Fast and Fully Automated Video Colorization

V S Rao Veeravasarapu
Center for Visual Information Technology
IIIT-Hyderabad

Jayanthi Sivaswamy
Center for Visual Information Technology
IIIT-Hyderabad

***Abstract*—Colorization is the process of adding colors to grayscale images. This is done to restore or enhance old films or photographs. Most of the techniques for the colorization of an image or video require manual designation of the locations to be colored and the colors themselves which is an expensive and time consuming process. In this paper, we propose a fast but effective fully automated technique for coloring the gray scale image sequences. We define a notion of a most informative frame which is to be coloured manually and exploit the motion field between frames for propagation of the colors to the remaining frames. The proposed technique attempts to provide a method to minimize the amount of labour required for colorization and to decrease the computational cost of this task. The most informative frame is one which has almost all the objects present in that scene. The motion field estimation is based on optical flow. A final refinement step uses similarity based colour filling. Extensive testing of the proposed technique on a large set of videos from movies and animation confirms that it is efficient and effective without any loss of quality.**

I. Introduction

Colorization is the process of addition of color to a black and white video or still image. A colored image is a vector valued function often represented as three separate channels (Y,C_b and C_r). A gray scale image, in contrast, is a scalar valued function. Thus, the colorization process requires mapping of a scalar to a vector valued function which has no unique solution.

A number of automated and semi-automated techniques exist for colorizing monochrome video [3]-[12]. Automatic colorization process has got more attention from 1980s. We divided existing methods into four major classes. (a) *Psuedo-colorizing Methods*: These systems [11] involve the use of so-called "pseudocolorizers", which operate, in the manner of an electronic look-up table, to assign a particular color to the luminance value that is sensed at each elemental position or pixel of a monochrome frame. (b) *User involved systems* [3], [6] also exist in which an operator can specify, such as via a computer terminal, particular colors that are to be assigned to specified regions of a frame being colorized. These systems generally tend to be complex and expensive and do not provide the operator with adequate flexibility (in selection of seed points and their locations, colors etc.) while performing the colorization tasks. (c) *Reference image based colorization methods*: [9], [10], [7] transfer colors from a user-selected source image to a target grayscale image. Ideally, the source image should be similar in structure to the image to be colourized. Reduction in manual search for suitable reference images is achieved by using an image database and a content based image retrieval methodology. The cost of colorizing a video sequence is directly proportional to the time that it takes to color each frame, so it is highly desirable to have a system that is not unduly expensive. This also facilitates the rapid production of colorized video frames. The prior art systems suffer from one or more of the following disadvantages: (a) slow processing speed, with each frame to be colorized requiring many minutes or hours, resulting in unduly high processing cost [9]; (b) lack of operator convenience and flexibility (selection of seed points/regions on each frame and colors to them) that is needed to facilitate obtainment of quality colorized video [8]; (c) high computational complexity and/or cost(money) of the colorizing system itself [9].

In the colorization frame work proposed in [3], the user marks some sample points, called seeds, in the first frame and selects the colors of each seed. This step is done using a brush like tool. These colours are spread to the bounded surrounding region using an optimization technique using a quadratic cost function. The user only needs to select a few training points for each object. The colour propagation utilises pattern recognition and classification. Different techniques [5], [12], [4] are proposed to determine the similar neighbors or bounded region. They operate on the assumption that neighboring pixels with similar intensities should have similar colors. But the only disadvantage of this strategy is that the user has to interact for each frame in which a *new* object enters the scene which can be often and hence be a high load on the user. In order to decrease the amount of user interaction, we propose a fully-automated process to colorize a sequence representing a *single scene* which is extendable to a general video sequence. For every single-scene sequence, we assume that there is a frame which contains all the objects. We call this frame as the Most Informative (MI) frame. We propose colorizing this MI frame by using an existing static image colorization method followed by propagation of colors from colorized MI frame to remaining frames based on the motion vectors between frames. Motion estimation is performed using an optical flow technique. All pixels which are not colorized at the end of the previous step is colorized by a refinement step to assign colors. A general video sequence can consist of several scenes. Hence, our colorization strategy requires pre-segmenting the entire video into several scenes using a shot detection method [2], prior to colorization.

978-1-4673-2014-6/12/$31.00 ©2012 IEEE

II. PROPOSED METHOD

Let the given scene/shot be denoted as $I(n)$. The proposed system for colorization of $I(n)$ mainly consists of three parts: (a) MI frame selection; (b) Colorizing the MI frame; (c) Optical flow (OF) computation for all frames by taking MI frame as the starting frame. Thus, if the $n = k$ is the MI frame, then OF is computed between the frame pairs $I(k \pm i)$, $I(k \pm (i+1))$; $i = 0, 1, 2..$; (d) Propagation of color; (e) Refinement for colorizing the remaining pixels. The flow chart of the proposed algorithm is given in Fig.1.

A. Most Informative (MI) frame selection

The MI frame is defined to be the frame which contains maximum number of objects present in that scene. Hence, this frame should have maximum spatial activity (entropy) and highest amount of edges across the sequence. We detected this frame by maximizing a score S which is defined as,

$$S(n) = w_h \frac{H(n)}{\sigma_h} + w_e \frac{E(n)}{\sigma_e} \tag{1}$$

Here, σ_h and σ_e are the standard deviations of $H(n)$ and $E(n)$ respectively. w_h and w_e are empirically determined weights. The first term $H(n)$ captures to the spatial activity and the second term $E(n)$ corresponds to the amount of edge content of the n^{th} frame. $H(n)$ is the entropy of the n^{th} frame and is given as

$$H(n) = -\sum_x p(x,n) \log_2(p(x,n)) \tag{2}$$

where $p(x,n)$ is the probability of the grey value x in the intensity histogram of the n^{th}-frame. For an 8-bit image, $H(n) \epsilon [0, 8]$.

The amount of edge content present in the n^{th}-frame is determined by computing the energy of the gradient of the frame I_n.

$$E(n) = \sum_x \sum_y |\frac{\partial I_n}{\partial x} + \frac{\partial I_n}{\partial y}|^2 \tag{3}$$

Finally, the MI frame is selected as the frame with the highest score.

$$k = argmax_{k \epsilon n} S(n) \tag{4}$$

where k is the frame number of MI frame.

The above summarized procedure for MI frame can also be used to automatically extract from a video sequence a single key frame representative of its content.

B. Colorizing the MI frame and Propagation

After the section of MI frame, we colorize the MI frame using a scribble based colorization process from [3] as explained earlier. As seen in the Fig.1, there are two loops to colorize the frames which are on left (<k) and right (> k) side of the MI frame. Let us explain the processing of right loop. The positions of the pixels in next frames ($> k$) which are displaced from the k^{th}-frame are next estimated using flow vectors followed by propagation of colors to the next frames according to motion vectors. Any missing pixels in this propagation (new incoming pixels of current frame) are colored with the help of a refinement step which contains rules for color assignment. This will also help decide the need for any user interaction for that frame. This loop will continue until the last frame (N) in the scene is colored. In the similar way, Left loop will continue until first frame in the scene colored.

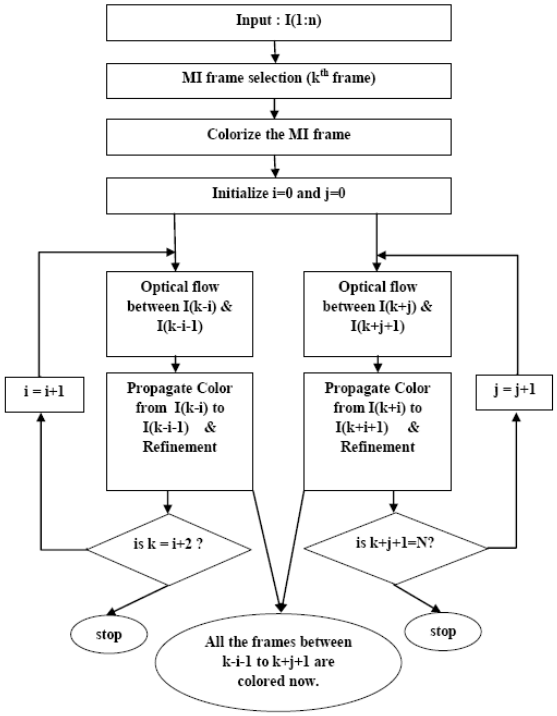

Fig. 1. Flow chart of the colorization scheme

In each loop, at each iteration one frame is colored. The following sections discusses about some common blocks in the both right and left loops.

C. Optical flow computation

Optical flow (OF) computation is a standard technique to estimate the motion field between two consecutive frames. OF is computed under the assumption that the brightness of the object remains constant from the initial point of $I(x,y,t)$ (in the current frame) towards the latest position of $I(x+\delta x, y+\delta y, t+\delta t)$ (in the next frame). Thus, the brightness constancy constraint of a point is represented as follows:

$$I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t) \tag{5}$$

The brightness constraint can be represented using the 1^{st} order Taylor series expansion of the right hand side of eqn.5 as,

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t \tag{6}$$

Replacing the right side of eqn.5 with the result in eqn.6, we arrive at the basic constraint equation for OF

$$I_x v_x + I_y v_y + I_t = 0 \tag{7}$$

The brightness constancy assumption is very sensitive to brightness changes that often appear in real cases. Therefore, it is important to introduce small variation of intensities within the initial intensity value itself. This is done via a gradient constancy assumption which is expressed as:

$$\nabla I(x, y, t) = \nabla I(x + v_x, y + v_y, t + 1) \tag{8}$$

$$\begin{aligned} E_{Data}(v_x, v_y) = \int_{\Omega}^{\psi} (|I(X + w) - I(X)|^2 + \\ \gamma |\nabla I(X + w) - \nabla I(X)|^2) dX \end{aligned} \tag{9}$$

where $w = [v_x, v_y]$, $X = [x, y]$ and γ is weighting parameter. To overcome aperture problem [1], an additional smoothness constraint is imposed on flow field which is based on the reasoning that the OF computation should not only be based on a single pixel displacement but also relate to the displacement of neighbouring pixels. In other words the smoothness constraint requires OF to be piecewise smooth. This smoothness criterion is computed using total variation of the piecewise smooth flow field which can be expressed as:

$$E_{smooth}(v_x, v_y) = \int_{\Omega}^{\psi} (|\nabla v_x|^2 + |\nabla v_y|^2) dx \tag{10}$$

The desired flow vectors v_x and v_y are determined by minimizing the energy functional $E(v_x, v_y)$ defined as

$$E(v_x, v_y) = E_{Data} + \alpha E_{Smooth} \tag{11}$$

$$(v_x, v_y) = argmin_{(v_x, v_y)}(E(v_x, v_y)) \tag{12}$$

where $\alpha > 0$ is the regularization parameter. The derivation of brightness and gradient constancy assumption can be expressed as:

The above summarized procedure of optical flow computation was used for motion field estimation to generate the desired motion vectors at every pixel.

D. Color propagation

This is the main task in the colorization process. We assign or propagate the color to a pixel in the current frame from its corresponding location in the neighbouring frame. For example, let us consider the right loop. In the first iteration (j=0), the colors of pixels in I(k+1) are found from the colors from I(k), which is the MI frame, according to the motion between the frame pairs. Likewise, in the next iteration (j=1), the pixels in I(k+2) will inherit colors from I(k+1) according to the motion between these frame pairs. This process repeats until the last frame (N). A similar process is followed to colour frames which are left neighbours of the MI frame in the left loop.

E. Refinement

It is possible that some pixels are missed in the process of color propagation between the frames. Colorizing them is the refinement process. This is based on a test for similarity between the greyvalues of the pixel to be colourized (missed pixel) and its neighbours which are already colourized: (1) If the intensity (Y) of the missed pixel is similar to its neighbouring pixel, then they should have same colors (chromatic values: C_b, C_r). (2) If a set of connected pixels are missing and the size of this cluster is more than 5x5, then it either signals the introduction of a new object into the scene or shadow formation due to a change in illumination. This is best resolved by involving the user. Hence, in this case, the user is asked to decide the color with scribble or marker.
The proposed scheme should result in the requirement of a relatively smaller number of user input. This was also seen to be true when experimenting with our large variety of videos.

III. RESULTS

The proposed colorization system was implemented in MATLAB-R2009a on Windows7 ultimate (32-bit OS), Processor Intel(R)Core2Duo 1.83 GHz, RAM 3GB. It was tested on videos from Levin's data base [3] and animated videos which were independently obtained. In all cases, the system was able to produce high quality colored videos in a short time. We present some sample in this section. In order to assess the performance of the proposed method a set of comparisons were carried out: Comparison with (i) state of art, (ii) ground truth and (iii) some challenging sequences like animation sequences.

A. Comparison with State of Art

Fig.2 shows some selected frames from a greyscale movie clip (containing 83 frames) and the corresponding colorized frames. Our system detected frame 18 as the MI frame and did not ask for user interaction for the entire movie clip except for MI frame colorization. The colorization process of this video took about 1 minute and 42 seconds. For comparison, results of colorization with the method in [3] is also shown. This method required 12 scribbles from the user. Our results indicate comparable quality of results. Since our method uses Optical Flow only to define the local temporal neighborhood, it is robust to tracking failures. Some zoomed details are also provided in the bottom row for a detailed comparison from which we can observe a small amount of overlapping between colors at the edges. This is because of the erroneous motion vectors given by OF method at edges. However, these will be invisible while playing a movie.

Sample MI (frame 27) and some of preceding (frame 5) and later (frame 33) frames for a scene captured by a camera in a car are provided in Fig.3. The reflection on the windscreen has a subtle change in colour which is successfully propagated in the distant (5^{th}) as well as relatively proximal (33^{rd}) frames. The effectiveness of the proposed system and color propagation are better observed in videos which have been made available at http://web.iiit.ac.in/~vsrao/colorization.

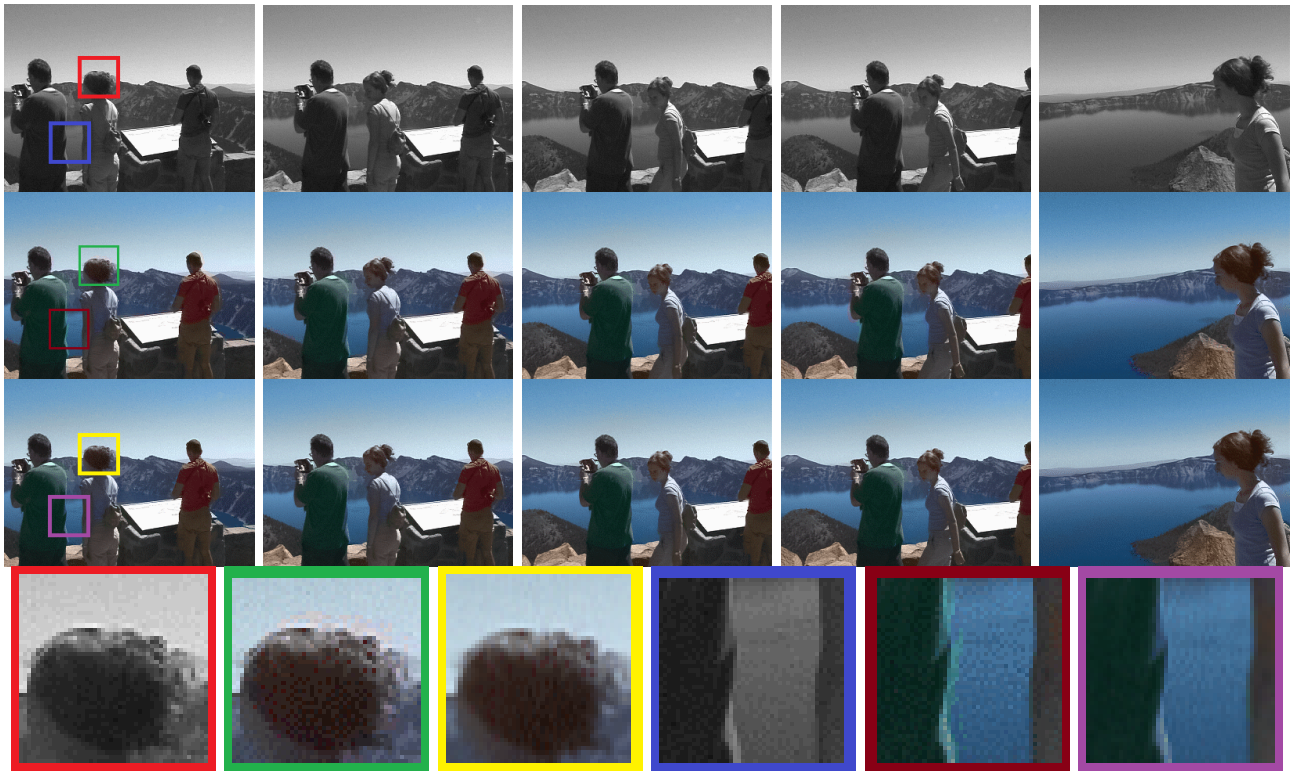

Fig. 2. Colorization of an Lake scene from a 83-frame clip. First row: greyscale input frames (1,9,18 (MI frame),67,83); Corresponding frames colorized by the proposed (second row) and scribble based methods [3] (third row); Sample zoomed regions are shown in the bottom row.

(a) 5 (b) 27 (c) 33

Fig. 3. Colorization of a car video scene. From left to right: 5^{th}, 27^{th} (MI frame) and 33^{rd} frames. Top row: greyscale input; Bottom row: colorized result.

B. Comparison with Ground-truth

We also evaluated our system on colour videos by colorising the greyscale version and using the original colour as ground truth for validation. Some sample frames of a party scene (consisting of 63 frames) are shown in Fig.4. For this scene, frame 19 was detected as the MI frame. No other user input was given. The results for the distal (5^{th} and (45^{th}) frames still appear to be very close in quality to the original colour frames. This is also illustrated by using $PSNR$ measure to quantify error in colourisation. The PSNR value for the n^{th} colorized frame is given by,

$$PSNR(n) = 20\log_{10}(\frac{255}{MSE(n)}) \quad (13)$$

where MSE(n) is the mean squared error between the original and the colorized n^{th}frame. Fig.5 shows the PSNR plot for 63 frames, with the x-axis representing the frame number.

Fig. 4. Colorization of a party video scene. From left to right: 5^{th}, 19^{th} (MI frame) and 45^{th} frames. First row: greyscale input; second row: colorized result; third row: ground truth.

Generally, the higher the PSNR value the more similar is the colored image to the original one. The plot peaks at 19^{th} frame which is the MI frame. The first and final frames have least PSNR. The PSNR degrades for non-MI frames due to the color propagation error. However, given the highly magnified scale for the y-axis in this plot, the degradation is only by 2 % which demonstrates the effectiveness of the propagation strategy. The entire colorization process of 63 frames took 93 seconds.

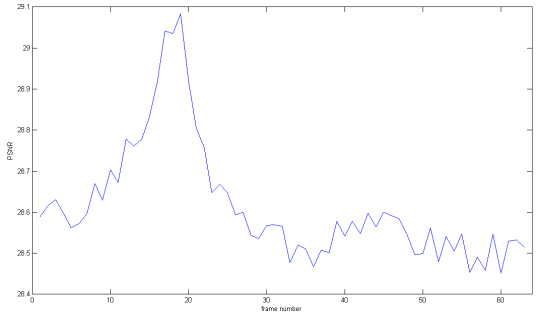

Fig. 5. PSNR plot for the party movie scene with 63 frames.

C. Animated movie scenes

Colorisation on animated movie scenes from *Finding Nemo* and *Megamind* were also tested. Fig.6 shows our result and ground truth (GT) version of frame 31 from a scene of *Finding Nemo* movie. We can clearly observe that there is very little overlap between color content of objects at edges. In general, animated movie scenes contain large motion fields. Hence, optical flow algorithms might be more erroneous in such cases. Large displacement optical flow methods are more suitable for such sequences, however they are computationally expensive. Sudden appearance of objects in a scene is another characteristic of such sequences. Accordingly, the required number of user interaction can increase for our system. In our experiments with a large dataset, the maximum number of times that user interacted to decide color with scribbles was found to be 16 for a clip of 274 frames from *Megamind* animated movie.

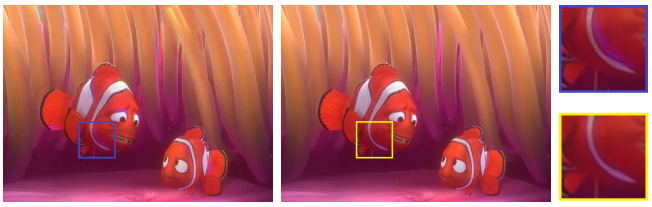

Fig. 6. Ground truth and colorized result for frame 31 of *Finding Nemo* video scene. (a) our result, (b) original frame (GT), (c) zoomed regions.

Run time Statistics: Key features of the proposed system are computational simplicity and a greatly reduced need for user interaction. This is demonstrated by the time required for computations in our method with that of Levin's method [3] to colorize the scene in Fig.3 which has 60 frames. [3] reports that the user interaction is used for the first frame and 11 other frames. The required computations are listed in Table 1. Here, U is the unit time taken by the user to specify colors to seed points in a frame which is a minimum of 30 seconds. F is the unit time for colorization of the MI frame and M1, M2 are the unit times taken for color propagation between two frames by [3] and our methods respectively. M2 is inclusive of the refinement time. The total time shown for [3] is the time takenby using the code available at http://www.cs.huji.ac.il/~yweiss/Colorization/.

Task	Method[3]	Our method
MI frame selection	0	0.8 sec
Colorizing key frame	U+F	U+F
User interactions	11(U+F)	0
Color propagation	48M1	59M2
Total Time	12U+12F+48M1 $\sim$ 17.04 min	U+F+0.8+59M2 $\sim$ 2.33 min

IV. CONCLUSIONS

The process of colorization remains a manually intensive and time consuming process. In this paper, we have suggested a method that helps graphic artists to colorize films with less manual effort. We propose a framework which capitalises on the notion that not all frames will have maximum information together with the fact that frames of a scene are related by a motion field. Thus, an artist needs to color automatically selected *most informative* frames (1 per scene) which is subsequently propagated using an optical flow-based algorithm. With the current framework, little more user effort is needed when the video contains more objects not all of which may be present in one frame such as capturing a scene with rotating camera or a still camera capturing a busy road (surveillance videos) scene. In such scenarios also, user effort for the proposed method is far less than that of other methods. Our future work aims at colorization of these kind of scenes with least user interactions.

REFERENCES

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV 2004*.

[2] Z. Cernekova, C. Kotropoulos, and I. Pitas. Video shot segmentation using singular value decomposition. In *ICASSP'03*. IEEE, 2003.

[3] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics*, volume 23, pages 689–694, 2004.

[4] Y. Li, M. Lizhuang, and W. Di. Fast colorization using edge and gradient constrains. *Proceedings of WSCG'07*, pages 309–315, 2007.

[5] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y. Xu, and H. Shum. Natural image colorization. In *Eurographics Symposium on Rendering*, 2007.

[6] NEURALTEK. Blackmagic photo colorization software, version 2.8. *http://www.blackmagic-color.com/*, 2003.

[7] E. Reinhard et al. Color transfer between images. *Computer Graphics and Applications, IEEE*, 21(5):34–41, 2001.

[8] G. Sapiro. Inpainting the colors. In *ICIP 2005*.

[9] Vieira et al. Fully automatic coloring of grayscale images. *Image and vision computing*, 25(1):50–60, 2007.

[10] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics*, volume 21. ACM, 2002.

[11] L. Williams and J. Bloomenthal. System for colorizing video with both pseudo-colors and selected colors, Aug. 26 1986. US Patent 4,608,596.

[12] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing-2006*, 15(5):1120–1129.