

# Penelusuran Citra pada Citra Bergerak dengan Interpolasi Newton

David Soendoro

Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
if17086@students.if.itb.ac.id

**Abstract**— Dalam melakukan pengenalan citra pada citra bergerak setiap citra harus diproses secara sekuensial. Untuk setiap pemrosesan citra tersebut, terkadang terdapat beberapa kegagalan deteksi sehingga hasil pengenalan menjadi terputus. Dalam mengatasi permasalahan ini, dapat digunakan perkiraan dengan melakukan interpolasi terhadap citra yang dikenali pada citra sebelum dan sesudah citra yang tidak dikenali. Salah satu metode untuk memperkirakan citra tersebut adalah dengan melakukan interpolasi. Interpolasi Newton merupakan salah satu bentuk interpolasi sederhana yang memiliki tingkat keakuratan cukup tinggi. Dengan menggunakan interpolasi Newton sebagai metode untuk memperkirakan lokasi citra yang tidak dikenali di antara lokasi citra yang dikenali didapatkan penelusuran citra yang baik. Metode interpolasi sederhana ini mampu memberikan hasil yang cukup akurat. Bagaimanapun, terkadang masih terjadi kesalahan interpolasi apabila tidak didapatkan interval yang sama untuk beberapa elemen di sekitar titik yang dicari.

**Kata kunci**- penelusuran citra, interpolasi Newton, citra bergerak.

## I. PENDAHULUAN

Penelusuran objek pada citra bergerak diperlukan untuk memperhalus tampilan penglihatan komputer pada saat mendeteksi suatu objek. Pada penglihatan komputer sering kali diperoleh hasil pada beberapa frame namun tidak seluruhnya. Hal ini mengakibatkan terjadinya loss pada frame-frame tertentu. Tentunya hal ini sangat mengganggu. Contoh aplikasi dari penelusuran objek pada citra bergerak ini adalah penelusuran bola tenis pada penganalisa pertandingan tenis[1]. Tanpa penelusuran, analisis pertandingan tersebut menjadi tidak sempurna. Bagaimanapun, masalah penelusuran objek pada citra bergerak dengan interpolasi Newton masih memiliki kelemahan yaitu hanya berlaku pada pergerakan objek yang normal. Apabila objek melakukan pergerakan zig-zag dengan cepat, interpolasi Newton yang polinom akan mengalami kesalahan perhitungan.

Bagian-bagian yang akan dibahas pada tulisan ini adalah sebagai berikut: pada seksi ke-dua, akan dibahas mengenai algoritma dan teori yang digunakan yaitu algoritma interpolasi Newton. Pada seksi ke-tiga dipaparkan hasil percobaan yang telah dilakukan. Terakhir, pada seksi ke-empat akan diberikan

kesimpulan dari percobaan tersebut dan saran untuk perbaikan dari algoritma yang telah dipaparkan.

## II. TEORI DAN ALGORITMA

Sebelum membahas algoritma interpolasi Newton sebagai penelusuran objek pada citra bergerak akan dijelaskan mengenai algoritma deteksi objek sederhana yang digunakan. Penjelasan mengenai algoritma deteksi tidak akan terlalu dalam karena bukan merupakan topik utama. Algoritma deteksi yang digunakan diambil dari tulisan [5]. Pada intinya adalah deteksi sudut-sudut pembentuk elips yang kemudian digabungkan menjadi satu.

Algoritma yang digunakan adalah interpolasi Newton sederhana. Interpolasi Newton dipilih karena komputasinya yang cukup cepat dan algoritmanya yang mudah digunakan serta cukup akurat. Adapun algoritma interpolasi polinom adalah sebagai berikut:

1. Bila diketahui pada saat  $x_i$  nilai  $f(x_i)$  adalah sekian di mana  $i$  adalah selang-selang di antara nilai yang dicari( $x$ ).
2. Buat tabel selisih sebesar  $n+1 * n+1$  di mana nilai  $n$  adalah derajat polinom Newton yang diinginkan.
3. Masukkan ke tabel selisih pada kolom pertama, seluruh nilai  $f(x_i)$ .
4. Lanjutkan dengan mengisi tabel selisih pada kolom berikutnya dengan menghitung selisih pada kolom sebelumnya, di mana nilai  $ST[i][k]$ , nilai tabel selisih pada kolom yang dicari adalah  $(ST[i+1][k-1] - ST[i][k-1]) / (x[i+k] - x[i])$ .
5. Setelah tabel selisih terbentuk, lakukan perhitungan pada baris di mana seluruh tabel selisih terisi. Jumlahkan nilai  $ST[0][i]$  di mana  $i$  adalah dari 0 hingga  $n + 1$ .
6. Untuk penjumlahan pada nilai  $ST[0][i]$  di mana  $i \neq 0$ , kalikan nilai selisih dengan selisih dari nilai yang dicari dengan  $x[i]$ .

Secara urut-urutan, di atas adalah algoritma interpolasi Newton pada umumnya. Algoritma tersebut dapat dituliskan dalam bentuk *pseudocode* seperti pada gambar berikut:

```

function Newton(x:real; n:integer):real;
{Menghitung y = p(x), dengan p(x) adalah polinom Newton
derajat n.Titik-titik data telah disimpan di dalam Larik
x[0..n] dan y[0..n] }

var
i, k : integer;
ST : array[0..30, 0..30] of real;
    {menyimpan tabel selisih terbagi}
jumlah, suku: real;

begin
for k:=0 to n do
    { simpan y[k] pada kolom 0 dari matriks ST }
    ST[k,0]:=y[k];
{end for}

for k:=1 to n do {buat tabel selisih terbagi}
    for i:=0 to n-k do
        ST[i,k]:=(ST[i+1,k-1] - ST[i,k-1])/(x[i+k]-x[i]);
    {end for}
{end for}

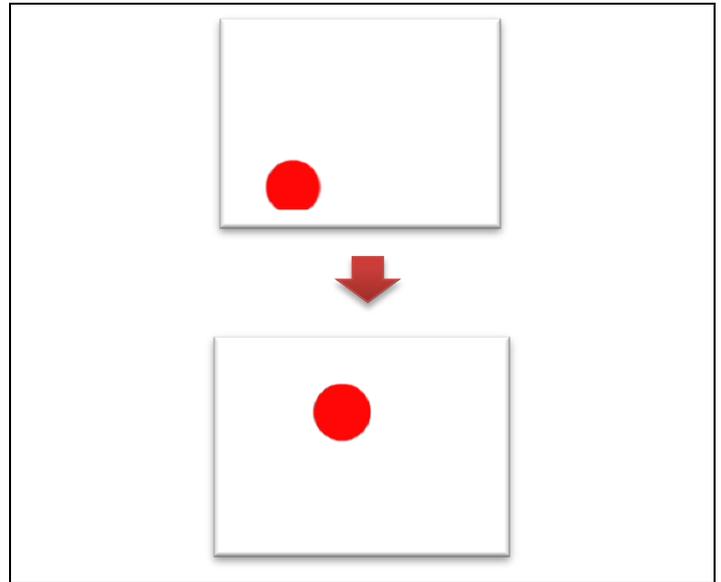
{hitung p(x) }
jumlah:=ST[0,0];
for i:=1 to n do
    begin
    suku:=ST[0,i];
    for k:=0 to i-1 do
        suku:=suku*(x-x[k])
    {end for}
    jumlah:=jumlah + suku;
    end;
Newton:=jumlah;

end;

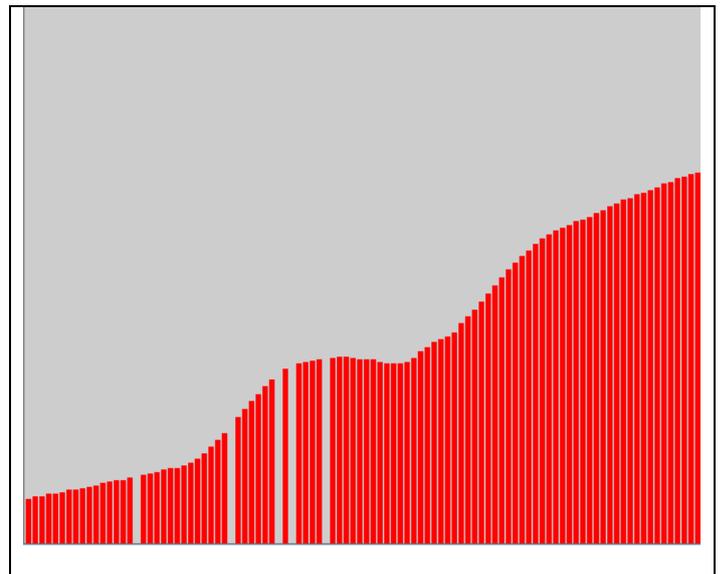
```

Algoritma 1 - Algoritma interpolasi Newton[2]

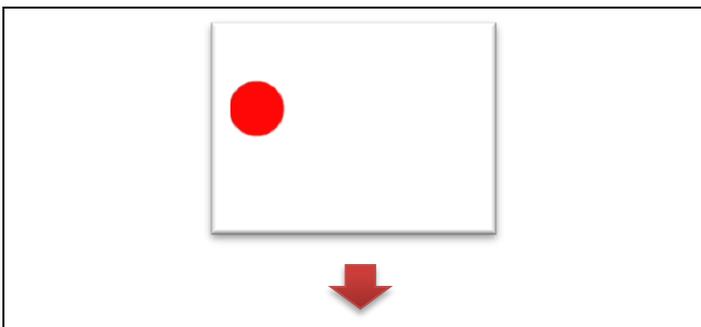
Implementasi algoritma tersebut pada persoalan penelusuran objek pada citra bergerak dapat dilakukan dengan melakukan interpolasi kepada titik x dan y dari objek yang terdeteksi. Mula-mula, buat histogram dari lokasi x dan y pada layar. Dengan menggunakan sampel citra bergerak sederhana seperti pada gambar 1, didapati letak sebuah benda adalah seperti pada gambar 2 dan 3.

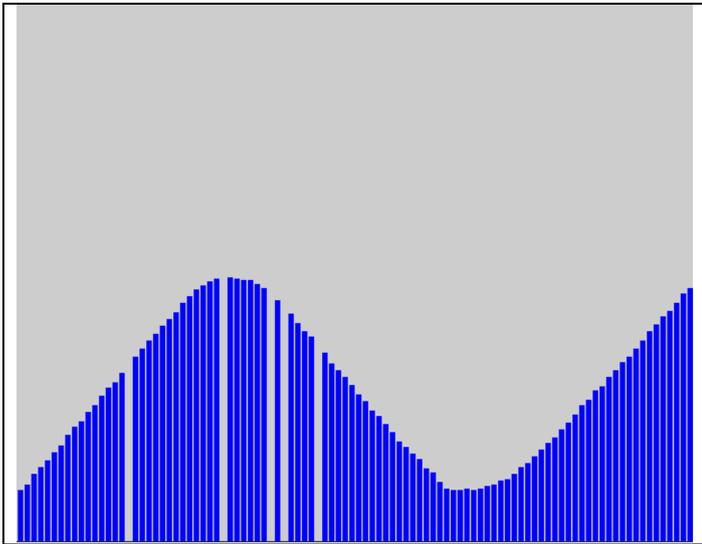


Gambar 1 - sampel citra bergerak



Gambar 2 - Histogram x hasil dari deteksi objek lingkaran merah

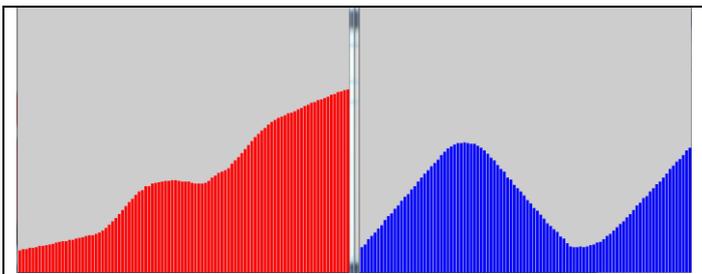




Gambar 3 - Histogram y hasil dari deteksi objek lingkaran merah

Setiap persegi panjang di atas merupakan letak objek lingkaran merah pada tiap citra-nya. Citra bergerak tersebut mengandung 100 citra. Histogram tersebut kemudian diinterpolasi untuk memperkirakan posisi objek lingkaran merah pada frame ke-sekian, di mana objek tidak terdeteksi.

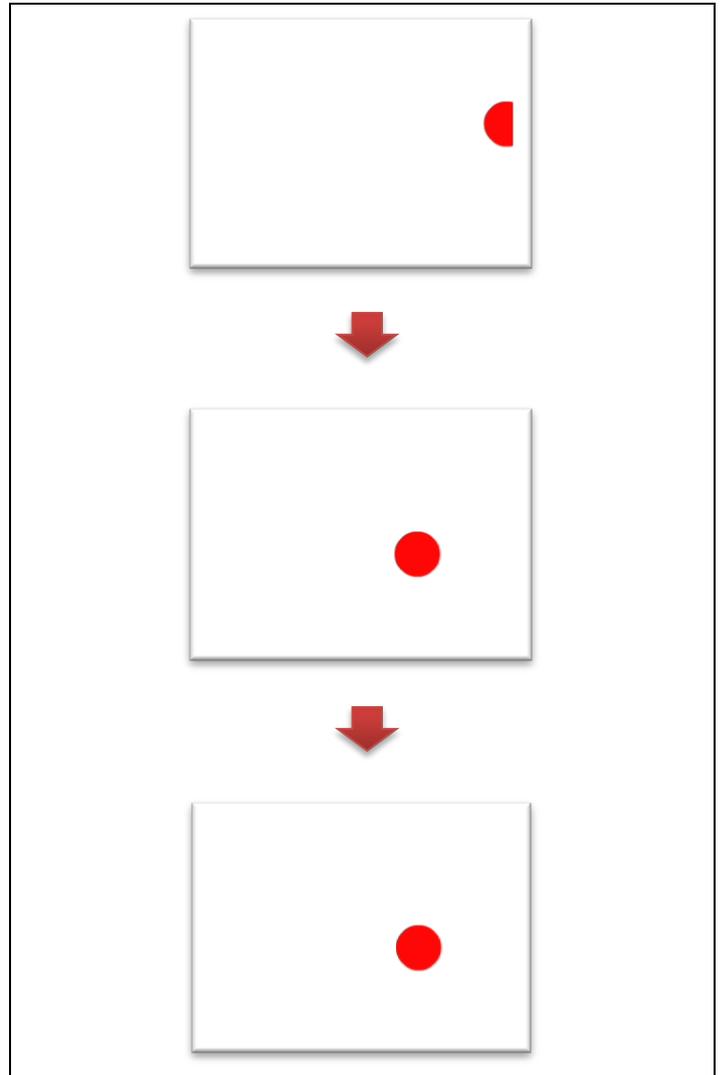
Hasil interpolasi histogram di atas adalah sebagai berikut:



Gambar 4 - hasil interpolasi posisi lingkaran merah pada citra bergerak

### III. PENGUJIAN

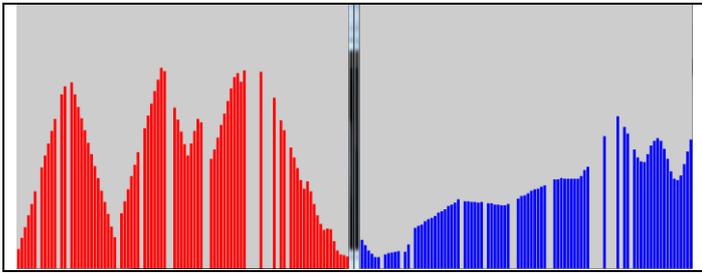
Pada pengujian, citra bergerak yang digunakan akan memiliki lebih banyak belokan dengan jumlah citra yang sama yaitu 100 buah citra. Berikut adalah citra bergerak yang akan digunakan pada pengujian kali ini:



Gambar 5 - citra bergerak lebih cepat untuk pengujian

Diperoleh histogram letak lingkaran merah pada citra bergerak tersebut seperti pada gambar 6, di mana histogram merah mewakili posisi x dan histogram y mewakili posisi y. Dibandingkan dengan pada saat pembuatan teori, histogram x dan y pada pengujian ini memiliki lebih banyak kekosongan.

Kekosongan deteksi banyak terjadi pada beberapa citra di belakang, hal ini diakibatkan pada citra belakang tersebut terjadi pergerakan zig-zag yang cepat. Hal serupa sering kali terjadi pada implementasi algoritma deteksi di lingkungan bergerak. Saat terjadi pergerakan citra dapat mengalami blur yang mengakibatkan tidak terdeteksinya objek.



Gambar 6 - histogram dari letak lingkaran merah

Hal ini menjadi persoalan tersendiri. Interpolasi Newton memiliki suatu kelemahan, yakni untuk memperoleh hasil yang akurat dibutuhkan nilai pada interval yang tetap. Sedangkan interval yang kita ketahui dari histogram di atas tidaklah tetap. Untuk mengatasi permasalahan tersebut, dilakukan pengecekan interval yang valid untuk titik yang dicari interpolasinya dengan mencoba interval dari 2 hingga ditemukan interval yang sesuai.

Algoritma yang digunakan adalah sebagai berikut:

```
void get4NearestPoints(int* xs, int* interval, int x, int
n, int* part_xs, int* retVal)
{
    int i, points_get = 0, n_prev = 0, n_next = 0;
    *interval = 2;

    for(i = 0; i < x; i++) {
        if(xs[i] != -1)
            n_prev++;
    }
    for(i = x; i < n; i++) {
        if(xs[i] != -1)
            n_next++;
    }

    while(points_get < 3) {
        if(n_prev / (*interval) < 2) {
            points_get = 0;

            // no other choice, return xs with interval
            for(i = 0; i < n && points_get < 4;
                i += (*interval)) {
                if(xs[i] != -1) {
                    retVal[points_get] = xs[i];
                    part_xs[points_get] = i;
                    points_get++;
                }
            }
            else {
                points_get = 0;
                i = 0;
                *interval = (*interval) + 1;
            }
        }
        continue;
    }
    if(n_next / *interval < 2) {
        points_get = 0;

        // no other choice, return xs from behind
        for(i = n; i > 0 && points_get < 4;
            i += (*interval)) {
            if(xs[i] != -1) {
                retVal[n - points_get] = xs[i];
                part_xs[n - points_get] = i;
                points_get++;
            }
        }
    }
}
```

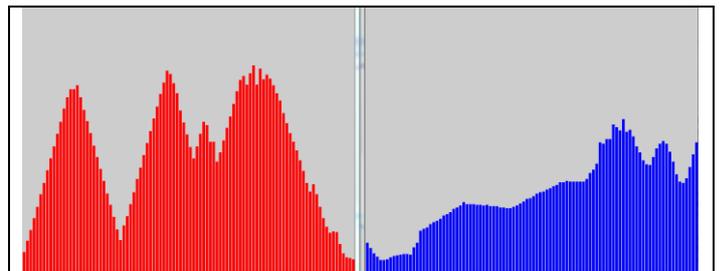
```

}
else {
    points_get = 0;
    i = n;
    *interval = (*interval) + 1;
}
}
continue;
}

if(xs[x - 2 * (*interval)] != -1 &&
xs[x - (*interval)] != -1 &&
xs[x + (*interval)] != -1 &&
xs[x + 2 * (*interval)] != -1) {
    retVal[0] = xs[x - 2 * (*interval)];
    part_xs[0] = x - 2 * (*interval);
    retVal[1] = xs[x - (*interval)];
    part_xs[1] = x - (*interval);
    retVal[2] = xs[x + (*interval)];
    part_xs[2] = x + (*interval);
    retVal[3] = xs[x + 2 * (*interval)];
    part_xs[3] = x + 2 * (*interval);
    points_get += 4;
}
else {
    *interval = (*interval) + 1;
}
}
}
}
```

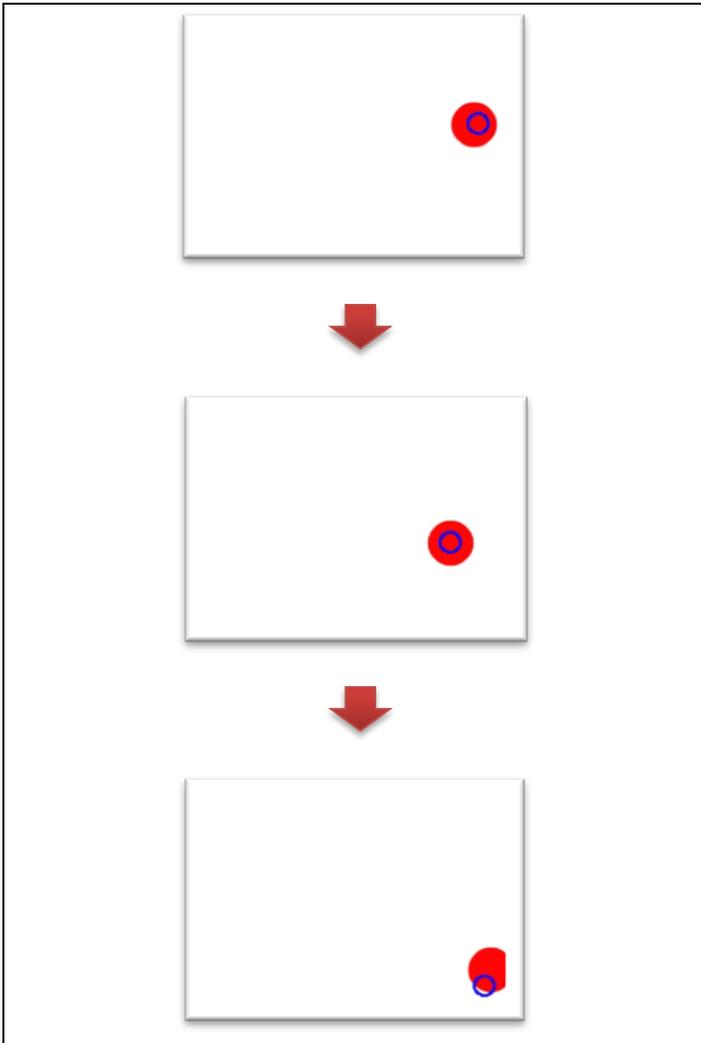
Algoritma 2 - algoritma untuk mendapatkan 4 titik terdekat dari sebuah titik yang ingin diinterpolasi

Dengan algoritma 2 di atas, dapat diperoleh titik titik interpolasi yang sesuai untuk menghasilkan nilai interpolasi yang mendekati. Menggunakan titik-titik di atas, diperoleh hasil interpolasi citra bergerak tersebut seperti pada gambar 7.



Gambar 7 - hasil interpolasi citra bergerak hasil pengujian

Titik-titik yang telah ditemukan ini kemudian digambarkan kembali pada citra bergerak dan menghasilkan citra bergerak seperti pada gambar 8.



Gambar 8 - Hasil penelusuran citra bergerak

Pada pengujian, terdapat ketidak tepatan pada beberapa citra terutama saat objek yang ditelusuri melakukan belokan secara berulang-ulang. Hal ini diakibatkan oleh salah sampel yang diambil oleh program. Salah sampel tersebut terjadi akibat gagal deteksi yang banyak terjadi di sekitar daerah yang terjadi perbelokan berulang-ulang.

Pada kasus banyaknya kegagalan deteksi akan mengakibatkan interval yang digunakan semakin besar karena interpolasi Newton mengharuskan interval yang tetap dari satu elemen ke elemen berikutnya. Akibatnya misalkan yang ingin diperoleh adalah posisi objek pada citra ke-50 maka lokasi objek pada citra yang digunakan apabila interval adalah 5 adalah posisi objek pada citra ke 42, 47, 52, dan 57 dan apabila interval semakin besar akan berarti semakin jauh data sampel yang diambil. Apabila data sampel ini melewati *local maxima* atau *local minima* yang seharusnya terjadi pada lingkungan terbatas tersebut maka dapat dipastikan interpolasi menjadi tidak tepat.

Akibatnya, pada pengujian dengan citra bergerak cepat terjadi beberapa kali ketidak tepatan pada penelusuran. Seperti pada gambar 8 bagian terakhir. Penelusuran memperkirakan titik tengah lingkaran berada pada kiri bawah titik lingkaran yang seharusnya.

Untuk memperjelas tulisan ini, hasil pengujian dapat dilihat di pranala berikut:

<http://www.youtube.com/watch?v=OAdNhnY42yA>

Pada citra bergerak yang terdapat pada pranala di atas, lingkaran hijau adalah hasil deteksi sedangkan lingkaran biru adalah hasil penelusuran. Dapat dilihat perbaikan dari deteksi saja dengan ditambahkannya penelusuran objek.

#### IV. KESIMPULAN DAN SARAN

Interpolasi Newton dapat digunakan untuk memperkirakan lokasi objek pada citra bergerak untuk citra-citra yang tidak terdeteksi. Hasil interpolasi Newton cukup baik namun memiliki kesalahan apabila data interpolasinya tidak mencukupi dan tidak ditemukan selang yang cukup dekat. Salah satu kelemahan interpolasi Newton adalah interval yang harus sama jumlahnya untuk memperoleh hasil yang akurat.

Untuk perbaikan dapat dipilih interpolasi yang mampu melakukan interpolasi dengan interval beragam. Selain itu, interpolasi juga tidak mampu menjawab permasalahan penelusuran pada waktu berjalannya aplikasi. Untuk mengatasi permasalahan ini mungkin model regresi nirlanjar akan lebih akurat.

Salah satu tulisan yang dibuat mengenai penelusuran objek dengan regresi adalah pada referensi [4]. Penelusuran objek dengan regresi dapat menangani permasalahan penelusuran secara *real-time*. Namun, dibandingkan dengan interpolasi hasil dari regresi lebih banyak menyimpang.

Penelusuran objek pada citra bergerak dengan interpolasi Newton adalah sebuah implementasi sederhana dari metode numerik dalam aplikasi yang sesungguhnya. Masih banyak hal dari penelusuran objek pada citra bergerak yang menggunakan metode-metode numerik. Topik ini dapat menjadi salah satu topik yang menarik untuk dikembangkan.

#### UCAPAN TERIMA KASIH

David Soendoro berterima kasih atas segala bantuan yang didapat dari teman-teman kuliah dalam meminjamkan sarana dalam mencari data hingga akhirnya makalah ini dapat tersusun. Terima kasih juga penulis tujuan bagi bapak Rinaldi Munir M.T., dosen mata kuliah Kriptografi yang telah mengajarkan materi sebagai dasar pembuatan makalah ini.

#### REFERENSI

- [1] <http://www.youtube.com/watch?v=iRIWw8GD0xc>; waktu akses: 08:07 12 Mei 2011.
- [2] Munir, R. "Metode Numerik Untuk Teknik Informatika". Diktat kuliah, Instut Teknologi Bandung, 1997. Halaman 162.

- [3] Mayol, W.W. and Murray, D.W.. "Tracking with General Regression". Accepted paper from [www.springerlink.com](http://www.springerlink.com). Accepted: 27 February 2007.
- [4] Zhu, W., Wang, S., Lin, R., and Levinson S.. "Tracking of Object with SVM Regression". Computer Society Conference on Computer Vision and Pattern Recognition. IEEE 2001.
- [5] Nguyen, T.M., Ahuja S., Wu, Q.M.J.. "A Real-Time Ellipse Detection Based on Edge Grouping"

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 13 Mei 2011

ttd

David Soendoro 13507086