PRESERVING PERCEPTUAL CONTRAST IN DECOLORIZATION WITH OPTIMIZED COLOR ORDERS

Bin Jin and Sabine Süsstrunk

School of Computer and Communication Sciences, EPFL, Switzerland

ABSTRACT

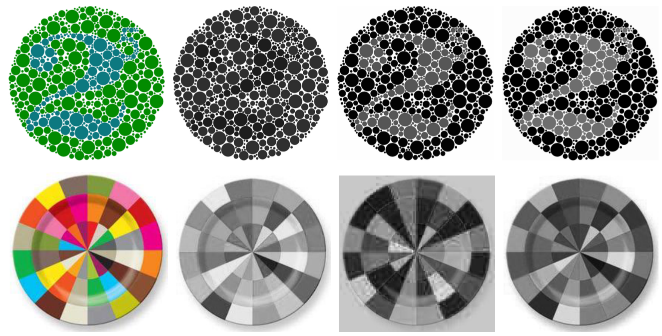
Converting a color image to a grayscale image, namely decolorization, is an important process for many real-world applications. Previous methods build contrast loss functions to minimize the contrast differences between the color images and the resultant grayscale images. In this paper, we improve upon a widely used decolorization method with two extensions. First, we relax the need for heuristics on color orders, which the baseline method relies on when computing the contrast differences. In our method, the color orders are incorporated into the loss function and are determined through optimization. Moreover, we apply a nonlinear function on the grayscale contrast to better model human perception of contrast. Both qualitative and quantitative results on the standard benchmark demonstrate the effectiveness of our two extensions.

***Index Terms*—** Decolorization, Color Orders, Non-linear Perception, Contrast Preservation

1. INTRODUCTION

Although color images are more widely used, grayscale image visualizations are still dominate in specific applications, such as black-and-white printing and digital-ink display. Decolorization, which refers to the process of converting three-channel color images into one-channel grayscale images, is thus fundamentally important for these usages. Since decolorization reduces the dimensions of the input signal, it inevitably results in information loss. The goal of decolorization is thus to maintain as much of the visually distinguishable information, namely contrast, as possible.

Many methods have been proposed to address this problem. Simple approaches, like directly extracting the L channel of the CIELAB colorspace, often do not capture the salient structures and suffer from significant detail loss, as shown in Fig. 1b. Later methods preserve the contrast of the color images by minimizing a contrast loss function. During optimization, heuristics are often adopted to determine the color orders, which change the signs of the contrast measure. A widely used heuristic is a bimodel distribution introduced in [1]. However, such a distribution may result in loss in details,



(a) Color images (b) CIEL (c) [1] (d) Ours

Fig. 1: Example decolorizations by different methods.

unfaithful grayscale values and ringing artifacts, as shown in Fig. 1c.

We extend upon the widely used decolorization method [1][1] in two aspects. First, instead of applying a heuristic bimodal distribution on the color orders, we propose to incorporate the color orders directly into the contrast loss function to obtain the optimal color orders as well as the grayscale values through optimization. Second, we propose to apply a nonlinear function on the grayscale contrast to model the nonlinear characteristics of human perception. The two extensions result in a notable improvement over [1] in the decolorization performance. As shown in Fig. 1d, the contrast and the visual details are better preserved in our method than [1]. Additionally, experiments on the standard decolorization benchemark demonstrate that our method, using both extensions, outperforms different state-of-the-art approaches.

2. RELATED WORK

Various decolorization methods [1–19] have been proposed, which can be categorized as local or global approaches.

Local methods assign different grayscale values to the same color to enhance the local contrast. [2, 3, 5, 16] use different approaches to obtain local contrast, which is then added to the luminance channel. Neumann et al. [4] regard the color and luminance contrast as a gradient field and obtain the grayscale image via fast direct integration. Saliency

[1] [1] is the default decolorization method in OpenCV 3.0.

measure is adopted in [6] where the authors focus on preserving the contrast in the salient regions. [19] use a Laplacian pyramid decomposition to measure the local information. These local decolorization methods, although preserving the local contrast, may occasionally distort the constant color regions [7].

Global methods, on the other hand, define linear or nonlinear mapping functions that map the same color to the same grayscale value. Bala et al [8] sort the colors and assign grayscale values accordingly. [9, 10] enforce the local and global contrast between neighbouring pixels with empirically determined color orders. [17] propose a constrained contrast mapping paradigm in the gradient domain to map the gradient from the color image to the grayscale image, and then reintegrate to generate the grayscale image. [15] derive the grayscale image by maximizing the gradient correlation between each channel of the color image and the grayscale image. Recent work [18] treat the decolorization function as the sum of three subspaces. [1, 13, 14] propose a contrast preserving function and adopt a bimodal distribution to approximate the color orders.

Our method falls within the global category. We propose two extensions to the global method in [1], which lead to state-of-the-art decolorization performance.

3. METHOD

In this section, we describe our decolorization algorithm, with the two proposed extensions detailed in section 3.2.1 and 3.2.2.

3.1. Mapping Function

As in [1], we adopt a polynominal function f to map each RGB color vector $\boldsymbol{v} = \{v_r, v_g, v_b\}$ to a grayscale value g:

$$g = f(v_r, v_g, v_b; \boldsymbol{\omega}) = \sum_i \omega_i m_i \tag{1}$$

where $\boldsymbol{\Omega} = \{\omega_i\}$ are the parameters of the mapping function, which we optimize on for each image. m_i is the ith element in $\{v_r, v_g, v_b, v_r v_g, v_g v_b, v_b v_r, v_r^2, v_g^2, v_b^2\}$. Although simple, such a polynomial function is powerful enough to accurately fit to highly nonlinear functions [1].

3.2. Contrast Loss Function

The goal of decolorization is to minimize the contrast loss after decolorization, which is defined as $\boldsymbol{E}$:

$$\boldsymbol{E} = \sum_{(x,y) \in \boldsymbol{P}} (\delta^g_{(x,y)} - \delta^c_{(x,y)})^2 \tag{2}$$

where x and y indicate two different pixels, $\boldsymbol{P}$ represents the set of all possible pixel pairs, $\delta^g_{(x,y)}$ is the contrast between the pixel pair (x, y) in the grayscale image, and $\delta^c_{(x,y)}$ is the corresponding contrast in the color image.

3.2.1. *Color orders*

In the baseline method [1], $\delta^g_{(x,y)}$ and $\delta^c_{(x,y)}$ are defined as:

$$\delta^g_{(x,y)} = g_x - g_y \tag{3}$$

$$\delta^c_{(x,y)} = s^c_{(x,y)} \cdot \sqrt{(L_x - L_y)^2 + (a_x - a_y)^2 + (b_x - b_y)^2} \tag{4}$$

where L, a, b are the corresponding values in CIELAB colorspace. $s^c_{(x,y)}$ is the sign function that determines the orders for a pair of colors, which consequently determines the order of the grayscale values after the optimization of $\boldsymbol{E}$. In [1], such a sign function is defined as a heuristic bimodal distribution, which may result in visual artifacts in the converted grayscale images as shown in Fig. 3.

In this paper, we remove the heuristics on color orders by moving the sign function from the color contrast to the grayscale contrast, and directly incorporate it into the optimization of $\boldsymbol{E}$. The new sign function on grayscale contrast is defined as:

$$s^g_{(x,y)} = \begin{cases} 1 & \text{if } r_x \geq r_y, g_x \geq g_y, b_x \geq b_y \\ -1 & \text{if } r_x < r_y, g_x < g_y, b_x < b_y \\ |\cdot| & \text{otherwise} \end{cases} \tag{5}$$

Where $|\cdot|$ represents the operation for computing absolute values. The grayscale contrast is hence computed as:

$$\delta^g_{(x,y)} = s^g_{(x,y)}(g_x - g_y) \tag{6}$$

After removing the sign function, the color contrast is directly computed as the classic color difference formula:

$$\delta^c_{(x,y)} = \sqrt{(L_x - L_y)^2 + (a_x - a_y)^2 + (b_x - b_y)^2} \tag{7}$$

With such a new sign function, for a pair of colors that have a clear order ($r_x \geq r_y, g_x \geq g_y, b_x \geq b_y$ or $r_x < r_y, g_x < g_y, b_x < b_y$), the corresponding grayscale values g_x, g_y will follow the same order after optimizing Eq. 2. Otherwise, the color orders are determined to be the global optimal ones that minimize the contrast loss $\boldsymbol{E}$.

3.2.2. *Non-linear perceptual contrast*

According to the Weber-Fechner law [20], visual perception of physical stimuli follows a logarithmic correspondence. For decolorization, the aim is to preserve the contrast perceived by the human visual system (HVS) the same in the grayscale image and the color image. Hence, to better align with human contrast perception, we propose a second extension to [1], where we consider nonlinearity of the percevied contrast. Since the encoding in the CIELAB colorspace is designed to already incorporate the HVS nonlinearity, a nonlinear function is applied only to the grayscale values. We obtain the perceptual grayscale values p_x as follows:

$$p_x = \frac{log(\beta g_x + 1)}{log(1 + \beta)} \tag{8}$$

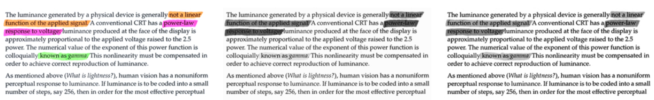
β is a parameter to control the nonlinearity. The logarithmic function ensures the range of p_x to be the same as that of g_x ($[0, 1]$). The perceptual grayscale contrast is thus computed as the difference between the perceptual grayscale values:

$$\delta^g_{(x,y)} = s^g_{(x,y)}(p_x - p_y) \tag{9}$$

According to Taylor's theorem, $p_x - p_y$ in Eq. 9 can be further simplified into multinomial, which makes the optimization process more stable:

$$p_x - p_y \approx \frac{1}{log(1+\beta)} \cdot \frac{\beta}{\beta g_y + 1} \cdot (g_x - g_y) \tag{10}$$

Using the nonlinear perceptual contrast instead of the linear contrast leads to better preservation of visual details. As illustrated in Fig. 2, the green mark in the color image is better preserved in the result using non-linear perceptual contrast than that with linear contrast.



(a) Color image (b) linear (c) nonlinear

Fig. 2: Comparison between linear and nonlinear contrast measure.

Combining the two extensions together, the global contrast loss $\boldsymbol{E}$ can be rewritten as:

$$\boldsymbol{E} = \sum_{(x,y)\in \boldsymbol{P}} \left(s^g_{(x,y)}(p_x - p_y) - \delta^c_{(x,y)}\right)^2 \tag{11}$$

where $s^g_{(x,y)}$ is defined in Eq. 5, $p_x - p_y$ and $\delta^c_{(x,y)}$ are computed according to Eq. 10 and 7, respectively.

3.3. Clustering Colors

To minimize Eq. 11, all pixel pairs need to be considered, which is time-consuming and redundant since many pixels carry the same color. Following the technique in [13], we speed up the computation by grouping similar colors together and consider different color group pairs. Specifically, we first perform K-means clustering in the CIELAB colorspace on all pixels. The number of classes is automatically determined by setting a threshold for minimum color distance between clusters (we empirically set it as 30 in all experiments as a good compromise between speed and accuracy). Such settings typically result in less than 50 classes per image. The contrast loss $\boldsymbol{E}$ is then directly computed on the cluster pairs instead of the pixel pairs, with each cluster represented by its mean color.

$$\begin{aligned}\boldsymbol{E} = &\sum_{(c_i,c_j)\in \boldsymbol{C}} \kappa_{(c_i,c_j)} \cdot \\ &\Bigg(s^g_{(c_i,c_j)}\Big(\frac{1}{log(1+\beta)} \cdot \frac{\beta}{\beta g_{c_j} + 1} \cdot (g_{c_i} - g_{c_j})\Big) - \\ &\sqrt{(L_{c_i} - L_{c_j})^2 + (a_{c_i} - a_{c_j})^2 + (b_{c_i} - b_{c_j})^2}\Bigg)^2\end{aligned} \tag{12}$$

where $\boldsymbol{C}$ represents the set of all cluster pairs. The weight $\kappa_{(c_i,c_j)}$ of a cluster pair is determined as:

$$\kappa_{(c_i,c_j)} = \frac{N_{c_i} N_{c_j}}{N_0^2} \tag{13}$$

where N_{c_i} and N_{c_j} are the number of pixels in the clusters c_i and c_j, respectively. N_0 is set to be $0.01Z$ with Z denoting the total number of pixels in the image.

3.4. Optimization

Due to the absolute operation used in Eq. 5, the contrast loss $\boldsymbol{E}$ is non-convex, and thus a numerical solution is difficult to derive. We adopt the Adam gradient descent method [21] for optimization, which was originally developed for optimizing highly non-convex functions in deep learning. Similar to [1], we first initialize $\boldsymbol{\Omega}$ as $\{0.33, 0.33, 0.33, 0, 0, 0, 0, 0, 0\}$. We perform 1000 iterations of gradient updates on the parameters $\boldsymbol{\Omega}$ when minimizing Eq. 12 (on average 3 seconds per image). The resultant grayscale image is then generated according to Eq. 1 with the optimized parameters $\boldsymbol{\Omega}$.

4. EXPERIMENTS

We validate our method on the standard decolorization benchmark [22], which has 24 colorful images covering both realistic and synthetic images. The parameter β is empirically set to 0.5 for all images. The learning rate of the Adam optimizer is set to 0.0005 for 1000 iterations.

4.1. Quantitative comparison

For quantitative comparison, we adopted the CCPR metric proposed in [13].

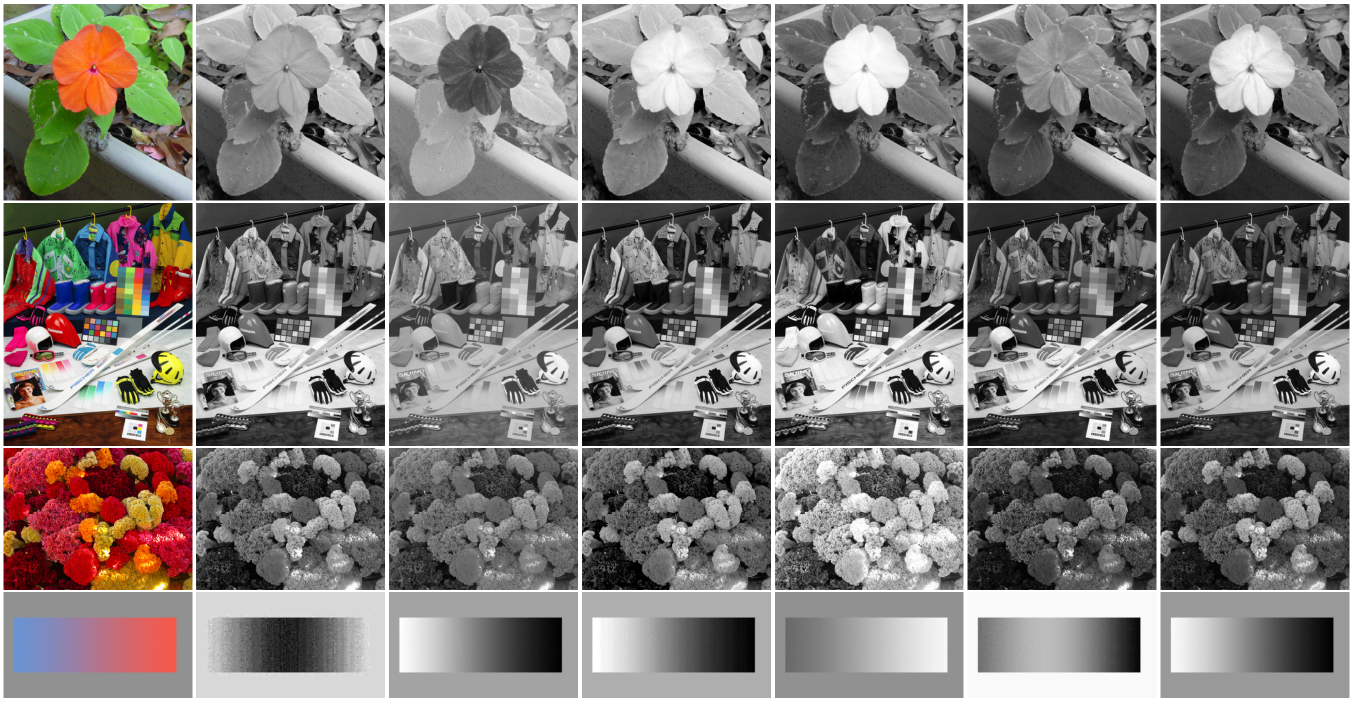
$$CCPR = \frac{\#\{(x,y)|(x,y) \in \boldsymbol{Q}, |g_x - g_y| \le \tau\}}{\|\boldsymbol{Q}\|} \tag{14}$$

where $\boldsymbol{Q}$ is the set of all pixel pairs whose color difference is larger than τ. $\|\boldsymbol{Q}\|$ is the number of elements in $\boldsymbol{Q}$. By computing the mean CCPR values with varying thresholds τ from 1 to 15, we show the CCPR curves[2] in Fig. 4. $Ours\_E1$ represents our method that only adopts the color orders extension while still using the linear contrast measure. $Ours\_E1E2$ represents our method that uses both extensions. We compare with [1, 6, 15, 18] since they report better decolorization performance than the earlier ones [2–5, 7–12, 14, 16, 17][3].

Compared with the baseline method [1], the $Ours\_E1$ method clearly improves the performance by a remarkable margin, demonstrating the effectiveness of the color orders extension. $Ours\_E1E2$ further improves over $Ours\_E1$,

[2]Higher CCPR curve represents better decolorization performance.

[3]for [13, 19], neither the results on this benchmark nor the code are released.



(a) **Color images** (b) **L in CIELAB** (c) **[1]** (d) **[6]** (e) **[15]** (f) **[18]** (g) **Ours_E1E2**

Fig. 3: Qualitative comparison with different state-of-the-art decolorization methods.

proving that using the nonlinear perceptual contrast model leads to better decolorization performance.

The $Ours\_E1$ method already achieves comparable performance against the state-of-the-art decolorization methods. By further incorporating the nonlinear perceptual contrast, the $Ours\_E1E2$ method clearly reaches the best CCPR values, even better than the methods [6, 15] that consider additional information like saliency or gradient.

4.2. Qualitative comparison

We show qualitative comparison in Fig. 3. Our method generates favorable or comparable results against other approaches. For example, in the first row, our decolorization result preserves the high contrast between the flower and the leaves while sill preserving the line structures on the flower and the leaves. Methods like [6, 15] generate high-contrast results while missing the details on the leaves and the flower. Results produced by [1, 18] present the details but suppress the high contrast of the flowers. For the image in the forth row, the gradually changing colors are well preserved in our result without any line artifacts as in [6] or contrast loss as in [15,18]. More examples can be found in the supplementary materials.

5. CONCLUSION

This paper presents a decolorization approach that extends upon a widely used method [1]. We first remove the heuristics for color orders presented in [1]. Instead, we directly incorporate the color orders into the contrast loss function and optimize it through a gradient descent technique. Moreover, we apply a nonlinear function on the grayscale contrast to align with the nonlinear characteristics of human perception. Using these two extensions, our method outperforms the state-of-the-art decolorization approaches on the standard decolorization benchmark.

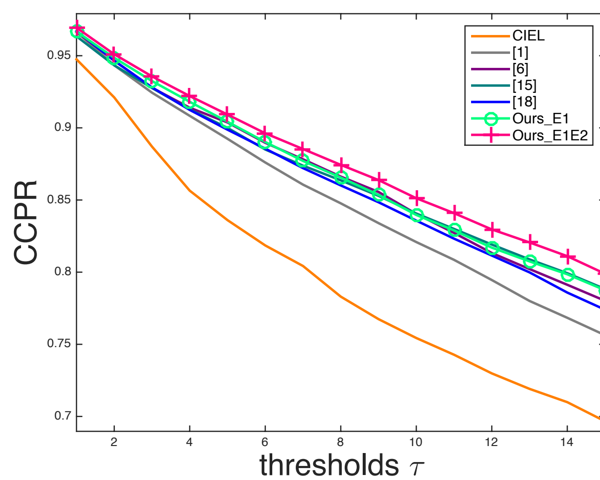


Fig. 4: Quantitative comparison with different state-of-the-art decolorization algorithms.

6. REFERENCES

[1] Cewu Lu, Li Xu, and Jiaya Jia, “Contrast preserving decolorization,” *IEEE International Conference on Computational Photography*, pp. 1–7, 2012.

[2] Raja Bala and Reiner Eschbach, “Spatial color-to-grayscale transform preserving chrominance edge information,” *Color and Imaging Conference*, vol. 2004, no. 1, pp. 82–86, 2004.

[3] Kaleigh Smith, Pierre-Edouard Landes, Joëlle Thollot, and Karol 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.

[4] Laszlo Neumann, M Čadík, and Antal Nemcsics, “An efficient perception-based adaptive color to gray transformation,” *The Third Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 73–80, 2007.

[5] Yibing Song, Linchao Bao, and Qingxiong Yang, “Real-time video decolorization using bilateral filtering,” *IEEE Winter Conference on Applications of Computer Vision*, pp. 159–166, 2014.

[6] Hao Du, Shengfeng He, Bin Sheng, Lizhuang Ma, and Rynson WH Lau, “Saliency-guided color-to-gray conversion using region-based optimization,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 434–443, 2015.

[7] Yongjin Kim, Cheolhun Jang, Julien Demouth, and Seungyong Lee, “Robust color-to-gray via nonlinear global mapping,” *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 161, 2009.

[8] Raja Bala and Karen M Braun, “Color-to-grayscale conversion to maintain discriminability,” *Electronic Imaging*, pp. 196–202, 2003.

[9] Amy A Gooch, Sven C Olsen, Jack Tumblin, and Bruce Gooch, “Color2gray: salience-preserving color removal,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 634–639, 2005.

[10] Jung Gap Kuk, Jae Hyun Ahn, and Nam Ik Cho, “A color to grayscale conversion considering local and global contrast,” *Asian Conference on Computer Vision*, pp. 513–524, 2010.

[11] Karl Rasche, Robert Geist, and James Westall, “Re-coloring images for gamuts of lower dimension,” *Computer Graphics Forum*, vol. 24, no. 3, pp. 423–432, 2005.

[12] Mark Grundland and Neil A Dodgson, “Decolorize: Fast, contrast enhancing, color to grayscale conversion,” *Pattern Recognition*, vol. 40, no. 11, pp. 2891–2896, 2007.

[13] Cewu Lu, Li Xu, and Jiaya Jia, “Contrast preserving decolorization with perception-based quality metrics,” *International Journal of Computer Vision*, vol. 110, no. 2, pp. 222–239, 2014.

[14] Cewu Lu, Li Xu, and Jiaya Jia, “Real-time contrast preserving decolorization,” *SIGGRAPH Asia Technical Briefs*, p. 34, 2012.

[15] Qiegen Liu, Peter X Liu, Weisi Xie, Yuhao Wang, and Dong Liang, “Gcsdecolor: gradient correlation similarity for efficient contrast preserving decolorization,” *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2889–2904, 2015.

[16] Cheryl Lau, Wolfgang Heidrich, and Rafal Mantiuk, “Cluster-based color space optimizations,” *IEEE International Conference on Computer Vision*, pp. 1172–1179, 2011.

[17] David Connah, Mark Samuel Drew, and Graham David Finlayson, “Spectral edge image fusion: Theory and applications,” *European Conference on Computer Vision*, pp. 65–80, 2014.

[18] Qiegen Liu, Peter Liu, Yuhao Wang, and Henry Leung, “Semi-parametric decolorization with laplacian-based perceptual quality metric,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[19] Cosmin Ancuti and Codruta O Ancuti, “Laplacian-guided image decolorization,” *IEEE International Conference on Image Processing*, pp. 4107–4111, 2016.

[20] Arthur S Reber, *The Penguin Dictionary of Psychology*, Penguin Press, 1995.

[21] Diederik Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[22] M Ĉadík, “Perceptual evaluation of color-to-grayscale image conversions,” *Computer Graphics Forum*, vol. 27, no. 7, pp. 1745–1754, 2008.