GRAPH MODELLING OF 3D GEOMETRIC INFORMATION FOR COLOR CONSISTENCY OF MULTIVIEW IMAGES

Manohar Kuse, Sunil Prasad Jaiswal

The Hong Kong University of Science and Technology, Hong Kong

ABSTRACT

In this paper we address the problem of multiview color consistency. We propose to use a graph model of 3d positions obtained using matched dense feature points. We define an energy functional on this model which captures relationship of the colors across views while also imposing a smoothness cost to obtain optimal colors for the 3d positions. We finally recolorize the images using these optimal colors at reprojected co-ordinates. An important feature of the proposed method is that it does not use a reference view. Finally we present a qualitative evaluation of our method with methods that use a reference view.

Index Terms— Color consistency, colorization, multiview colors.

1. INTRODUCTION

In recent times several works on 3d reconstruction from image-sets have been published [1]. 3D reconstruction pipelines generally involve a set of images from different view points, from which 3D models of the environment are generated. In computer graphics community it is referred to as Image-based-modeling. Most of the applications of it is modeling and visualizing an environment in 3d. In this work we explore the possibility of achieving color synchronization across views by use of 3d model of the scene.

Color blending of spatially adjacent views is a commonly employed technique to alleviate the problem of inconsistency in color. However in case the images have different color tones (as shown in Fig.1), no amount of blending can give a photo-realistic effect.

Several approaches have been proposed for color consistency between a pair of images. Xu *et al.* [2] present a review of various approaches for color correction of a pair of image. Of particular interest was the approach by Tai *et al.* [3]. They make use of the EM algorithm for obtaining probabilistic segmentation of the image. Then using correspondences between the images, they propose to map the Gaussian components between the images and transfer their colors using the Reinhard color transfer algorithm [4].

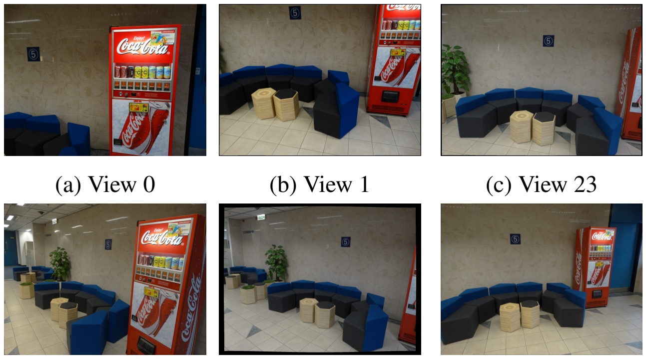
A simple way to extend the pairwise colorization to multiple images is to select one of the view as a reference view. Using the colors in the reference view transfer the colors to other views. These approaches are sensitive to the reference frame chosen. The reference view may not contain all the colors shades present in the scene causing unusual colorization of the target image.

To the best of our knowledge, until now only one study (by Moulon *et al.* [5]) attempts a global all view color consistency. They starts by finding common pixels in the image sequence $I_i,\ i = 1, \cdots, n$. Histograms are used to model distribution of colors between matched regions. A gain g_i and offset o_i is used per image to normalize the shutter speed and the aperture time which are determined by an optimization framework. Optimal g_i and o_i are used for intensity transformation of the original sequence. A drawback of their scheme is that it takes a view as a reference view. This might not be suitable if the views are very different from each other. Yet another issue with the formulation is that they assume the intensity of the patch in image i is related to that of corresponding patch from image j with an affine function. This may not necessarily be true. Further this method assumes that all the intensities can be transferred independently. It does not take into account the intensities of the neighboring pixels for the color transfer.

The inputs to our system are a set of images of different appearance and unknown capture conditions. The objective of this project is to re-colorize the images and ensure consistency of colors across multiple views (see Fig. 1). An important feature of the proposed method is that it does not set a view as a reference view to recolor other views. Instead it relies on the 3d reconstruction to choose optimal colors at dense keypoints.

2. PROPOSED APPROACH

We propose a scheme for global color consistency. The most prominent feature of the scheme is that it does not require a reference view. We present an approach that maps the visible 2d points into 3d positions. Consistency of color is ensured by the proposed local photometric consistency optimization process on the 3d points. The first step is to extract EXIF information from the images to obtain the calibration parameters and relative camera pose. A sparse set of keypoints are initially mapped to their 3d positions. Details of this can be



(a) View 0 (b) View 1 (c) View 23

(d) View 26 (e) View 27 (f) View 28

Fig. 1: Showing selected views of the dataset. Inconsistency in color. For example, shades of blue of the chairs, shade of back wall are different across views.

found in section 2.1. Then these are repeatedly expanded to nearby pixel correspondences to obtain a dense set of rectangular patches (see section 2.2). Once we have a dense set of matches in multiple views, we formulate an optimization problem to ensure local photometric consistency on the 3d points (section 2.3). With a consistent set of colors we recolorize the image based on a successful colorization frame work presented by Levin *et al.* [6].

2.1. Reconstruction of Sparse Geometry

The objective here is to geometrically register a set of images and extract the 3d information for a set of interest points. The approach proceeds by detecting interest keypoints in each image. SIFT keypoints [7] were used to find features in each images because of its invariance towards image transformation and varying lightening conditions. Matches of keypoint across pair of images is found. And then finally a robust structure from motion (SfM) procedure is used to recover the camera parameters. We rely on the "Bundler"[1] system for structure from motion. Bundler takes a set of images, image features, and image matches as input, and produces a 3D reconstruction of the (sparse) scene geometry and camera pose as output. The system is described in [8, 9] and uses the Sparse Bundle Adjustment package of Lourakis and Argyros [10] as the underlying optimization engine.

2.2. Multi-view Dense Stereo

Furukawa and Ponce [11] proposed a technique that takes calibrated set of images and produces dense set of patches covering the surfaces visible in input images. They implemented multi-view stereopsis as a match, expand and filter procedure. It essentially spreads the initial matches to nearby pixels to obtain a dense set of patches. PMVS[2] is a multi-view stereo software that takes a set of images and camera parameters, then reconstructs 3D structure of an object or a scene visible in the images.

2.3. Local Volumetric Photo-consistency

Let $P_0, P_1, \ldots, P_N$ denote N 3d points obtained using PMVS software as noted in section 2.2. A point P_i is visible in a set of views (which is a subset of all available views) denoted by V_{P_i}. $v_j^{(P_i)}$ $j = 1 \ldots |V_{P_i}|$ are the elements of the set V_{P_i}. Let r_i^s denote the projection of point P_i on the view s ($s \in V_{P_i}$). From the basic principles of projective geometry we know that –

$$r_i^s = M_s P_i$$

Where M_s denote the 3×4 projection matrix corresponding to view s. Note that the projection matrices can be readily obtained using the camera parameters and camera pose obtained using the bundler package (SfM) as described in section 2.1. In our approach we achieve color synchronization across views by computing an optimal set of colors for the 3d points. For this purpose we propose to use a graph model. Further we define an energy functional over the graph model. Finally we show that the proposed combinatorial optimization framework reduces to the labeling problem, which is an NP-hard problem.

As stated earlier, we model the 3d arrangement of points as a graph model $G = (P, E)$. Let the 3d points be the nodes of the graph. Thus, $P = \{P_i | \ i = 0, \ldots, N\}$. The edge set of the graph contains the k nearest neighbors of the nodes. Thus $E_i = \{k \text{ nearest neighbors of} P_i\}$. $E = \{E_i | \ i = 0, \ldots, N\}$. For efficiency we find the approximate nearest neighbors (ANN) using kd-tree package of Arya *et al.* [12]. Let l_i ($l_i \in \{0, 1, \ldots, 255\}$) denote the label of the node associated with P_i.

We define the Energy functional for the graph G as –

$$\Delta(G) = \xi \sum_{i=0}^{N} \Phi(P_i, l_i) + \sum_{(i,j) \in E} \Psi(l_i, l_j) \tag{1}$$

Where, $\Phi(P_i, l_i)$ is the cost of setting the node associated with P_i as l. $\Psi(l_i, l_j)$ is the smoothness penalty with respect to its neighbors in the graph. We define the cost function as follows.

$$\begin{aligned} \Phi(P_i, l_i) &= \sum_{s \in V_{P_i}} \left(c(r_i^s) - l_i\right) \\ &= |V_{P_i}| \Big[\frac{1}{|V_{P_i}|} \Big(\sum_{s \in V_{P_i}} c(r_i^s) \Big) - l_i \Big] \end{aligned} \tag{2}$$

$c(r_i^s)$ is the color intensity of one of the channels. Setting $\xi = \frac{1}{|V_{P_i}|}$. Also observe that the internal first term in equation 2 is the average observed color of P_i. For robustness we

[1] http://www.cs.cornell.edu/~snavely/bundler/
[2] http://www.di.ens.fr/pmvs/

compute this average in La^*b^* color space instead of usual RGB space and then reconvert the averages to RGB space.

$$\Psi(l_i, l_j) = \begin{cases} \lambda_{i,j} & l_i = l_j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The objective is to find a labeling (ie. set of labels associated with every node) such that the energy is minimized. It is a well known fact that computing the minimizer of equation 1 is a NP-hard problem when number of possibilities of a label is greater than 2 [13]. Recently, Gridchyn and Kolmogorov [14] proposed an efficient technique to minimize an energy function with Potts smoothness term (equation 3). We rely on this optimization engine[3] to find a minimizer for equation 1.

Minimizing the energy will give labellings that is as close as possible to the original reprojected colors and also shall give spatial smoothness of colors. The labels in this case denote the intensities of a color channel. Thus, similar energy functionals need to be minimized for each of the color channels (viz. R,G,B).

Finally we set the optimal labels, as color intensities for each of the reprojected points on the images. Thus we obtain images with optimal set of colors at dense feature points.

2.4. Re-colorization

Once we obtain a globally consistent set of colors at the interest points as described in section 2.3, we re-colorize each of the images using these newly obtained color values at interest points. The re-colorization framework is inspired from Levin *et al.* [6]. The algorithm is given as input the gray scale intensity image $Y(x, y)$ at every (x, y). In addition to that it has color information at a few interest points. The algorithm outputs the two color channels $U(x, y)$ and $V(x, y)$ at every (x, y). To simplify the notation we use boldface $\boldsymbol{r}$ to denote the pair (x, y).

Next we define an objective function which imposes that the two neighboring pixels $\boldsymbol{r}$ and $\boldsymbol{g}$ have similar colors if their intensities are similar. Thus, we wish to minimize the difference between the color $U(\boldsymbol{r})$ at pixel $\boldsymbol{r}$ and weighted average of the colors at neighboring pixels :

$$J(U) = \sum_{\boldsymbol{r}} \big(U(\boldsymbol{r}) - \sum_{\boldsymbol{g} \in N(\boldsymbol{r})} w_{\boldsymbol{rg}} U(\boldsymbol{g})\big)^2$$

Where $w_{\boldsymbol{rg}}$ is a weighting function that sums to one, large when $Y(\boldsymbol{r})$ is similar to $Y(\boldsymbol{g})$ and small when two intensities are different. $w_{\boldsymbol{rg}} \propto e^{-\frac{(Y(\boldsymbol{r})-Y(\boldsymbol{g}))^2}{2\sigma^2}}$. Such functions are also referred to as affinity function in the literature. By inspection the cost function $J(U)$ can be expressed in a quadratic form, ie. $J(U) = U^T(D - W)U$. Where W is an $npixels \times npixels$ matrix whose elements W_{rg} denote the weights w_{rg}. D is a diagonal matrix containing row-wise sums. Note that W is a sparse matrix with any row (or column) r of it containing non-zero entries in positions which are neighbors of pixel r in the image. To recolorize the image I_s (ie. view s) we minimize $J_s(U)$ and $J_s(V)$ subject to the constraints that $U(c_{s,j}) = p_{s,j}$ and $V(c_{s,j}) = q_{s,j}$. $J_s(U)$ and $J_s(U)$ denote the cost functions for image I_s. It is to be noted that $p_{s,j}$ and $q_{s,j}$ are known constants representing the U and V color components respectively at j^{th} interest points of s^{th} view. Note that these were obtained as described in section 2.3. For an image I_s, the problem can be stated as :

$$\underset{U}{\text{minimize}}\ U_s^T (D_s - W_s)\, U_s$$
$$\text{subject to } U(c_{s,j}) = u_{s,j},\ j = 1 \dots n_s$$

Where, U_s represents the vectorized U color values of I_s. The equality constraint represents the known color values at interest points. It can be written in matrix form as $AU = u$. A is a large sparse matrix of size $n_s \times npixels$. This can be re-written as a standard quadratic programming problem with equality constraints. Define $B = 2(D - W)$.

Levin *et al.* [6] has suggested to find the optimum value of the above function by use of graph cuts. However, since this is a quadratic program with equality constraints, it can be reduced to a problem of solving system of linear equations. We provide the necessary and sufficient conditions under which an optimal solution can be computed. Since objective function and the constraints are differentiable and continuous the solution U^* to this system satisfies the Karush-Kuhn-Tucker (KKT) conditions [15]. The U^* that minimizes the above objective must satisfy the conditions of primal feasibility and zero gradient of the Lagrangian.

$$AU^* = u \quad (4)$$

$$\frac{\partial L(U, \nu)}{\partial U} = B^T U^* + A^T \nu^* = 0 \quad (5)$$

$L(U, \nu) = U^T BU + \nu^T(AU - u)$ is the Lagrangian function of the primal optimization problem. The equation 4 and equation 5 can be written up together as a matrix equation.

$$\begin{bmatrix} A & 0 \\ B & A^T \end{bmatrix} \times \begin{bmatrix} U^* \\ \nu^* \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

This equation is a large square sparse system of linear equation. The optimal U can be obtained by solving the linear equation. An iterative method for solution of a sparse linear equation like the Generalized minimum residual method can be employed to obtain the optimal U. Following the same exact principle an optimum V for I_s can also be obtained for re-colorization of I_s.

3. RESULTS

We implement our method and apply it to a variety of examples on an Intel i7 3.4 Ghz workstation. However due to space

[3] Available at http://pub.ist.ac.at/~vnk/software.html

(a) View 0 (b) View 2 (c) View 26

(d) View 0 (e) View 15 (f) View 27

(g) View 5 (h) View 6 (i) View 24

(j) View 0 (k) View 26 (l) View 28

Fig. 2: Selected patches of recolorized Images. 1^{st} row showing the coke dispenser. 2^{nd} row showing a set of chairs. 3^{rd} row showing the background wall and 4^{th} row showing a colorful sticker on the dispenser.

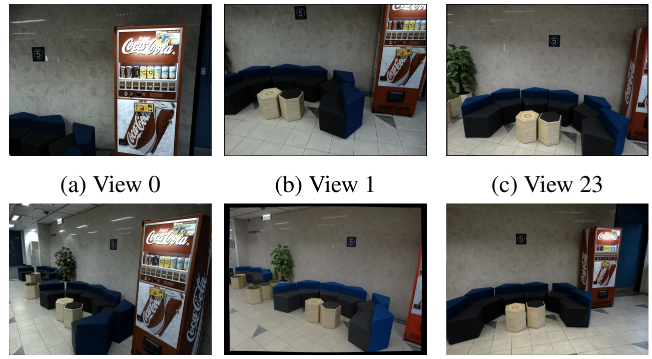
limitations we present only 1 scene (referred to as *coke*) example. It consists of 4 sets of 7 images (2000×1500 pixels) all of same scene clicked using a Sony DSC-HX300 camera under 4 different lightening settings. Upon acceptance of the paper more results will be available online on our website[4].

For our experiments we used $k = 4$ nearest neighbors for construction of the nearest neighbor graph of 3d points. For robustness we use a clipped datacost, thus we use the datacost as $min(\Phi(P_i, l_i), \tau)$. We set $\tau = 100$. We also used a simple and adaptive technique to set $\lambda_{i,j}$. If the colors at node$_i$ and node$_j$ are different we set $\lambda_{i,j}$ as $\lambda_{low}(= 5)$. For similar colors we set $\lambda_{i,j}$ as $\lambda_{high}(= 20)$. The difference of colors is quantified as the euclidean distance of the difference of the 2 chrominance channels ($threshold = 40$ was used). If the chrominance of the adjacent nodes is very different the nodes belong to different object and in this case the smoothness constraint is to be relaxed, thus we set a lower penalty.

Figure 2 shows some of the interest zone of the recolorized images from various views. We observe that the color details are preserved, unlike in the case of color transfer (Fig. 3) where we observe lot of false colorization and loss of color details. In case of the results obtained using the proposed method, we observe that the colors of the background wall (row-3 in Fig.1) are lot similar unlike the original image set (Fig.1) in which they have different color tones.

We compare the results obtained with the color transfer by setting one of the views as reference. We selected view-5 as a reference view and transferred the colors to other views. Since the view had little of red it has resulted in false looking colors. Since the reference view does not contain enough of red color, we get false shades of red after color transfer as can be observed in Fig. 3.

[4]http://aaa.bbb.com



(a) View 0 (b) View 1 (c) View 23

(d) View 26 (e) View 27 (f) View 28

Fig. 3: Result of adapting pairwise color transfer[4] scheme onto multiview using view-27 as reference image

4. CONCLUSION

We present a scheme to re-colorize a set of images from multiple views in a color consistent way. We obtain a set of consistent colors from dense set of image interest points. A graph model is defined over the 3d points of the scene. An objective function is defined with a smoothness penalty which measures the difference of colors between reprojections of spatial neighbors. Using these optimal set of consistent colors at reprojections, we colorize the image.

The proposed approach is different from the one presented by Moulon *et al.* [5] is that we do not need a reference view. Further, unlike Moulon *et al.*, the color transformations are not restricted by only an affine function. Instead, the colorization process iterates until the colors are properly propagated across the entire image. Experiments and comparisons show that our method leads to high-quality color consistent across views.

5. REFERENCES

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski, “Building Rome in a day.,”

in *ICCV*. 2009, pp. 72–79, IEEE.

[2] Wei Xu and Jane Mulligan, "Performance evaluation of color correction approaches for automatic multi-view image and video stitching," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 263–270.

[3] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang, "Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization.," in *CVPR (1)*. 2005, pp. 747–754, IEEE Computer Society.

[4] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley, "Color Transfer between Images.," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.

[5] Pierre Moulon, Bruno Duisit, Pascal Monasse, et al., "Global Multiple-View Color Consistency," in *Proceedings of Conference on Visual Media Production*, 2013.

[6] Anat Levin, Dani Lischinski, and Yair Weiss, "Colorization using optimization.," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2003, vol. 20.

[8] Noah Snavely, Steven M. Seitz, and Richard Szeliski, "Photo tourism: exploring photo collections in 3D.," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.

[9] Noah Snavely, Steven M. Seitz, and Richard Szeliski, "Modeling the World from Internet Photo Collections.," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[10] Manolis Lourakis and Antonis Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," Tech. Rep., Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece, 2004.

[11] Yasutaka Furukawa and Jean Ponce, "Accurate, Dense, and Robust Multi-View Stereopsis.," in *CVPR*. 2007, IEEE Computer Society.

[12] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions.," *J. ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[13] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[14] Igor Gridchyn and Vladimir Kolmogorov, "Potts model, parametric maxflow and k-submodular functions," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2320–2327.

[15] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.