pada penelitian ini algoritma yang digunakan untuk perkiraan gerakan adalah algoritma *three step*. Gambar 1 memperlihatkan tahapan dari algoritma tersebut. Metoda ini digunakan karena kesederhanaannya, *robust*, mudah diimplementasikan, dan mempunyai unjuk kerja yang mendekati optimal[4].

*Motion vector* diindikasikan dengan nilai-nilai x dan y. Dengan asumsi bahwa sumbu x sejajar dan sumbu y tegak lurus terhadap permukaan bumi, sumbu x menyatakan arah ke kanan atau ke kiri, sedangkan sumbu y menyatakan arah ke atas atau bawah. Perkiraan gerakan menggunakan nilai *motion vector* sebagai salah satu parameter yang digunakan untuk menjejaki objek yang bergerak. Perkiraan ini mengindikasikan arah pergerakan objek tersebut. Parameter lain yang digunakan untuk mempertegas pergerakan objek adalah jumlah blok dalam frame yang bergerak.



Gambar 1. Algoritma *Three Steps Search*

Terdapat beberapa kamera tetap yang digunakan untuk mencuplik citra. *Motion vector* dari citra-citra yang berasal dari sebuah kamera akan dihitung, kemudian hasilnya akan menentukan kamera berikutnya yang akan diambil citranya untuk dianalisis. Bila nilai *motion vector* pada sumbu x adalah positif maka kamera berikutnya yang akan diambil citranya berada di sebelah kanan dari kamera sekarang, sebaliknya bila nilai *motion vector* pada sumbu x adalah negatif maka citra akan diambil dari kamera yang berada di sebelah kiri kamera sekarang.

Kamera-kamera diletakkan pada posisi tertentu sebagai simpul-simpul dalam peta yang didasarkan pada koordinat kardinal. Arah pandang kamera adalah Selatan, Timur, Utara, atau Barat. Bila arah pandang kamera Utara dan pergerakan objek adalah ke Barat, maka algoritma akan mengambil deretan citra berikutnya dari kamera yang berada di sebelah kiri kamera sekarang. Oleh karena itu penempatan dan arah pandangan tiap kamera akan menentukan asal dari deretan citra berikutnya yang akan diproses. Bila tidak ada pergerakan objek yang ditangkap oleh kamera, algoritma akan memeriksa semua kamera secara berurutan hingga menemukan objek bergerak.

\*) Staf Pengajar Jurusan Teknik Elektro Fakultas Teknik Undip

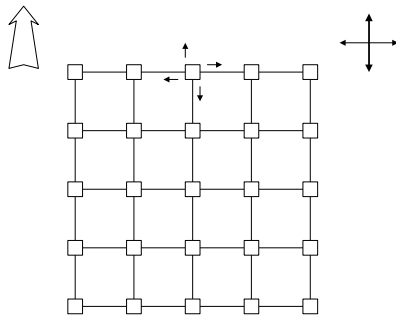
## Metodologi Penelitian

Metodologi penelitian yang digunakan adalah dengan menerapkan algoritma dari sistem yang dikembangkan. Secara umum algoritma terdiri dari 4 tahap. Pada bagian ini diberikan gambaran rinci dari algoritma yang digunakan tersebut.

### 1. Menentukan Posisi dan Arah Pandang Kamera

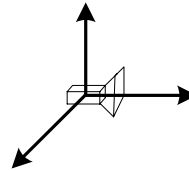
Penempatan kamera sebagai satu simpul akan menentukan arah pergerakan objek secara nyata. Nilai nyata dari *motion vector* ditentukan dengan persyaratan ini. Selain itu juga koordinat kardinal digunakan sebagai dasar arah riil dari pergerakan objek. Posisi atas pada area peta adalah Utara, bawah adalah Selatan, kanan adalah Timur, dan kiri adalah Barat, seperti diperlihatkan pada Gambar 2. Koordinat dari tiap-tiap simpul dinyatakan dengan  $(x,z)$ . Koordinat kamera di posisi atas kiri adalah  $(x1,z1)$ , kamera di sebelah kanannya adalah  $(x2,z1)$ , dan seterusnya.

Arah pandang kamera adalah sejajar dengan permukaan tanah. Dengan asumsi bahwa sebuah objek bergerak hanya secara horisontal, maka dapat hanya menggunakan *motion vector* pada arah sumbu x untuk memperkirakan sebuah objek bergerak ke kanan atau ke kiri.



Gambar 2. Koordinat simpul dinyatakan dalam  $(x,z)$

Gambar 3 memperlihatkan tiga dimensi dari arah pandang kamera pada suatu posisi. Dari gambar tersebut terdapat 3 sumbu, yaitu sumbu x, sumbu z, dan sumbu y. Sumbu x dan sumbu z paralel terhadap bidang horisontal, sedangkan sumbu y tegak lurus terhadap permukaan bumi. Dari Gambar 2 dan 3 dapat digunakan untuk menentukan arah pergerakan objek. Gambar 3 mengilustrasikan arah pandang kamera adalah Utara, oleh sebab itu sumbu z paralel dengan sumbu  $z'$  dan sumbu x paralel dengan sumbu  $x'$ .



Gambar 3. Ilustrasi arah pandang dari kamera

### 2. Memperoleh *Motion Vector* dan Jumlah Blok

Tahapan ini menggunakan algoritma *three step search* untuk memproses 2 frame  $f(t)$  dan  $f(t-1)$ , dimana  $f(t-1)$  menyatakan gambar referensi, dan  $f(t)$  menyatakan gambar sekarang. Tiap frame dibagi menjadi sekelompok blok. Tiap blok mempunyai 4 parameter yaitu :

$(x,y)$  : indeks blok dalam frame

$B\_DX$ =lebar Blok

$B\_DY$ =tinggi Blok

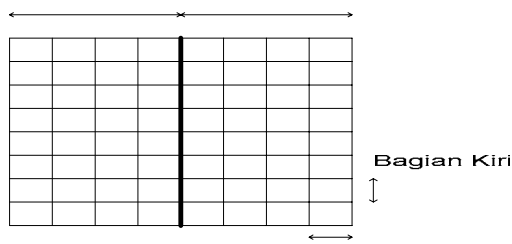
Dengan demikian tiap frame terdiri atas sejumlah maksimum blok kolom dan blok baris. Nilai-nilai tersebut diperoleh dengan membagi lebar frame terhadap lebar blok dan tinggi frame terhadap tinggi blok masing-masing.

Tiap blok pada frame  $(t-1)$  dimungkinkan mempunyai sebuah blok yang sama pada frame  $(t)$ . Perbedaan indeks antara 2 blok akan memunculkan nilai pergerakan. Vektor-vektor tersebut disimpan sebagai sebuah *motion vector*. Array dari *motion vector* terdiri atas *motion vector* dari semua blok [1]. Penentuan perbedaan sebuah blok dilakukan dengan menghitung nilai Minimum Average Distortion (MAD). Pada langkah pertama dan kedua ditentukan nilai  $MINTHRESHOLD$ , sedangkan pada langkah ketiga ditambahkan nilai  $MAXTHRESHOLD$ .

Nilai *motion vector* pada sumbu x dan sumbu y diperoleh masing-masing dari nilai  $DV_x$  dan  $DV_y$ . Bila  $DV_x$  sama dengan nol mengindikasikan blok kolom tidak bergerak dan  $DV_y$  sama dengan nol mengindikasikan blok baris tidak bergerak.

*Motion vector* dari tahap ini digunakan untuk menentukan arah pergerakan objek. Algoritma hanya menggunakan nilai *motion vector* pada sumbu x untuk menentukan pergerakan ke arah sisi kanan atau sisi kiri. Nilai ini diperoleh dari *motion vector* dari semua blok dibagi dengan jumlah blok yang bergerak.

Jumlah blok yang bergerak dihitung untuk memperkuat keputusan tentang pergerakan objek, karena dimungkinkan kamera menangkap lebih dari sebuah objek dalam sebuah frame, misalnya terdapat seorang manusia dan kucing yang melewati pandangan kamera, algoritma harus memutuskan objek mana yang harus diikuti.



Gambar 4. Pembagian frame

Frame yang merepresentasikan *motion vector* dibagi menjadi 2 bagian, yaitu bagian kanan dan bagian kiri, Gambar 4. Algoritma menghitung *motion vector* pada masing-masing bagian. Langkah ini mempunyai tujuan untuk menangkap kecenderungan arah pergerakan sebuah objek. Untuk sebuah kasus bila objek melintasi kamera dari sisi kiri ke sisi kanan, akan menjadi mudah untuk menentukan tujuan dari gerakan objek.

### 3. Konversi *Motion Vector* Pada Sumbu X

Berpatokan pada permukaan bumi, *motion vector* pada sebuah node di sumbu x harus dikonversikan ke dalam arah mata angin. Konversi ini bergantung pada tempat dan arah pandangan kamera.

Algoritma untuk mengkonversi posisi sumbu x pada simpul yang aktif adalah sebagai berikut :

Bila arah pandangan kamera adalah Utara maka sumbu x paralel dengan sumbu x', sehingga simpul sekarang dipertimbangkan dalam deretan sumbu x dan *motion vector* adalah sama dengan arah pergerakan riil dari objek.

Bila arah pandangan kamera adalah Timur maka sumbu x paralel dengan minus sumbu z', sehingga simpul sekarang dipertimbangkan sebagai deretan dalam sumbu z dan *motion vector* tegak lurus terhadap arah pergerakan riil objek.

Bila arah pandangan kamera adalah Selatan maka sumbu x paralel dengan minus sumbu x', sehingga simpul sekarang dipertimbangkan sebagai deretan sumbu x dan *motion vector* adalah berlawanan dengan arah riil pergerakan objek.

Bila arah pandangan kamera adalah Barat maka sumbu x paralel dengan sumbu z' sehingga simpul sekarang diperimbang sebagai deretan sumbu z dan *motion vector* adalah tegak lurus berlawanan dengan arah riil pergerakan objek.

### 4. Berpindah ke Kamera Lain

Dari langkah 3 algoritma akan memeriksa nilai dari *motion vector* dan kemudian akan memutuskan untuk mengambil deretan citra dari kamera yang lainnya. Bila tidak ada objek yang bergerak yang tertangkap di kamera, algoritma akan mengambil citra dari simpul deretan berikutnya secara berurutan.

Bila ada objek yang bergerak muncul pada suatu kamera, algoritma akan mempertimbangkan tidak hanya *motion vector* pada sumbu x tetap juga jumlah blok yang bergerak melintasi tepi dari frame untuk menentukan nilai arahnya. Berdasarkan nilai arah tersebut algoritma menghitung koordinat posisi simpul sekarang dan kemudian mencari kamera berikutnya mengikuti pergerakan objek.

## Hasil dan Pembahasan

Langkah pertama algoritma adalah menentukan posisi dan arah pandangan semua kamera yang ada. Implementasi posisi dilakukan dalam koordinat x dan z, dan juga arah mata angin pandangan kamera. Tiap data disimpan dalam variabel array, yaitu XCAM[maxCam], ZCAM[maxCam], dan VCAM[maxCAM].

Perbedaan indeks antara 2 blok menyatakan *motion vector* dalam nilai DVx dan DVy. Implementasi *motion vector* dengan menggunakan class MOTIONVECTOR sebagai berikut :

```
class MOTIONVECTOR {
public : int DVx;
        int DVy;
        int NZeC;
        int NZeR;
        int bleft;
        int bright;
};
```

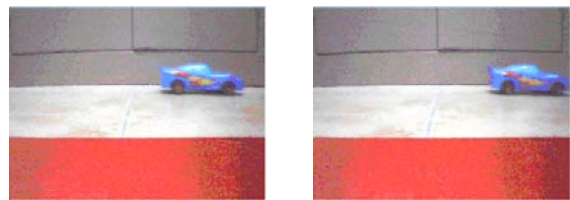
dimana DVx adalah perbedaan kolom, DVy adalah perbedaan baris, NZeC adalah jumlah blok kolom yang bergerak, NZeR adalah jumlah blok baris yang bergerak, bleft adalah jumlah blok yang bergerak melintasi tepi kiri frame, dan bright adalah jumlah blok yang bergerak melintasi tepi kanan frame.

Selain menghitung *motion vector*, algoritma juga menghitung blok kolom yang bergerak, blok baris yang bergerak, dan jumlah blok yang melintas tepi frame. Dari data-data tersebut algoritma dapat menentukan arah dari pergerakan objek. Variabel arah sama dengan 1 bila kecenderungan objek bergerak adalah ke arah sisi kanan, sebaliknya akan bernilai -1 bila kecenderungan pergerakan objek adalah ke arah kiri, selain itu variabel arah sama dengan nol bila pergerakan objek masih berada dalam pandangan kamera sekarang. Nilai variabel arah ditentukan oleh nilai-nilai DVx, bleft, and bright. Adapun implementasi yang dilakukan adalah sebagai berikut :

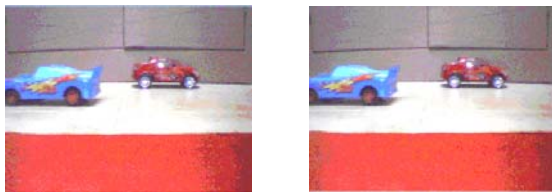
```
MOTIONVECTOR mvr;
if ((mvr.DVx < 0) && (mvr.bleft>0)) arah = -1;
if ((mvr.DVx > 0) && (mvr.bright>0)) arah = 1;
```

Algoritma menghitung arah pandangan kamera sekarang dan melakukan perhitungan iterasi indeks kamera XCAM dan ZCAM untuk menghitung posisi koordinatnya. Simpul berikut dihitung dari indeks tersebut dan

nilai arah. Nilai arah sama dengan -1 dan 1 mengindikasikan simpul berikut berada di sisi kiri dan sisi kanan.



Gambar 6. Frame-frame pada Test 2



Gambar 5. Frame-frame pada Test 1

Algoritma *three step* digunakan dalam penelitian ini dengan mengamati beberapa parameter berbeda yaitu ukuran jendela pencarian dan ukuran blok. Percobaan dilakukan di dalam ruang laboratorium. Luas ruangan yang digunakan untuk observasi adalah 2 x 3 meter persegi. Jumlah kamera yang digunakan sebanyak tiga buah kamera tetap. Kamera-kamera tersebut diletakkan sebaris, lihat Gambar 2. Koordinat kamera-kamera tersebut adalah  $(x_1, z_1)$ ,  $(x_2, z_1)$ , dan  $(x_3, z_1)$ , dengan arah pandangan kamera masing-masing adalah Selatan, Utara, dan Selatan.

Gambar 5 dan Gambar 6 memperlihatkan frame-frame pada dua test dari sejumlah observasi yang dilakukan. Masing-masing gambar sisi kiri adalah frame  $(t-1)$  dan sisi kanan adalah frame  $(t)$ .

Gambar 5 memperlihatkan dua objek yang bergerak berlawanan arah satu sama lainnya. Pada percobaan ini algoritma menghasilkan nilai-nilai untuk *motion vector*, blok kolom yang bergerak, dan jumlah blok yang melintasi tepi tiap bagian frame, dan kecenderungan arah switching ke kamera berikutnya, lihat Tabel 1.

Tabel 1. Hasil dari Test 1

B_DX	S_DX	bagian kiri frame			bagian kanan frame			arah
		DVx,Dvy	NZeC	bleft	DVx,Dvy	NZeC	bright	
8	16	-9,-3	109	18	9,-2	50	0	-1
8	8	-8,0	96	13	4,0	46	0	-1
8	4	-4,0	77	9	5,0	49	0	-1
4	16	-7,-1	338	26	7,-2	128	0	-1
4	8	-6,0	312	28	3,0	154	0	-1
4	4	-1,0	341	25	3,0	157	0	-1
2	16	-6,-1	1444	17	2,-1	628	0	-1
2	8	-5,0	1105	11	1,0	910	0	-1
2	4	-1,0	1253	55	0,0	945	0	-1

Gambar 6 menunjukkan sebuah objek yang melintasi kamera dari sisi kiri menuju kanan. Hasil dari percobaan ini dicatat dalam Tabel 2.

Tabel 2. Hasil dari Test 2

B_DX	S_DX	bagian kiri frame			bagian kanan frame			arah
		DVx,Dvy	NZeC	bleft	DVx,Dvy	NZeC	bright	
8	16	0,0	0	0	3,-1	91	2	1
8	8	0,0	0	0	9,-1	80	1	1
8	4	0,0	0	0	2,0	61	2	1
4	16	0,0	0	0	4,0	228	2	1
4	8	0,0	0	0	1,0	255	8	1
4	4	0,0	0	0	1,-1	249	11	1
2	16	0,0	0	0	1,-2	821	43	1
2	8	0,0	0	0	1,0	958	40	1
2	4	0,0	0	0	1,0	969	32	1

### Kesimpulan

Dari hasil yang ditunjukkan oleh Tabel 1 dan Tabel 2, mengindikasikan algoritma memberikan hasil keputusan yang benar. Nilai pada kolom arah memperlihatkan ke-cenderungan pergerakan objek, nilai-nilai tersebut akan mengarahkan switching kamera berikutnya, untuk menjejaki objek yang bergerak.

Dari berbagai ukuran blok yang diamati, kolom 1 dan kolom 2, terlihat tidak ada efek terhadap performa algoritma. Selama pengamatan terlihat semakin kecil ukuran blok akan semakin meningkatkan beban komputasi. Ukuran blok 16 x 16 menjadi pilihan bagus untuk efisiensi proses komputasi pada *three step search*.

### Daftar Pustaka

1. Mansour A Alzuair dan Bandar Al Rashed, (2008), "Tracking Using Motion Estimation", *Proceedings of International Conference on Computer and Communication Engineering*, Kuala Lumpur.
2. S Lopez, GM Caillo, JF Lopez dan R Sarmiento, (2005), "A High Quality/Low Computational Cost Technique for Block Matching Estimation", *Proceeding of the Design, Automation and Test in Europe Conference and Exhibition*.
3. H. Bhaskar, R.L. Kingsland, dan S.Singh, (2006), "Multi Resolution Based Motion Estimation for Object Tracking Using Genetic Algorithm", *IET International Conference on Visual Information Engineering (VIE2006) (CP522)*, hal.583-588, Bangalore, India.

4. Golam Sowar, Manzur Murshed, dan Laurence Dooley, (2004), "Fast Block-Based True Motion Estimation Using Distance Dependent Thresholds", *Journal of Research and Practice in Information Technology*, Vol. 36, No. 3.
5. Himanshu Shah dan Darryl Morrel, (2005), "An Adaptive Zoom Algorithm for Tracking Target Using Pan Tilt Zoom Cameras", *Conference Record of The Thirty Ninth Asilomar Conference*, October 28 – Nov 1, hal. 59-63.
6. Techmer, A., (2001), "Countour based Motion Esti-mation and Object Tracking for real time Applications Image Processing", *Procee-ding International Conference on Volume 3, Issue*, hal. 648-651.

