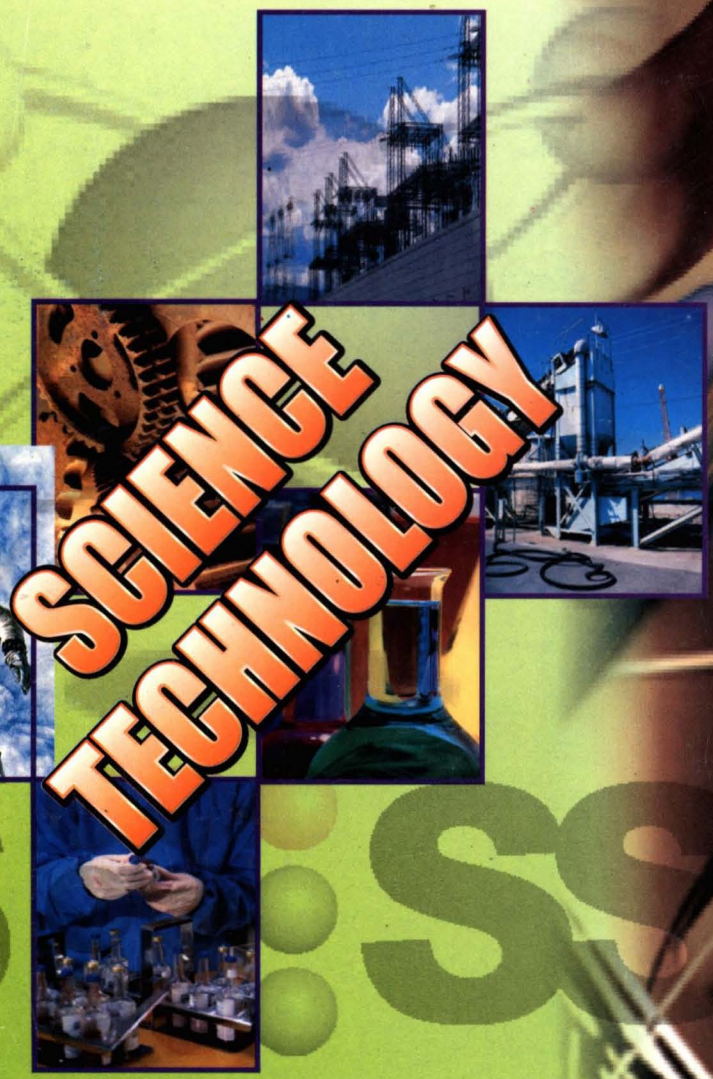
UNIVERSITI TEKNOLOGI MARA

ISO 9001:2000 No Sijil :0500132

Globalising Knowledge and Information



NATIONAL SEMINAR ON SCIENCE TECHNOLOGY & SOCIAL SCIENCES 2006

30-31 May 2006

Swiss Garden Resort & Spa

Kuantan, Pahang

Kaedah Penjanaan Bayang Berbantukan OPENGL®

Umar Baba
Mohd. Nain Hj Awang
Mazwin Tan

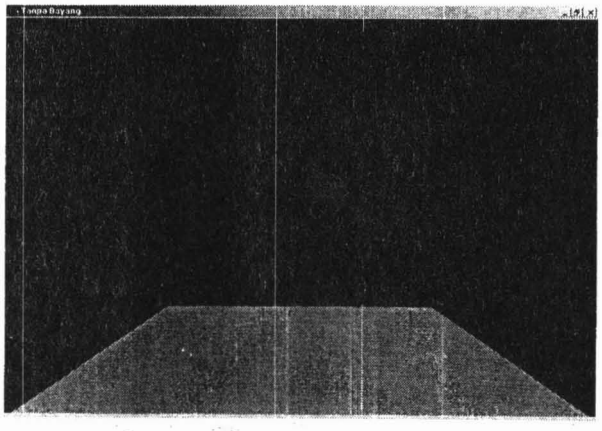
ABSTRAK

Kini komputer dapat menjana gambar yang sukar dibezakan dengan gambar yang diambil oleh kamera. Untuk menjana gambar sedemikian banyak faktor diambilkira. Satu daripadanya ialah bayang-bayang objek yang terdapat dalam sesuatu gambar. Bayang akan diperoleh apabila terdapat objek yang menghalang cahaya dari sumber cahaya seperti lampu menerangi sesuatu kawasan. Kawasan yang terhalang cahaya adalah kawasan bayang dan biasanya diwarnakan dengan warna yang gelap atau hitam. Penentuan kawasan bayang ini melibatkan kedudukan sumber cahaya dan kedudukan pemerhati yang dianggap melihat gambar. Terdapat beberapa kaedah menjana bayang pada komputer seperti kaedah unjuran, isipadu bayang dan pemetaan bayang. Kertas kerja ini akan membincangkan hanya kaedah unjuran. Dalam kaedah unjuran ini dua teknik dibincangkan iaitu teknik palsu dan teknik unjuran sesatah. Penjanaan bayang disempurnakan dengan bantuan antara muka grafik OpenGL menerusi bahasa pengaturcaraan Microsoft Visual C++ versi 6.0. OpenGL ialah suatu antara muka grafik yang terdiri daripada sekumpulan perpustakaan grafik yang mampu menjana grafik 2 dimensi dan 3 dimensi yang bermutu tinggi. Untuk mengatasi masalah komunikasi dengan sistem pengoperasi tetingkap OpenGL digunakan bersama-sama perpustakaan Graphics Library Utilities Toolkit (GLUT). Jadi pengaturcara tidak perlu bimbang tentang cara-cara berkomunikasi dengan sistem pengoperasi yang berasaskan tetingkap. Pengaturcara hanya perlu tahu perintah-perintah dari GLUT yang sesuai untuk mengendalikan tetingkap dan kebanyakan masa dihabiskan dalam penjanaan grafik sahaja. Kertas kerja ini juga membincangkan perintah-perintah OpenGL yang diperlukan untuk menghasilkan bayang pada kedua dua teknik unjuran.

Kata kunci: *Bayang-bayang, pengaturcaraan, OpenGL*

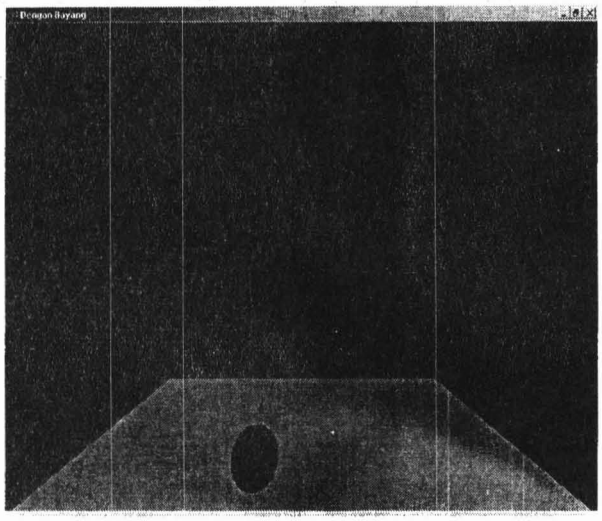
Pengenalan

Pada awal perkembangan grafik komputer objek tiga dimensi yang dijana oleh komputer tidak mempunyai bayang. Ini menyebabkan objek kelihatan terapung dalam persekitarannya seperti dalam Rajah 1.



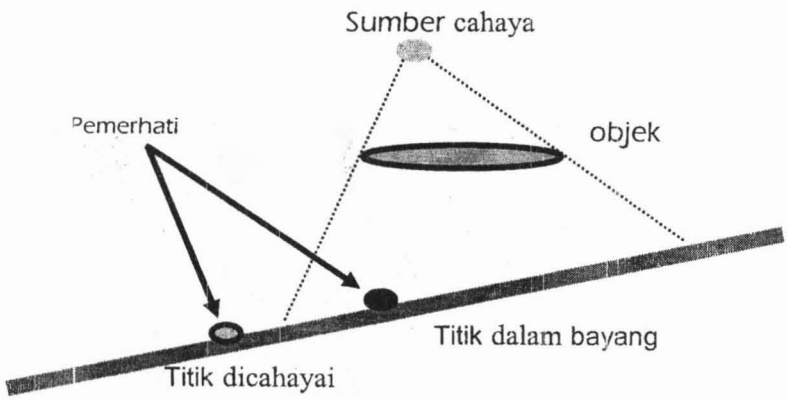
Rajah 1: Objek Tanpa Bayang

Gambar janaan komputer yang lengkap dengan bayang-bayang objek lebih mendekati realiti berbanding gambar tanpa bayang-bayang. Kewujudan baying, juga dapat memberikan maklumat tambahan tentang objek yang dijana komputer seperti dalam Rajah 2.



Rajah 2: Objek Dengan Bayang

Bayang ditakrifkan sebagai kawasan yang tidak terkena cahaya kerana cahaya terhalang oleh objek legap seperti dalam Rajah 3. Gambar tanpa bayang boleh dianggap bahawa pemerhati berada di tempat sumber cahaya. Jika pemerhati tidak berada di tempat sumber cahaya, dia akan melihat bayang pada gambar tersebut. Oleh itu proses membuat bayang ialah proses mengecam kawasan-kawasan bayang dan mewarnakan kawasan tersebut dengan warna yang gelap biasanya hitam. Ada berbagai cara mengecam kawasan bayang ini. Crow (1977) telah menjana bayang dengan keadah isipadu bayang. Williams (1978) telah mencadangkan kaedah pemetaan bayang dan Blinn (1988) telah memperkenalkan kaedah unjuran. Kertas kerja ini akan menumpukan perhatian kepada kaedah penjanaan bayang berasaskan unjuran. Dalam kaedah ini dua teknik telah dikaji iaitu teknik palsu dan teknik unjuran sesatah. Kedua-dua teknik dilaksanakan dengan bantuan antara muka grafik OpenGL melalui bahasa pengaturcaraan Microsoft Visual C++ versi 6.0.



Rajah 3: Titik dalam Bayang jika Cahaya dihalangi.

OpenGL adalah suatu API (Application Programmer's Interface) grafik yang diperkenalkan oleh Syarikat Silicon Graphics pada tahun 1992. Ia berupaya menjana gambar 2D dan 3D yang bermutu. Malangnya penggunaannya terhad kepada mereka yang mahir dengan pengaturcaraan berasaskan objek dan tetingkap. Apabila fail perpustakaan GLUT diperkenalkan, baharulah penjanaan grafik menjadi mudah kerana pengaturcara tidak lagi perlu risau tentang pengaturcaraan berasaskan objek dan tetingkap kerana telah dikendalikan dengan cekap oleh fail-fail perpustakaan GLUT. Pengaturcara hanya perlu tahu menggunakan perintah-perintah dari fail GLUT yang sesuai sahaja. Semua gambar dalam kertas kerja ini dipaparkan dengan bantuan OpenGL.

OpenGL boleh digunakan dalam berbagai sistem pengoperasi dan bahasa pengaturcaraan, Baba (2003) telah menjelaskan cara-cara memasang dan menggunakan OpenGL pada sistem pengoperasi Microsoft Windows dan juga

cara-cara menggunakan perintah-perintah OpenGL dalam pengaturcaraan melalui Microsoft Visual C++ versi 6.0. Perintah-perintah OpenGL yang minimum untuk menghasilkan aturcara grafik dalam Microsoft Windows diterangkan oleh Baba (2004). Selanjutnya Baba dan Awang (2004) juga telah menerangkan cara-cara menjana objek 3D. dan beranimasi.

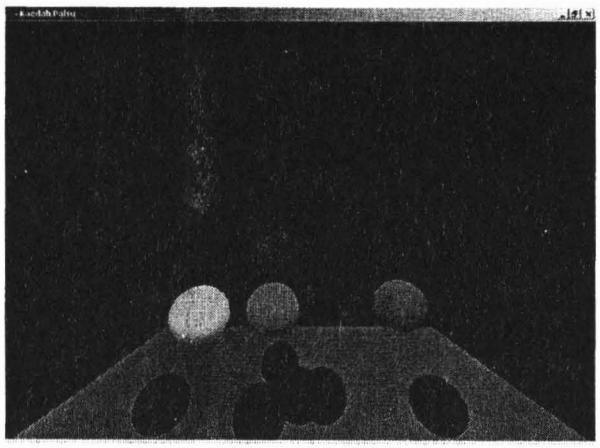
Kaedah penjanaan bayang dengan bantuan antara muka grafik OpenGL telah memudahkan pembinaan aturcara untuk menghasilkan gambar yang mengandungi bayang. Ia menjadi lebih pantas dan mudah. Kebolehan menghasilkan gambar berbayang yang berkualiti dengan cepat dan mudah sangat diperlukan terutamanya dalam industri penghasilan gambar-gambar beranimasi.

Dalam membincangkan penggunaan perintah-perintah OpenGL, hanya yang melibatkan penjanaan bayang sahaja dibincangkan disini. Semua perintah-perintah asas yang diperlukan untuk membina gambar 3D termasuk perintah-perintah dari perpustakaan GLUT tidak dibincangkan di sini. Semua ini boleh dirujuk dari Baba(2003), Baba(2004) dan Baba dan Awang (2004).

Kaedah Penjanaan Bayang

Teknik Palsu

Teknik yang paling mudah. Bagi teknik ini kawasan bayang dicam dengan mengandaikan kedudukan sumber cahaya dan seterusnya kawasan bayang pada satah diperoleh. Di kawasan bayang, objek itu sendiri dilukis sekali lagi dengan menggunakan perintah OpenGL, glShadeModel(GL_FLAT). Perintah ini akan melukis objek dikawasan bayang dan objek kelihatan rata dan diberi warna hitam. Amnya kedudukan dan saiz bayang tidak tepat tetapi menurut Woo et al. (1990) teknik ini sangat berkesan bagi beberapa aplikasi seperti penjanaan gambar dalam permainan video. Objek sebenar akan dilukis dengan perintah glShadeModel(GL_SMOOTH). Rajah 4 adalah gambar yang dijana mengikut kaedah ini.



Rajah 4: Bayang yang dijana dengan Teknik Palsu

Dalam teknik ini algoritma grafik boleh disusun seperti berikut:

i) Lukis latar belakang
ii) Lukis objek
iii) Lukis bayang objek

Jadi secara kasarnya aturcaranya tidak sukar.

Teknik Unjuran Sesatah

Bagi kaedah ini tempat sumber cahaya hendaklah ditentukan, kemudian cahaya akan dipancarkan ke objek dan terus ke satah latar belakang. Kawasan bayang pada satah dicam melalui pengiraan menggunakan konsep unjuran. Garis

cahaya dianggap sebagai garis unjuran. Kawasan yang terhalang daripada cahaya adalah kawasan bayang. Untuk memudahkan pengiraan satah latar belakang dianggap berada pada satah xy, Jika $L(x_l, y_l, z_l)$ kedudukan sumber cahaya, $P(x_p, y_p, z_p)$ titik pada objek dan $S(x_s, y_s, z_s)$ titik bayang maka persamaan garis berbentuk vektor ialah

$$S = P + \alpha(P - L)$$

Oleh kerana $z_s = 0$ maka $\alpha = \dfrac{-z_p}{z_p - z_l}$

$$x_s = \frac{x_l z_p - x_p z_l}{z_p - z_l} \quad \text{dan} \quad y_s = \frac{y_l z_p - y_p z_l}{z_p - z_l}$$

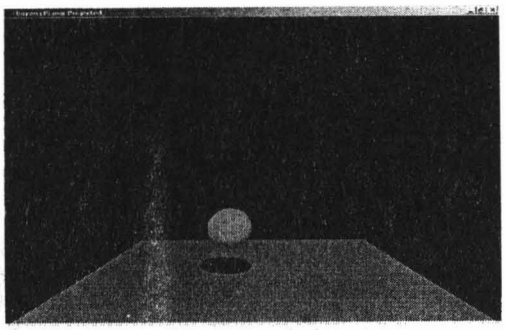
Dari itu diperoleh matriks penjelmaan untuk bayang dalam koordinat homogen

$$M = \begin{pmatrix} z_l & 0 & 0 & 0 \\ 0 & z_i & 0 & 0 \\ -x_l & -y_l & 0 & -1 \\ 0 & 0 & 0 & z_i \end{pmatrix}$$

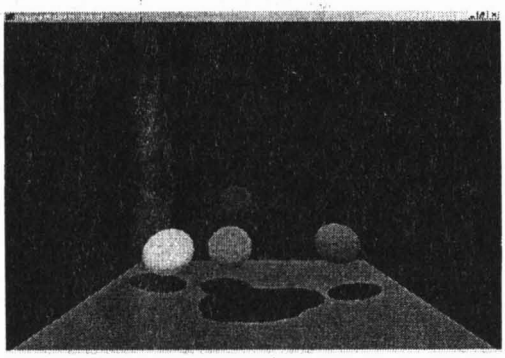
Maka sebarang titik P pada objek akan mempunyai titik bayang S pada satah yang diperoleh seperti berikut:

$$S = P * M$$

Teknik ini menghasilkan bayang pada hanya permukaan rata. Ia menjana bayang yang betul bagi sesuatu objek. Gambar-gambar yang dihasilkan dalam Rajah 5 dan Rajah 6 adalah mengikut kaedah unjuran sesatah.



Rajah 5: Bayang satu Objek dijana dengan Teknik Unjuran Sesatah



Rajah 6: Bayang lebih dari satu Objek dijana dengan Teknik Unjuran Sesatah.

Dalam teknik ini algoritma grafik disusun seperti berikut:

Tetapkan kedudukan sumber cahaya
Lukis latar belakang
Lukis objek
Dapatkan matriks unjuran untuk bayang
Dapatkan kawasan bayang
Lukis bayang.

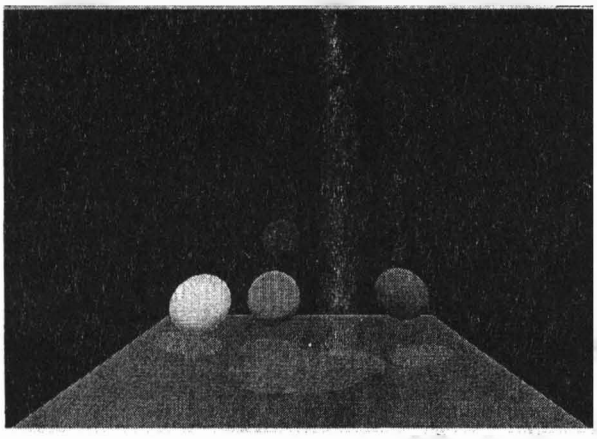
Di sini kerja yang banyak dilakukan untuk mengira matriks unjuran dan kawasan bayang mengikut kedudukan sumber cahaya. Perintah-perintah OpenGL telah digunakan dalam teknik adalah seperti berikut:

glLightfv digunakan untuk menakrifkan lampu dan kedudukan lampu
glNormal digunakan untuk mengira normal poligon atau satah
glShadeModel digunakan untuk melorek objek
glMultMatrix adalah matriks penjelmaan yang digunakan untuk mendapatkan kawasan bayang
glPolygonOffset untuk membuang nilai-nilai kedalaman yang dijana oleh perenderan poligon. Perintah ini digunakan bersama-sama perintah

glEnable(GL_POLYGON_OFFSET_FILL) untuk memulakan glPolygonOffset
glDisable (GL_POLYGON_OFFSET_FILL).untuk menamatkan glPoligonOffset

Bagi kedua-dua teknik, bayang yang terhasil kelihatan hitam. Ini tidak sesuai dengan apa yang dilihat hari-hari. Ada dua cara mengatasinya. Pertama dengan menggunakan warna yang kurang gelap sebelum melukis bayang. Ini dilakukan dengan meletakan parameter yang sesuai dalam perintah OpenGL glColor( ). Cara ini memerlukan kemahiran pengaturcara dalam memilih warna yang sesuai. Rajah 7 menunjukkan warna bayang hasil dari cara ini. Cara kedua ialah menggunakan kemudahan yang tersedia dalam OpenGL bagi mengurangkan kegelapan warna

bayang iaitu dengan menggunakan penimbal stensil dan beberapa perintah-perintah OpenGL yang lain. Cara ini menghasilkan warna bayang yang lebih baik. Rajah 8 menunjukkan warna bayang yang dihasilkan menggunakan penimbal stensil.



Rajah 7: Warna Bayang dilukis dengan Pilihan Warna dalam glColor ( )

Untuk menggunakan penimbal stensil beberapa langkah perlu diambil iaitu

i) Dalam perintah glutInitDisplayMode masukkan opsyen GLUT_STENCIL yang bertujuan memberitahu komputer supaya menyediakan suatu penimbal untuk stensil.

ii) Penimbal dibersihkan dengan memasukkan opsyen GL_STENCIL_BUFFER_BIT ke dalam perintah glClear( ).

iii) Sebelum bayang dilukis panggil perintah-perintah

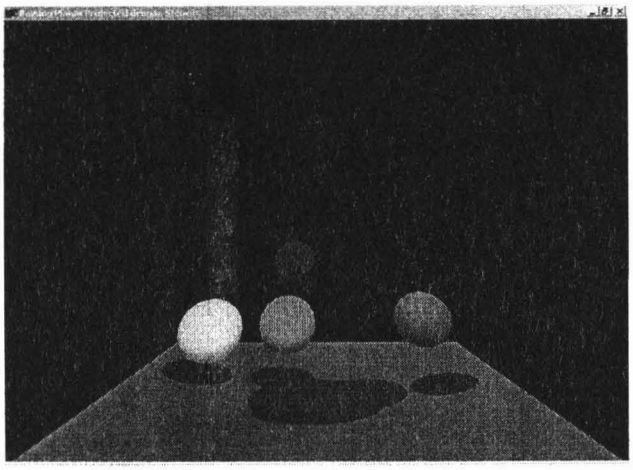
a. glEnable(GL_STENCIL_TEST)
b. glStencilFunc( )
c. glStencilOp( )
d. glEnable(GL_BLEND)

untuk menggunakan penimbal stensil

iv) Selepas bayang diwarnakan, panggil perintah-perintah

e. glDisable(GL_STENCIL_TEST)
f. glDisable(GL_BLEND)

untuk menamatkan penggunaan penimbal stensil



Rajah 8: Warna Bayang menggunakan Penimbal Stensil

Kesimpulan

Teknik palsu menghasilkan bayang yang merupakan bentuk hampiran unjuran objek ke atas satah berdasarkan agakan pengaturcara. Bagi objek yang mudah agakan pengaturcara mungkin baik tetapi bagi objek yang kompleks mungkin agakan pengaturcara kurang baik. Teknik unjuran sesatah menghasilkan bayang yang baik kerana kawasan bayang diperoleh melalui pengiraan unjuran objek dari arah sumber cahaya ke atas satah unjuran. Bayang yang terhasil sentiasa mengambil bentuk objek asal. Teknik-teknik ini tidak boleh digunakan untuk menjana bayang pada permukaan tidak rata. Bagi penjanaan bayang ke atas permukaan tak rata kaedah lain perlu dicari.

Memilih teknik penjanaan bayang ke atas permukaan rata bergantung kepada aplikasi. Teknik palsu mungkin sesuai bagi komputer tak berkuasa tinggi, pembinaan permainan video yang tak memerlukan bentuk bayang yang baik tetapi perlukan penjanaan bayang yang cepat dan objek dalam gambar tak banyak.Teknik unjuran sesatah memerlukan komputer berkuasa lebih tinggi kerana pengiraan bayang memerlukan masa dan ruang ingatan yang banyak. Ia mungkin sesuai untuk menghasilkan gambar statik yang mempunyai banyak objek dan penghasilan gambar bermutu tinggi.

Warna bayang boleh dikurangkan kehitamannya sama ada menggunakan pemilihan warna yang kurang gelap atau menggunakan penimbal stensil di mana OpenGL sendiri mengadun warna hitam supaya sesuai dengan persekitaran.

Perakuan

Kami sukalah memperakukan bahawa penyelidikan ini telah dijalankan di bawah geran penyelidikan Jangka Pendek USM 304/PMATHS/636016.

Rujukan

Baba, U. (2003). Panduan menggunakan OpenGL® pada Microsoft windows. *Laporan Teknik no. M02/03. Pusat Pengajian Sains Matematik*. Universiti Sains Malaysia.

Baba, U. (2004). *OpenGL® as a tool for graphics generation on a windows environment*. Paper presented at the Regional Conference on Ecological and Environmental Modeling (ECOMOD 2004), Georgetown, 15-16 September.

Baba, U. & Awang, M.N. (2004). *Penghasilan gambar beranimasi menggunakan OpenGL®*. Kertas kerja dibentangkan di Simposium Kebangsaan Sains Matematik ke-XII, Universiti Islam Antarabangsa, Gombak, 23-24 Disember.

Blinn, J.F. (1988). Me and my (fake) shadow. *IEEE Computer Graphics and Applications*, 8(1): pp. 82-86.

Crow, F.C, (1977). Shadow algorithms for computer graphics. *Computer Graphics,* 11(3): pp. 242-248.

Williams, L. (1978). Casting curved shadows on curved surfaces. *ComputerGraphics,* vol.12 : pp. 270-174.

Woo, A., Poulin, P. & Fournier, A. (1990). A survey of shadow algorithms. *IEEE Computer Graphics and Applications*, 10(6): pp. 13-32.

UMAR BABA, Pusat Pengajian Sains Matematik, Universiti Sains Malaysia, 11800 Minden Pulau Pinang. umar@cs.usm.my

MOHD NAIN HJ AWANG, Pusat Pengajian Jarak Jauh, Universiti Sains Malaysia, 11800 Minden Pulau Pinang. mnain@usm.my

MAZWIN TAN, Centre of Foundation Studies, International University College of Technology Twintech, 52200 Kuala Lumpur. mazwin_tan@yahoo.com, mazwin_cfs@iuctt.edu.my