



PEMBANGKITAN CAHAYA VIRTUAL DINAMIS PADA AUGMENTED REALITY MENGGUNAKAN CANNY EDGE DETECTION, CONTOUR FINDING DAN UNITY LIGHT RENDERER

Yoze Rizki¹⁾, Mochamad Hariadi²⁾

¹⁾Ilmu Komputer, Universitas Muhammadiyah Riau,
Pekanbaru, Indonesia

²⁾Jurusan Teknik Elektro, Institut Teknologi Sepuluh November,
Surabaya, Indonesia

email: yozerizki@umri.ac.id⁽¹⁾ mochar@ee.its.ac.id⁽²⁾

ABSTRAK

Dalam *Augmented Reality*, Faktor pencahayaan objek menjadi hal perlu diperhatikan. Pencahayaan objek *virtual* yang selama ini diatur secara manual dianggap kurang realistis. Sistem pembangkitan cahaya yang dinamis dan *realtime* dibutuhkan untuk membuat sebuah aplikasi *Augmented Reality* lebih realistis. Dengan adanya pembangkitan cahaya virtual dinamis, pencahayaan objek AR mampu dibangkitkan pada posisi, dan intensitas warna cahaya yang sesuai dengan sumber cahaya dari lingkungan nyata disekitar objek AR. Pada penelitian ini dibuat sebuah sistem pembangkitan cahaya dengan acuan warna cahaya dan posisi cahaya pada lingkungan nyata. Pengambilan warna cahaya sumber dilakukan dengan mengambil nilai warna pada sebuah pixel dengan intensitas kecerahan tertinggi Dan pengambilan posisi sumber cahaya diambil dengan menentukan titik axis dari sebuah piksel pada marker yang memiliki tingkat kecerahan tertinggi. Dari hasil percobaan 1 sampai dengan percobaan 4 didapatkan hasil persentase kesamaan posisi 92.10% dari posisi sebenarnya. Dari hasil percobaan warna, didapatkan hasil persentase kesamaan warna cahaya hasil dibandingkan dengan warna cahaya sumber adalah 66.66%. Persentase kesamaan warna cahaya yang rendah disebabkan pantulan cahaya pada kertas marker yang bernilai keabuan tinggi (> 180) dan sumber cahaya lain sehingga mempengaruhi hasil cahaya yang dibangkitkan oleh *game engine* Unity 3D pada simulasi..

Kata kunci: *Augmented Reality*, Cahaya Virtual, RGB.

ABSTRACT

In Augmented Reality, the object lighting factor becomes a matter of concern. Lighting of virtual objects that have been manually generated is considered less realistic. Real time dynamic light generation system is needed to make an Augmented Reality application more realistic. With the generation of dynamic virtual light, AR objects lighting can be generated at the position and intensity of light colors that match the light source from the real environment around the AR object. In this study a light generation system was made with reference to the color intensity of light and the direction of light in the real environment. Retrieval of the light source color is done by retrieving the color value of a pixel with the highest intensity of brightness. Retrieval of the position of the light source is done by determining the axis of the pixel on the marker image which has the highest brightness level. From the results of 1st experiment through 4th experiment, the percentage of position equality is 92.10% from the actual position. From the results of the color experiment, it was found that the percentage of the light color of the results compared with the color of the source light was 66.66%. Low percentage of color similarity caused by light reflection on high gray value on marker (> 180), and other light sources that affect the light output generated by the Unity3D game engine in the simulation

Keyword : *Augmented Reality, Virtual Light, RGB*

1.1 Latar Belakang

Pada Augmented reality, seringkali pengembang tidak memperhatikan aspek pencahayaan. Aplikasi akan diluncurkan dan disebarkan ke publik asalkan objek yang muncul sudah dapat terlihat dengan jelas oleh pengguna. Penelitian ini dibuat, dengan tujuan meningkatkan nilai realistis pada sebuah objek dimensi tiga pada sebuah sistem *augmented reality* melalui sistem komposisi pencahayaan dinamis.

objek dimensi tiga yang muncul diatas *marker AR* yang selama ini diterangi cahaya virtual yang parameternya diatur manual, seringkali tidak sesuai dengan parameter sebenarnya pada lingkungan nyata. Parameter yang dimaksud dapat berupa arah datangnya cahaya, warna cahaya yang menerangi, intensitas cahaya, dan tipe cahaya. Apabila cahaya pada lingkungan nyata muncul dari bagian kiri objek, maka cahaya sistem yang dibangkitkan juga muncul dari bagian kiri objek. Begitu pula untuk parameter lainnya seperti warna cahaya, yang akan dibangkitkan sesuai dengan warna yang menerangi objek.

Pada penelitian ini akan dibahas mengenai menghasilkan sebuah sistem pencahayaan dinamis yang mampu dibangkitkan pada posisi dan warna cahaya yang sesuai dengan sumber cahaya pada lingkungan nyata. Yang pada penerapannya dapat digabungkan dengan sistem *augmented reality*.

I. DASAR TEORI

A. Pencahayaan Pada objek Virtual

Terdapat beberapa jenis cahaya yang diterapkan dalam dunia skenario *virtual* atau *game* dimensi tiga. Dalam penelitian ini digunakan sistem pencahayaan yang terdapat pada Unity3D^[1], diantaranya:

1) *Point lights*

Point light juga dikenal dengan *Omni*, merupakan tipe pencahayaan yang menyebar dari satu titik ke segala arah. Seperti sebuah bola lampu dalam sebuah ruangan.

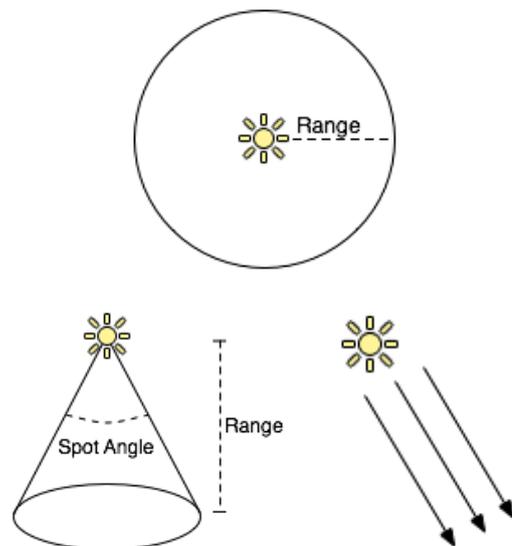
2) *Spotlights*

Sering juga disebut dengan *Spot*. Tipe ini dapat dikontrol dan diatur dengan

maksimal. Pada tipe cahaya ini selain arah, dapat pula diatur tingkat kehalusan cahaya atau sering disebut dengan *softness edges decay* dan *attenuation*.

3) *Directional lights*

Tipe ini paling sering digunakan untuk mensimulasikan cahaya langsung matahari. Disebut juga sebagai *direct lamp*, *sunlight*, *infinite*. Di Blender dikenal dengan *Sun*, dan di Unity3D dikenal dengan *Directional light*. Tipe ini mensimulasikan cahaya kepada semua objek pada sudut yang sama, walaupun antara satu objek dengan lainnya terletak ditempat yang berbeda.



Gambar 1. *Pointlight*, *Spotlight* dan *Directional light* (Sumber: Unity 3D. Unity: *Light*)^[2]

B. Histogram Citra

Informasi penting mengenai isi citra digital dapat diketahui dengan membuat histogram citra. Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relatives*) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar. Karena itu, histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara

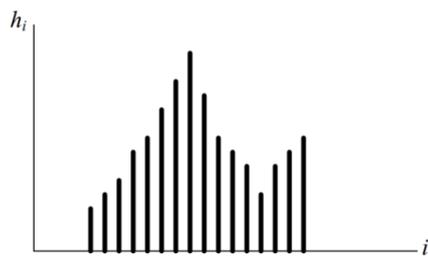
kualitatif maupun kuantitatif(Sumber: Chakravarti, Rishav)^[3].

Secara matematis histogram citra dihitung dengan rumus

$$h_i = \frac{n_i}{n}, \quad i = 0, 1, \dots, L - 1$$

yang dalam hal ini,
 n_i = jumlah pixel yang memiliki derajat keabuan i
 n = jumlah seluruh pixel di dalam citra

Plot h_i versus f_i dinamakan histogram. Gambar 2 adalah contoh sebuah histogram citra. Secara grafis histogram ditampilkan dengan diagram batang. Perhatikan dari (1) bahwa nilai n_i telah dinormalkan dengan membaginya dengan n . Nilai h_i berada di dalam selang 0 sampai 1.

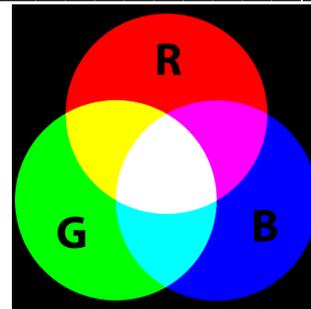


Gambar 2. Grafik histogram dari sebuah citra^[3]

C. Format RGB dan Grayscale

RGB adalah suatu model warna yang terdiri atas 3 buah warna: merah (*Red*), hijau (*Green*), dan biru (*Blue*), yang ditambahkan dengan berbagai cara untuk menghasilkan bermacam-macam warna.

Sebuah warna dalam RGB digambarkan dengan menentukan seberapa banyak masing-masing warna merah, hijau, dan biru yang dicampurkan. Warna ini dituliskan dalam bentuk triplet RGB (r, g, b), setiap bagiannya dapat bervariasi dari nol sampai nilai maksimum yang ditetapkan(Sumber S. Sabine; B. Robert; S. Steve)^[4].



Gambar 3. Penggabungan Komposisi Warna RGB menghasilkan CMY^[4].

Bila sebuah citra RGB memiliki tiga komposisi *red*, *green* dan *blue* pada setiap pixelnya, sebuah citra digital *grayscale* adalah suatu citra dimana nilai dari setiap pixel merupakan sampel tunggal. Dengan kata lain komposisi *red*, *green* dan *blue* pada tiap pixel citra *grayscale* memiliki nilai yang sama. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat.. Citra *grayscale* disimpan dalam format 8 bit untuk setiap sampel pixel, yang memungkinkan sebanyak 256 intensitas. Dimana nilai 0 menunjukkan tingkat kecerahan paling tinggi dan nilai 255 menunjukkan nilai kehitaman paling tinggi. Pada aplikasi lain seperti pada aplikasi medical imaging dan remote sensing biasa juga digunakan format 10,12 maupun 16 bit(Sumber V. Karthikeyani; K.Duraiswamy, P.Kamalakkannan)^[5].

D. Canny Edge Detection

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang cepat atau tiba-tiba (besar) dalam jarak yang singkat. Tujuan mendeteksi tepi sendiri adalah untuk mengelompokkan objek-objek dalam citra, dan juga digunakan untuk menganalisis citra lebih lanjut. Ada banyak algoritma yang digunakan untuk mendeteksi tepi, salah satu diantaranya adalah deteksi tepi *Canny* (*Canny Edge detection*)^[6]. *Canny edge detector* dikembangkan menggunakan algoritma untuk mendeteksi berbagai tepi dalam gambar.

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\Theta = \text{atan2}(G_y, G_x)$$

Dengan

Gx = arah horizontal, dan

Gy = arah vertikal

Perhitungan dalam menentukan intensitas gradien pada sebuah citra dapat diselesaikan dengan (2).

Kategori algoritma yang dikembangkan dalam deteksi tepi Canny adalah:

1. Deteksi

Kemungkinan mendeteksi titik tepi yang benar harus dimaksimalkan sementara kemungkinan salah mendeteksi titik tepi harus diminimalkan.

2. Lokalisasi

Tepi terdeteksi harus sedekat mungkin dengan tepi yang nyata.

3. Jumlah tanggapan : Satu tepi nyata tidak harus menghasilkan lebih dari satu ujung yang terdeteksi.

Algoritma Canny berjalan dalam 5 langkah yang terpisah yaitu:

1. *Smoothing*

Mengaburkan gambar untuk menghilangkan *noise finding*

2. gradien

Tepian harus ditandai pada gambar memiliki gradien yang besar.

3. *Non maximum suppression*

Hanya maxima lokal yang harus ditandai sebagai *edge*.

4. *Double thresholding*

Tepian yang berpotensi ditentukan oleh *thresholding*.

5. *Edge Tracking by hysteresis*

Tepian final ditentukan dengan menekan semua sisi yang tidak terhubung dengan tepian yang sangat kuat.

nilai warna pada pixel dengan intensitas kecerahan tertinggi. Dan pengambilan posisi sumber cahaya diambil dengan menentukan titik koordinat lokasi pixel dengan tingkat kecerahan tertinggi. Penelitian dilakukan dengan menerapkan pola kotak dengan tingkat kecerahan tinggi yang akan memantulkan warna cahaya yang dideteksi dengan metode *edge detection* sebelum citra diolah.

1. Dalam sistem akan dilacak objek *rectangle* dimana *marker augmented reality* berada dengan *edge detection*.

2. Setelah kotak telah dideteksi, maka dilakukan pemotongan terhadap citra yang terdeteksi sebagai kotak.

3. Setelah hasil pemotongan didapatkan, citra terpotong digunakan untuk proses pengolahan citra untuk menentukan hasil yang diinginkan.

4. Proses penentuan warna cahaya dilakukan pada citra yang belum mengalami perubahan ke format *grayscale* agar didapatkan nilai RGB dari titik dengan nilai kecerahan terbesar.

5. Nilai yang telah didapatkan diterapkan pada lingkungan Unity3D dengan menggunakan penanda yang merupakan objek yang sama dengan objek *rectangle* yang telah diproses.

Nilai posisi yang didapatkan adalah berbasis koordinat x dan y pada gambar, sementara pada unity, posisi tersebut dimasukkan dalam koordinat pada lingkungan Unity3D. posisi cahaya harus sesuai dengan koordinat *marker*. maka dilakukan modifikasi pada posisi yang didapat dengan membagi nilai x dan y yang didapat dengan lebar dan tinggi *marker* yang telah dimasukkan dalam editor unity. dengan begini, posisi cahaya akan sesuai dengan ukuran *marker*.

Sementara nilai Z pada posisi sumber cahaya simulasi ditentukan dengan nilai $z > 0$ dan $z < 5$ agar cahaya terlihat pada *marker*.

II. METODE PENELITIAN

Pada penelitian ini akan dibuat sebuah sistem pembangkitan cahaya dengan komposisi dinamis untuk diterapkan dalam *augmented reality* dengan hasil komposisi warna cahaya dan posisi sumber cahaya. Pengambilan warna cahaya sumber dilakukan dengan mengambil



Gambar 4. Bagan utama sistem

A. Pengolahan Intensitas cahaya

Setelah hasil pemotongan didapatkan, citra terpotong digunakan untuk proses pengolahan citra untuk menentukan hasil yang diinginkan.

1. Mengubah format citra yang telah dipotong menjadi Grayscale. dengan menggunakan format grayscale, nilai yang sama untuk setiap komposisi R, G dan B nya. sehingga perbedaan warna cahaya yang masuk akan menjadi nilai keabuan citra dan memungkinkan untuk menghitung nilai kecerahan dari citra tersebut.
2. Mendapatkan nilai dengan nilai kecerahan terbesar pada citra berformat gray. untuk tiap skalar X dan Y yang ada pada citra, yang memiliki nilai kecerahan berbeda

antara 0 hingga 255. maka dilakukan pencarian nilai dengan memperbaharui nilai setiap ditemukan nilai yang lebih besar dari nilai terbesar yang telah ditangkap.

3. Menetapkan titik koordinat pada titik dengan nilai kecerahan terbesar. posisi X dan Y sebagai koordinat dimensi dua dengan penambahan input nilai Z di posisi $Z > 0$.

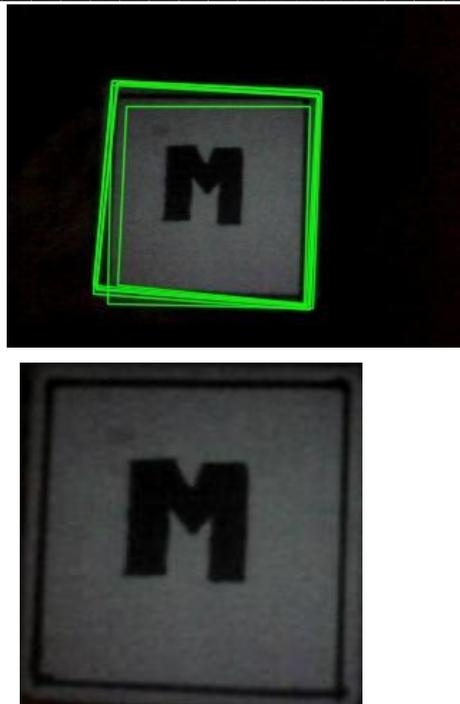
B. Penentuan warna cahaya

Proses penentuan warna cahaya dilakukan pada citra yang belum mengalami perubahan ke format grayscale agar didapatkan nilai RGB dari titik dengan nilai kecerahan terbesar.

1. Mengambil citra hasil pemotongan berformat RGB.
2. Mengambil nilai RGB pada titik dengan nilai kecerahan terbesar. Nilai posisi yang telah didapatkan pada proses pengolahan cahaya dijadikan titik yang diambil nilai RGB nya. sehingga didapat sebuah nilai warna pada titik yang terterang.

III. ANALISA HASIL DAN PEMBAHASAN

Pada Simulasi didapatkan bahwa sistem telah berjalan dengan pendeteksian intensitas cahaya sebagai berikut:



Gambar 5 Citra Hasil Simulasi captured, tracked, and cropped

Dalam penelitian ini diujikan beberapa kondisi pencahayaan dengan posisi cahaya dan warna cahaya yang berbeda-beda lalu disimulasikan pada Unity3D.

A. Pengujian Posisi Sumber Cahaya

Percobaan yang dilakukan bertujuan untuk mendapatkan nilai posisi cahaya yang tepat dari berbagai posisi sumber cahaya dengan mengambil sepuluh frame di setiap percobaan. Pengujian dilakukan pada 4 arah terhadap sumbu X dan Y yang diujikan pada setiap percobaan. yaitu:

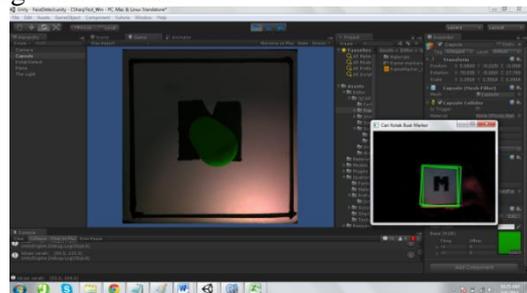
1. posisi cahaya sumber di titik $Y < height/2$
2. posisi cahaya sumber di titik $Y > height/2$
3. posisi cahaya sumber di titik $X < width/2$
4. posisi cahaya sumber di titik $X > width/2$

dengan ilustrasi sebagai berikut



Gambar 6. Ilustrasi pengujian posisi sumber cahaya

cahaya virtual dibangkitkan di titik piksel dengan intensitas kecerahan tertinggi yang sebelumnya dikonversi menjadi grayscale. Pengujian dilakukan dengan menghitung jarak antara ujung citra marker dengan piksel yang memiliki intensitas cahaya tertinggi. dengan kondisi Sumber cahaya akan tepat berada di ujung citra marker.



Gambar 7 Simulasi Posisi sumber cahaya di bagian bawah citra, dimana $Y > tinggi marker/2$

TABEL I
 PERCOBAAN DENGAN CAHAYA BERADA DI UJUNG BAWAH CITRA, DIMANA $Y > TINGGI MARKER / 2$

Height citra	koordinat Y dari piksel dengan intensitas cahaya tertinggi
109	103
116	105
103	100
103	90
104	99
108	100

109	103
104	98
104	95
110	97

TABEL 3
HASIL PENGUJIAN POSISI SUMBER CAHAYA

Percobaan ke N	persentase
Percobaan 1	92.55%
Percobaan 2	88.12%
Percobaan 3	95.63%
Percobaan 4	96.58%

Dari hasil percobaan 1 sampai dengan percobaan 4 didapatkan bahwa rata rata persentase gabungan ketepatan posisi cahaya yaitu sebesar 92.10% dari posisi sebenarnya, dengan titik pangkal serta titik terujung dari lebar dan tinggi citra sebagai titik acuan pengujian.

B. Pengujian Warna Cahaya

Percobaan yang dilakukan bertujuan untuk mendapatkan nilai RGB dari sumber cahaya yang akurat dari sumber cahaya dengan frekuensi sepuluh frame setiap percobaan. Dengan 3 warna sumber cahaya yaitu merah, kuning, hijau.

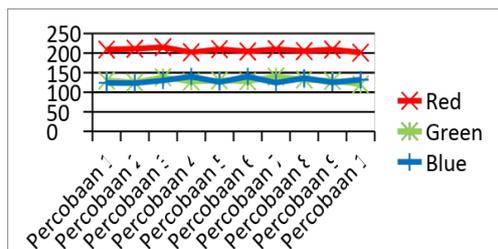
Batas-batas toleransi threshold untuk segmentasi warna pada citra adalah:

TABEL 4
BATAS TOLERANSI THRESHOLD RGB UNTUK SEGMENTASI CITRA

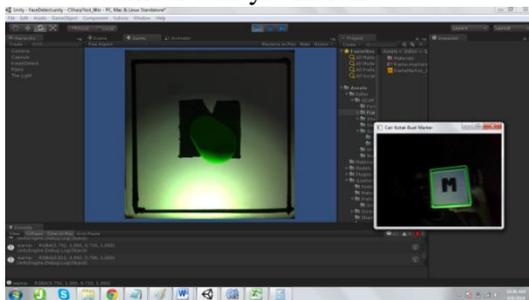
Warna	Batas R	Batas G	Batas B
Merah	180 – 255	0 – 150	0 – 150
Kuning	180 - 255	180 - 255	0 – 150
Hijau	0 – 150	180 - 255	0 – 150

TABEL 5
 PENGUJIAN DENGAN SUMBER CAHAYA WARNA MERAH

percobaan	Jumlah error
Sumber cahaya warna merah	0/30
Sumber cahaya warna Kuning	10/30
Sumber cahaya warna Hijau	20/30
Banyak error	30/90
Persentase error	33.33%



Gambar 8. Grafik hasil pengujian warna sumber cahaya: merah.



Gambar 9. Tampilan simulasi percobaan warna hijau

Dapat dilihat bahwa batas toleransi yang dilanggar cukup besar. Dapat dilihat apabila cahaya simulasi diterapkan pada obyek 3D, obyek akan terlihat diterangi oleh cahaya berwarna keputih-putihan sehingga tidak mudah diidentifikasi oleh mata

manusia. Beberapa penyebab terjadinya hal ini adalah:

1. Penerapan kertas *marker* berwarna putih dengan nilai keabuan lebih dari 180
2. Sumber cahaya lain disekitar *marker*.

Dari hasil percobaan warna merah hingga percobaan warna hijau pada pengujian warna cahaya , didapatkan hasil persentase kemiripan

warna cahaya hasil dibandingkan dengan warna cahaya sumber adalah 66.66%.

IV. KESIMPULAN DAN SARAN

Posisi sumber cahaya yang telah dirancang, hasil percobaan 1 sampai dengan percobaan 4 didapatkan hasil rata rata persentase 92.10% dari posisi sebenarnya. Dari hasil percobaan warna, didapatkan hasil persentase kemiripan warna cahaya hasil dibandingkan dengan warna cahaya sumber adalah 66.66%. Hal ini disebabkan pantulan cahaya pada kertas marker yang bernilai keabuan tinggi (> 180) dan sumber cahaya lain sehingga mempengaruhi hasil cahaya yang dibangkitkan oleh *game engine* Unity 3D pada simulasi.

DAFTAR PUSTAKA

[1] Labschutz, Matthias; Kros, Katharina, "Content Creation for a 3D Game with Maya and Unity 3D Systems", Institute of Computer Graphics and Algorithms, Vienna University of Technology, Vienna, Austria, 2011.

[2] Unity 3D. Unity: *Light*. <http://docs.unity3d.com/Manual/class-Light.html>, 2010.

[3] Chakravarti, Rishav; Meng, Xiannong, "A Study of Color Histogram Based Image retrieving", Department of Computer Science, Bucknell University, Lewisburg, PA, 2009.

[4] S. Sabine; B. Robert; S. Steve, "Standard RGB Color Spaces", Laboratory of audio-visual Communication (EPFL), Xerox Architecture Center, Apple Computer, Lausanne, Switzerland; Webster, NY; Cupertino, CA, 1999

[5] V. Karthikeyani; K. Duraiswamy; P. Kamalakkannan, "Conversion of Gray-scale image to Color Image with and without *Texture Synthesis*", K.S. Rangasamy College of Technology, Tiruchengode, Tamil Nadu, India, 2007.