International Journal of Electronics and Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 2, Issue 5, Nov 2013, 45-54
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

IMAGE MANIPULATION TECHNIQUES USING MATLAB

SIDDHARTH BHATT, SHILEN JHAVERI, VISMAY PANDIT & JINESH DOSHI

U.G. Student, Department of Electronics and Telecommunication, DJSCOE, Vile-Parle (W), Mumbai, Maharashtra, India

ABSTRACT

Colors can be added to grayscale images in order to increase the visual appeal of these images. The task of “colorizing” a grayscale image involves assigning three dimensional (RGB) pixel values to an image which varies along only one dimension (luminance or intensity).Hence rather than choosing RGB colors from a palette to color individual components, we transfer the entire color distribution of the source to the target image by matching the luminance information between the images. High Dynamic Range Imaging (HDRI) is a set of techniques that allows a greater dynamic range of luminance (the range of values between light and dark areas of a scene) than normal digital imaging techniques. The intention of HDRI is to accurately represent the wide range of intensity levels found in real scenes ranging from direct sunlight to shadows. Information stored in high dynamic range images usually corresponds to the physical values of luminance or radiance that can be observed in the real world. This is different from traditional digital images, which represent colors that should appear on a monitor or a paper print. We describe a process for creating an image mosaic—a collection of small images arranged in such a way that when they are seen together from a distance they suggest a larger image. To visually suggest the larger form, the small images are arranged to match a large picture as much as possible, and then their colors are adjusted to better suggest the overall form. Here arrangement of the small images is automatic.

KEYWORDS: Grayscale, RGB, HDR, Imaging, Luminance Mosaics

INTRODUCTION

Grayscale to RGB Conversion

Colorization of the grayscale images such as old black and white photos, classic movies or scientific illustrations enhances the information content by exploiting the variations in chromaticity as well as luminance. Since different colors may have the same luminance value but vary in hue or saturation, the problem of colorizing grayscale images has no inherently “correct” solution. Due to these ambiguities, human interaction plays an important role in the colorization process. Where the mapping of luminance values to color values is automatic, the choice of the reference color image is commonly determined by human decision.

COLOR TRANSFER ALGORITHM

In this section, we describe the general algorithm for transferring color. The general procedure for color transfer requires a few simple steps. First each image is converted into the YCbCr color space. The YCbCr color space provides three decors related, principal channels corresponding an achromatic luminance channel and two chromatic channels and, which roughly correspond to yellow-blue and red-green opponent channels. Thus, changes made in one color channel should minimally affect values in the other channels. The reason the color space is selected in the current procedure is because it provides a chromatic channel for both the images. This allows us to selectively transfer the two chromatic channels from the color image to the grayscale image without cross-channel artifacts.

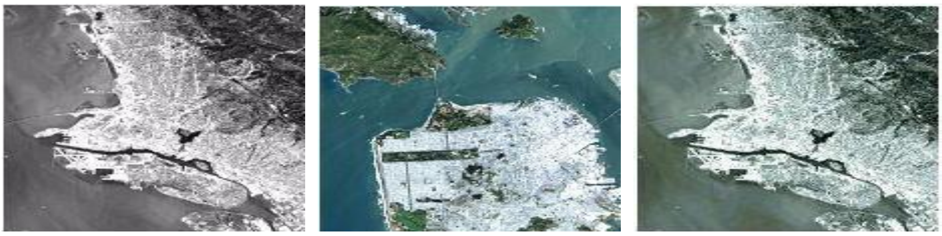
Next, we go through each pixel in the grayscale image in scan-line order and select the best matching sample in the color image. The best match is determined by continuously finding the difference in the luminance values of the current Grayscale pixel and the luminance values of the all upcoming Color pixels. The Color pixel having the minimum difference (error), is chosen as the best matching pixel and corresponding chromaticity values are transferred to that Grayscale pixel. Hence in this manner, each and every Grayscale pixel is compared with all the RGB pixels and the chromaticity value of the best matching pixel is transferred to it in order to form the final image. This automatic, global matching procedure works reasonably well on images when corresponding color regions between the two images also correspond in luminance values. However, regions in the target image that do not have a close luminance value to an appropriate structure in the source image will not appear correct. Hence in order to color the Grayscale image properly, the user should know this fact that the source image to be chosen should have a close similarity in the luminance values with the Grayscale image.

RESULTS ANALYSIS

Figures 1 a-c shows the results of colorizing foliage, face and landscape photographs. While Figure 2 a-c shows the final results of the algorithm applied to a variety of facial images. The technique Works well on scenes where the image is divided into distinct luminance clusters or where each of the regions has distinct textures. In general, however, the current technique does not work very well with faces. Although colors are transferred well into the target image, the algorithm is not able to classify the difference between skin and lips and sometimes clothes and hair.

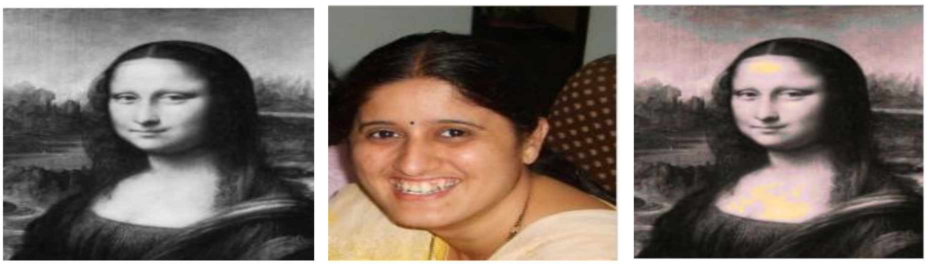
FURTHER IMPROVEMENT

In the method discussed above instead of using the single pixel value, we can use the neighborhood processing for comparing the luminance values which can result in better colorization. Colorization of the videos can also be done using the above method. To colorize all the frames in a scene, we first transfer the color from the source RGB image to a single frame. Once this frame is colorized successfully, rest of the frames can be colored using the same algorithm. Finally, one could also utilize a database of basis texture swatches for the initial Color transfer in the user-guided stage of the colorization process.



(a) (b) (c)

Figure 1



(a) (b) (c)

Figure 2

HDR Imaging

The dynamic range (or the luminosity range) is the ratio (contrast) between the brightest and the darkest part of the photographed scene. The human vision can accommodate (in long term) the dynamic range of 10^{14}:1 (from starlight to bright objects in full sunlight) but covers the dynamic range of about 100,000 : 1 at one time, just the top dynamic range of most real-world scenes. In comparison the dynamic range of typical digital cameras is about 1000-4000: 1 and in case of computer displays it is even much worse. As a result, even in correctly exposed photographs, shadows are too dark while highlights are too bright creating a loss of significant details. Hence, HDR images increases visual fidelity by allowing a greater dynamic range of luminance of an image than standard digital imaging techniques or photographic methods and represents the varying levels of intensities more correctly.

The problem caused due to the dynamic range of the capturing equipment is illustrated in figure 3. In the first image the inside part of the room is the subject, whereas the window is the surrounding. Much is seen inside the window, but the details on shelf and the window are not seen clearly. In the second image the window and items on shelf are clear enough but nothing is visible inside the room. The limitations of dynamic range of camera do not allow accommodating both the features in one image.

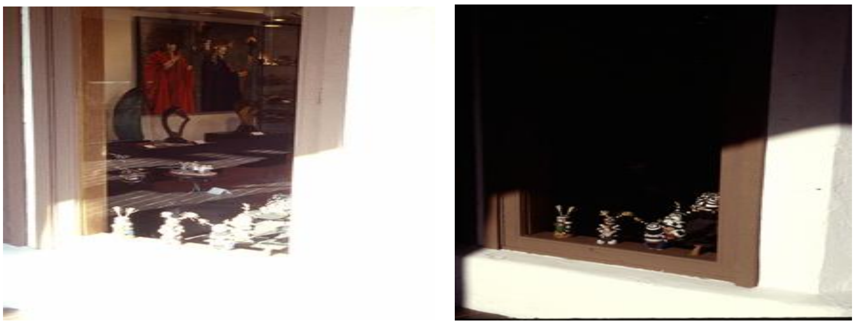


Figure 3: Limitations of Dynamic Range of the Capturing Equipment

ALGORITHM

Hence, HDRI basically consists of following 2 steps:

Recovering High Dynamic Range Radiance Maps from Photographs

When one photographs a scene, either with film or an electronic imaging array, and digitizes the photograph to obtain a two dimensional array of brightness values, these values are rarely true measurements of relative radiance in the scene. For example, if one pixel has twice the value of another, it is unlikely that it observed twice the radiance. Instead, there is usually an unknown, nonlinear mapping that determines how radiance in the scene becomes pixel values in the image. This nonlinear mapping is hard to know beforehand because it is actually the composition of several nonlinear mappings that occur in the photographic process.

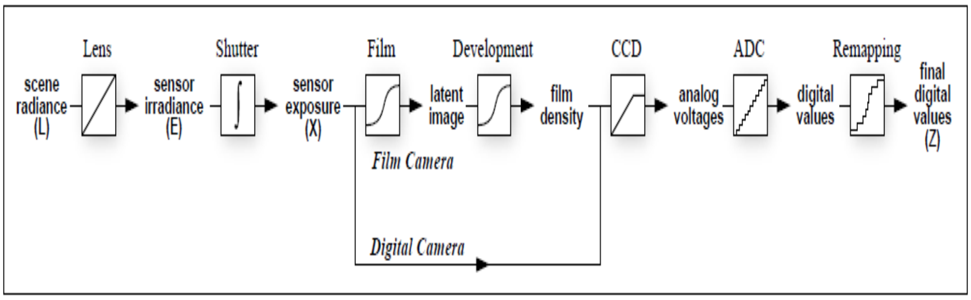


Figure 4

In a conventional camera, the film is first exposed to light to form a latent image, then developed to change this latent image into variations in transparency, or *density*, on the film. The film can then be digitized using a film scanner, which projects light through the film onto an electronic light-sensitive array, converting the image to electricalvoltages. These voltages are digitized, and then manipulated before finally being written to the storage medium.

In the first stage of the process, the film response to variations in exposure X (= $E\Delta t$; i.e. X is the product of the irradiance E the film receives and the exposure time Δt) is a non-linear function, called the "*characteristic curve*" of the film. The development, scanning and digitization processes usually introduce their own nonlinearities which compose to give the aggregate nonlinear relationship between the image pixel exposures X and their values ***Z***(the final digital value after processing).

A simple technique for recovering this response function, up to a scale factor, using nothing more than a set of photographs taken with varying, known exposure durations has been the proposed work by *Debevec & Malik* [1]. With this mapping, the pixel values from all the available photographs are used to construct an accurate map of the radiance in the scene, up to a factor of scale. This radiance map will cover the entire dynamic range captured by the original photographs.

This algorithm is based on exploiting a physical property of imaging systems, both photochemical and electronic, known as *reciprocity*. The response of a photographic film to variations in exposure is summarized by the characteristic curve (or Hurter-Driffield curve) [1], which is a graph of the optical density D of the processed film against the logarithm of the exposure X to which it has been subjected. The exposure X=EΔt. Key to the very concept of the characteristic curve is the assumption that only the product EΔt is important, and that halving E and doubling Δt will not change the resulting optical density D. Under extreme conditions (very large or very low Δt), the reciprocity assumption can break down, a situation described as reciprocity failure.

After the development, scanning and digitization processes, we obtain a digital number Z, which is a nonlinear function of the original exposure X at the pixel. Let us call this function *f*, which is the composition of the characteristic curve of the film as well as all the nonlinearities introduced by the later processing steps. The first goal is to recover this function *f*. Then, we can compute the exposure X at each pixel, as X=f^{-1}(Z). A reasonable assumption made is that the function f is monotonically increasing [1], so its inverse f^{-1} is well defined. Knowing the exposure X and the exposure time Δt, the irradiance E is recovered as E=X/Δt, which will be taken to be proportional to the radiance L in the scene.

The input to the algorithm is a number of digitized photographs taken from the same vantage point with different known exposure durations Δtj. It is assumed that the scene is static and that this process is completed quickly enough that lighting changes can be safely ignored. It can then be assumed that the film irradiance values Ei for each pixel i are constant. Pixel values are denoted by Zij where i is a spatial index over pixels and j indexes over exposure times Δtj. Thus the film reciprocity equation is:

$$Zij = f(Ei\Delta tj) \tag{1}$$

Since it is assumed *f* is monotonic, it is invertible, and so we can rewrite (1) as:

$$f^{-1}(Zij) = Ei\Delta tj \tag{2}$$

Taking the natural logarithm of both sides of (2), we have:

$$\ln f^{-1}(Zij) = \ln Ei + \ln\Delta tj \tag{3}$$

To simplify notation, let us define a function, $g = \ln f-1$ We then have the set of equations:

$$g(Zij) = \ln Ei + \ln \Delta tj \qquad (4)$$

In this set of equations, the *Zij* are known, along with their corresponding Δ*tj* The unknowns are the irradiances *Ei* as well as the function *g*, although it is assumed that *g* is smooth and monotonic.

Our aim now is to recover the function *g* and the irradiances *Ei* that best satisfy the set of equations arising from Equation (4) in a least-squared error sense. It can be verified that recovering *g* only requires recovering the finite number of values that *g*(*Z*) can take since the domain of *Z*, pixel brightness values, is finite.

Let:

Zmin and *Zmax* be the least and greatest pixel values (integers),

N be the number of pixel locations, and

P be the number of photographs.

Thus the existing problem can be formulated as one of finding the (*Zmin*−*Zmax*+1) values of *g*(*Z*) and the *N* values of ln*Ei* that minimize the following quadratic objective function:

$$o = \sum_{i=1}^{N}\sum_{j=1}^{P}[[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]]^2 + \lambda \sum_{z=Z_{min}+1}^{z=Z_{max}-1} g''(z)^2 \qquad (5)$$

The first term ensures that the solution satisfies the set of equations arising from Equation (4) in a least squares sense. The second term is a smoothness term on the sum of squared values of the second derivative of *g* to ensure that the function *g* is smooth. In this discrete setting,

g''(*z*) = *g*(*z* −1)−2*g*(*z*)+*g*(*z* +1) is used. This smoothness term is essential to the formulation in that it provides coupling between the values *g*(*z*) in the minimization. The scalar *λ* weights the smoothness term relative to the data fitting term, and should be chosen appropriately for the amount of noise expected in the *Zij* measurements. Because it is quadratic in the *Ei*'s and *g*(*z*)'s, minimizing *o* is a straightforward linear least squares problem. The over-determined system of linear equations is robustly solved using the singular value decomposition (SVD) method.

CONSTRUCTION OF HIGH DYNAMIC RANGE RADIANCE MAP

Once the response curve *g* is recovered, it can be used to quickly convert pixel values to relative radiance values, assuming the exposure Δ*tj* is known. From equation (4), we obtain:

$$\ln Ei = g(Zij) - \ln \Delta tj \qquad (7)$$

For robustness, and to recover high dynamic range radiance values, we should use all the available exposures for a particular pixel to compute its radiance. For this, we reuse the weighting function in Equation (6) to give higher weight to exposures in which the pixels value is closer to the middle of the response function:

$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})} \qquad (8)$$

Combining the multiple exposures has the effect of reducing noise in the recovered radiance values. It also reduces the effects of imaging artifacts such as film grain.

TONE MAPPING

After obtaining the HDR scene values, the next task was to tone map these values onto a LDR matrix for being able to see it on the screen. While digital imaging technology now enables us to capture full dynamic range of the real world scene, still we are limited by the low dynamic range displays. Thus the scene can be visualized on a display monitor only after the captured high dynamic range is compressed to available range of the display device. This has been referred to as the tone mapping problem in the literature.

ALGORITHM

In paper (2), the authors have proposed a simple tone mapping operator which allows us to preserve the visual content of the real-world scene without the user having to manually set a number of parameters. The method is conceptually simple, and easy to use. It uses a $s - function$ type operator which takes into account both the global average of the image, as well as local luminance in the immediate neighbourhood of each pixel. The local luminance is computed using a median filter. A detail in the image can result from two possible reasons− either the pixel is brighter than its surrounding pixels or the pixel is darker than its surrounding pixels. If luminance Y at a pixel is more than YL, the local luminance in its immediate neighborhood, we would like to increase the displayable value YD at that point, so that it appears brighter against its neighborhood. On the other hand, if the luminance Y at a point is less than the local luminance YL in its immediate neighborhood, we would like to reduce the displayable value YD, so that it appears darker than its neighborhood.

YD(x,y)=Y(x,y)/[Y(x,y)+CL(x,y)]

where CL is the contrast luminance at a point (x,y) obtained by modifying the global contrast factor GC with a term involving the logarithm of ratio of local low-pass filtered value YL to original luminance value,

CL=YL[log($\boldsymbol{\delta}$+YL/Y)]+GC

where $\boldsymbol{\delta}$ is a very small value and is included to avoid singularity while doing the log operation.

where GC is the global contrast factor computed through

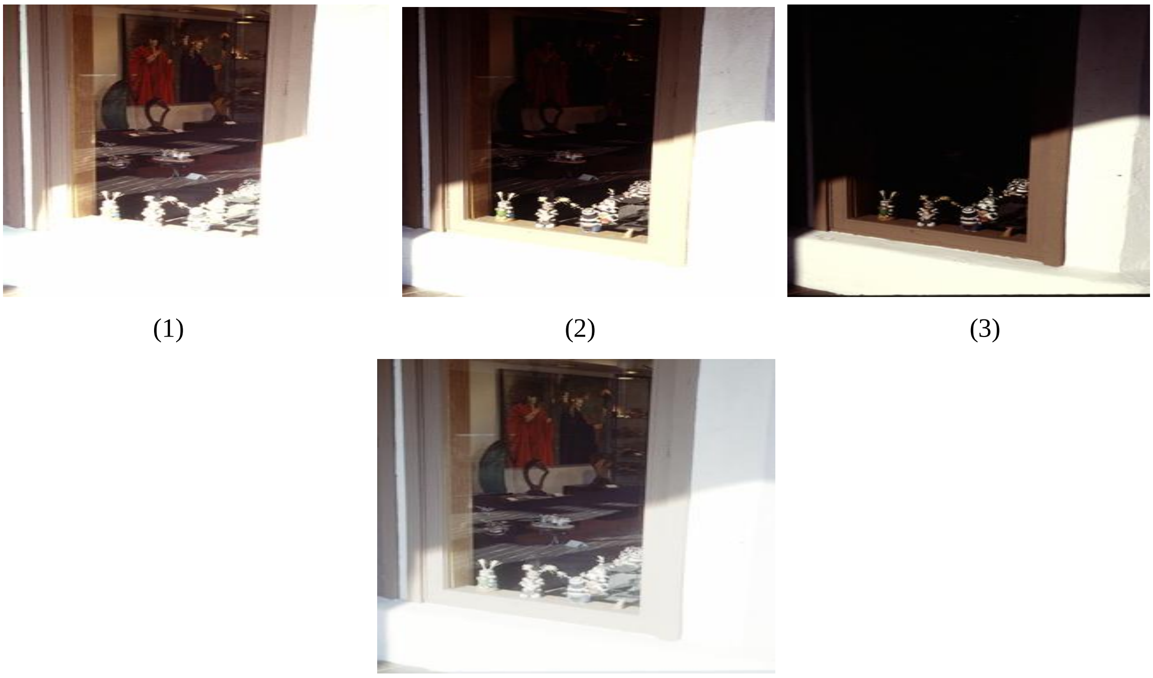
GC=c YA

where YA is the average luminance value of the whole image and c is a multiplying factor.

This would have the effect of bringing the high luminance value closer to 1, while the low luminance values would be a fraction closer to zero. The choice of factor c has an effect on bridging the gap between the two ends of the luminance value.

RESULTS ANALYSIS

The proposed scheme was simulated on the window image. A series of 3 different exposures of the image was taken as input which is shown below. The exposures of the three images were in decreasing order. The darkest image has the least exposure time while the brightest has the maximum. Then the radiance map was found. Then the HDR image was generated from the inverse of the found function. Then the HDR image was tone mapped. It was observed that the proposed scheme gave satisfactory results.



(1) (2) (3)

(4)

Figure 5

HDR TONE MAPPED IMAGE

Mosaicing

In this section we describe how to create an image mosaic. Specifically, we'll address the problem: given a collection of tile images *Ti* and a target image *I,* create an image mosaic *M* that resembles *I* at a coarse scale but is composed of a tiling of the *Ti*'s. We solve the problem in four steps:

(1) Choose images, (2) choose a tiling grid, (3) find an arrangement for the image tiles within the grid (4) Display the final image.

ALGORITHM

Choosing Images

The choice of subject matter is purely an artistic one.

- A "base" image. This is the picture that emerges if you take a step back and squint your eyes at your finished mosaic.
- A collection of "tile" images. These are the little sub-images which will make up the larger image.

Process the Tiles and Resize the Base Image

Shrink the tile images, if necessary, so that they are all a uniform size with equal width and height. The processed tile images are saved to a separate directory. In order to have an easy time of creating a grid of tiles which is the same size as the base image, we crop the base image so that its dimensions are a multiple of the size of a (processed) tile. To find an appropriate arrangement for the image tiles, the first question that needs to be addressed is what kind of tiling will be used for the mosaic. We have investigated only semi regular rectangular grids for simplicity, although other tilings are possible. Furthermore, since the image tiles themselves tend to be rectangles, a rectangular tiling is a natural choice. Arranged in a rectangular grid, all the small image tiles must have the same aspect ratio. Thus, it is important to choose an aspect ratio appropriate for the entire set of image tiles, as they will either have to be cropped**.**

Arranging the Image Tiles

Having selected a tiling grid, we need to place individual image tiles into the grid. We have explored a number of arrangement options:

- Use the same tile everywhere.
- Choose a random arrangement of the image tiles.
- Arrange different tiles manually by eye.
- Place image tiles by matching their average colors to the region of the target image that they cover. This is the case we have used.

This is implemented as follows:

- Create a blank image of the same size as the base image.
- Extract the mean intensities in each of four segments (upper left, upper right, lower left and lower right) of each of the image tile for each color band.
- Divide the base image into tiles of exact same size as that of the image tiles.
- Extract the mean intensities in each of four segments of the first tile of the base image for each color band.
- Select the image tile with the minimum absolute error of the intensities of image tiles and the first tile of the base image.
- Place the image tile found in step(4) in the corresponding position in the blank image .
- Repeat step (3)- step(6) till all the tiles are placed in the blank image by image tiles.

RESULTS ANALYSIS

Figure 6 contains an image composed of smaller images. There is relative patchiness in the image. As the size of image tiles decreases and number of different image tiles increases, the quality of the final mage improves drastically, but at the cost of more memory space and time.

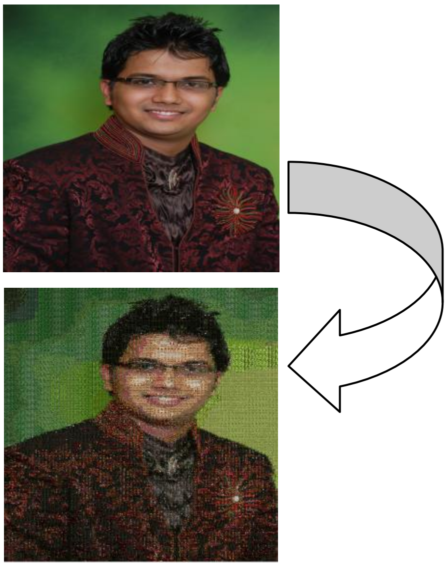


Figure 6

CONCLUSIONS

In this paper we have formulated a new, general, fast, and user-Friendly approach to the problem of colorizing grayscale images using MATLAB 2007 while the standard methods uses the global color palette which is more difficult to implement.

A simple implementation of HDR along with tone mapping was done using MATLAB 2007 and the results achieved were satisfactory. Here, we have also explained a very simple, fast and easy way to make image mosaics.

REFERENCES

1. Debevec, P. E., & Malik, J. (n.d.). Recovering High Dynamic Range Radiance Maps from Photographs.
2. Biswas, K. K., & Pattanaik, S. (n.d.). A Simple Spatial Tone Mapping Operator for High Dynamic.
3. Pattanaik, S. N., Ferwerda, J. A., Fairchild, M. D., & Greenberg, D. P. (1998). A multiscale model of adaptation and spatial vision for realistic images display. *SIGGRAPH, Annual Conference Series, ACM* , 267-276.
4. Reinhard, E., Stark, M., Shirley, P., & Ferwerda, J. (2002). Photographic Tone reproduction for digital images. *SIGGRAPH, Annual Conf. Series, ACM* , 267-276.
5. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson Education, second edition, 1992.
6. Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*. Pearson Education, fourth edition, 2008.
7. Govind Haldankar, Atul Tikare and Jayprabha Patil,*‘Converting Gray-Scale Image to Color Image’*. Proceedings of SPITIEEE Colloquium and International Conference, Mumbai, India, Vol. 1, 189.
8. www.mathworks.com/matlabcentral
9. Yue Zhang, Mario A. Nascimento, Osmar R. Za¨ıane, ‘Building Image Mosaics An Application of Content-Based Image Retrieval’. Dept. of Computing Science, Univ.of Alberta, Canada.
10. Adam Finkelstein and Marisa Range, *‘Image Mosaics’*. Computer Science Department, Princeton University.