Optik 126 (2015) 368–372

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Color images enhancement for edge information protection based on second order Taylor series expansion approximation

Zhigang Zhou, Lihong Chen*, Xinrong Hu

Department of Mathematics and Computer College, Wuhan Textile University, Wuhan 430073, PR China

ARTICLE INFO

Article history:
Received 12 November 2013
Accepted 20 August 2014

Keywords:
Adaptive enhancement
Gaussian bilateral filter
Taylor series expansion
Edge

ABSTRACT

This article addresses the problem of color image enhancement for edge information protection. A new nonlinear adaptive enhancement (NNAE) algorithm is presented to resolve the problem in parallel procedure for low or high intensity and poor contrast (LIPC or HIPC) images. The NNAE algorithm consists of three steps. First, a RGB color image is converted to an intensity image, then an adaptive intensity adjustment with local contrast enhancement is performed parallelly, by using a single scale shift-variant Gaussian bilateral filter and the second order Taylor series expansion approximation technology, and finally colors are restored. A significant advantage of NNAE is that the edge information of enhanced images can be enhanced or preserved for LIPC or HIPC images. NNAE is only compared with Parallel Nonlinear Adaptive Enhancement (PNAE) algorithms. The experimental results show that the visual effects of NNAE are same or better than PNAE, and the enhanced images have more edges information changes by NNAE.

1. Introduction

Color images provide more and richer information for visual perception than gray images. How to improve visual quality for low intensity with poor contrast (LIPC) color images and high intensity with poor contrast (HIPC) color images gained increasing attention in image and video processing [1]. How to preferably enhance LIPC and HIPC color images and simultaneously enhance or preserve edges information is our focus in the paper.

For LIPC and HIPC image enhancement, two innovative techniques named OBLCAE (Overall Brightness and Local Contrast Adaptive Enhancement) [2] and PNAE (Parallel Nonlinear Adaptive Enhancement) [3] algorithms were proposed in 2013 and 2014 respectively. Local neighborhood grayscale variance value and local neighborhood average gray value are used in OBLCAE to adaptively enhance local contrast with the purpose of reducing the detail information loss caused by global brightness enhancement. OBLCAE algorithm was only used to enhance LIPC images and cannot enhance HIPC images because of its monotone increasing nonlinear intensity transfer function. By using a single scale Gaussian filter and first order Taylor series expansion approximation technology, an adaptive intensity adjustment with local contrast enhancement is performed parallelly in PANE for LIPC and HIPC image enhancement. Although PNAE algorithm had achieved good effects for some LIPC and HIPC image enhancement, the visual effects of enhanced image still need to be further improved. A new nonlinear adaptive enhancement (NNAE) algorithm, as an improved algorithm for PNAE, is proposed in the article. The main differences between NNAE and PNAE are summarized in the following:

(1) In NNAE, considering the spatial proximity and intensity similarity of the pixels in the local neighborhood, a single scale shift-variant Gaussian bilateral filter is used to produce the mean intensity image. But in PNAE, a single scale Gaussian smoothing operator is used to produce the mean intensity image. The NNAE provides more complete information about the overall intensity and local contrast than PNAE even though the computational efficiency of bilateral filter is lower compared to the multi-scale Gaussian filter.
(2) In NNAE, the intensity mapping function is approximated by the second order Taylor series expansion instead of the first order Taylor series expansion used in PANE. The reason to do so is that large error may occur when a nonlinear function is approximated by a linear function. At the same time we need the second order term to get more detailed edges information since we pay more attention to the visual enhancement results, even though it may slightly affect the computational efficiency.

* Corresponding author.
E-mail addresses: zzghust@163.com (Z. Zhou), lhchenz@163.com (L. Chen).

http://dx.doi.org/10.1016/j.ijleo.2014.08.126

In the following section, the NNAE algorithm is discussed in details. Experimental results of NNAE are discussed in Section 3, followed by the conclusions and discussions of future work in Section 4.

2. The NNAE algorithm

In order to better explain NNAE algorithm, we first review PNAE algorithm. The PNAE algorithm consists of three steps.

First, a RGB color image is converted to an intensity image. The intensity transfer formula is given by:

$$I(x, y) = \frac{76.245 I_r(x, y) + 149.685 I_g(x, y) + 29.07 I_b(x, y)}{255}, \tag{1}$$

where $I_r(x, y)$, $I_g(x, y)$ and $I_b(x, y)$ are the red, green and blue components of a pixel located at (x, y) in the RGB color image. The intensity image is further normalized to:

$$I_{\text{in}}(x, y) = \frac{I(x, y)}{255}. \tag{2}$$

Then an adaptive intensity adjustment with contrast enhancement based on the local neighborhood is performed parallelly on the intensity image. The adaptive intensity mapping function is given by:

$$T(I_{\text{in}}(x, y)) = I_{\text{in}}(x, y)^p, \quad I_{\text{in}}(x, y) \in [0, 1]. \tag{3}$$

The power p is given as:

$$p = c_1 \frac{I_{\text{ave}}(x, y) + \varepsilon}{[1 - I_{\text{ave}}(x, y)] + \varepsilon} + c_2, \tag{4}$$

where $I_{\text{ave}}(x, y) \in [0, 1]$ is the normalized local mean intensity value of the pixel at location (x, y), c_1 and c_2 are constants determined empirically, and $\varepsilon = 0.01$ is a numerical stability factor introduced to avoid division by zero when $I_{\text{ave}}(x, y) = 1$. The formula of intensity adjustment with local contrast enhancement used in PNAE is given by:

$$\begin{aligned} &C_{\text{out}}^{\text{enh}}[I_{\text{in}}(x, y)] \\ &= \bar{I}(x, y) T[I_{\text{in}}(x, y)] + \bar{I}(x, y) T'[I_{\text{in}}(x, y)][I_{\text{in}}(x, y) - I_{\text{ave}}(x, y)] \\ &= C_{\text{out1}} + C_{\text{out2}} \end{aligned} \tag{5}$$

where $C_{\text{out}}^{\text{enh}}[I_{\text{in}}(x, y)]$ denotes the result of intensity adjustment with local contrast enhancement for $I_{\text{in}}(x, y)$, $\bar{I}(x, y)$ is given by:

$$\bar{I}(x, y) = \frac{I_{\text{in}}(x, y)}{I_{\text{ave}}(x, y)}, \tag{6}$$

and $T'[I_{\text{in}}(x, y)]$ denotes the first order derivative of the mapping function (3). The local average of the image $I_{\text{ave}}(x, y)$ in formula (6) is computed by:

$$I_{\text{ave}}(x, y) = \sum_{(m,n) \in \Omega} w_{mn} T[I_{\text{in}}(m, n)], \tag{7}$$

where (x, y) is the center pixel of the $M \times M$ neighborhood Ω, $I_{\text{in}}(m, n)$ is the intensity value of the pixel in the location (m, n) of the original intensity image, and ω_{mn} is a single scale Gaussian smoothing operator with one neighborhood, as the weight of the pixel in the location (m, n), and given by:

$$\omega_{mn} = K^{-1} \exp\left[-\frac{(m - x)^2 + (n - y)^2}{2\sigma^2}\right], \quad (m, n) \in \Omega, \tag{8}$$

where σ is the standard derivation of ω_{mn}, and K is the normalization factor.

In the last, colors are restored to produce the enhancement result by using formula (9):

$$P_{\text{out}}^{\text{RGB}}(x, y) = \beta(x, y) P_{\text{in}}^{\text{RGB}}(x, y), \tag{9}$$

where $P_{\text{in}}^{\text{RGB}} = [R_{\text{in}}, G_{\text{in}}, B_{\text{in}}]$ and $P_{\text{out}}^{\text{RGB}} = [R_{\text{out}}, G_{\text{out}}, B_{\text{out}}]$ denote the input and output color values of each pixel in RGB color space, respectively, $\beta(x, y)$ is given by:

$$\beta(x, y) = \frac{C_{\text{out}}^{\text{enh}}[I_{\text{in}}(x, y)] + \varepsilon}{I_{\text{in}}(x, y) + \varepsilon}, \tag{10}$$

and $\varepsilon = 0.01$ is a numerical stability factor introduced to avoid division by zero when $I_{\text{in}}(x, y) = 0$.

The main difference between NNAE and PNAE is the second step. Now we show how to calculate $I_{\text{ave}}(x, y)$ with a different method. The bilateral filter proposed by Tomasi and Manduchi [4] is a nonlinear filter that smoothes the noise while preserving edge structures. The idea of the bilateral filter has been adopted for many applications not only in the area of image de-noising, but also in computer graphics, video processing, image interpolation, dynamic range compression, illumination estimation, etc. [5]. The value of $I_{\text{ave}}(x, y)$ in NNAE is computed still using the formula (7), but ω_{mn} is a shift-variant Gaussian bilateral filter given by:

$$\omega_{mn} = K^{-1} \exp\left[-\frac{(m - x)^2 + (n - y)^2}{2\sigma_d^2} - \frac{(I_{\text{in}}(m, n) - I_{\text{in}}(x, y))^2}{2\sigma_r^2}\right], \quad (m, n) \in \Omega, \tag{11}$$

$$\sum_{(m,n) \in \Omega} w_{mn} = 1, \tag{12}$$

where σ_d and σ_r are the standard derivations of ω_{mn}, and K is the normalization factor given by:

$$K = \sum_{(m,n) \in \Omega} \exp\left[-\frac{(m - x)^2 + (n - y)^2}{2\sigma_d^2} - \frac{(I_{\text{in}}(m, n) - I_{\text{in}}(x, y))^2}{2\sigma_r^2}\right]. \tag{13}$$

The value of ω_{mn} is obtained by using two Gaussian smoothing functions. It considers the similarity of locations and intensity values between the center pixel and the pixels in the neighborhood. Function (11) gives bigger weights to those pixels that are similar to the center pixel in location and intensity value, and gives small weights to those pixels that have big difference on intensity values even if their positions is very close to the center pixel. Because there are huge differences on intensity values between the center pixel and the pixels on the edges near the center pixel, function (11) gives small weights to those pixels on the edges, and therefore it can effectively overcome the influence of the pixels on the edges to those pixels not on the edges in the enhanced image. The effects of σ_d and σ_r will be discussed in details in Section 3 too.

The mathematical condition to preserve the local contrast is described in [6] by:

$$\frac{g_{\text{out}}(x, y)}{g_{\text{ave}}(x, y)} = \frac{I_{\text{in}}(x, y)}{I_{\text{ave}}(x, y)}, \tag{14}$$

where $g_{\text{out}}(x, y)$ and $g_{\text{ave}}(x, y)$ denote the output intensity and the output local intensity average of the pixel (x, y) respectively. The terms of $I_{\text{in}}(x, y)$ and $I_{\text{ave}}(x, y)$ are defined as before. By using the second order Taylor series expansion approximation technology, the adaptive intensity mapping function (3) can be approximated by the second order Taylor series expansion as follows:

$$\begin{aligned} T[I_{\text{in}}(m, n)] \cong{}& T[I_{\text{in}}(x, y)] + T'[I_{\text{in}}(x, y)] \times [I_{\text{in}}(m, n) - I_{\text{in}}(x, y)] \\ &+ \frac{T''[I_{\text{in}}(x, y)]}{2} \times [I_{\text{in}}(m, n) - I_{\text{in}}(x, y)]^2. \end{aligned} \tag{15}$$