

PENERAPAN METODE FAST MARCHING PADA PERHITUNGAN GEODESIC DISTANCE PERMUKAAN OBYEK TRIANGULAR MESH

Rully Soelaiman¹, Eddy Tjandra² dan I Made Agus Setiawan²

¹Jurusan Sistem Informasi, Fakultas Teknologi Informasi ITS

²Jurusan Teknik Informatika, Fakultas Teknologi Informasi ITS

e-mail: rully@its-sby.edu

ABSTRAK: Dalam makalah ini akan dibahas penerapan algoritma perhitungan *geodesic distance* pada permukaan obyek triangular mesh untuk dibuktikan tingkat keakuratan dan efisiensinya. Algoritma yang diterapkan adalah *Fast Marching Method on Triangulated Domain* (FMM on TD) yang berjalan dengan kompleksitas waktu $O(n \lg n)$, dimana n adalah jumlah titik pada permukaan. Inti dari algoritma ini adalah melakukan front propagation dari titik awal ke segala arah yang mungkin sampai diperoleh titik akhir. Setiap bergerak maju algoritma ini selalu menghitung nilai jarak suatu titik terhadap titik awal. Setelah proses perhitungan *geodesic distance* selesai, dilakukan proses pembuatan *geodesic path*. Inti dari proses ini adalah melakukan *back propagation* pada permukaan dari titik akhir sampai diperoleh titik awal. Berdasarkan uji coba, tingkat keakuratan algoritma FMM on TD adalah lebih dari 95%. Keakuratan ini dipengaruhi oleh jumlah segitiga pembentuk permukaan. Semakin banyak segitiga semakin akurat *geodesic distance* yang dihasilkan, tetapi waktu yang dibutuhkan untuk melakukan proses perhitungan menjadi semakin lama.

Kata kunci: *computational geometry, geodesic distance, geodesic path, triangular mesh, fast marching method on triangulated domain, front propagation, back propagation.*

ABSTRACT: In this paper, we will describe an algorithm for *geodesic distance* calculation that applied at the surface of triangular mesh object, to analyze its efficiency and accuracy. We applied the *Fast Marching Method on Triangulated Domain* (FMM On TD) with $O(n \lg n)$ time complexity, where n is the number of point/triangle vertex that represent the surface. The main task of this algorithm is to do the front propagation step from the starting point to all possible direction to obtained the final point. Every step of this algorithm, will calculate the distance from the current point to the starting point. After the calculation of the *geodesic distance* finished, the next process is to build the *geodesic path*. The main task of this step is to do the *back propagation* step at the object surface from the final point to the starting point. Based on the experimental result, the accuracy of the FMM on TD algorithm is more than 95%. This accuracy depend on the number of the triangles that represent the surface. The accuracy of the *geodesic path* and the computational times are depends on the number of the triangles that used to represent the surface

Keywords: *Geodesic Distance, Geodesic Path, Triangular Mesh, Fast Marching Method on Triangulated Domain, Front Propagation, Back Propagation.*

PENDAHULUAN

Geodesic Distance adalah jarak terdekat antara pasangan titik pada permukaan obyek 3 dimensi. Jarak ini dihitung tanpa melewati bagian dalam dari obyek 3 dimensi tersebut. *Geodesic Path* adalah lintasan pada permukaan obyek 3 dimensi yang jaraknya direpresentasikan oleh *Geodesic Distance*. *Triangular Mesh* adalah representasi permukaan 3D (*surface*) yang dibentuk oleh segitiga-segitiga.

Pada tahun 1998, Sethian dan Kimmel [3] menggunakan *Fast Marching Method* (FMM) untuk menentukan fungsi jarak dari satu titik ke semua titik lainnya pada permukaan dengan tingkat kompleksitas waktu $O(n \lg n)$. Sampai sekarang mungkin

algoritma FMM inilah yang mempunyai tingkat kompleksitas waktu yang paling cepat. Tetapi tidak seperti algoritma-algoritma lainnya, algoritma ini tidak menghasilkan *Geodesic Distance* dan *Geodesic Path* yang sebenarnya, melainkan hanya perkiraan.

Dalam penelitian ini akan dilakukan implementasi dan analisis algoritma *Fast Marching Method on Triangulated Domain* (FMM on TD), yang merupakan pengembangan dari algoritma FMM dasar, sebagai metode untuk perhitungan *Geodesic Distance* pada permukaan obyek *Triangular Mesh*. Algoritma FMM on TD merupakan bentuk khusus dari algoritma FMM yang hanya diterapkan pada obyek *triangular mesh*. Penjelasan algoritma ini ada pada bagian 2. Kemudian nilai *Geodesic Distance*

hasil dari proses FMM dijadikan inputan proses pembuatan *Geodesic Path*. Penjelasan proses ini ada pada bagian 3. Pada bagian 4 akan ditampilkan hasil dari beberapa uji coba yang dilakukan dan kesimpulan dijabarkan pada bagian 5.

FAST MARCHING METHOD

Dalam permasalahan perhitungan *Geodesic Distance*, secara matematis tujuan dari algoritma *Fast Marching Method* ini adalah mencari pendekatan penyelesaian persamaan Eikonal:

$$|\nabla T|F = 1 \quad (1)$$

dengan T adalah fungsi jarak dari suatu titik terhadap titik awal (*starting vertex*), dimana nilai T pada titik awal sama dengan 0 dan F adalah fungsi kecepatan dari propagasi, dimana F dapat berupa *cost* (biaya) yang dibutuhkan untuk berpindah dari satu titik ke titik lainnya, tetapi dalam permasalahan geometri yang hanya mempertimbangkan jarak saja nilai $F = 1$.

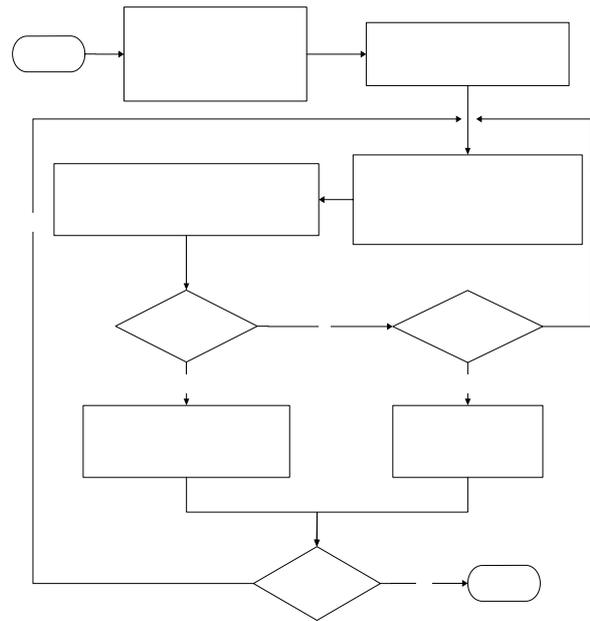
Inti dari algoritma *Fast Marching Method* ini adalah melakukan propagasi menyebar/maju (*front propagation*) dari sebuah titik awal ke segala arah yang mungkin. Setiap bergerak maju algoritma ini selalu menghitung dan menyimpan nilai jarak suatu titik terhadap titik awal, nilai jarak tersebut disimpan sebagai properti dari titik tersebut. Proses ini berhenti sampai titik akhir telah dihitung nilai jaraknya.

Setiap titik mempunyai beberapa properti yaitu:

- T , menyatakan nilai *geodesic distance* titik tersebut dari titik awal.
- Status, ada 3 macam status, yaitu :
 - o *Far*, artinya titik tersebut belum pernah dihitung nilai *geodesic distance*-nya.
 - o *Close*, artinya titik tersebut pernah diproses, tetapi nilai *geodesic distance*-nya belum final. Nilai *geodesic distance* tersebut dapat berubah atau di-*update* lagi.
 - o *Fix*, artinya titik tersebut telah diproses. Nilai *geodesic distance*-nya sudah final, tidak dapat di-*update* lagi.

Gambar 1 menunjukkan diagram alir proses FMM on TD. Proses ini dibagi menjadi 2 tahap yaitu: tahap inisialisasi dan tahap perulangan. Berikut langkah-langkah dalam tahap inisialisasi:

- Semua titik yang ada statusnya ditetapkan menjadi *Far*.
- Nilai *geodesic distance* atau nilai T dari tiap titik ditetapkan menjadi tidak hingga (*infinite*).
- Pada titik awal (*Starting vertex*), nilai T ditetapkan = 0.



Gambar 1. Diagram Alir dari Proses FMM on TD untuk Perhitungan Geodesic Distance

Berikut langkah-langkah dari tahap perulangan:

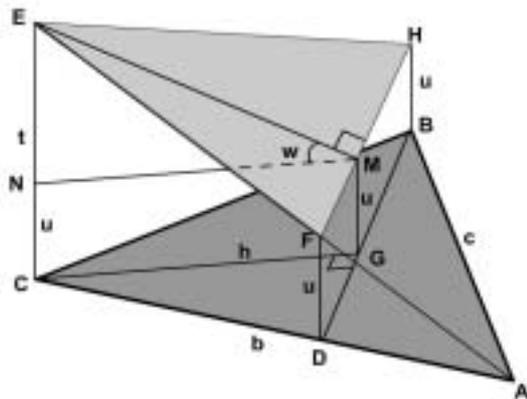
- Ambil titik trial yang merupakan titik pada himpunan closed vertex dengan nilai T terkecil/minimum.
- Tetapkan status titik trial menjadi fix.
- Tetapkan juga status tiap titik tetangga dari titik trial (titik yang terhubung oleh satu edge / garis penghubung dengan titik trial) yang bukan anggota dari himpunan fixed vertex menjadi close. Hitung nilai T dari tiap titik tersebut. Apabila T yang baru dihitung lebih kecil dari T yang lama maka update nilai T dengan nilai T yang baru, jika tidak maka nilai T yang lama tidak perlu diganti.
- Kembali ke langkah pertama tahap perulangan.

Update Sethian's Method

Proses ini dilakukan waktu menghitung nilai T di suatu titik dengan bantuan nilai T dari 2 titik lainnya yang berada pada segitiga yang sama. Misalkan titik yang ingin dihitung nilai T nya adalah titik C dan 2 titik lain yang berada dalam satu segitiga dengan titik C adalah titik A dan B , maka nilai T pada titik C dapat dihitung menggunakan rumus:

$$T(C) = T(A) + t \quad (2)$$

Tujuan dari *Update Sethian's Method* adalah untuk mencari nilai t . Caranya ialah dengan membangun sebuah bidang $\triangle EFH$ di atas $\triangle ABC$ dimana $\triangle EFH$ tersebut mempunyai kemiringan terhadap $\triangle ABC = 1$. Sudut $EMN = 45^\circ$ (gambar 2). Didefinisikan juga nilai $u = T(B) - T(A)$.



Gambar 2. Konstruksi ΔEFH yang Mempunyai Gradien Terhadap $\Delta ABC = 1$

Dari gambar 2, maka dapat dirumuskan bahwa:

$$\frac{t - u}{h} = w \tag{3}$$

dimana h adalah panjang garis MN dan w adalah kemiringan ΔEFH terhadap ΔABC , jadi $w = \tan \angle EMN = 1$. Dengan bantuan persamaan $t/b = DF/AD = u/AD$, nilai AD dapat ditulis: $AD = bu/t$. Kemudian nilai CD dapat ditulis: $CD = b - AD = b - bu/t = b(t - u)/t$. Pada ΔABC , dengan dalil Sinus: $\sin \phi = CD \sin \theta / BD$ dan dalil Cosinus: $BD^2 = a^2 + CD^2 - 2aCD \cos \theta$, maka diperoleh:

$$h = a \sin \phi = a \frac{CD}{BD} \sin \theta = \frac{a \cdot CD \sin \theta}{\sqrt{a^2 + CD^2 - 2aCD \cos \theta}} \tag{4}$$

Kemudian dengan mengganti nilai CD dan memasukkan persamaan (3) ke dalam persamaan (4) akan diperoleh:

$$(a^2 + b^2 - 2ab \cos \theta)t^2 + 2bu(a \cos \theta - b)t + b^2(u^2 - w^2 a^2 \sin^2 \theta) = 0 \tag{5}$$

Nilai t yang dicari merupakan solusi dari persamaan (5). Agar titik update G tetap berada di dalam ΔABC maka harus memenuhi syarat persamaan:

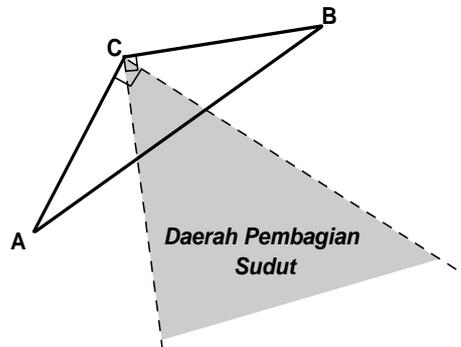
$$a \cos \theta \leq \frac{b(t - u)}{t} \leq \frac{a}{\cos \theta} \tag{6}$$

Selain itu solusi nilai t dari persamaan (5) harus memenuhi: $u < t$. Jika kedua syarat tersebut terpenuhi maka untuk menghitung nilai T pada titik C digunakan $T(C) = \min\{T(C), T(A) + t\}$, jika tidak terpenuhi maka nilai T pada titik C dihitung dengan $T(C) = \min\{T(C), T(A) + b, T(B) + a\}$.

Unfolding Triangles

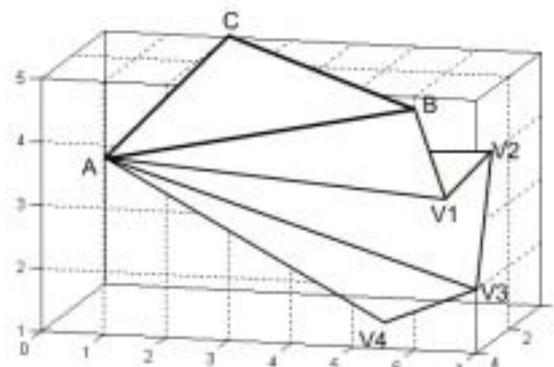
Proses ini dilakukan ketika sudut C pada segitiga besarnya lebih dari 90° atau sudut C merupakan sudut tumpul akan mengakibatkan nilai $\cos \theta$

negatif. Untuk itu jika sudut C merupakan sudut tumpul, maka perlu dilakukan proses *Unfolding Triangles* untuk membagi sudut C tersebut menjadi 2 buah sudut lancip.

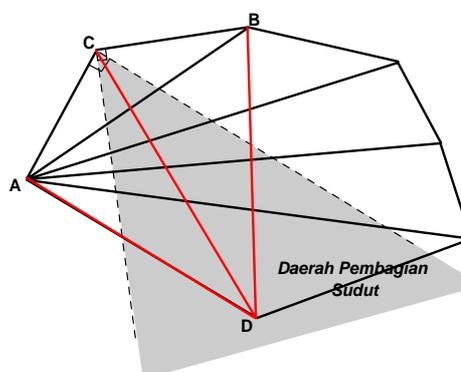


Gambar 3. ΔABC dengan Sudut C adalah Sudut Tumpul

Ide dari proses *Unfolding Triangles* ini adalah membuat daerah pembagian sudut (gambar 3), lalu memperluas daerah segitiga mula-mula (ΔABC) dengan membentangkan (*unfolding*) segitiga-segitiga yang bersebelahan sampai diperoleh sebuah titik D yang berada pada daerah pembagian sudut. Dengan garis yang menghubungkan titik D dan titik C , maka sudut C tersebut telah dibagi menjadi 2 buah sudut lancip yaitu : $\angle ACD$ dan $\angle BCD$ (gambar 5).



Gambar 4. Permukaan sebelum Proses Unfolding Triangles



Gambar 5. Permukaan setelah Proses Unfolding Triangles

Setelah proses *Unfolding Triangles* ini selesai, maka dilakukan proses *Update Sethian's Method* pada ΔACD dan dilanjutkan pada ΔBCD , nilai T terkecil yang diambil untuk menjadi nilai $T(C)$.

Analisis Algoritma FMM on TD

Algoritma FMM on TD mempunyai kompleksitas waktu $O(n \lg n)$ dimana n adalah jumlah titik yang ada, nilai kompleksitas ini didapat dari:

- Waktu untuk meng-update nilai T dari semua titik yang ada = $O(n)$
Algoritma FMM on TD akan menghitung semua nilai T dari tiap titik yang ada sampai nilai T pada titik akhir (*end vertex*) telah diperoleh atau status titik akhir tersebut telah menjadi *fix*. Proses meng-update semua nilai T dari tiap titik tersebut berjalan dengan kompleksitas waktu $O(n)$.
- Waktu untuk memelihara struktur data min-priority queue dari himpunan *close vertex* = $O(\lg n)$
Di dalam tiap tahap perulangan dilakukan operasi-operasi push-heap, pop-heap dan decrease-key terhadap min-priority queue yang menyimpan himpunan *close vertex*. Tiap operasi tersebut berjalan dengan kompleksitas waktu $O(\lg n)$.

PEMBUATAN GEODESIC PATH

Karena *geodesic path* merupakan representasi dari *geodesic distance*, maka sebelum melakukan proses pembuatan *geodesic path*, perlu dilakukan proses FMM on TD untuk menghitung *geodesic distance* dari titik awal ke semua titik lainnya.

Inti dari pembuatan *geodesic path* adalah melakukan *back propagation* (penelusuran balik) dimulai dari titik akhir sepanjang penurunan gradien dari *distance map* sampai diperoleh titik awal. *Distance Map* adalah matriks berukuran n yang berisi *geodesic distance* tiap titik dari titik awal. n adalah jumlah titik. Secara matematis, pembuatan *geodesic path* adalah memecahkan *Ordinary Differential Equation (ODE)*:

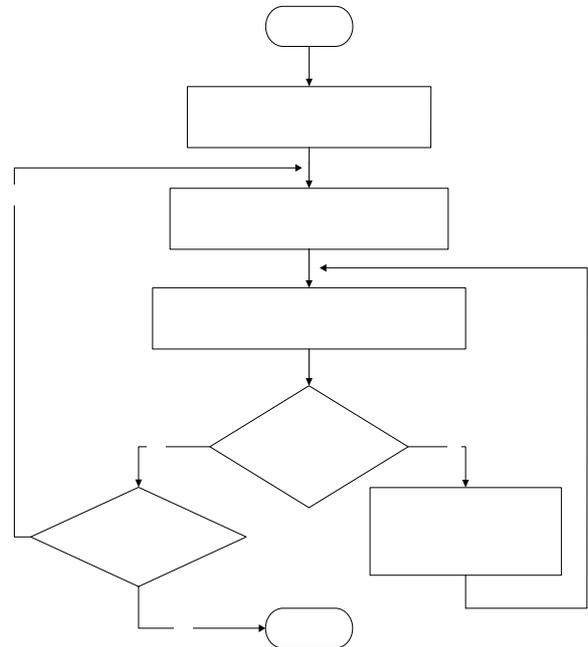
$$\frac{dX(s)}{ds} = -\nabla T$$

$$X(0) = v_0 \tag{7}$$

dimana $X(s)$ adalah fungsi kurva parametrik yang merepresentasikan lintasan *geodesic path* pada permukaan obyek 3 dimensi dan v_0 adalah titik awal.

Ada 2 hal yang perlu diperhatikan dalam melakukan pembuatan *geodesic path* yaitu: *Current Vertex* dan *Current Face*. *Current Vertex* adalah titik yang akan terus dipropagasi. Setiap kali bergerak, posisi *Current Vertex* tersebut disimpan, karena

posisi-posisi inilah yang nantinya membentuk *geodesic path*. Sedangkan *Current Face* adalah segitiga dimana *Current Vertex* akan dipropagasi.



Gambar 6. Diagram Alir Proses Pembuatan Geodesic Path

Gambar 6 menunjukkan diagram alir proses pembuatan *Geodesic Path*. Berikut langkah-langkah untuk pembuatan *geodesic path*:

- Tetapkan *Current Vertex* sebagai vertex yang akan dipropagasi. Sebagai inisialisasi awal *Current Vertex* adalah titik akhir.
- Lakukan Prosedur Khusus untuk memilih *Current Face* yaitu segitiga dimana *Current Vertex* akan dipropagasi.
- Lakukan Prosedur Umum untuk propagasi *Current Vertex* pada *Current Face* sampai *Current Vertex* di tepi *Current Face*, atau dengan kata lain sampai *Current Vertex* hampir keluar dari *Current Face*.
- Jika setelah dipropagasi *Current Vertex* berada di tengah *edge*, maka *Current Face* di-set segitiga yang merupakan tetangga dari *Current Face* sebelumnya melalui *edge* yang memuat *Current Vertex* dan lakukan kembali Prosedur Umum dengan *Current Face* yang baru.
- Jika setelah dipropagasi *Current Vertex* berada tepat pada sebuah vertex, maka lakukan Prosedur Khusus terlebih dahulu sebelum Prosedur Umum.
- Ulangi Prosedur Umum dan Prosedur Khusus di atas sampai *Current Vertex* mencapai titik awal.

Prosedur Khusus

Tujuan dari prosedur ini adalah memilih segitiga yang memuat *Current Vertex* sebagai segitiga di mana *Current Vertex* akan dipropagasikan. Langkah-langkah Prosedur Khusus adalah sebagai berikut:

- Pada *Current Vertex* pilih *vertex* tetangga dengan nilai T yang terkecil untuk menjadi *Vertex* acuan.
- Pilih segitiga yang memuat *Current Vertex* dan *Vertex* acuan, dimana *vertex* lain pada segitiga tersebut mempunyai nilai T terkecil dibandingkan pada segitiga lainnya.
- Set *Current Face* dengan segitiga yang dipilih tersebut.

Prosedur Umum

Prosedur merupakan inti dari proses pembuatan *geodesic path* karena pada prosedur ini dilakukan proses propagasi *Current Vertex*. Prosedur ini terdiri dari 2 proses utama yaitu:

- Inisialisasi *Triangular Interpolation Quadratic* pada *Current Face*.
- Propagasi Balik *Current Vertex* pada *Current Face*.

Inisialisasi *Triangular Interpolation Quadratic*

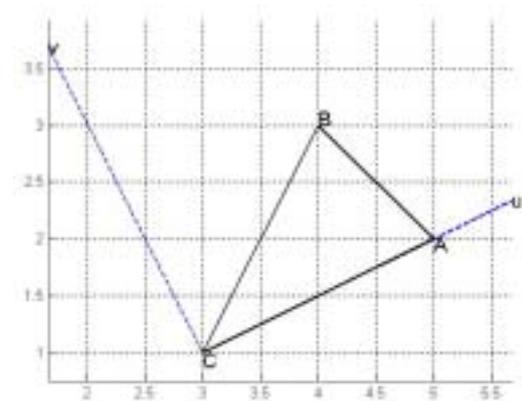
Tujuan dari langkah ini ialah untuk mencari fungsi T yang merupakan fungsi jarak suatu titik terhadap titik awal. Langkah *Inisialisasi Triangular Interpolation Quadratic* ini dilakukan di tiap segitiga yang dilewati ketika *Current Vertex* dipropagasi.

Misalkan titik yang akan dipropagasi adalah titik P dan segitiga yang memuat titik P tersebut adalah ΔABC , lalu dari ΔABC tersebut dibentuk basis ortonormal R^2 dengan titik C sebagai pusat basis. Vektor basis u diperoleh dari normalisasi vektor CA, yaitu dengan rumus:

$$u = \text{normalize}(\overline{CA}) = \overline{CA} / \text{length}(\overline{CA}) \tag{8}$$

dan vektor basis v diperoleh melalui rumus:

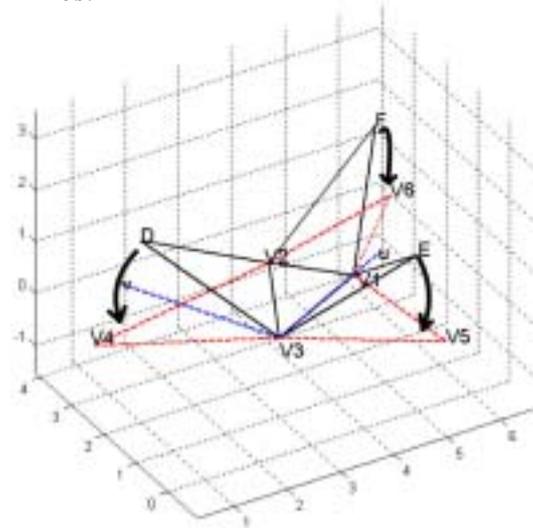
$$v = \text{normalize}(u \times \overline{CB} \times u) \tag{9}$$



Gambar 7. Ortonormal Basis R^2 pada ΔABC

Pada gambar 7, tampak ortonormal basis R^2 dengan titik C sebagai pusat basis, vektor u dan vektor v sebagai vektor basis yang saling tegak lurus.

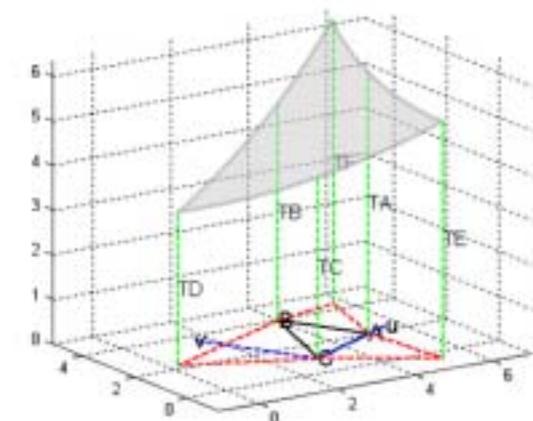
Setelah basis ortonormal dibentuk, dilakukan proses *unfolding triangles* dari ketiga tetangga dari ΔABC (gambar 8) agar menjadi sebidang dengan ΔABC sehingga diperoleh 6 buah titik $V_i(x_i, y_i)$, $i=\{1,2,3,4,5,6\}$ pada ortonormal basis R^2 yang telah dibuat sebelumnya. Tiga titik pertama mengacu pada titik A,B,C dan Tiga titik selanjutnya mengacu pada titik-titik dari ketiga segitiga yang merupakan tetangga dari ΔABC .



Gambar 8. Pendataran Segitiga-Segitiga Tetangga ke Basis R^2 pada ΔABC

Dari enam buah titik $V_i(x_i, y_i)$, $i=\{1,2,3,4,5,6\}$, tetapkan nilai fungsi $T(x,y)$ dengan nilai *geodesic distance* (nilai T_i) tiap titik tersebut yang telah dihasilkan dari proses FMM on TD. Dengan memasang tiap titik $V_i(x_i, y_i)$ dan T_i maka dapat diperoleh persamaan permukaan kuadratik (gambar 9) yang melalui ke-enam titik tersebut:

$$T(x, y) = a + bx + cy + dxy + ex^2 + fy^2 \tag{10}$$



Gambar 9. Permukaan Kuadratik yang Merepresentasikan Fungsi Jarak $T(x,y)$

Dengan menggunakan persamaan permukaan tersebut, maka fungsi vektor gradien dari setiap titik di dalam ortonormal basis ΔABC dapat dihitung dengan :

$$\nabla T(x, y) = \left\{ \frac{\partial T(x, y)}{\partial x}, \frac{\partial T(x, y)}{\partial y} \right\}$$

$$\nabla T(x, y) = \{b + dy + 2ex, c + dx + 2fy\} \quad (11)$$

vektor gradien ini kemudian dapat digunakan untuk mempropagasi titik P di dalam ΔABC sampai titik P tersebut keluar dari ΔABC

Propagasi Balik Current Vertex pada Current Face

Proses propagasi ini bertujuan untuk memprediksi lintasan *geodesic path* pada *current face*. Suatu pendekatan numerik yang dilakukan untuk memprediksi lintasan tersebut adalah dengan mempropagasi mundur *current vertex* sampai titik tersebut keluar dari *current face*.

Pendekatan Numerik tersebut dapat dirumuskan sebagai:

$$P' = P - \delta \times \nabla T(P) \quad (12)$$

dengan P adalah *current vertex* dan δ adalah konstanta skalar yang menunjukkan besar step pengurangan dari kurva. Setiap titik P yang diperoleh disimpan ke dalam suatu *list point* pembentuk *geodesic path*.

Pada gambar 10 (tengah) tampak hasil *back propagation* dengan *current vertex* asal yaitu titik A dan *current face* adalah ΔABC . Garis dari A yang memotong ΔABC menunjukkan lintasan hasil propagasi titik A, sedangkan pada gambar 10 (atas dan bawah) tampak inset dengan perbesaran 5 kali dari gambar 10 (tengah). Pada gambar ini tampak titik-titik yang merupakan hasil propagasi titik A menggunakan rumus persamaan (12) dengan $\delta = 0.01$.

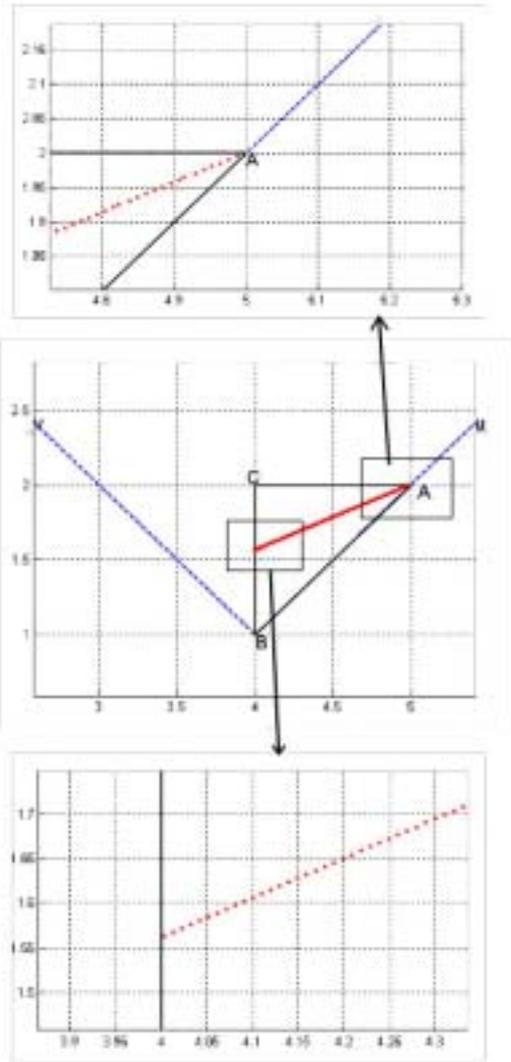
Koordinat Barycentric

Pada sebuah segitiga yang dibentuk oleh 3 titik (V_1, V_2, V_3), suatu titik P di dalam sebuah segitiga dapat dinyatakan secara unik dengan koordinat Barycentric (a_1, a_2, a_3).

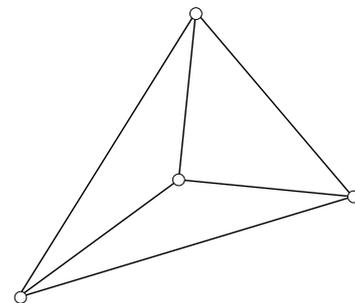
$$P = a_1V_1 + a_2V_2 + a_3V_3 \quad (13)$$

dimana:

$$a_1 + a_2 + a_3 = 1 \text{ dan } 0 \leq a_1, a_2, a_3 \leq 1$$



Gambar 10. Hasil Propagasi Balik dari Titik A pada ΔABC



Gambar 11. Koordinat Barycentric Titik P Terhadap Segitiga $V_1 V_2 V_3$

Pada gambar 11 ditunjukkan bahwa A_1 adalah luas dari ΔPV_2V_3 , A_2 adalah luas dari ΔPV_1V_3 , dan A_3 adalah luas dari ΔPV_1V_2 . Dengan nilai A_1, A_2 dan A_3 , nilai a_1, a_2, a_3 dapat dicari dengan rumus:

$$a_i = \frac{A_i}{A_1 + A_2 + A_3}, \quad i = \{1, 2, 3\} \quad (14)$$

Dalam pemakaiannya, koordinat Barycentric ini cukup istimewa karena memiliki kondisi khusus yaitu:

- Jika salah satu dari nilai a_1 , a_2 , atau a_3 bernilai 0, maka titik P berada tepat pada salah satu sisi segitiga $V_1V_2V_3$.
- Jika salah satu dari nilai a_1 , a_2 , atau a_3 lebih besar dari 1 atau kurang dari 0, maka titik P berada diluar segitiga $V_1V_2V_3$.

Di dalam proses pembuatan Geodesic Path, koordinat Barycentric ini digunakan untuk menyimpan hasil proses *back propagation* sebuah titik (*Current Vertex*) pada sebuah segitiga (*Current Face*). Selain digunakan untuk penyimpanan, koordinat Barycentric ini juga berguna untuk menentukan apakah proses *back propagation* dari current vertex itu telah berakhir atau belum. Proses *propagation* berakhir apabila current vertex telah keluar dari segitiga.

HASIL UJI COBA

Uji Coba yang dilakukan terdiri atas 3 bagian yaitu:

- Uji coba kebenaran
- Uji coba kecepatan eksekusi
- Uji coba visualisasi

Uji Coba Kebenaran

Uji coba kebenaran dilakukan untuk menguji validitas dari *geodesic distance* yang dihitung menggunakan algoritma FMM on TD. Percobaan yang dilakukan adalah melakukan proses FMM on TD untuk menghitung *geodesic distance* dari sebuah titik awal ke semua titik lainnya (*single source shortest path problem*). Evaluasi yang dilakukan terhadap hasil uji coba ini meliputi perhitungan rata-rata kesalahan (*error rate*). Rumus perhitungan tingkat kesalahan adalah sebagai berikut:

$$err = \frac{abs(A - B)}{B} \times 100\% \quad (15)$$

dimana:

err = tingkat kesalahan.

A = *geodesic distance* hasil dari FMM on TD.

B = *geodesic distance* sebenarnya.

Uji coba ini dilakukan terhadap bentuk model permukaan bidang datar. *Geodesic Distance* pada permukaan bidang datar adalah jarak euclidian. Pada

bentuk model tersebut dibuat 8 file model yang memiliki resolusi yang berbeda. Semakin tinggi resolusi maka semakin banyak jumlah titik/vertex dan segitiga/face yang terdapat pada model tersebut.

Pada tiap model dilakukan sekali percobaan, kemudian dihitung *error rate*-nya. Hasil percobaan pada bentuk permukaan bidang datar dapat dilihat pada tabel 1.

Tabel 1. Hasil Uji Coba pada Bentuk Permukaan Bidang Datar dengan Berbagai Resolusi

Nama File	Jumlah Titik	Jumlah Segitiga	Error Rate
flat.ply	9	8	7,9203%
flat1.ply	25	32	7,5089%
flat2.ply	81	128	6,0580%
flat3.ply	289	512	4,3407%
flat4.ply	1089	2048	2,8702%
flat5.ply	4225	8192	1,7988%
flat6.ply	16641	32768	1,0885%
flat7.ply	66049	131072	0,6413%
Rata-rata			4,0283375%

Uji Coba Kecepatan Eksekusi

Uji coba yang dilakukan adalah melakukan proses FMM on TD untuk menghitung *geodesic distance* dari sebuah titik awal ke semua titik lainnya (*single source shortest path problem*). Evaluasi yang dilakukan terhadap hasil uji coba ini adalah perhitungan rata-rata kecepatan eksekusi.

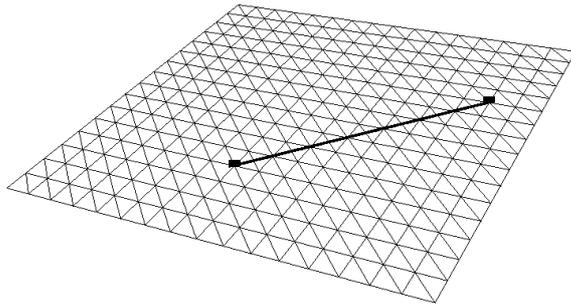
Uji coba ini dilakukan dengan menggunakan 10 file model yang berbeda-beda resolusinya, diurutkan dari resolusi rendah ke resolusi tinggi. Pada tiap model, percobaan dilakukan sebanyak 10 kali dengan inputan titik awal yang berbeda-beda ditentukan secara acak (*random*). Hasil percobaan dapat dilihat pada tabel 2.

Tabel 2. Hasil Uji Coba pada berbagai file model dengan berbagai resolusi

Nama File	Jumlah Titik	Rata-rata waktu (ms)
dragon.ply	5352	158
bunny.ply	17460	525,1
foot.ply	25845	682,9
hand.ply	38219	1041,9
horse.ply	48478	1399,9
buddha.ply	59958	1884,1
shoe.ply	78239	2202,5
topo.ply	88596	2671,9
mountain.ply	152709	4740,3

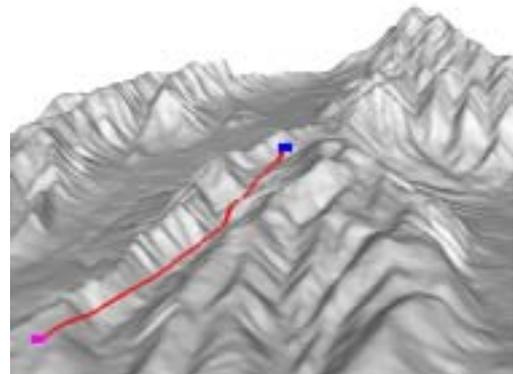
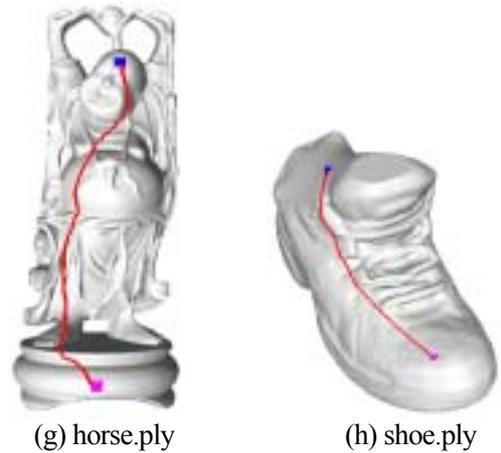
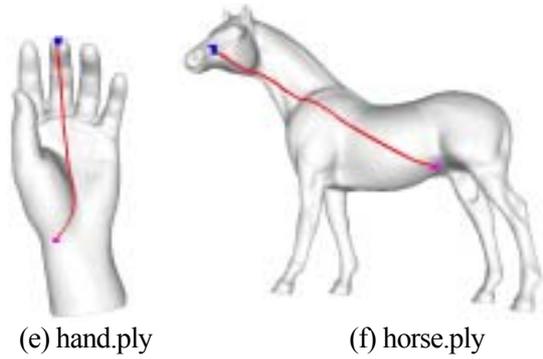
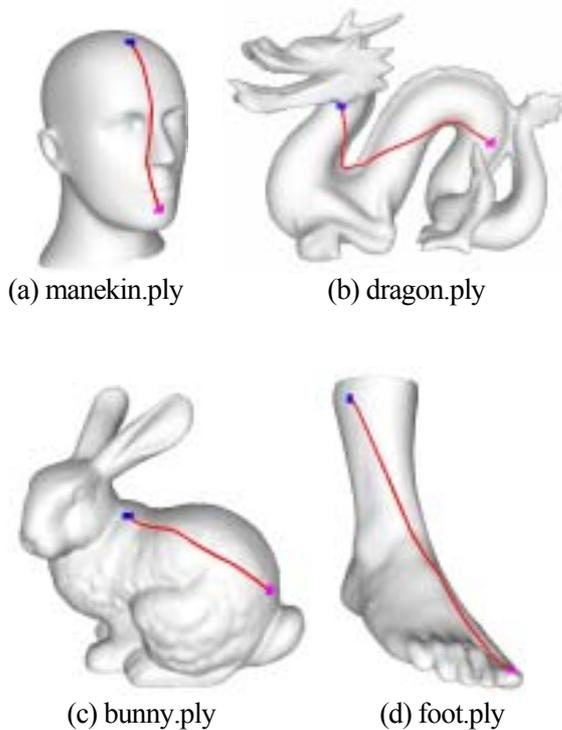
Uji Coba Visualisasi

Pada uji coba ini dilakukan proses pembuatan *geodesic path* dan kemudian divisualisasikan pada model. *Geodesic path* pada permukaan bidang datar adalah sebuah garis lurus yang menghubungkan kedua titik tersebut. Gambar 12 adalah hasil uji coba pembuatan *geodesic path* pada permukaan bidang datar berupa garis lurus yang menghubungkan titik awal dan titik akhir.

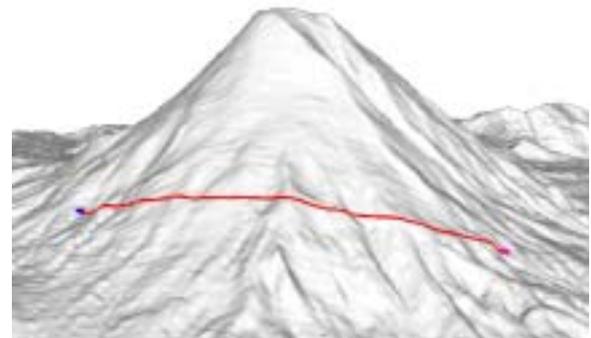


Gambar 12. Hasil Uji Coba Pembuatan Geodesic Path pada Permukaan Bidang Datar

Selain pada permukaan bidang datar, uji coba visualisasi *geodesic path* juga dilakukan pada 10 file model yang digunakan pada uji coba kecepatan eksekusi. Hasil dari ujicoba ini dapat dilihat pada gambar 13.



(i) topografi.ply



(j) mountain.ply

Gambar 13. Hasil Uji Coba Visualisasi Pembuatan Geodesic Path pada Berbagai File Model

KESIMPULAN

Dari hasil ujicoba yang telah dilakukan dapat diambil kesimpulan:

- Algoritma *Fast Marching Method on Triangulated Domain* dapat digunakan sebagai pendekatan untuk menghitung geodesic distance antara 2 titik pada permukaan obyek *triangular mesh* dengan tingkat keakuratan lebih dari 95 %.
- Kecepatan eksekusi algoritma FMM on TD dipengaruhi oleh jumlah titik. Kompleksitas waktunya adalah $O(n \lg n)$, dimana n adalah jumlah titik yang dimiliki oleh model. Jadi semakin banyak titik yang ada maka semakin lama waktu yang dibutuhkan untuk melakukan perhitungan.
- Banyaknya segitiga yang membentuk *triangular mesh* berpengaruh pada *error rate* dari hasil algoritma FMM on TD. Semakin banyak segitiga yang ada semakin sedikit *error rate* yang dihasilkan.

DAFTAR PUSTAKA

1. Sethian J. A., *Level Set Methods and Fast Marching Methods*, 2nd ed, Cambridge University Press, 1999.
2. Sethian J. A. *Fast Marching Methods*, SIAM Review, no. 41, July 1999.
3. Kimmel R. and Sethian J. A., *Computing Geodesic Paths on Manifolds*. Proc. Natl. Acad. Sci., vol. 95, no. 15, pp. 8431-8435, 1998.
4. Gil Zigelman, Ron Kimmel, and Nahum Kiryati, "Texture Mapping Using Surface Flattening via Multidimensional Scaling". *IEEE Transaction on Visualization and Computer Graphics*, Vol 8, No 2, April-June 2002
5. Adi L. and Kimmel R., *Interactive Edge Integration on Painted Surfaces*. Computer Science, Technion-Israel Institute of Technology, 2003.
6. Peyré G. and Cohen L. D., *Geodesic Computations for Fast and Accurate Surface Flattening*. Proc. IEEE VLISM, 2003
7. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, 2nd Edition, The MIT Press, 2001.
8. Marcin Novotni and Reinhard Klein, *Computing Geodesic Distances on Triangular Meshes*. Institut für Informatik II, Bonn, Germany, 2002.