

Image Matching untuk Penggabungan Panoramic Picture

Willy Gotama / 13510072

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13511072@std.stei.itb.ac.com

Abstrak – Makalah ini membahas tentang algoritma baru yang saya buat dan dapat digunakan untuk mencocokkan dua buah gambar yang memiliki irisan. Penerapan algoritma ini dapat digunakan terutama pada proses penggabungan pada dua buah foto agar dapat menjadi panoramic picture seiring dengan berkembangnya bidang ilmu strategi algoritma, kebutuhan akan algoritma yang canggih dan mampu melakukan proses image matching dengan lebih cepat. Hal – hal yang sebelumnya tidak mampu kita lakukan misalnya seperti menggabungkan gambar secara automatic kini dapat dilakukan bahkan dalam hitungan detik. Namun kebutuhan akan algoritma picture matching yang lebih cepat dan akurat tidak pernah berhenti. Berdasarkan pada hal ini, saya menulis makalah ini untuk menginformasikan ide algoritma saya untuk proses image matching.

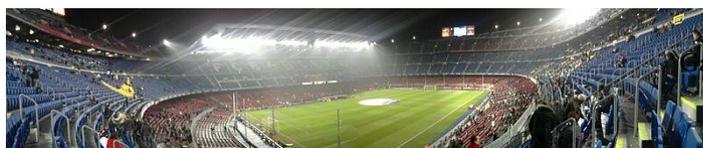
Kata kunci – Panoramic, Image Matching, Brute Force, Boyer Moore, KMP

I. PENDAHULUAN



Gambar 1.1 Sydney Harbour Bridge Panoramic Picture
http://en.wikipedia.org/wiki/Panoramic_photography

Panoramic Photography adalah salah satu teknik fotografi yang menghasilkan gambar yang lebar dan mencakup area pandang yang luas. Panoramic Photography biasa digunakan untuk memfoto pemandangan alam, jembatan, pemandangan kota maupun lapangan olahraga. Gambar yang dihasilkan oleh Panoramic Photography biasanya disebut Panoramic Picture. Panoramic Picture menimbulkan kesan yang mengandung nilai estetika yang tinggi serta memberi kesan yang lebih nyata terhadap gambar.



Gambar 1.2 Panoramic Picture 3D
http://en.wikipedia.org/wiki/Panoramic_photography

Pada awal mulanya butuh keahlian yang sangat tinggi serta ketekunan dari seorang Fotografer untuk dapat

menghasilkan sebuah Panoramic Picture. Proses Pembuatan sebuah Panoramic Picture mencakup Pegambilan gambar dan penggabungan gambar secara manual.

Proses pengambilan gambar harus dilakukan dari posisi yang sama namun dengan sudut yang berbeda. Pengambilan gambar juga harus dilakukan pada rentang waktu yang sedekat mungkin agar pencahayaan dan moment foto sama.

Proses penggabungan gambar dilakukan dengan cara menempelkan foto-foto yang telah dihasilkan sehingga menghasilkan sebuah foto yang lebar yang merupakan gabungan dari beberapa foto.



Gambar 1.3 Panoramic Picture Jaman Dulu
by Martin Behrmanx(1851)

http://en.wikipedia.org/wiki/Panoramic_photography

Seiring Berkembangnya zaman, Proses Pembuatan Panoramic Picture menjadi semakin mudah dengan adanya dukungan dari Kamera Panoramic. Kamera Panoramic adalah kamera yang mampu memutar lensanya sehingga proses pengambilan foto dan penggabungan foto langsung dilakukan pada Kamera. Namun Kamera Panoramic memiliki harga yang cukup mahal sehingga tidak banyak orang yang dapat memiliki Kamera Panoramic.



Gambar 1.4 Contoh Kamera Panoramic Zaman dahulu
<http://www.antiqewoodcameras.com>

Untuk menangani masalah biaya pada Kamera Panoramic, muncullah beberapa software yang mampu menggabungkan gambar sehingga dapat menjadi gambar Panoramic. Software-software ini (Software Panoramic) biasa di-implementasikan langsung pada kamera, namun ada juga dijalankan di komputer. Software Panoramic

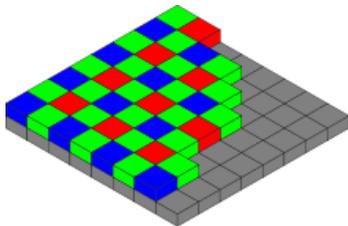
menggunakan berbagai metode picture matching untuk mencocokkan bagian gambar dan menggabungkannya.

Pada makalah ini saya akan membuat Algoritma batu yang merupakan gabungan dari beberapa algoritma sadar string matching untuk mencocokkan dua gambar digital sehingga dapat menjadi sebuah Panoramic Picture

II. TEORIDASAR

Gambar Digital tersusun atas pixel – pixel yang sangat kecil. Pixel-pixel ini tersimpan dalam bentuk seperti Matrix [A][B] dengan A adalah jumlah maksimum pixel dalam setiap baris Gambar dan B adalah jumlah maksimum pixel dalam setiap kolom gambar. Pixel-pixel ini memiliki warna masing-masing yang jika dilihat secara keseluruhan akan memunculkan suatu gambar.

Tiap-tiap pixel memiliki tiga elemen penting penentu warna pixel yaitu R , G dan B. R merupakan elemen penentu seberapa merah suatu Pixel, G merupakan elemen penentu seberapa hijau suatu Pixel dan B merupakan elemen penentu seberapa biru suatu Pixel. Tiap-tiap elemen ini memiliki nilai dari 0 sampai 255 dan jika digabungkan dapat menghasilkan $256^3 = 16777216$ jenis warna.



Gambar 2.1 Pixel

http://en.wikipedia.org/wiki/Digital_picture

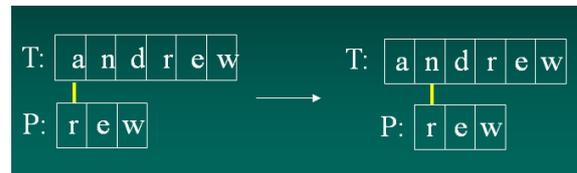
Untuk Mencocokkan dua buah gambar digital kita dapat mencocokkan setiap Pixelnya seperti Proses pencocokkan karakter pada string matching. Namun karena gambar digital memiliki struktur Map dan bukan Array maka algoritma pencocokan dua buah gambar harus dikembangkan lebih lanjut.

Algoritma baru yang saya buat melibatkan dua buah algoritma dasar yaitu Brute Force dan Boyer Moore.

A. Brute Force

Algoritma Brute Force adalah Algoritma yang menyelesaikan masalah dengan mencoba semua kemungkinan penyelesaian yang ada. Penyelesaian masalah dengan Algoritma Brute force cenderung lebih sederhana dan mudah dimengerti namun membutuhkan waktu yang lama.

Brute Force pada string matching mencari keberadaan pattern pada sebuah string dengan mencocokkan tiap – tiap karakternya , jika ada karakter yang tidak cocok maka pattern akan digeser 1 posisi ke kanan dan proses pencarian dilanjutkan.



Gambar 2.2 Contoh Brute Force pada string matching Slide Perkuliahan IF2211 String Matching

String Matching dengan menggunakan algoritma Brute Force memiliki kompleksitas $O(mn)$ dengan m adalah panjang pattern yang dicari dan n adalah panjang text yang menjadi dasar pencarian pada string matching. Best Case dari Algoritma Brute Force adalah $O(m)$ sementara average case nya adalah $O(m + n)$.

Kelebihan dari algoritma Brute Force adalah cepat jika jenis dari elemen yang dicocokkan banyak. Selain itu , algoritma Brute Force tidak perlu melakukan preprocessing terlebih dahulu sebelum melakukan pencarian.

Kelemahan dari algoritma bruteforce adalah lambat jika jenis dari elemen yang dicocokkan sedikit. Brute Force juga banyak melakukan pengecekan yang tidak efisien sehingga jumlah proses dan waktu eksekusi nya cenderung lebih lama dari algoritma string matching lainnya

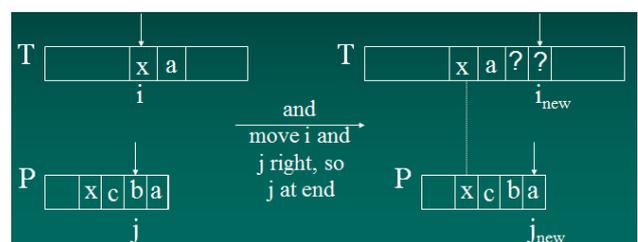
B. Boyer Moore

Algoritma Boyer Moore adalah algoritma sting matching yang cukup efisien dan senderugn cepat proses eksekusinya. Algoritma Boyer Moore memiliki 2 tehnik dasar yaitu looking-glass dan character-jump

Looking glass adalah pencocokkan karakter yang dimulai dari karakter terakhir dari pattern dan berjalan mundur jika character cocok. Pattern dinyatakan ketemu jika looking-glass sampai pada karakter pertama pattern dan karakter tersebut juga cocok.

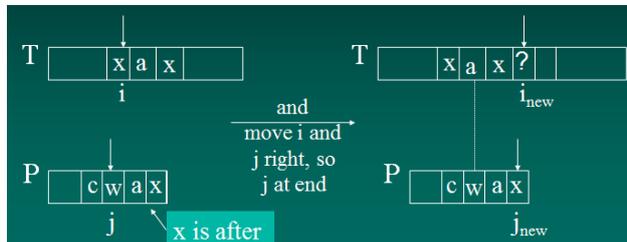
Character-jump adalah pemindahan pattern ke suatu posisi. Character-jump adalah dasar yang membuat Boyer Moore cepat. Ada 3 kasus yang ditemukan pada Character-jump.

Kasus 1, jika karakter penyebab ketidakcocokan ada pada pattern namun ada di sebelah kiri posisi karakter sekarang . Pada kasui ini langkah yang diambil adalah geser pattern sampai karakter penyebab ketidakcocokkan tersebut berada pada posisi yang sama dengan karakter terakhir pada pattern yang sama dengan karakter penyebab ketidakcocokkan.



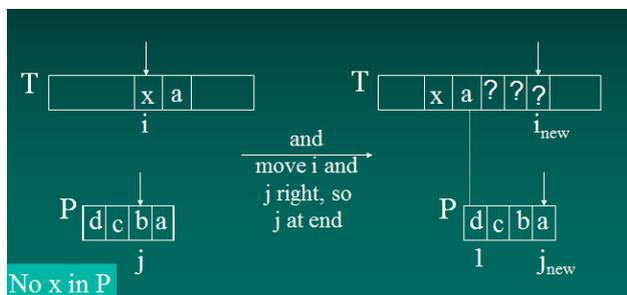
Gambar 2.3 Contoh Boyer Moore Case 1

Kasus 2, jika karakter penyebab ketidakcocokan ada pada pattern namun ada di sebelah kanan posisi karakter sekarang . Pada kasu ini langkah yang diambil adalah geser pattern satu posisi ke kanan lalu kembalikan posisi pengecekan karakter pada ujung dari pattern.



Gambar 2.4 Contoh Boyer Moore Case 2 Slide Perkuliahan IF2211 String Matching

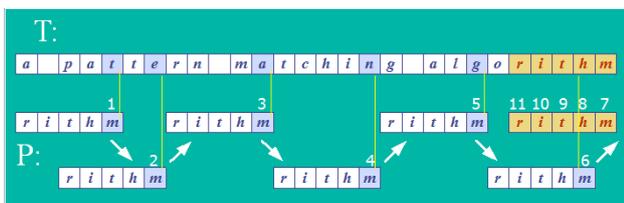
Kasus 3, jika , jika karakter penyebab ketidakcocokan tidak ada pada pattern. Pada kasus ini pattern digeser sehingga awal pattern berada tepat di sebelah kanan dari karakter penyebab ketidakcocokkan. Lalu kembalilakukan pencocokkan pada akhir pattern.



Gambar 2.5 Contoh Boyer Moore Case 3 Slide Perkuliahan IF2211 String Matching

Selain looking-glass dan charcter-jump , Boyer Moore harus melakukan preparation terlebih dahhulu sebelik melakukan string matching. Preparasi ayng perlu dilakukan adalah mengisi table Last Occurrence yang berisi nilai posisi terakhir kemunculan sebua karakter pada pattern.

String Matching dengan menggunakan algoritma Boyer Moore memiliki kompleksitas $O(mn + A)$ dengan m adalah panjang pattern yang dicari, n adalah panjang text yang menjadi dasar pencarian pada string matching dan A adalah jumlah jenis karakter yang dicocokkan.



Gambar 2.6 Contoh Eksekusi Boyer Moore Slide Perkuliahan IF2211 String Matching

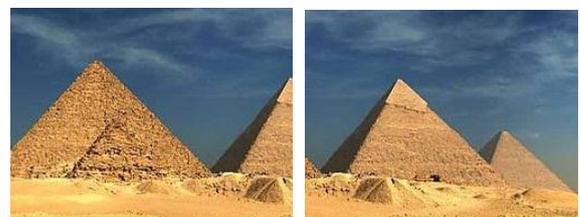
Kelebihan dari algoritma Boyer Moore adalah cepat jika jenis dari elemen yang dicocokkan banyak. Selain itu , character-jump pada algoritma Boyer Moore sangat efektif dan cepat dalam menelusiri Text sehingga waktu eksekusi cenderung cepat.

Kelemahan dari algoritma Boyer Moore adalah lambat jika jenis dari elemen yang dicocokkan sedikit. Boyer Moore juga membutuhkan preses preprocessing yang membutuhkan resource memory yang cukup besar tergantung dari panjang pattern dan jumlah jenis karakter.

III. ALGORITMA

Kasus penggabungan dua buah gambar ini berbeda dari string matching biasa karena yang harus di match tidak hanya satu baris. Selain itu jumlah pattern dan text tidak tetap, karena semangkin digeser jumlah pattern dan text akan semangkin bertambah.

Algoritma Image Matching yang saya buat adalah algoritma picture matching yang menggabungkan 2 buah algoritma diatas dan menghasilkan algoritma yang cukup efektif dan efisien dalam melakukan pengecekan pada dua buah gambar digital yang hendak di gabungkan pada sisi kiri atau kanan nya.



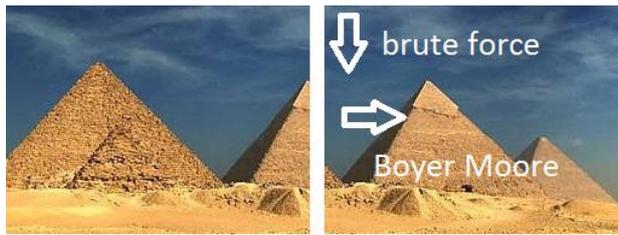
Gambar 3.1 Contoh Gambar yang hendak digabung http://en.wikipedia.org/wiki/Panoramic_photography

Algoritma ini menggunakan Brute Force untuk mencocokkan gambar secara vertikal dan menggunakan algoritma Boyer Moore untuk mencocokkan picture secara horizontal.

Dengan asumsi bahwa kedua gambar yang hendak digabungkan memiliki resolusi dan jumlah pixel pada sisi vertikal dan horizontal yang sama Pencocokkan gambar secara vertikal menggunakan algoritma Brute Force. KPM dan Boyer Moore lebih efektif dari Brute Force jika ada proses pergeseran, namun karena panjang Pattern dan Text sama , maka tidak akan ada pergeseran. Sementara itu KMP dan Boyer Moore membutuhkan preprocessing terlebih dahulu. Karena itu saya memilih Brute Force sebagai Algoritma Untuk mencocokkan gambar secara verikal.

Sementara untuk Pencocokkan gambar secara horizontal, saya memilih untuk menggunakan Boyer Moore karena panjang pattern dan text berbeda. Maka Boyer Moore lebih baik dari Greedy dalam hal ini, sementara Boyer Moore juga lebih baik dari pada KMP dalam kasus ini karena jenis pixel sangat banyak (16777216). Karena itu saya memilih Boyer Moore

sebagai Algoritma Untuk mencocokkan gambar secara horizontal.



Gambar 3.2 Proses Penggabungan

http://en.wikipedia.org/wiki/Panoramic_photography

Algoritma utama Pencarian berjalan dengan cara menggeser gambar. Asumsikan ada 2 buah gambar Pict1 dan Pict2 dengan Pict1 ada di sebelah kiri dari Pict2. set 2 pointer, pointer pertama menunjuk ke kolom terakhir dari Pict1 dan pointer kedua menunjuk ke kolom pertama dari Pict2.

Lalu lakukan Brute Force untuk kolom pada pointer pertama dan pointer kedua. Jika sama, iterasi pointer pertama ke kanan dan pointer kedua ke kiri. lakukan terus Brute Force sampai kedua pointer menunjuk ke luar dari gambar atau ditemukan kesalahan.

Jika kedua pointer menunjuk ke luar dari gambar maka gambar match dan dapat di gabungkan pada posisi pointer pertama dan kedua sebelum di iterasi.

Jika ditemukan kesalahan, lakukan Boyer Moore untuk baris yang menyebabkan kesalahan. Boyer Moore hanya dilakukan pada baris yang menyebabkan kesalahan pada Brute Force karena penyebab kesalahan tersebut memiliki kemungkinan Boyer Moore yang lebih kecil untuk cocok.

Jika jika Boyer Moore tidak menemukan kecocokkan maka gambar tidak dapat di gabungkan. Sementara jika Boyer Moore menemukan kecocokkan maka pindahkan posisi pointer pertama dan kedua ke posisi hasil Boyer Moore. Setelah itu kembali lakukan Brute Force dan berulang sampai gambar dapat dinyatakan dapat digabungkan atau tidak dapat digabungkan .



Gambar 3.3 Hasil Penggabungan

http://en.wikipedia.org/wiki/Panoramic_photography

Sebagai Pengembangan dari algoritma diatas dapat ditentukan panjang pattern yang akan dicocokkan , kita dapat pula menumpukan beberapa pattern terakhir penyebab kesalahan.

Panjang pattern dan jumlah pattern akan mempengaruhi performansi dari algoritma ini. Semakin banyak jumlah dan panjang pattern mungkin mempercepat proses, namun dapat pula membuat proses lambat karena preprocessing yang terlalu banyak.

IV. PSEUDO CODE

pixes adalah type dengan 3 parameter

R untuk nilai Red

G untuk nilai Green

B untuk nilai Blue

picture = pixel[a][b]

a = jumlah pixel horizontal

b = jumlah pixel vertical

LIMIT_A = limit panjang pattern untuk BM

LIMIT_B = limit jumah pattern untuk BM

TABLE BMX[LIMIT_A][LIMIT_B]

// INISIASI TABEL BM dengan Pixel(-1,-1,-1)

function BF (pixel[] A,pixel[] B) : int

```
{
  int i= 0;
  for (int j =0; j < A.length; j++)
  {
    if(A[i] != B[i])
    {
      i = j
      break
    }
  }
  return j;
}
```

function BM (pixel[] Pattern, pixel[] Text) : int

```
{
  int temp
  int BMtable[256][256][256]

  // isi table BM
  for (int i=0; i < Pattern.length; i++)
  {
    BMtable[Pattern[i].R][Pattern[i].G][Pattern[i].B] =
  }
  j = 0;
  k = 0;
  while (k < Text.length)
  {
    // jika pixel sama
    if (Text[k] == Pattern[j])
    {
      if (j == 0)
      {
        temp = k + pattern.length
        break
      }
    }
    else
    {
      j--
    }
  }
}
```

```

        k--
        }
    }
    else
    {
        if
        (BMtable[Pattern[j].R][Pattern[j].G][Pattern[j].B] == 0) //
        case 3
        {
            k = k+j
            j = pattern.length
        }
        else
        (BMtable[Pattern[j].R][Pattern[j].G][Pattern[j].B] > j) //
        case 2
        {
            k = k + pattern.length - j + 1
            j = pattern.length
        }
        else // case 1
        {
            k = k + pattern.length
            j = pattern.length
        }
    }
    }
    temp = k;
    return
}

int pos = 0;
int pos2;
while (pos < pict1.length)
{
    pos2 = pos;
    while (x == 0 && pos2 >= 0)
    {
        int x = BF(Pict1[text.length - pos2],Pict2[pos2])
        pos2--
        if (pos2 < 0) {return pos} // ketemu
    }
    // set BMX
    for (int z=0; z < LIMIT_A; z++)
    {
        BMX[pos % LIMIT_B][z] = Pict1(pixel[x][pos + z])
    }
    for (int z=0; z < LIMIT_A; z++)
    {
        if (BMX[z][0] != pixel(-1,-1,-1))
        {
            pos = max (pos,BM(BMX[z],Pict2))
        }
    }
}
}

```

V. ANALISA DAN KESIMPULAN

Dari hasil analisa dan percobaan terhadap algoritma Image Matching diatas. Proses pencocokkan image berjalan dengan kecepatan eksekusi yang cukup memuaskan.

Perbedaan waktu Eksekusi lumayan jauh jauh jika digunakan pada gambar dengan jumlah pixel banyak maupun sedikit. Hal ini berarti proses sangat bergantung dari besarnya gambar.

Sementara untuk beberapa gambar dengan banyak warna yang sama seperti gambar langit, proses berjalan dengan kecepatan yang tidak terlalu berbeda. Hal ini berarti algoritma ini dapat dengan efektif mencari bagian yang berbeda dan mencocokkannya.

Pada algoritma ini karena Algoritma Brute Force dan Boyer Moore yang digunakan agak berbeda maka Orde dari algoritma ini tidak sesederhana perkalian dari kedua orde diatas. Orde dari Brite Force pada proses ini adalah $O(Y*X*X)$. X adalah lebar gambar dan Y adalah tinggi gambar. Orde untuk Boyer Moore pad algoritma ini adalah $O(LIMIT_B*X*LIMIT_A + LIMIT_B*A)$. LIMIT_A adalah panjang pattern untuk Boyer Moore dan LIMIT_B adalah panjang pattern untuk Boyer Moore. Orde untuk Orde dari algoritma ini adalah perkalian dari Orde Brute Force dan Orde dari algoritma Boyer Moore pada algoritma ini yaitu $O(Y*X*X + X*LIMIT_A*LIMIT_B)$ dengan bestcase adalah $O(Y)$.

Meskipun dari orde dan kecepatan Algoritma ini tidak lebih baik daripada algoritma yang sudah ada saat ini dan belum dapat mempertimbangkan factor srpia atau flare dan efek” photo lainnya. namun algoritma ini bisa menjadi pertimbangan untuk dikembangkan lebih lanjut.

REFERENCES

- [1] Munir, Rinaldi, *Matematika Diskrit*. Bandung: Penerbit Informatika, 2004.
- [2] Munir, Rinaldi, Diktat Kuliah IF 3051 Strategi Algoritma. Bandung: Program Studi Teknik Infomatika Institut Teknologi Bandung, 2009.
- [3]<http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/stmik.htm>, diakses tanggal 22 Desember 2013, 11.00 WIB
- [4] “Panoramic Picture”, URL: http://en.wikipedia.org/wiki/Panoramic_photography diakses tanggal 20 Desember 2013, 11.00 WIB
- [5] “Digital Picture”, URL: http://en.wikipedia.org/wiki/Digital_picture, diakses tanggal 22 Desember 2013, 11.00 WIB
- [6] <http://www.antiquewoodcameras.com>, diakses tanggal 22 Desember 2013, 11.00 WIB

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, 20 Desember 2013

A handwritten signature in black ink, consisting of several loops and a long horizontal stroke at the end.

Willy Gotama