ISSN:0975-9646

An Automatic Approach Towards Coloring Videos Using Monochrome Texture Descriptor & Parallel Processing

Gauri R. Joshi[1], Prof. Mayur S. Burange[2]

[1]Department of Computer Science and Engineering,
P.R. Pote Engg/Sant. Gadage Baba Amravati University, India.

[2]Department of Computer Science and Engineering,
P.R. Pote Engg/Sant. Gadage Baba Amravati University, India.

Abstract-We present a new scheme for video colorization using optimization in rotation-aware new YCbCr feature space. Most current methods of video colorization incur temporal artifacts and prohibitive processing costs, while this approach is designed in a spatiotemporal manner to preserve temporal coherence. The parallel implementation on graphics hardware is also facilitated to achieve realtime performance of color optimization. By adaptively clustering video frames and extending filtering to chrominance flow computation, we can achieve real-time color propagation within and between frames. Temporal coherence is further refined through user scribbles in video frames. The experimental results demonstrate that our proposed approach is efficient in producing high-quality colorized videos.

Keywords—YCbCr space, parallel optimization, user strokes, video colorization

1. INTRODUCTION

The term colorization is introduced by the Wilson Markle in 1970. He introduces this term to elaborate the automated or computer assisted process that he invented for putting the colors to the black and white movie or tv programs. But now a days the term colorization is generally describes any method or technique for adding color to any monochrome still images or the footages. Traditionally, the colorization process is very difficult and time consuming and also requires artistic skills to coloring the images Although substantial efforts has been taken to colorize the images very less amount of efforts are devoted to colorize the video. Because it is more difficult task to colorize the video than image. As the video is the 3-dimensional and the color is to be propagated in the 3-D space. Coloring a video is very complex as it has to balance constraints between spatio-temporal color coherence and processing costs [1],[2]. The optimization of the colored video is very difficult and time consuming as the data appear here is in 3-D space.

The technique that are developed till date to coloring the video, make the color propagation from keyframes to subsequent frames by relying on the optical flow. The key frames are the frames where the sudden changes occurred in the frame. However the optical flow causes some artifacts in some of the frames which consequently reflected in the colorization results. To remove the coloriztion error from the frame uses the optimization based on the one-pass linear system solver [1]. But such errors cant not be easily eliminated through user input or post processing approach.

In this paper, proposing a more efficient method to colorize the video in optimising way using parallel color propagation. From the previously proposed method our key observation is that since color propagation assigns similar color to the similar pixels. In the higher dimensional pixel feature space, video colorization constitutes smooth function rather than in the video space.

Rather than using previous method that solves color optimization in very large linear system, our method iteratively propagates the colors among the pixels in the graphs which results in the greater refinement and parallel propagation. This method of parallel propagating color gives better result than previous one.

The colors may shift erratically between the frames as methods of static images applied to video frames. This may cause visual fatigue to the viewers. So. To solve this problem, we propose the leverages user scribbles through a new feature space formed through rotation aware gabor filtering. In this new feature space instead of using the optical flow to propagate the color between the consecutive frames gabor flow is used to compute the temporal connectivity of the pixel graphs. In comparison to previous work two notable contributions are made in this paper:

1) A method for maintaining temporal color coherence using optimized chrominance flow;

2) A parallelized strategy for fully solving the optimization of video colorization.

2. LITERATURE SURVEY

In the image colorization technique, the colorization is usually achieved on the basis of color scribble inputs through the user. In colorization technique user needs to give some input in the given image to describe which part of image is colored by which color. In the technique in [1], the user only needs to color small number of pixels in the selected area. The algorithm automatically propagates the color throughout the image intensity boundaries. In this technique artist or the user don’t need to precisely define the boundaries. While in other colorization approach, colorization process is broadly divide in to two stages i.e color labelling and color mapping. The regions that approximately share the same pixel values are grouped in to coherent regions and to further fine tune the color in the images the color mapping step is introduced. In this technique the intensities are used to propagate the color [2],[4] ,[5]. In [5] we learned the relation between the grayscale image and its color version. Colorizing one or more input grayscale images, based on a partially segmented reference color image. The partially segmented means one or more mutually disjoined regions which has been assigned with different unique label [6],[7] respectively. By using the input scribbles given by user another technique is developed to colorize the real size images [3],[8]-[9]. The color of this scribbles are then spread algorithmically over the image. For example , a weighted average of scribble color is used to colorize pixels [3] and the weight is proportional to the geodesic distance between a pixels and its scribbles.

The technique based on the gabor feature space is introduced recently, but only suitable for single image domain [10]. Color distribution can be optimized across the entire image since the pixels are still in the 2-d space. But this strategy fails as it still introduce the cross frames and flickering artifacts or the errors. The key to handle this problem is color propogation in temporal video space not only in the 2-d image space. One technique is introduce to do this using artificial neural network but when used to complete this temporal propogation it also increases computational costs and hinders parallelization [11]. Some of the most efficient methods in calculating the inconsistency of stereoimagaing and motion in video sequences is optical flow algorithm [12]-[15]. However, these algorithm faced with severe challenges that are caused by the occlusion. Some methods are put forward to handle over smoothing , such as anisotropic diffusion with image or flow adaptiveness [17], [16]–[18], and isotropic diffusion with non quadratic regularization [13],[14].

A novel video cutout system which, unlike previous systems, uses adaptive local classifiers for segmenting dynamic video objects. By localizing the classifiers, our system achieves significantly better results than previous systems for complicated videos, including complex color distributions, dynamic backgrounds and non-rigid foreground deformations [19].

3. PROPOSED METHODOLOGY

The essential task of video colorization is to assign colors to pixels based on the intuition that pixels with high similarity should have similar colors. The key to our approach is to establish pixel similarity in the video's grayscale channel. To accomplish this, a novel pipeline is introduced that measures pixel similarity across video frames and allows parallel color optimization among pixels. The colorization of all the three inputs i.e static image or real time image or the viedo stream is done by using luminance matching .The proposed work can be divided in to the following modules.

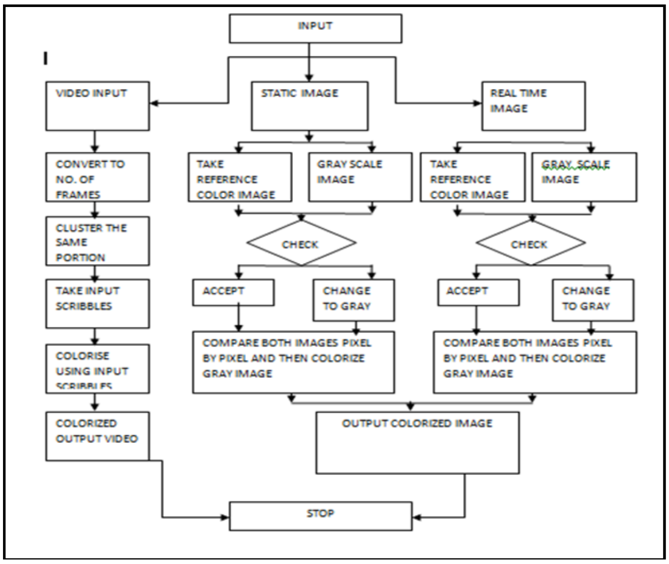


Fig 3.1: Flow diagram of proposed work

3.1 Static Image Colorization:

In this module the static image colorization is done. The static image means the still image, here in this module the one static reference color image is taken and then after that one gray scale image is taken then with reference to the color image colorization of the gray scale image is done. Each and every pixels of the gray scale image is compared with reference color image and after that minimum and minimum value are calculated. From the min and max we calculate the minimum difference in the pixels. Then where the minimum difference found put the color from color image to gray one.

3.2 Real Time Image Colorization:

In this module the input image taken is also the static image but the image that is taken as input will be the real time image. While choosing for the image for reference image if selected option is real time then the camera automatically gets on and one can take image in real time. And if for the gray scale image also real time image is also be taken and if by mistake someone choose color image instead of color one then the image automatically change to gray scale. The procedure is same to colorise the gray image by luminance matching.

3.3 Video Colorization:

The video colorization is the main aim of ours rather than coloring the static image. In this module, the input is taken as the video stream. As video is nothing but the series of number of images and images are nothing but frames. So in the next step the video is converted in to number of frames. After converting in to frames clustering is done on the basis of luminance matching. Luminance matching is done on the basis of pixels comparison. Then the input scribble step is carried out, in this step users need to give the input scribbles of the color they want to put in the particular region. When the user gives input the color automatically gets propagate throughout the frame using texture similarity. Accordingly all the frames get colorized in this manner and the frames are again synchronised to make the video. And the final output of the stage is the colorized video of the given gray scale one.

In the proposed methodology, we first convert the both gray and the color image in to the YCbCr color space. The reason to convert image from RGB color space to YCbCr color space is that we want to match the texture of both images and then colorize the image accordingly. And in the YCbCr color space texture can be extracted easily.

$$[v,x,y] = min2d(abs(py-pg(i,j)))$$

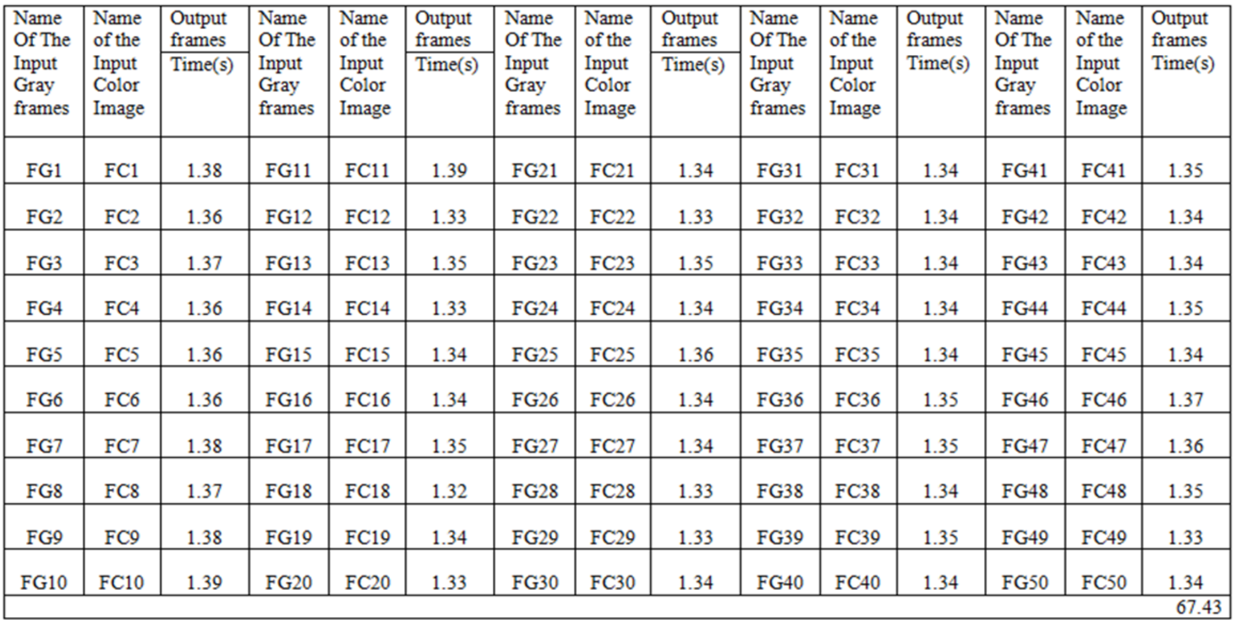
The minimum difference in the pixel is found and color is put from color image to gray one.

$$gr(i,j,2)=pal(x,y,2)..........(1)$$

$$gr(i,j,3)=pal(x,y,3)..........(2)$$

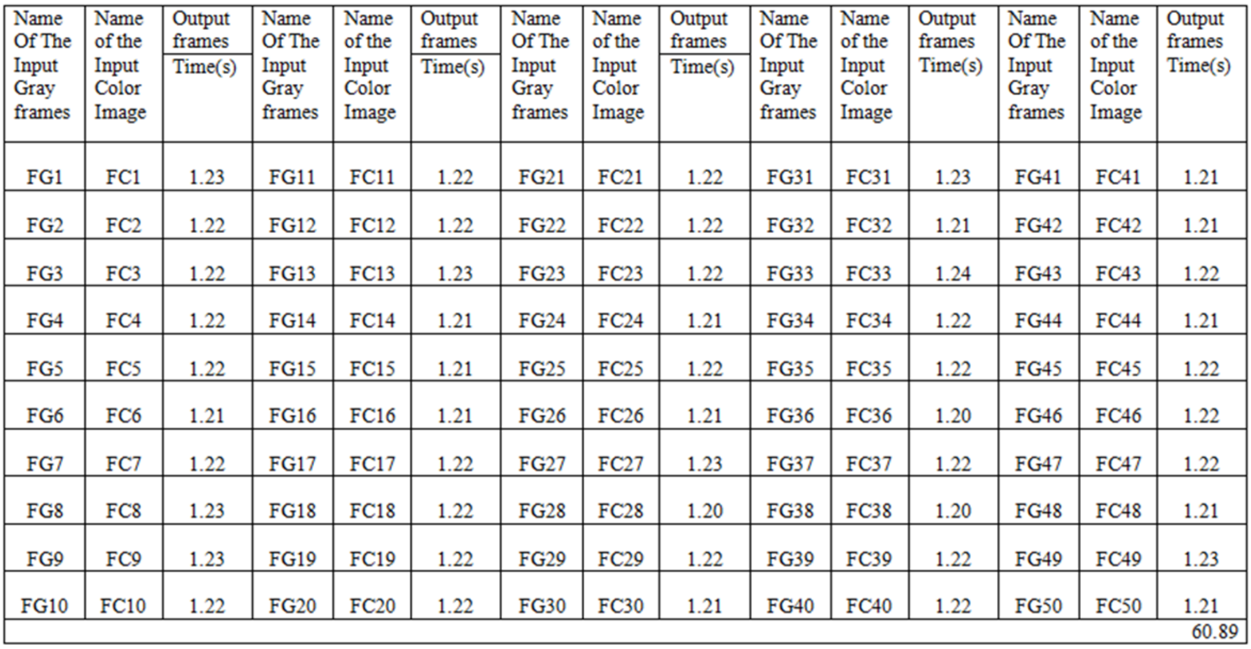
Above two equation shows how chrominance and luminance value is put in the gray image and the Y that is texture is keep same.

4. RESULT AND ANALYSIS:



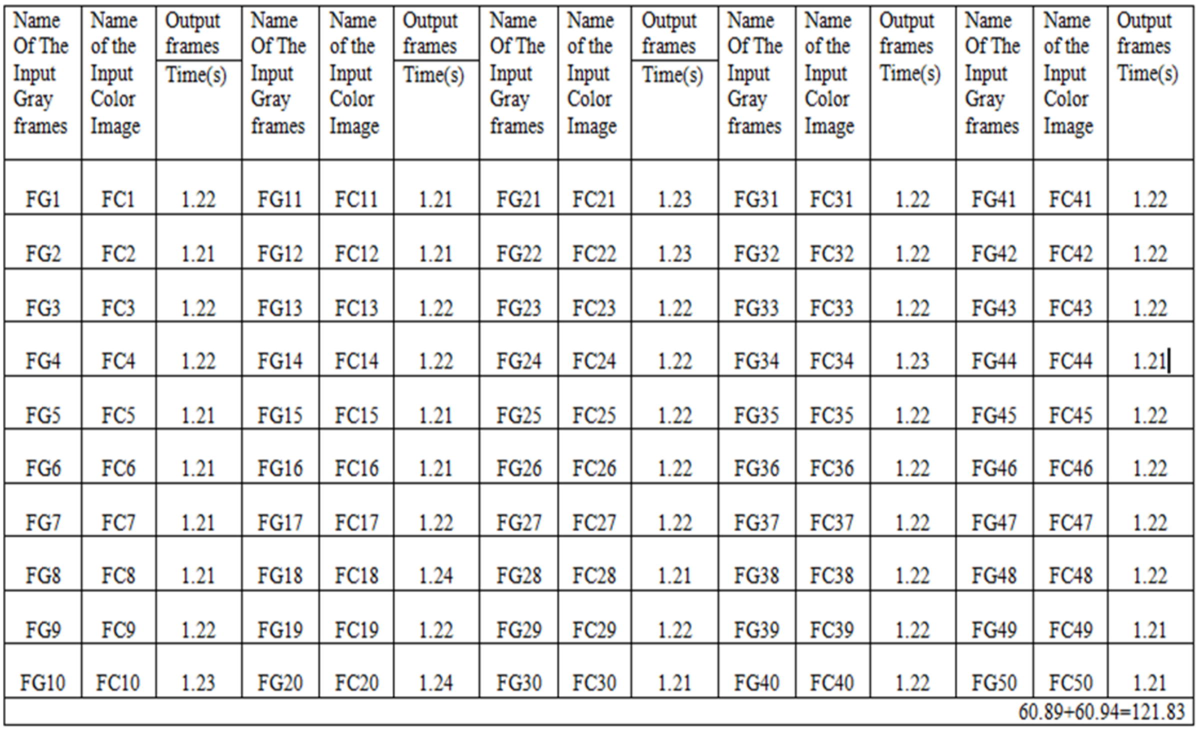
Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)
FG1	FC1	1.38	FG11	FC11	1.39	FG21	FC21	1.34	FG31	FC31	1.34	FG41	FC41	1.35
FG2	FC2	1.36	FG12	FC12	1.33	FG22	FC22	1.33	FG32	FC32	1.34	FG42	FC42	1.34
FG3	FC3	1.37	FG13	FC13	1.35	FG23	FC23	1.35	FG33	FC33	1.34	FG43	FC43	1.34
FG4	FC4	1.36	FG14	FC14	1.33	FG24	FC24	1.34	FG34	FC34	1.34	FG44	FC44	1.35
FG5	FC5	1.36	FG15	FC15	1.34	FG25	FC25	1.36	FG35	FC35	1.34	FG45	FC45	1.34
FG6	FC6	1.36	FG16	FC16	1.34	FG26	FC26	1.34	FG36	FC36	1.35	FG46	FC46	1.37
FG7	FC7	1.38	FG17	FC17	1.35	FG27	FC27	1.34	FG37	FC37	1.35	FG47	FC47	1.36
FG8	FC8	1.37	FG18	FC18	1.32	FG28	FC28	1.33	FG38	FC38	1.34	FG48	FC48	1.35
FG9	FC9	1.38	FG19	FC19	1.34	FG29	FC29	1.33	FG39	FC39	1.35	FG49	FC49	1.33
FG10	FC10	1.39	FG20	FC20	1.33	FG30	FC30	1.34	FG40	FC40	1.34	FG50	FC50	1.34
														67.43

Table 4.5: Time Constrain for Video Having 50 Frames



Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)
FG1	FC1	1.23	FG11	FC11	1.22	FG21	FC21	1.22	FG31	FC31	1.23	FG41	FC41	1.21
FG2	FC2	1.22	FG12	FC12	1.22	FG22	FC22	1.22	FG32	FC32	1.21	FG42	FC42	1.21
FG3	FC3	1.22	FG13	FC13	1.23	FG23	FC23	1.22	FG33	FC33	1.24	FG43	FC43	1.22
FG4	FC4	1.22	FG14	FC14	1.21	FG24	FC24	1.21	FG34	FC34	1.22	FG44	FC44	1.21
FG5	FC5	1.22	FG15	FC15	1.21	FG25	FC25	1.22	FG35	FC35	1.22	FG45	FC45	1.22
FG6	FC6	1.21	FG16	FC16	1.21	FG26	FC26	1.21	FG36	FC36	1.20	FG46	FC46	1.22
FG7	FC7	1.22	FG17	FC17	1.22	FG27	FC27	1.23	FG37	FC37	1.22	FG47	FC47	1.22
FG8	FC8	1.23	FG18	FC18	1.22	FG28	FC28	1.20	FG38	FC38	1.20	FG48	FC48	1.21
FG9	FC9	1.23	FG19	FC19	1.22	FG29	FC29	1.22	FG39	FC39	1.22	FG49	FC49	1.23
FG10	FC10	1.22	FG20	FC20	1.22	FG30	FC30	1.21	FG40	FC40	1.22	FG50	FC50	1.21
														60.89

Table 4.6: Time Constrain for Video Having 50 of 100 Frames



Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)	Name Of The Input Gray frames	Name of the Input Color Image	Output frames Time(s)
FG1	FC1	1.22	FG11	FC11	1.21	FG21	FC21	1.23	FG31	FC31	1.22	FG41	FC41	1.22
FG2	FC2	1.21	FG12	FC12	1.21	FG22	FC22	1.23	FG32	FC32	1.22	FG42	FC42	1.22
FG3	FC3	1.22	FG13	FC13	1.22	FG23	FC23	1.22	FG33	FC33	1.22	FG43	FC43	1.22
FG4	FC4	1.22	FG14	FC14	1.22	FG24	FC24	1.22	FG34	FC34	1.23	FG44	FC44	1.21
FG5	FC5	1.21	FG15	FC15	1.21	FG25	FC25	1.22	FG35	FC35	1.22	FG45	FC45	1.22
FG6	FC6	1.21	FG16	FC16	1.21	FG26	FC26	1.22	FG36	FC36	1.22	FG46	FC46	1.22
FG7	FC7	1.21	FG17	FC17	1.22	FG27	FC27	1.22	FG37	FC37	1.22	FG47	FC47	1.22
FG8	FC8	1.21	FG18	FC18	1.24	FG28	FC28	1.21	FG38	FC38	1.22	FG48	FC48	1.22
FG9	FC9	1.22	FG19	FC19	1.22	FG29	FC29	1.22	FG39	FC39	1.22	FG49	FC49	1.21
FG10	FC10	1.23	FG20	FC20	1.24	FG30	FC30	1.21	FG40	FC40	1.22	FG50	FC50	1.21
60.89+60.94=121.83														

Table 4.3: Time Constrain for Video Having 100 Frames

Table 4.1 shows the 50 frames of the video and time take to colorize that frames. In the next two tables 4.2 and 4.3 shows the video having 100 frames and respective time to colorize the video having 100 frames. From the above tables the technique present in the given paper is more efficient. It takes very less time to colorize the video.

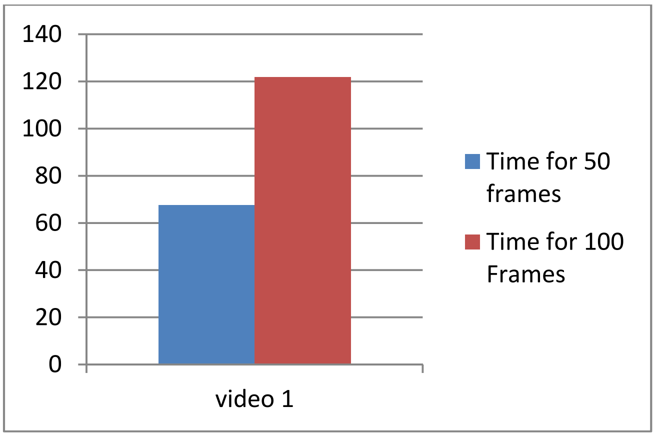


Fig. 4.4: Graph Shows Comparison of Time for Video at 50th Frames and 100th frames

5. CONCLUSION AND FUTURE SCOPE:

The main challenge of colorization is to assign similar colors to texture-similar regions. In this proposed technology, a novel approach to video colorization is proposed, which uses the New Feature Space to achieve good matching results. This method is highly parallelizable. This is applicable to various video data, especially the videos of natural scenes. Especially we are using more sophisticated monochrome texture descriptors in video sequences to improve the color propagation capabilities of the approach.

In future there is lot of scope to improve colorization of video. Because in the propagation of color there may remains artifacts due to user input scribbles. Color can be added to a range of scientific images for illustrative and educational purposes. In medicine, image modalities which only acquire grayscale images such Magnetic Resonance Imaging (MRI), X-ray and Computerized Tomography (CT) images can be enhanced with color for presentations and demonstrations.

The future research topics include the inclusion of texture information in the colorizing process. Secondly, the future scope includes the handling of un-concluded objects in video colorization.

REFERENCES

[1] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[2] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum , “Natural image colorization,” in *Proc. 18th Eurograph. Conf. Rendering Tech.*, 2007, pp. 309–320.

[3] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, May 2006.

[4] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” in *Proc. SIGGRAPH*, 2002, pp. 277–280.

[5] Y.-W. Tai, J. Jia, and C.-K. Tang, “Local color transfer via probabilistic segmentation by expectation-maximization,” in *Proc. IEEE Comput. Soc.Conf. CVPR*, vol. 1. 2005, pp. 747–754.

[6] R. Ironi, D. Cohen-Or, and D. Lischinski, “Colorization by example,” in *Proc. Rendering Techniques*. Eurographics, Konstanz, Germany, 2005, pp. 201–210.

[7] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, “Intrinsic colorization,” *ACM Trans. Graph.*, vol. 27, no. 5, pp. 152:1–152:9, 2008.

[8] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.- L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proc. ACM Multimedia*, 2005, pp. 351–354.

[9] V. Konushin and V. Vezhnevets, "Interactive image colorization and recoloring based on coupled map lattices," in *Proc. Graphicon*, vol. 4. 2006, pp. 231–234.

[10] B. Sheng, H. Sun, S. Chen, X. Liu, and E. Wu, "Colorization using the rotation-invariant feature space," *IEEE Comput. Graph. Appl.*, vol. 31, no. 2, pp. 24–35, Mar. 2011.

[11] M. Koleini, S. Monadjemi, and P. Moallem, "Film colorization using texture feature coding and artificial neural networks," *J. Multimedia*, vol. 4, no. 4, p. 240, 2009.

[12] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artif.Intell., vol. 17, nos. 1–3, pp. 185–203, 1981.

[13] M. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Comput. Vision Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[14] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *Lecture Notes Comput. Sci.*, vol. 3024, pp. 25–36, May 2004.

[15] A. Bruhn, J. Weickert, and C. Schn¨orr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *Int. J. Comput. Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[16] H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 5, pp. 565–593, May 1986.

[17] L. Alvarez, R. Deriche, J. S´anchez, and J. Weickert, "Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach," *J. Visual Commun. Image Representation*, vol. 13, nos. 1–2, pp. 3–21, 2002.

[18] N. Grammalidis and M. Strintzis, "Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 328–344, Mar. 1998.

[19] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video SnapCut: Robust video object cutout using localized classifiers," in *Proc. ACM SIGGRAPH Papers*, 2009, p. 70.