Laplace operator based multi-channel image filters learning

Yantao LIU*,**, Meng LIU*, Bin LIU*, Jingang SUN* and Xiuping LIU*
* Department of Mathematical Science, Dalian University of Technology
No.2 Linggong Road, Ganjingzi District, Dalian City, Liaoning Province, 116024, China
E-mail: xpliu@dlut.edu.cn
** Department of Science, Dalian Nationalities University
No. 18, Liaohe West Road, Jinzhou New District, Dalian, Liaoning Province, 116600, China

Received 26 January 2016

Abstract
Learning an image filter from a pair of images composed of an original image and its filtered version has attracted increasing attention in the image processing region. However, the existing methods only take the color information into consideration to learn an image filter, which cannot accurately depict the luminance and some style changes of images so that the generated learning results are inaccurate. This paper proposes a Laplace operator based multi-channel learning approach to learn a multi-channel filter from an original image and its filtered version. Different from previous method, we adopt a strategy to learn a multi-channel filter which can accurately reconstruct the filtered image. In addition, benefiting from the Laplace regularization based multi-channel learning model, we can make the neighbor pixels of each channel have the similar weights and learn more homogeneous weight maps for each channel filter, thereby generating more accurate results with less time. Due to the multi-channel representation of our learnt filter, we can edit the luminance style and the color style of images by changing the weights of different channel filter. Furthermore, we can edit the effect of the image filters by tuning the weights of some basis filters.

At last, extensive experiments well validate the performance of our method over state-of-the-art methods in terms of accuracy and speed.

Key words: Laplace operator, Multi-channel learning, Image filters, Image editing, Multi-channel filter

1. Introduction

Image filtering is a very general concept, including artistic filters, curves, tone adjustment (Lischinski et al., 2006; Farbman et al., 2008), color transfer (Reinhard et al., 2001; Xiang et al., 2009; Cusano et al., 2012; Wu et al., 2013), image enhancement (Son et al., 2014; Wojcik, 1994), and so forth. It is the key element in many graphics algorithms as well as an important tool in many image editing applications. However, the exact functioning principles behind many image filters in commercial softwares, such as Adobe Photoshop, are undisclosed. Besides, we need to manually tune a lot of parameters for achieving a desired style of image, which might require expert knowledge and can be time consuming. Therefore, learning image filters becomes an important and challenging task.

Hertzmann et al. (2001) and Tang et al. (2012) propose non-parametric texture synthesis algorithms to obtain "texture-like" effects from appropriate example pairs. However, they cannot perform well on many other typical image filter categories. Berthouzoz et al. (2011) use multiple training examples to learn photo manipulation macros, but it requires Photoshop macros to be recorded. Kaiming et al. (2013) construct a linear combination of local mappings within windows of the guided images to approximate the input image. Recently, Huang et al. (2014) propose a model which is the state-of-the-art method to learn an image filter from a single pair of input images. They model a compound filter by a linear combination of basis filters from filter bank. Nevertheless, the speed of their learning algorithm is very slow. Sahba and Tizhoosh (2003) also use a linear combination of four filters to produce an improved denoising filter for a given input image by a reinforced learning algorithm. Though their model is very simple, it cannot automatically learn the filter.

To tackle these problems, we propose a multi-channel learning algorithm based on the Laplace operator to learn a multi-channel filter from an example pair. The main contributions of our approach lie in three aspects: firstly, we learn
a multi-channel filter from an example pair with multi-channel learning algorithm to enhance the accuracy of learning results; secondly, we utilize a Laplace regularization based learning model to obtain more homogeneous weight maps for each channel filter, which improves the learning accuracy and speed; thirdly, the multi-channel representation of our filters enables us to respectively tune the weights of luminance channel and color channels, thereby achieving color editing function. The main pipeline of our paper is shown in Fig. 1.

The rest of our paper is organized as follows. We describe the details of our learning algorithm in section 2. Experimental results of the proposed algorithm and the comparisons are shown in section 3. Applications will be presented in section 4 and the conclusion is made in section 5.

2. Our Algorithm

Our method contains three steps. Firstly, we use color space conversion technique to divide the input images into a luminance channel and two color channels for learning a multi-channel filter. Secondly, we select a basis filter bank which is used for constructing each channel filter. Lastly, we use a Laplace operator based learning model to learn the weight matrices for each channel filter.

2.1. Color Space Converting

YIQ color space (Schwarz et al., 1987) has the advantage of separating the luminance and color information, and is very efficient to convert from the RGB color space to YIQ color space. Besides, the clustering feature of the YIQ color space is good. Thus, we convert an input example pair containing an original image A and its filtered version A' into the YIQ color space in our algorithm, where Y is a luminance channel and both I and Q channels are “color difference” components. As we all know, different filter effects can be achieved by means of modifying the luminance or the color information of images. Motivated by this fact, we introduce a multi-channel filter for better approximating the latent filter.
2.2. Basis Filters Selection

We define each channel of our multi-channel filter as a linear combination of basis filters:

$$c(p) = \sum_i c_i(p) w_i(p),$$  \hspace{1cm} (1)

where $p$ is the pixel coordinate, $c_i$ is the result of applying one of the basis filter to the corresponding channel of the input unfiltered image $A$, and $w_i(p)$ is the spatially varying parameters which need to be learned in our method.

Sahba and Tizhoosh (2003) linearly combine four filters to form a denoising filter by a reinforced learning algorithm for a given input image, while the approximation effect depends on the learnt weights. Inspired by this strategy, we select a fixed filter bank as our basis filter bank. Our basis filter bank consists of nine basis filters which are Average, Gaussian, Horizontal Motion Blur, Vertical Motion Blur, Horizontal Sobel, Vertical Sobel, Bilateral, Channel Offset and Identity filter. These filters are commonly used in image processing field. The Identity filter is the corresponding channel of image $A$, which passes through the channel unchanged. Channel Offset filter provides a constant channel offset for each corresponding channel like in Huang et al. (2014). In our method, we set the constants of the three Channel Offset filters to 0.01. The parameters of other basis filters are set to their default values.

2.3. Learning Model

The problem of determining the optimal weight maps for a channel of image $A$ and the corresponding channel of the filtered image $A'$ is regarded as an energy minimization model. We formulate the learning model as follows:

$$w^* = \arg \min_w \lambda (\sum_{i=1}^9 |c_i(p)w_i(p) - c'_i(p)|^2 + \sum_{i=1}^9 |L w_i(p)|) + \alpha \sum_{i=1}^9 |w_i(p)|,$$  \hspace{1cm} (2)

where $p$ is the pixel coordinate, $c_i$ is the result of applying one of the basis filter to the corresponding channel of the input unfiltered image $A$, $c'_i$ is the corresponding channel of the image $A'$ and $w_i$ is the spatially varying parameters which need to be learned in our method. Note that,

$$|L w_i(x, y)| = |w_i(x+1, y) + w_i(x-1, y) + w_i(x, y+1) + w_i(x, y-1) - 4w_i(x, y)|.$$  \hspace{1cm} (3)

Supposing the size of the input unfiltered image $A$ in the YIQ space is $m \times n \times 3$, then the sizes of the luminance and the color channel images are $m \times n$. The Eq. (2) can be further rewritten in the matrix form:

$$w^* = \arg \min_W \lambda \| \text{approximate} W - C \|^2 + \| L W \|_1 + \alpha \| W \|_1,$$  \hspace{1cm} (4)

where $C$ is a $mn \times 9mn$ matrix that denotes the results of applying basis filters to the corresponding channel of image $A$, $W$ is a $9mn \times 1$ matrix denotes the unknown basis weights, $C'$ is a $mn \times 1$ vector denotes pixel values from the corresponding
Fig. 3  Quantitative comparison of Laplace operator based learning model (maroon lines) with Gradient operator based model (blue lines). The horizontal axis represents different kind of image filters in the Photoshop which are applied to the input images. (a) The SSIM value curve of the two methods’ learning results. (b) The RMS error curve of two learning models.

channel of image $A'$ in the YIQ color space, $L$ is the matrix of the Laplace operator in Eq. (3), and $Ψ = (L\alpha I)^T$, where $I$ is the identity matrix. Here, $\|\bullet\|_2$ is the $L_2$ norm which can be defined as the sum of squares of all elements and $\|\bullet\|_1$ is the $L_1$ norm which can be defined as the sum of absolute values of all the elements. In our model, $\lambda$ and $\alpha$ are two balance parameters, $\lambda$ is set to 100 and $\alpha$ is set to 0.01.

Three terms in our model are related to the model in (Rudin et al., 1992 and Huang et al., 2014). The first one is used to approximate the latent channel filter. The second one aims to make neighbor pixels have similar weights and form homogeneous weight matrixes for each channel. The third one is a sparse item which makes the weights of every pixel concentrate on few basis filters and ensures the numerical stability of our learning model.

As Laplace operator is a second derivative and an isotropic diffusion operator (Wang, 2007), we choose it instead of the gradient operator as the smoothing term of our learning model, which can make more neighbor pixels have similar weights and yield more homogeneous weight maps for each channel. Therefore, the Laplace operator based model improves the learning results in the aspect of accuracy and speed.

Our model is a $\ell_1$ regularized convex problem. The minimum result can be obtained using Split Bregman method (Goldstein and Osher, 2009) and (Chen et al. 2008). The following three equations are alternately iterated until convergence:

$$W^{k+1} = \min W \frac{\lambda}{2} \|CW - C'\|_2^2 + \frac{\gamma}{2} \|s - ΨW - t\|_2^2,$$

$$s^{k+1} = \min s \|s\|_1 + \frac{\gamma}{2} \|s - ΨW^{k+1} - t\|_2^2,$$

$$t^{k+1} = t^k + ΨW^{k+1} - s^{k+1},$$

where $k$ is the current iteration number, $s$ and $t$ are two additional vectors, and $\gamma = 10$ is a relaxation constant which only affects the convergence rate.

Equation (5) can be solved by setting the derivative of its function with respect to $W$ to be zero. We use shrink operator (Goldstein and Osher, 2009) to solve Eq. (6), and Eq. (7) is direct.

The weights of each channel filter can be obtained by applying our model to luminance and two color channels, respectively. After that, we use Eq. (1) to reconstruct luminance channel and color channels. Finally, we convert the obtained image into RGB color space which approximates filtered image $A'$.

3. Experiments and Comparison

In this paper, we use both Structure Similarity Image Metric (SSIM) (Wang et al., 2004) value and Root Mean Square (RMS) error to measure the difference between the reconstructed and the edited image $A'$. The SSIM is used to measure the structure similarity and the best performance value is one. Moreover, we utilize the RMS error to measure the difference between the reconstructed and the edited image, and the best performance value is zero. Our algorithm is
Table 1 The comparison of the learning time between Laplace operator based model and Gradient based model on different image filters. Some learning time of Laplace based model is shown in red and Gradient based model is shown in blue.

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 166.0514s    | 41.3048s     | **215.7203s** | 46.8155s | 34.725s   | 34.8485s   |
| Laplace Operator | 26.079s      | 6.0065s      | **21.8295s**  | 7.2692s | 5.4744s   | 5.4564s    |

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 33.7207s    | 67.994s      | **35.2141s** | 34.0212s | 34.4871s  | 34.7078s   |
| Laplace Operator | 5.4847s      | 11.7977s     | **5.7254s**  | 5.4166s | 5.5393s   | 6.1051s    |

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 35.0461s    | 34.3389s     | **33.0131s** | 34.4725s | 34.3868s  | **34.0265s** |
| Laplace Operator | 5.3977s      | 5.4109s      | **5.6801s**  | 5.6316s | 5.3528s   | 6.5485s    |

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 34.5231s    | 34.0427s     | **35.0243s** | 34.9431s | 34.2415s  | 34.4045s   |
| Laplace Operator | 6.7649s      | 5.5092s      | **6.4966s**  | 6.2467s | 5.3869s   | 5.6292s    |

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 318.7772s   | 36.1409s     | 35.3287s     | 102.2925s | 74.2485s  | **410.0638s** |
| Laplace Operator | 35.0925s     | 6.8811s      | **6.5402s**  | 18.0917s | 12.4114s  | 35.9997s   |

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 34.6574s    | 253.3597s    | 425.6817s     | 240.2008s | 298.9446s | 438.7355s  |
| Laplace Operator | **5.1332s**  | 44.8353s     | 36.7895s     | 52.5488s | 37.8784s  | 44.2044s   |

| Filter          | Color Adjust | Curve Adjust | Color Pencil | Cut Out | Dry Brush | Film Grain |
|-----------------|--------------|--------------|--------------|---------|-----------|------------|
| Gradient Operator | 157.3234s   | 142.0868s    | 443.0793s     | **43.9385s** | 43.5435s  | 42.8664s   |
| Laplace Operator | 27.9966s     | 76.0482s     | 36.2423s     | **43.7069s** | 6.7757s   | 6.3707s    |

implemented with Matlab (R2010b) and all experiments are ran on a PC with an Intel(R) Xeon(R) E5630 2.53GHz CPU and 24GB RAM.

Figure 2 demonstrates that our multi-channel filter can greatly approximate filtered images. The first row shows the original images and the images in the second row are the filtered versions of the corresponding images in the first row. The corresponding filters are Curves, Color Adjustment, Curves, Black and White, Curves and Paint Daubs filters, respectively. The bottom shows the results obtained by the proposed method. From left to right of the third row, the corresponding SSIM values are 1, 0.9998, 1, 1, 1 and 1, respectively, and the corresponding RMS value are 0.0149, 2.2371, 0, 0, 0 and 0. It is obvious that we can accurately learn different image filters from the given image pairs.

3.1. Validation of the Laplace Operator based Model

During the learning process we intend to keep the spatial continuity, thus we should make the neighbour pixels have similar weights. As Laplace operator is a second derivative operator and an isotropic diffusion operator (Wang, 2007), we choose $\|\bar{L}W\|_1$ as the smoothing term of our learning model, which can make adjacent pixels have similar weights and yield more homogeneous weight maps for each channel. To demonstrate effectiveness of the Laplace operator, we compare the model based upon the Laplace as smoothing term with the model using gradient operator as the smoothing term.

![Fig. 4 Comparison of Multi-channel learning model (maroon bar chart) with Non-multichannel based model (blue bar chart). The horizontal axis represents different kind of image filters in the Photoshop which are applied to the input images. (a) The SSIM value bar chart of the two methods. (b) The RMS error bar chart of the two methods’ learning results.](image-url)
term. The RMS error curve and the SSIM value curve of the reconstructed results obtained by these two models are shown in Fig. 3, in which the maroon curves are the results of Laplace based model and the blue ones are the results of Gradient based method. In the Fig. 3 (a), we can find that the Laplace operator based model can perfectly reconstruct the input images with most value equal to one. In the RMS curve Fig. 3 (b), the values of the Laplace operator based method are mostly near zero. Furthermore, the computation time of the Laplace based method and the Gradient based model are shown in the Table 1. It clearly shows that the Laplace based method is much more efficient.

### 3.2. Evaluation of Multi-channel Learning Algorithm

To test the effectiveness of our method, we compare our learning method with the Laplace operator based non multi-channel learning method in the YIQ color space through the SSIM measurement and the RMS error. The resulting bar charts are shown in Fig. 4, in which the maroon bar chart represents multi-channel learning method and the blue bar chart states the non multi-channel method. In the Fig. 4 (a), we find that our multi-channel method can well preserve the structure similarity of the results with the SSIM value near one. And Fig. 4 (b) shows that our multi-channel method generates more accurate learning results with the RMS value mostly equal to zero. Besides, we also compare the learning time of these two models, and the results are shown in Table 2. Although most learning time of our model is nearly two or three seconds higher than the non multi-channel, the learning accuracy is higher than non multi-channel based method. For some image filters, our method not only need less learning time, but also can achieve higher accuracy, see the time marked in red in Table 2. Generally speaking, our method outperforms the non multi-channel learning method.

### 3.3. Comparison with the State-of-the-Art Methods

The approach in (Huang et al., 2014) (referred as PMF) learns a parametric meta-filter from an example image pair by an automatically learning method. In Fig. 5, we quantitatively evaluate the proposed method (maroon bar chart) and PMF method (blue bar chart) on a large variety of composite filters. Figure 5 (a) shows that our method can robustly generate more accuracy learning result with the SSIM value near one. Moreover, both the SSIM value and RMS error bar charts clearly demonstrate the superior overall performance of our method. Table 3 lists the learning time of our method and PMF model for different image filters. As shown in Table 3, our method is much faster than the PMF. The experiments demonstrate the proposed method achieves significant improvement over PMF model.
4. Application

4.1. Style transfer

Our multi-channel filter can be applied to edit the color style of an image B, so that the filtered version of B can approximate the color style of filtered image $A'$. Firstly, we need to establish the pixel correspondence between A and B. Namely, each pixel in image A should be matched to a pixel in image B. Then, we copy the pixel-wise weights of basis filters which is relevant to image A to the weights of basis filters belonging to image B. Finally, we utilize the obtained weights to generate the final results.

In our paper, we use SiftFlow (Liu et al., 2011) algorithm to establish pixel correspondence that globally aligns the two images and well preserve the spatial coherence. Note that, the filtered images in our paper are generated by Photoshop filters, and the same filters are used to obtain the ground truth images. Moreover, Huang et al. (2014) propose a filter transfer method which can make a given image has similar filter effect with the input filtered image.

In Fig. 6, we show some comparison results with the filter transfer results of Huang’s method and the method of (Hertzmann et al., 2001). The first and second rows show the comparisons of the color transfer results and the bottom row shows results of changing a color image into a Black and White images. In the Fig. 6 (d), we find that the results of (Hertzmann et al., 2001) lose some color information and detail information. This is due to the method in (Hertzmann et
al., 2001) is designed for texture-like images, and it cannot learn well for color transform filters. Besides, the PMF method generates obvious noises on the mountain in the top of Fig. 6 (e) and undesired artifacts on the grass in the second one of Fig. 6 (e). Because the PMF method learns the weights for the three channels together in the RGB color space, which cannot correctly depict the change of the luminance and the color. Therefore, they cannot successfully learn a filter from an example pair with a color image and its filtered version. Although the method of (Hertzmann et al., 2001) can transfer the color to the new input images, it is designed for texture-like filters. Therefore, they generate images which are blurred and lose detail information. Consequently, it is apparent that our method performs favorably against both of the two models.

4.2. Color editing

Our filter’s multi-channel representation enables us to respectively tune the parameters of luminance channel and color channels, thereby obtaining filter editing and color editing functions. Figure 7 and Figure 8 show some color and luminance variations, which are achieved by tuning per-pixel weights of each channel filter. Figure 7 (d) is achieved by tuning the weights of Y channel filter. Figure 7 (e) and Figure 7 (f) are generated by uniformly adjusting the weights of I and Q channel filters. Figure 8 shows an example of luminance change, in which Figure 8 (d) is achieved by uniformly decreasing the weights of every basis filter and Figure 8 (d) and (e) are obtained by uniformly increasing the weights of each basis filter.

In Figure 9, we show some filter and color variations which are achieved by tuning per-pixel weights of each channel filter and the global parameters of basis filters. Figure 9 (c) shows a reduced filter which is generated by uniformly

---

Fig. 6 Visual comparison among our transfer method, PMF’s filter transfer method and Hertzmann et al. (2001) on color adjustment transfer. (a) Original images. (b) Filtered versions. (c) New input images. (d) Results of Hertzmann et al. (2001). (e) Results of Huang et al. (2014). (f) Our results. (g) Ground truth.

Fig. 7 Color editing results. (a) The original images. (b) The filtered images. (c) Our learnt results. (d) The results of editing Y channel. (e) Color editing results obtained by tuning the weights of I channel. (f) Results obtained by tuning the weights of Q channel.
Fig. 8 Luminance editing results. (a) The original images. (b) The filtered images. (c) Our learnt results. (d) The results of decreasing Y channel to 0.6x. (e) The results of increasing Y channel to 1.5x. (f) The results of increasing Y channel to 2x.

Fig. 9 Filter and color editing results for Motion Blur Effect. (a) The original images. (b) The image filtered by Motion Blur filter. (c) The reduced Motion Blur effect of (b). (d) The strengthened Motion Blur effect of (b). (e) Color editing result obtained by tuning the weights of I color channel. (f) Color editing result obtained by tuning the weights of Q color channels.

decrease the weights of all Motion filters in the three channels. Figure 9 (d) shows a strengthen filter which is achieved by uniformly increase the weights of all Motion filter in the three channels. Figure 9 (e) and Figure 9 (f) show the results of a manipulation for the per-pixel weights of color channels.

5. Conclusions

In this paper, we propose a novel multi-channel learning algorithm to learn an image filter from an example image pair. Different from previous methods, we decompose the input images into luminance and color channels to learn a multi-channel filter. Each channel filter is represented as a spatially varying linear combination of basis filters from a fixed filter bank. We further introduce a Laplace operator based learning model to learn the parameters of each channel filter. Experiments on a large variety of common image filters show that our method performs favorably against the state-of-the-art methods in both accuracy and efficiency. In the future work, we will investigate a more reliable correspondence map between general images to improve our style transfer process.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61370143) and the National Natural Science Foundation of China (F61202050).

References

Berthouzoz, F., Li, W., Dontcheva, M. and Agrawala, M., A framework for content-adaptive photo manipulation macros: application to face, landscape, and global manipulations, Acm Transactions on Graphics, Vol.30, No.5 (2011), pp.1317-1329.
Chen, Y., Davis, T. A., Hager, W. W. and Rajamanickam, S., Algorithm 887: cholmod, supernodal sparse cholesky factorization and update/downdate, Acm Transactions on Mathematical Software, Vol.35, No.3 (2008), pp.1-14.
Cusano, C., Gasparini, F. and Schettini, R., Color transfer using semantic image annotation, Digital Photography VIII, Vol.8299, No.3 92012), pp.872-886.
Farbman, Z., Fattal, R., Lischinski, D. and Szeliski, R., Edge-preserving decompositions for multi-scale tone and detail manipulation, Acm Transactions on Graphics, Vol.27, No.3 (2008), pp.15-19.
Goldstein, T. and Osher, S., The split bregman method for $\ell_1$-regularized problems, Siam Journal on Imaging Sciences, Vol.2, No.2 (2009), pp.323-343.
Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B. and Salesin, D. H., Image analogies, Proceedings of the Conference on Computer Graphics and Interactive Techniques (2001), pp.327–340.
Huang, S. S., Zhang, G. X., Lai, Y. K., Kopf, J., Cohen-Or, D. and Hu, S. M., Parametric meta-filter modeling from a single example pair, Visual Computer International Journal of Computer Graphics, Vol.30, No.6-8 (2014), pp.673-684.
Kaiming, H., Jian, S. and Xiaou, T., Guided image filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.35, No.6 (2013), pp.1397-1409.
Lischinski, D., Farbman, Z., Uyttendaele, M. and Szeliski, R., Interactive local adjustment of tonal values, Acm Transactions on Graphics, Vol.25, No.3 (2006), pp.646-653.
Liu, C., Yuen, J. and Torralba, A., Sift flow: dense correspondence across scenes and its applications, IEEE Transactions on Software Engineering, Vol.33, No.5, pp.978-994.
Reinhard, E., Adhikhmin, M., Gooch, B. and Shirley, P., Color transfer between images, IEEE Computer Graphics and Applications, Vol.21, No.5 (2001), pp.34-41.
Rudin, L. I., Osher, S. and Fatemi, E., Nonlinear total variation based noise removal algorithms, Physica D-nonlinear Phenomena, Vol.60, No.1-4 (1992), pp.259-268.
Sahba, Fand Tizhoosh, H. R., Filter Fusion For Image Enhancement Using Reinforcement Learning, Proceedings of the Canadian Conference on Electrical and Computer Engineering (2003), pp.847 - 850.
Schwarz, M. W., Cowan, W. B. and Beatty, J. C., An experimental comparison of rgb, yiq, lab, hsv, and opponent color models, Acm Transactions on Graphics, Vol.6, No.2 (1987), pp.123-158.
Son, M., Lee, Y., Kang, H. and Lee, S., Art-photographic detail enhancement, Computer Graphics Forum, Vol.33, No.2 (2014), pp.391-400.
Tang, Y., Shi, X., Xiao, T. and Fan, J., An improved image analogy method based on adaptive cuda-accelerated neighborhood matching framework, Visual Computer, Vol.28, No.6-8 (2012), pp.743-753.
Wang, X., Laplacian operator-based edge detectors, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.29, No.5 (2007), pp.886-890.
Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P., Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing, Vol.13, No.4 (2004), pp.600 - 612.
Wojcik, Z. M., Application of rough sets for edge enhancing image filters, Proceedings of the International Conference on Image Processing (ICIP-94) (1994), pp.525-529.
Wu, F., Dong, W., Kong, Y., Mei, X., Paul, J. and Zhang, X., Content-Based Colour Transfer, Computer Graphics Forum, Vol.32 (2013), pp.190C203.
Xiang, Y., Zou, B. and Li, H., Selective color transfer with multi-source images, Pattern Recognition Letters, Vol.30, No.7 (2009), pp.682-689.