COLOR TO GRAYSCALE IMAGE CONVERSION USING MODULATION DOMAIN QUADRATIC PROGRAMMING

Chuong T. Nguyen and Joseph P. Havlicek

School of Electrical and Computer Engineering
University of Oklahoma, Norman, OK 73019 USA

ABSTRACT

We propose a new polynomial-time grayscale conversion algorithm applicable to general color images. The output grayscale image is modeled as a linear combination of three color channels where the mixing coefficients are computed by a constrained quadratic programming scheme using modulation domain features of the input color image. The optimization is formulated such that local color space distances between pixels in the input image are approximately preserved in the output grayscale image. Our experimental results demonstrate that the proposed method performs favorably compared to state-of-the-art conversion algorithms, often producing a grayscale image with better visual distinctions between patches that are close in the color space of the original image.

Index Terms— grayscale conversion, quadratic constrained least squares, AM-FM

1. INTRODUCTION

Despite the ubiquity of low-cost storage, computational bandwidth, and network bandwidth, grayscale representations remain important for many fundamental image processing and computer vision tasks such as edge detection, feature extraction, and image and video quality assessment. One of the main reasons for this is that intensity or luminance often captures much of the visually important information present in a color image. Efficient color to grayscale conversion algorithms can reduce costs and enhance printed results in book production and provide important visual cues in aids for the color blind [1]. In addition, for a wide variety of applications grayscale representations can offer important advantages in terms of both spatial and temporal complexity relative to their color counterparts.

The process of converting an image from color to grayscale can be viewed as a mapping or projection from a high dimensional vector space into a lower dimensional space. Hence, the conversion involves inherent information loss: some colors that are distinct in the original image will generally be mapped to a single intensity in the grayscale image. Consequently, important visual information may be lost in the grayscale conversion process. This loss can impact the performance of subsequent processing such as edge detection or object recognition. Kanan and Cottrell [2] compared the performance of 13 grayscale conversion algorithms in an image recognition application and concluded that recognition performance is tangibly affected by the grayscale conversion step.

Grayscale conversion algorithms can be categorized into two major classes. The first class computes the grayscale image using color transformations from the acquired color space, usually RGB, to a new color space where the luminance channel is separated from chrominance channels. For example, the NTSC standard uses the YIQ color space where the Y channel is luminosity defined as a linear combination of the RGB channels. The grayscale image can also be extracted from the L (lightness) channel of the CIE L*a*b* color space. The main problem with these color transformation approaches is that distinct colors having the same luminance are mapped into the same gray level. Therefore, object boundaries and edges can be lost. Recently, there is a growing interest in formulating the grayscale conversion process using optimization-based approaches. The key idea is to compute a grayscale image from the color channels such that the *distance* between color pixels is preserved after the conversion, e.g., if the distance between two color pixels is d, then the distance between these pixels in the grayscale image should be close to d in the L^2 sense. With this second approach, the conversion results depend on the chosen color distance metric.

In this paper, we propose a new constrained optimization grayscaler (COG) using quadratic programming. The grayscale image is defined as a linear combination of the input color channels where the mixing coefficients are computed by a constrained quadratic programming scheme involving modulation domain image features. We calculate the signed color distance between pixels in the L*a*b* color space, where the polarity is deduced based on the Helmholtz-Kohlrausch effect. We tested the proposed algorithm on the popular image dataset described in [3] and compared the results both qualitatively and quantitatively in terms of contrast preservation against a variety of state-of-the-art grayscale conversion algorithms. The quantitative results show that the proposed COG on average preserves contrast better than the best known competing methods. While the average contrast preservation performance of COG is only marginally better than the computationally more expensive algorithm of Lu, Xu, and Jia [4], Fig. 1 shows that the results of these two approaches can be perceptually quite different.

2. BACKGROUND

The simplest conversion method is to compute the grayscale image by averaging the three input color channels. While this method is fast and can produce reasonable results, it fails in cases where two different input colors are mapped into one gray level. Rather than using equal weights, the three mixing coefficients can alternatively be assigned based on human visual perception. For example, the Matlab *rgb2gray* function computes the grayscale image as a weighted combination of the RGB input channels with mixing coefficients $\alpha_R = 0.2990$, $\alpha_G = 0.5870$, and $\alpha_B = 0.1240$. Alternatively, one can obtain a grayscale image by converting the color input image to a nonlinear color space such as CIE L*a*b* and then extracting the lightness channel L as the converted grayscale image. However, this technique can not differentiate between different colors that have the same luminance (i.e., the *iso-luminance* colors problem).

Bala and Eschbach [5] addressed the iso-luminance colors problem by injecting high frequency chrominance details into the luminance channel to aid in differentiating between adjacent colors. Rasche, Geist, and Westall [6] introduced a dimension compression mapping between input colors and output gray levels. Grundland and Dodgson [7] defined a parametric piecewise linear mapping between the color image and the grayscale image. The grayscale image is computed by fusing the information from the luminance and chrominance channels in the YPQ color space where Y is the luminance channel, P represents yellow-blue, and Q represents red-green.

State-of-the-art grayscale conversion results have been achieved recently by applying optimization-based approaches. These algorithms produce a grayscale image by preserving color differences in the grayscale image in some sense. Gooch, et al. [8] defined a nonlinear signed distance between color pixels and preserved this distance in the grayscale image by solving a global large scale least-squares optimization problem. Kim, et al. [9] proposed a nonlinear global mapping grayscale conversion. They computed the grayscale image using a least-squares optimization not unlike the one in [8]. In addition, they incorporated the Nayatani model [10] of the Helmholtz-Kohlrausch (HK) effect into the the signed distance computation. Smith, et al. [11] also used the HK model in their algorithm. Song, et al. [1] modeled the grayscale image as a Markov random field and cast the grayscale conversion problem as a labeling process using a probabilistic graphical model approach. Lu, Xu, and Jia [4] proposed a weak-order color model for grayscale conversion where they formulated the mapping between the color image and the grayscale image as a parameterized multivariate polynomial function. They then computed the polynomial parameters using an iterative least-squares optimization method. Lau, Heidrich, and Mantiuk [12] used a clustering-based method and a least-squares optimization for the grayscale conversion process. This approach is similar to that of [8], except that the distance computation occurs only on a limited number of clusters instead of on the whole image.

3. CONSTRAINED OPTIMIZATION-BASED GRAYSCALER (COG)

We adopt the CIE94 definition of color distance as in [8] and [9], noting that improved accuracy can be obtained with the more complex CIEDE2000 color distance definition [13]. Nevertheless, we use the CIE94 distance here in the interest of simplicity. We compute the distance between two pixels in the CIE L*a*b* color space, hereafter referred to as *LAB*. Let L, a, and b denote the three LAB color channels. Let $\mathbf{m}$ and $\mathbf{n}$ denote 2D spatial pixel coordinates in the image and assume that the input color images are in the RGB color space. In the remainder of this section, we give a procedural description of the proposed COG algorithm.[1]

Step 1: Convert the input RGB image to the perceptually uniform LAB color space [14, pp. 32-38], which consists of a lightness channel L and two chrominance channels a and b.

Step 2: Compute modulation domain features from each channel. We model the three LAB color channels as three components in the modulation domain feature space. The modulation domain space (AM-FM image model) represents a 2D image as a three component vector consisting of an amplitude modulation (AM) component and two frequency modulation (FM) components. The AM-FM representation is attractive because it provides perceptually motivated, intuitive interpretation of local image contrast and local texture characteristics. Specifically, the AM function captures local image contrast and the FM specifies local texture orientation and spacing that are derived from the local phase function. This model has been successfully used in many image and video processing applications such as texture analysis, image synthesis, and target tracking [15–17]. The LAB space AM-FM model is given by

$$\begin{aligned} L(\mathbf{m}) &= A_L(\mathbf{m})\cos[\varphi_L(\mathbf{m})] + L_o, \\ a(\mathbf{m}) &= A_a(\mathbf{m})\cos[\varphi_a(\mathbf{m})] + a_o, \\ b(\mathbf{m}) &= A_b(\mathbf{m})\cos[\varphi_b(\mathbf{m})] + b_o, \end{aligned} \tag{1}$$

where $A_L(\mathbf{m})$ is the amplitude modulation (AM) function of the lightness channel, which models local contrast, and $\varphi_L(\mathbf{m})$ is the phase modulation (PM) function. The local texture orientation and granularity are manifest explicitly in the frequency modulation (FM) function $\nabla\varphi_L(\mathbf{m})$. The modulating functions of the a and b channels are interpreted analogously to those of the L channel. In (1), the constants L_o, a_o, and b_o are DC offsets obtained by de-meaning the LAB color channels prior to AM-FM modeling.

Step 3: Construct dual-domain feature vectors $F(\mathbf{m})$ to characterize the local image structure jointly in the LAB color space and the modulation domain. Let

$$A_t(\mathbf{m}) = \sqrt{A_L^2(\mathbf{m}) + A_a^2(\mathbf{m}) + A_b^2(\mathbf{m})} \tag{2}$$

be the color AM vector modulus. The feature vectors are given by

$$F(\mathbf{m}) = [F_L(\mathbf{m})\ F_a(\mathbf{m})\ F_b(\mathbf{m})]^T, \tag{3}$$

where

$$\begin{aligned} F_L(\mathbf{m}) &= L(\mathbf{m})/100, \\ F_a(\mathbf{m}) &= a(\mathbf{m})/A_t(\mathbf{m}), \\ F_b(\mathbf{m}) &= b(\mathbf{m})/A_t(\mathbf{m}). \end{aligned} \tag{4}$$

We then define the unsigned color distance between two pixels at spatial locations $\mathbf{m}$ and $\mathbf{n}$ according to

$$|d(\mathbf{m},\mathbf{n})| = \sqrt{[F(\mathbf{m}) - F(\mathbf{n})]^T[F(\mathbf{m}) - F(\mathbf{n})]}. \tag{5}$$

Step 4: Compute the signed color distance $d(\mathbf{m},\mathbf{n})$ between pixels in the input image. Let $L_{\text{VAC}}(\mathbf{m})$ be the Variable-Achromatic-Color lightness channel computed by the KL prediction model [10], which seeks to provide additional cues for differentiating iso-luminant colors. Associate to the unsigned distance $|d(\mathbf{m},\mathbf{n})|$ in (5) difference operators ΔL_{VAC}, ΔL, Δa, and Δb defined by, e.g., $\Delta L = L(\mathbf{m}) - L(\mathbf{n})$. As in [9, 11], we take

$$\text{sgn}[d(\mathbf{m},\mathbf{n})] = \begin{cases} \text{sgn}(\Delta L_{\text{VAC}}), & \Delta L_{\text{VAC}} \neq 0, \\ \text{sgn}(\Delta L), & \Delta L_{\text{VAC}} = 0, \Delta L \neq 0, \\ \text{sgn}[(\Delta a)^3 + (\Delta b)^3], & \text{otherwise.} \end{cases} \tag{6}$$

The signed color distance is then given by

$$d(\mathbf{m},\mathbf{n}) = \text{sgn}[d(\mathbf{m},\mathbf{n})] \cdot |d(\mathbf{m},\mathbf{n})|. \tag{7}$$

Step 5: Compute the output grayscale image according to

$$y(\mathbf{m}) = \alpha_r R(\mathbf{m}) + \alpha_g G(\mathbf{m}) + \alpha_b B(\mathbf{m}), \tag{8}$$

where $\boldsymbol{\alpha} = [\alpha_r\ \ \alpha_g\ \ \alpha_b]^T$ is a vector of mixing coefficients and $R(\mathbf{m})$, $G(\mathbf{m})$, and $B(\mathbf{m})$ are the original red, green, and blue intensity values from the input image. Define a grayscale distance Δy

[1] As a practical matter, the page limit prevents us from providing the theoretical and perceptual justifications for each step in this paper.

between the pixels in the output grayscale image at spatial locations $\mathbf{m}$ and $\mathbf{n}$ by

$$\begin{aligned}\Delta y &= y(\mathbf{m}) - y(\mathbf{n}) \\ &= \begin{bmatrix} R(\mathbf{m}) - R(\mathbf{n}) \\ G(\mathbf{m}) - G(\mathbf{n}) \\ B(\mathbf{m}) - B(\mathbf{n}) \end{bmatrix}^T \begin{bmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{bmatrix} \\ &\triangleq \mathbf{D}_{\mathbf{m},\mathbf{n}}\boldsymbol{\alpha}, \end{aligned} \tag{9}$$

where $\mathbf{D}_{\mathbf{m},\mathbf{n}} \in \mathbb{R}^{1\times 3}$. For each $\mathbf{m}$, we allow $\mathbf{n}$ to range over the four-connected neighbors of $\mathbf{m}$ and we construct a matrix $\mathbf{D}$ by stacking the vectors $\mathbf{D}_{\mathbf{m},\mathbf{n}}$ corresponding to the four neighbors of all pixels $\mathbf{m}$ in the original image. For an $M \times N$ input image, the matrix $\mathbf{D}$ has dimensions $4MN \times 3$. The mixing coefficients $\boldsymbol{\alpha}$ are then obtained as the solution to the constrained quadratic programming problem

$$\begin{aligned} &\arg\min_{\boldsymbol{\alpha}} ||\mathbf{D}\boldsymbol{\alpha} - \mathbf{d}||_2^2 \\ &\text{subject to } \mathbf{1}\boldsymbol{\alpha} = 1, \end{aligned} \tag{10}$$

where the vector $\mathbf{d}$ is obtained by stacking the distances (7) corresponding to the four-neighbors $\mathbf{n}$ of all pixels $\mathbf{m}$ in the image and where $\mathbf{1} = [1\ 1\ 1]$ is a row vector with all entries equal to one in order to ensure that the mixing coefficients sum to unity. The individual mixing coefficients are not restricted to be in $[0\ 1]$.

The matrix $\mathbf{D}$ in (10) represents the color features extracted from the input image. In addition to the features computed from the AM-FM image model, we incoporate into the matrix $\mathbf{D}$ several secondary low-level features such as local variance, local mean, edges, local laplacian, and multiresolution.

4. RESULTS AND DISCUSSION

We evaluated the performance of the proposed COG algorithm against the popular set of test images due to Čadík [3]. The set consists of 24 color images and has been used to benchmark state-of-the-art algorithms including those in [4, 7, 9, 11]. Grayscale conversion results for eight of the test images are shown in Fig. 1. For each example, the original color image is shown in the first column and the COG conversion result is shown in the last column. For comparison, grayscale images obtained using the competing algorithms of Gooch, et al. [8], Grundland and Dodgson [7], Smith, et al. [11], Kim, et al. [9], and Lu el al. [4] are shown in columns two through six.

We used the color contrast preserving ratio (CCPR) [4] to quantitatively compare the performance of all six methods. The results are given in Table 1. The range of CCPR is between zero and one, where a higher CCPR score indicates better contrast preserving performance. With respect to the CCPR measure, the algorithm of Lu, et al. [4] and the proposed COG algorithm perform significantly better than the other four methods tested, with the Lu, et al. algorithm delivering the best performance in 11 cases and COG delivering the best performance in eight cases. However, the COG algorithm performed slightly better that that of Lu, et al. on average.

Visually, we argue that the conversion results delivered by COG are appreciably better than those of the methods due to Gooch, et al. [8], Grundland and Dodgson [7], Smith, et al. [11] and Kim, et al. [9] for this set of test images. We argue moreover that COG performs about equally as well as the state-of-the-art method of Lu, et al. [4] while providing a substantially lower computational complexity. It must also be kept in mind that the CCPR figure of merit may not always correspond well with human visual perception. For example, in our opinion the conversion results shown in the third and seventh rows of Fig. 1 for the COG algorithm are superior to those produced by the method of Lu, et al. [4]. However, Lu, et al. achieved a better CCPR score in both cases.

Table 1. Color contrast preservation comparison

No.	CIEY	Grunland07	Smith08	Kim09	Lu12	Ours
1	0.4556	0.4785	0.4932	0.4725	**0.5335**	0.5070
2	0.8726	0.9472	0.8626	0.8879	**0.9672**	0.9660
3	0.8216	**0.8809**	0.8456	0.8303	0.8653	0.8421
4	0.5750	0.5843	0.6258	0.6119	0.6274	**0.6410**
5	0.7262	0.7469	0.7559	0.7474	0.7189	**0.7561**
6	0.3168	0.6075	0.3485	0.3633	**0.7088**	0.4792
7	0.3729	0.5974	0.3725	0.4299	**0.8786**	0.8720
8	0.0621	0.3848	0.1594	0.4541	**0.6685**	0.5365
9	0.4536	0.6215	0.5882	0.5058	0.4003	**0.6436**
10	0.5803	**0.6349**	0.6137	0.5945	0.5701	0.6288
11	0.6036	0.6790	0.6885	0.5137	**0.7264**	0.6523
12	0.0422	0.3329	0.1299	0.3243	**0.8347**	0.7830
13	0.4477	0.5274	0.5070	0.4805	0.3816	**0.5420**
14	0.6621	0.6459	**0.7413**	0.7107	0.6681	0.6541
15	0.5765	0.6623	0.6417	0.6025	**0.7163**	0.7011
16	0.6383	0.6990	0.7021	0.6750	0.6848	**0.7133**
17	0.2313	0.6030	0.2952	0.5130	**0.8696**	0.6175
18	0.6089	0.6007	0.6470	0.6339	0.4946	**0.6495**
19	0.6360	0.6675	**0.7004**	0.6697	0.5206	0.6490
20	0.4319	0.6742	0.5260	0.4685	0.5690	**0.7740**
21	0.9061	0.9248	0.8997	0.9166	0.8520	**0.9259**
22	0.6012	0.6225	0.6706	0.6353	**0.7393**	0.6508
23	0.5850	0.4884	0.6218	0.6096	**0.7070**	0.6806
24	0.7039	0.7805	**0.8121**	0.7686	0.8002	0.7079
Avg	0.5380	0.6413	0.5937	0.6008	0.6876	**0.6906**

Finally, we performed qualitative experiment to illustrate the performance gain that results from incorporating modulation domain features into COG. The original color *Monet* image is shown in Fig. 2(a). The grayscale image obtained by simply extracting the L channel of the LAB color space is shown in Fig. 2(b), while the output of the Matlab rgb2gray routine is given in Fig. 2(c). The COG result obtained by using the LAB representation alone in (4) without modulation domain information is given in Fig. 2(d). The grayscale image in Fig. 2(e) was produced by the full COG implementation, including modulation domain features in (4). Observe in particular that the sun and its reflection in the water are washed out in the grayscale images of Fig. 2(b) and (c), while the grayscale representations delivered by COG in Fig. 2(d) and (e) are substantially more faithful to the original color image. Visually, we argue that the incorporation of modulation domain features increases the the fidelity of the full COG result in Fig. 2(e) relative to Fig 2(d).

5. CONCLUSION

We proposed a fast, high quality constrained optimization grayscaler (COG) for converting color images to grayscale. COG computes the mixing coefficients for blending the red, green, and blue components of the input image using constrained quadratic programming to enforce agreement of the grayscale distances between the pixels of the output image with the color distances between the pixels of the input image, where the color distances are defined using novel modulation domain features that are perceptually motivated and are more robust than the CIE L*a*b* representation for preserving the local texture structure and contrast of the input image. Our experimental results demonstrated that the COG algorithm is capable of delivering high fidelity grayscale images that are competitive with the best existing state-of-the-art grayscale conversion techniques while also providing polynomial-time computational complexity.

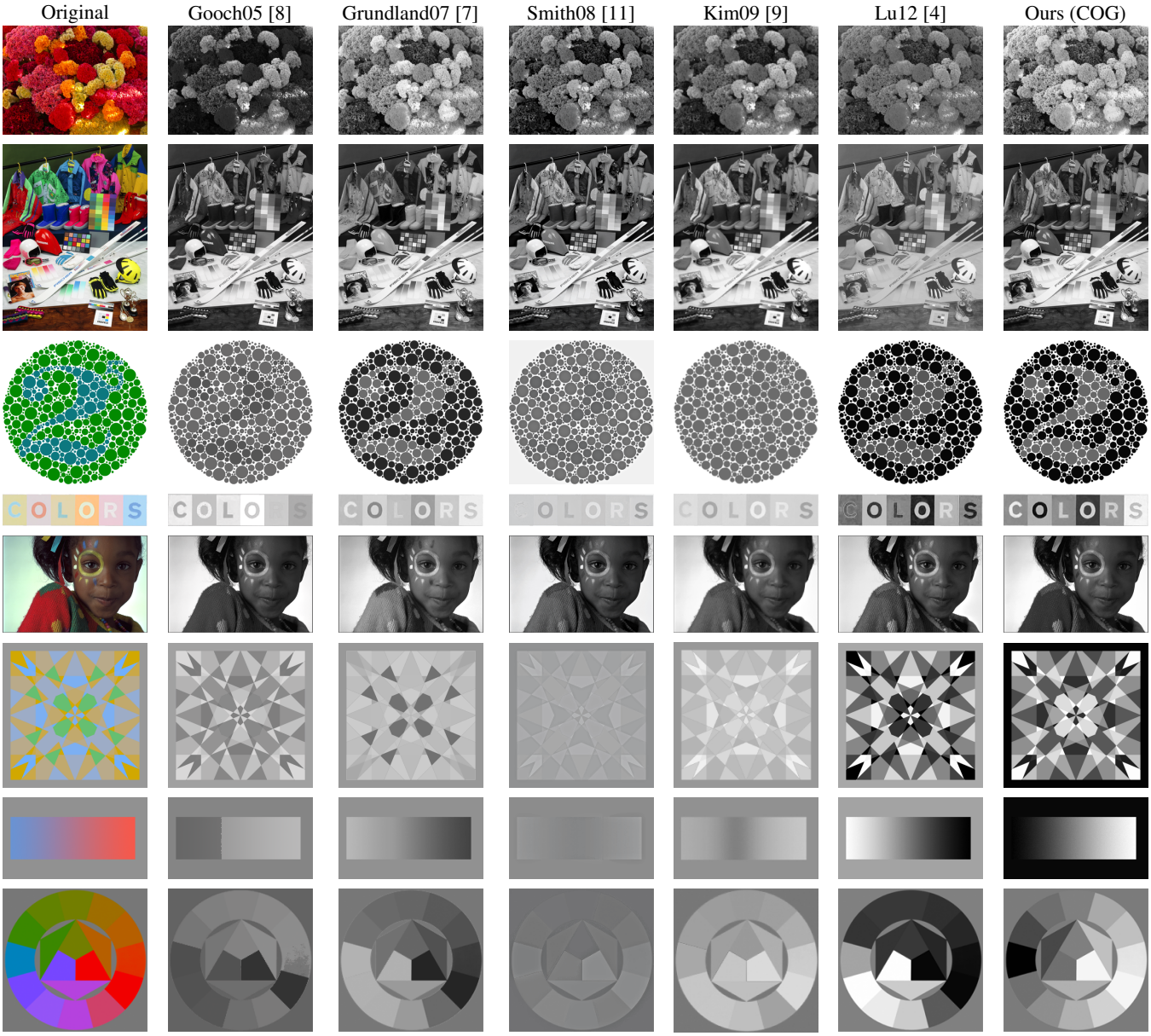

Fig. 1. Grayscale conversion results: (a) Original image. (b) Gooch, et al. [8]. (c) Grundland Dodgson [7]. (d) Smith, et al. [11]. (e) Kim, et al. [9]. (f) Lu, et al. [4]. (g) Ours (COG).

Fig. 2. Grayscale conversion results for the *Monet* image. (a) Original image. (b) CIEY channel. (c) Output of Matlab rgb2gray function. (d) COG result using LAB features only without modulation domain information. (e) Full COG implementation with AM-FM features.

6. REFERENCES

[1] M. Song, D. Tao, C. Chan, X. Li, and C. Chen, "Color to gray: Visual cue preservation," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 32, no. 9, pp. 1537–1552, Sept. 2010.

[2] C. Kanan and G. W. Cottrell, "Color-to-grayscale: Does the method matter in image recognition," *PLOS ONE*, vol. 1, no. 1, pp. 1–7, Jan. 2012.

[3] M. Čadík, "Perceptual evaluation of color-to-grayscale image conversions," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1745–1754, 2008.

[4] C. Lu, L. Xu, and J. Jia, "Contrast preserving decolorization," in *Proc. IEEE Int'l. Conf. Comput. Photography*, Apr. 28-29 2012, pp. 1–7.

[5] R. Bala and R. Eschbach, "Spatial color-to-grayscale transformation preserving chrominance edge information," in *Proc. IST/SID 12th Color Imaging Conf.*, 2004, pp. 82–86.

[6] K. Rasche, R. Geist, and J. Westall, "Re-coloring images for gamuts of lower dimension," *Computer Graphics Forum*, vol. 24, no. 3, pp. 423–432, 2005.

[7] M. Grundland and N.A. Dodgson, "Decolorize: Fast, contrast enhancing, color to grayscale conversion," *Pattern Recog.*, vol. 40, no. 11, pp. 2891–2896, Nov. 2007.

[8] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch, "Color2gray: Salience-preserving color removal," *ACM Trans. Graphics (Proc. SIGGRAPH 2005)*, vol. 24, no. 3, pp. 634–639, Jul. 2005.

[9] Y. Kim, C. Jang, J. Demouth, and S. Lee, "Robust color-to-gray via nonlinear global mapping," *ACM Trans. Graphics (Proc. SIGGRAPH 2009)*, vol. 24, no. 3, pp. 634–639, Jul. 2005.

[10] Y. Nayatani, "Simple estimation methods for the Helmholtz-Kohlrausch effect," *Color Research App.*, vol. 22, no. 6, pp. 385–401, Dec. 1997.

[11] K. Smith, P.E. Landes, J. Thollot, and K. Myszkowski, "Apparent greyscale: A simple and fast conversion to perceptually accurate images and video.," *Computer Graphics Forum*, vol. 27, no. 2, pp. 193–200, 2008.

[12] C. Lau, W. Heidrich, and R. Mantiuk, "Cluster-based color space optimizations," in *Proc. IEEE Int'l. Conf. Computer Vision*, Barcelona, Spain, Nov. 6-13 2011, pp. 1172–1179.

[13] M.R. Luo, G. Cui, and B. Rigg, "The development of the CIE 2000 colour-difference formula: CIEDE2000," *Color Res. Appl.*, vol. 26, no. 5, pp. 340–350, Oct. 2001.

[14] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer-Verlag, Princeton, NJ, 2000.

[15] P. Maragos and A. C. Bovik, "Image demodulation using multidimensional energy separation," *J. Opt. Soc. Amer. A*, vol. 12, no. 9, pp. 1867–1876, Sep. 1995.

[16] J.P. Havlicek, D.S. Harding, and A.C. Bovik, "Multidimensional quasi-eigenfunction approximations and multicomponent am-fm models," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 227–242, Feb. 2000.

[17] C. T. Nguyen, P.A. Campbell, and J. P. Havlicek, "FM filters for modulation domain image processing," in *Proc. IEEE Int'l. Conf. Image Proc.*, Cairo, Egypt, Nov. 7-11, 2009, pp. 3973–3976.