

LEARNING TRADITIONAL FILTERS BASED ON IMAGE EXAMPLES

Sarab M. Hameed

University of Baghdad, College of Science Computer Science Department.

Sarab_majeed@yahoo.com

Abstract

This paper presents an approach for image filtering process depends on image examples. The approach involves of preparing a training images, in which a pair of images, with one image purported to be a filtered version of the other, as a training data. The filtered property is to be automatically learned and transferred to another target image. The basic idea behind learning approach is depended on luminance neighborhood statistics. Statistics pertaining to each pixel in the target pair are to be compared against statistics for every pixel in the source pair, and the closest match is to be found and its property (i.e., luminance information) is applied to the target image in order to create a filtered image. This approach is simple and provides results comparable to that obtained in image analogies of the Hertzmann et al.

Keywords: blur filter, sharpen filter, image analogy.

Introduction

Traditionally, digital image processing includes variety of image operation which can be performed on image using convolution operation. Some of these are exploited for removing edges in the image and other sharp details, examples are the mean, mode and median. Other convolution masks can be used for amplifying, or enhance the edges. Examples are sobel, Roberts, and prewitt. Reader can refer to [1,2] for further information.

Rather than selecting form myriad of different types of filters and parameters, a user can simply supply an appropriate exemplar and say; in effect make it look like this. Recently, researches are interested in transferring properties from one image to another. One such algorithm is that of Hertzmann et al. [3]. Image analogies provide a very natural means of specifying image transformations. Image analogies are based on a various algorithms from different areas. It combines techniques from machine learning, rendering and texture synthesis. Texture synthesis, originally introduced by Heeger and Bergen [4], is a area that is being widely researched in the last few years. Ashikhmin [5] introduced just before the development of image analogies a texture synthesis method that works very fast and

produces good results. The image analogies work is mainly based on this algorithm combined with Wei and Levoy's work [6]. Similar research as image analogies has been done by Freeman et al. [7] where they use Markov Random Fields (MRFs) for scene learning.

While image analogies are clearly a desirable goal, it is not so clear how they might be achieved. For one thing, a crucial aspect of image analogies problem statement is the definition of the similarity used to measure not only the relationship between each unfiltered image and its respective filtered version, but also the relationship between the source pair and the target pair when taken as a whole. This issue is tricky, in that we want to use some metric that is able to preserve recognizable features of the original image, while at the same time is broad enough to be applied to some completely different target image.

In this paper, we are interested in transferring some properties including, blurring, sharpening, and embossing from one image to another and introduce an approach that performs this goal. The rest of this paper is organized as follows. Section two presents description of learning of the proposed approach. Section three depicts

some results using making comparison with Hertzmann algorithm. Finally, Section four introduces conclusion.

The Learning Approach

This section states how our approach works for transferring blurring, sharpening, and embossing filtered properties from one image to another without using explicit filter. The proposed approach is illustrated in figure (1), takes three images, the source image (A), the filtered source image (A'), and target image (B), and produce the smoothed image (B') as output. The B' is produced by learning the filter that is applied on A' and then apply it on B'. The steps of the learning approach can be stated as follows.

Selecting Image Feature: Image feature selection and representation is a large open problem and an active area of research in machine learning. One way is to compute, store and process the luminance at each pixel, and use it instead of RGB in the distance metric as they generally, did not contain enough data to match between a pair of image. The luminance can be computed in many ways; here, we use l from $l\alpha\beta$ color space that was proposed by Ruderman et al [8, 9] and developed to minimize correlation between the three coordinate axes of the color space. The color space provides three de-correlated, principal channels corresponding to an achromatic luminance channel l and two chromatic channels α and β , which roughly correspond to yellow-blue and green-red opponent channels.

The RGB color space is initially converted to LMS space, which corresponds to the bands of sensitivity of the human cones and was thus, used Runderman et al. as the basis to define $l\alpha\beta$ space. This initial transformation corresponds to multiplication by a 3 x 3 matrix as expressed in equation (1) [10]:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

The conversion from LMS to $l\alpha\beta$ is performed through the sequential application of the logarithmic transformation expressed

in equation (2) and the linear transformation expressed in equation (3).

$$[L, M, S] = [\log L, \log M, \log S] \quad (2)$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (3)$$

The transformation from RGB to $l\alpha\beta$ is applied to the input images A, A' and B to compute the luminance component (i.e. feature).

Calculate the Similarity metric: After computing the luminance for images A, A', and B images, we have to find the pixel $p \in A$ that is closet match to pixel $q \in B$. For each pixel $q \in B$, scan the source image A in raster scan order (from left to right and from top to bottom) and select the pixel that satisfy the closest matching. This approach uses 3x3 windows size and sum square luminance differences between the pixel of image A and B (i.e. norm $\| \cdot \| ^2$ is computed as a weighted distance). Consider a source pixel p and a target pixel q then the square sum of the pixel-wise luminance value difference is calculated as illustrated in equation 4:

$$match(p, q) = \min \sum_{x=1}^{x=3} \sum_{y=1}^{y=3} (l_p(x, y) - l_q(x, y))^2 \quad (4)$$

Transferring Property between Images: Transfer the luminance feature of pixel $p \in A'$ to pixel $q \in B'$ and the color information is recovered by copying the α and β channels of the input B image to the target B'. Suppose that the pixel $p_{i,j} \in A$ is the closet match to pixel $q_{m,k} \in B$, then the luminance value of pixel $p \in A'$ is transfer to pixel $q \in B'$, and the chromatic components of pixel $q_{m,k} \in B$ is transfer to pixel $q \in B'$, as formulated in equation 5.

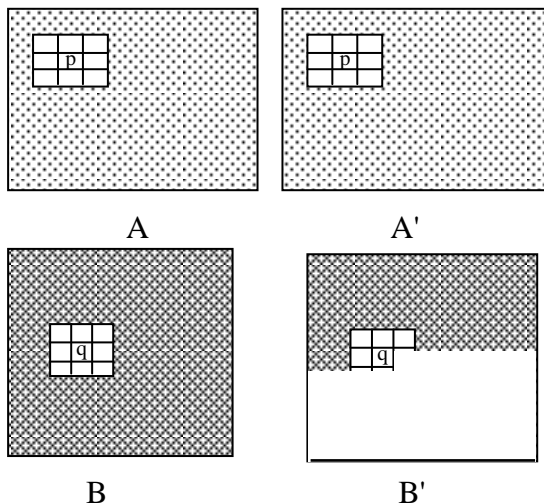
$$\begin{aligned} l(q_{k,m}) \in B' &= l(p_{i,j}) \in A' \\ \alpha(q_{k,m}) \in B' &= \alpha(q_{k,m}) \in B \\ \beta(q_{k,m}) \in B' &= \beta(q_{k,m}) \in B \end{aligned} \quad (5)$$

Finally, Transformation from $l\alpha\beta$ color space RGB color space: Covert image B' to RGB color space include: first, convert $l\alpha\beta$ to LMS using the equation (6):

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} \quad (6)$$

Then, after raising the pixel values to the power ten to go back to linear space, we can convert the data from LMS to RGB using equation (7).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.467 & -3.587 & 0.119 \\ -1.218 & 2.380 & -0.162 \\ 0.049 & -0.243 & 1.20 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (7)$$



**Figure (1) : The learning Approach:
Transfer the feature of pixel p in A' to
pixel q in B' that closet match**

Results

This section reported some results obtained from the learned approach which transfers filtered property including blurring, sharpening, and embossing from one image to another, transferring filtered properties depends on an example. Figure 2 depict some learning approach results. A' are obtained from Adobe Photoshop. These illustrate blurring (top), sharpening (middle), and embossing (bottom) filters. Figure 3 depicts the comparison results of our approach with Hertzmann algorithm.

Conclusions

In this paper, we introduce a simple approach that attempt to automatically learn filters from training data .This approach can be used for wide variety of image and its experimental results show that this approach give an acceptable results comparable to that obtained from image analogies of Hertzmann et al. However, the time required for our approach varied and depends on the size of source image and target image (i.e. increased exponentially). This open the door for other research to increase the efficiency of the algorithm by adopting other search algorithm that search the data in, for example logarithmic.

References

- [1] R.C. Gonzalez, and R.E. Woods "Digital Image Processing", Wesley Publishing 2001.
- [2] S.E. Umbaugh, "Computer Vision and Image Processing", Prentice Hall, Inc., 1998. [3] A. Hertzmann , C. E. Jacobs, N. Oliver , B. Curless and D. H Salesin , Image analogies. roceedings. ACM Press / ACM SIGGRAPH ,2001, pp 327-240.
- [4] H.J. David and J. R. Bergen, "Pyramid-based Texture Analysis/ Synthesis", proceedings of the 22nd annual conference on Computer graphics and interactive techniques. ACM Press, 1999, pp 229–238.
- [5] M. Ashikhmin, "Synthesizing Natural Textures", ACM Symposium on Interactive 3D Graphics, 2001, pp 217-226.
- [6] L. Y. Wei and M. Levoy, "Fast Texture Synthesis Using Tree-Structured Vector Quantization", proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co, 2000, pp 479–488.
- [7] W. T.Freeman W.T, E.C. Pasztor and O.T. Carmichael, "Learning Low-Level Vision", International Journal of Computer Vision., Vol. 40, Issue 1, 2000, pp 24–57.
- [8] L. F. M. Vieira, R. D. Vilela, E. R. do Nascimento, F. A. Fernandes Jr., R. L. Carceroni, and A. A. Araujo,

- "Automatically Choosing Source Color Images for Coloring Grayscale Images", IEEE Computer Society, 2004, pp 3-4.
- [9] Z. Pan, Z. Dong, and M. Zhang "A new Algorithm for Adding Color to Video or Animation Clips", *Journal of WSCG* 2004, pp2-3.
- [10] E. Reinhard, M. Ashikmin, B. Gooch, and P. Shirely, "Color Transfer between Images" IEEE Computer Graphics and Applications, 2001, pp 34-41,

المستخلص

يقدم هذا البحث طريقة لعملية ترشيح الصورة بالاعتماد على امثلة صور. تتضمن الطريقة تحضير صور التدريب، والتي هي زوج من الصور احدهما تمثل النسخة المرشحة للآخرى كبيانات تدريب. خاصية الترشيح ستتعلم بصورة اوتوماتكية وتنتقل هذه الخاصية الى صورة الهدف. إن الفكرة الأساسية وراء طريقة التعلّم مُعتمَدة على إحصائيات الجوار. الإحصائيات تُخصّص إلى كل نقطة شاشة في زوج الهدف ستُقارن ضدّ الإحصائيات لكل نقطة شاشة في الزوج المصدري، ونجد النظير الأقرب وخاصيته (معلومة الاضاء) تطبق إلى صورة الهدف لكي تكون صورة مرشحة. هذه الطريقة بسيطة واعطت نتائج مشابهة إلى تلك حصلت عليها في تناظرات الصورة لـ Hertzmann وآخرون.

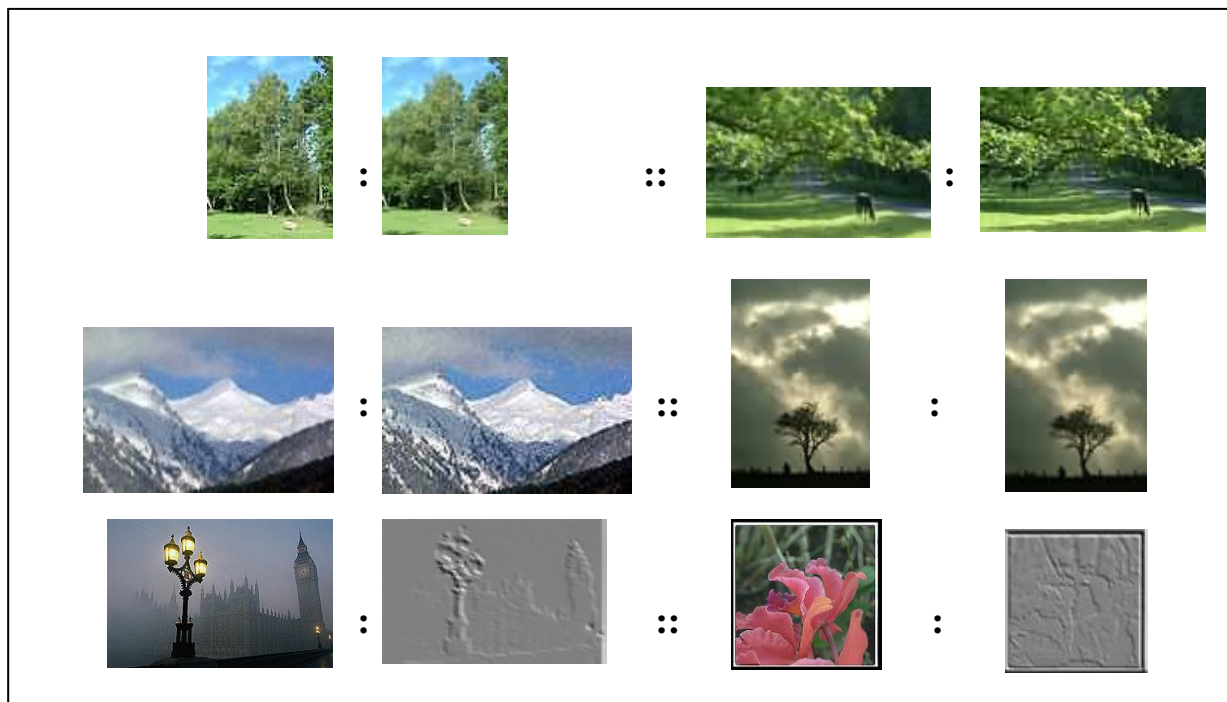


Fig.(2): Some learning approach results. Each result is depicted as A: A' :: B:B'

Source Image	Filtered Source Image	Target Image	Filtered Target Image by Hertzmann Algorithm	Filtered Target Image by Our approach

Figure 2: Comparison results between Hertzmann and our approach. The 1st row is applied blurring filters and the 2nd row is applied embossing filter.