

## Video To Animated Cartoon Conversion

<sup>1</sup>Munira Akther & <sup>2</sup>Md. Jahirul Islam

<sup>1,2</sup>Department of Computer Science & Engineering, Shahjalal University of Science & Technology, Sylhet, Bangladesh

---

**Abstract:** the logic of animation making from real video fully depends on various image processing techniques and various kinds of filtering like Gaussian filtering, bilateral filtering, and flow based bilateral filtering, mean-shift filtering etc. It also includes edge detecting methods like canny edge detector, dog edge detector, flow based dog edge detector etc. All this technique works differently on the measurement of quality and run time factor. In this research we have gone through all this techniques and proposed a suitable approach which is convenient to the subject.

**Index Terms:** Animation, Image Processing, Bilateral Filtering, Edge Detection, Bigo Notaion.

---

### I. Introduction

A video is the continuous sequence of still images. So we have to use the methodology of converting a still image into a cartoon like look. Digital image refers to a two-dimensional light intensity function  $f(x, y)$  where,  $x$  and  $y$  denote spatial coordinates and the value of  $f(x, y)$  at  $(x, y)$  is proportional to the brightness (or gray level) of the image at that point. Color and edge of an image is the most important concept for making an anim/cartoon like image. The whole process of color flattening and edge detection is the part of image abstraction process. Several image abstraction techniques have been offered from various scholars in several years. In this paper we have gone through those techniques and found the suitable abstraction process for making animated video. We also focused on the time consuming fact of the abstraction process and offer a complete methodology for making animation which fulfill the abstraction according to quality and timing.

The remainder of the paper organized as follows: section 2 discuss the related work of animation making, section 3 shortly discuss about the various of image abstraction techniques, section 4 discuss about fastening approach according to quality and runtime, section 5 contains results and comparative analysis, in section 6 we explain about our future planning, we conclude in section 7.

### II. Related Work

There are many work have been done by different times based on image and video cartoonization and abstraction. They mainly focused on mathematical and geometrical calculation for making abstracted images.

C.Tomasi et al. [1998] propose a non-iterative scheme for edge preserving smoothing that is non-iterative and simple. Although they claims no correlation with neuro-physiological observations, point out that the scheme could be implemented by a single layer of neuron-like devices that perform the operation once per image. [1]

Jan Eric et al. [2008] proposed a framework of automatic non-photorealistic image processing techniques that creates stylistic illustrations from color images, videos and 3D renderings. To smooth low-contrast regions while preserving edges, a new fast separated implementation of the bilateral filter applied. To extract salient important edges a one-dimensional difference-of-Gaussians filter applied in direction of the gradient. [7]

Winnemoller et al. [2006] use iterated bilateral filtering for abstraction and difference-of-Gaussians (DoG) for edge extraction. This procedure also applies luminance and color opponency. They reduce contrast in low-contrast regions using an approximation to anisotropic diffusion, and artificially increase contrast in higher contrast regions with difference-of-Gaussian edges. [3]

The abstraction process includes several mathematical average calculations which make the total process very slow. Several fastening approach have been proposed to make the abstraction process little bit faster. Durand et al. propose a process where bilateral filter expressed in the framework of robust statistics and show how it relates to anisotropic diffusion. Then bilateral filter accelerate by using a piecewise-linear approximation in the intensity domain and appropriate sub-sampling. This results in a speed-up of two orders of magnitude. [5]

Weiss et al. used a hierarchy of partial distributed histograms using a tier-based approach. Even though complexity has been lowered, simplicity has been lost due to filter size and optimal histogram count specific implementation requirements. This method is limited to rectangular spatial kernels and box filters. Another concern is the imperfect frequency response of their spatial box filter. [12]

Porikli et al. They construct an integral histogram and use the integral histogram to find the bilateral convolution response on rectangular box filter with uniform domain kernel, when the intensity differences can be weighted with any arbitrary range function. [10]

K.N Chowdhry et al. Fast O (1) bilateral filtering using trigonometric range kernels is a fast implementation of the bilateral filter proposed trigonometric functions for the range kernel of the bilateral filter, and by exploiting their so-called shiftability property. In particular, a fast implementation of the Gaussian bilateral filter was realized by approximating the Gaussian range kernel using raised cosines. [13]

### III. Abstraction processes for making animated videos

Abstraction process is the important part for cartoonizing an image. Images are abstracted by filtering with various mathematical equation. This segment will discuss the image abstraction technique which we have implemented and experimented in various ways to make an image into anim like look.

#### Abstraction through Bilateral Filter

At first we have start with bilateral filter because bilateral filter is the basic part of image abstraction. Tomasi and Manduchi [1998] have suggested computing the bilateral filter on a perceptually uniform feature space, such as CIE Lab[14] so that image contrast is adjusted depending on just noticeable differences. Bilateral filtering smoothes images while preserving edges, by means of a nonlinear combination of nearby image values. The method is non iterative, local, and simple. [1]

$$h(x) = k_d^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\epsilon) c(\epsilon, x) d\epsilon \dots \dots \dots (1)$$

Where  $c(\xi, x)$  measures the geometric closeness between the neighborhood center  $x$  and a nearby point. The bold font  $f$  and emphasizes the fact that both input and output images may be multiband. If low-pass filtering is to preserve the dc component of low-pass signals then the following equation obtained,

$$k_d(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\epsilon, x) d\epsilon \dots \dots \dots (2)$$

Range filtering is similarly defined:

$$h(x) = k_r^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\epsilon) s(f(\epsilon), f(x)) d\epsilon \dots \dots \dots (3)$$

Thus the similarity function  $s$  operates in the range of the image function  $f$  while the closeness function  $c$  operates in the domain of  $f$ . the normalization constant (2) is replaced by

$$k_r(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(f(\epsilon), f(x)) d\epsilon \dots \dots \dots (4)$$

Contrary to what occurs with the closeness function  $c$ , the normalization for the similarity function  $s$  depends on the image  $f$ .

The appropriate solution is to combine domain and range filtering thereby enforcing both geometric and photometric locality. Combined filtering can be described as follows:

$$h(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\epsilon) c(\epsilon, x) s(f(\epsilon), f(x)) d\epsilon \dots \dots \dots (5)$$

With the normalization

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\epsilon, x) s(f(\epsilon), f(x)) d\epsilon \dots \dots \dots (6)$$

The result of bilateral filter almost fulfill the initial phase cartoonizing. We experienced the difference between input image and filtered image. Bilateral smoothes the image in low contrast area sharpen the high contrast area.

#### Real Time Video Abstraction

Though bilateral almost fulfill the purpose of cartoonizing, but we have to use abstraction process to make it fully cartoonized. Winne mölleretal [2006] use iterated bilateral filtering for abstraction and difference-of-Gaussians (DoG) for edge extraction. This procedure abstracts imagery by modifying the contrast of visually important features, namely luminance and color opponency. They reduce contrast in low-contrast regions using an approximation to anisotropic diffusion, and artificially increase contrast in higher contrast regions with difference-of-Gaussian edges. [3]

#### Extended Nonlinear Diffusion

Given an input image  $f(\cdot)$ , which maps pixel locations into some feature space, by defining the following filter,  $H(\cdot)$ :

$$H(\hat{x}, \sigma_d, \sigma_r) = \frac{\int e^{-\frac{1}{2}(\frac{\|\hat{x}-x\|}{\sigma_d})^2} w(x, \hat{x})f(x)dx}{\int e^{-\frac{1}{2}(\frac{\|\hat{x}-x\|}{\sigma_d})^2} w(x, \hat{x})dx} \dots \dots \dots (1)$$

In this formulation,  $\hat{x}$  is a pixel location,  $x$  are neighboring pixels, and  $\sigma_d$  is related to the blur radius. Increasing  $\sigma_d$  results in more blurring, but if  $\sigma_d$  is too large features may blur across significant boundaries. The range weighting function,  $w(\cdot)$ , determines Where in the image contrasts are smoothed or sharpened by iterative Applications of  $H(\cdot)$ . [3]

$$w(x, \hat{x}, \sigma_r) = (1 - m(\hat{x})) \cdot w'(x, \hat{x}, \sigma_r) + m(\hat{x}) \cdot u(\hat{x}) \dots \dots \dots (2)$$

$$w'(x, \hat{x}, \sigma_r) = e^{-\frac{1}{2}(\frac{\|f(\hat{x})-f(x)\|}{\sigma_r})^2} \dots \dots \dots (3)$$

For the real-time, automatic case, set  $m(\cdot) = 0$ , such that  $w(\cdot) = w_0(\cdot)$  and Equation 1 becomes the familiar bilateral filter, where  $\sigma_r$  determine show contrasts will be preserved or blurred.

**Edge detection**

They postulated that retinal cells (center), which are stimulated while their surrounding cells are not stimulated, could act as neural implementations of this edge detector. A computationally simple approximation is the difference-of-Gaussians (DoG) operator. Rather than using a binary model of cell-activation, DoG edges defined using as lightly smoothed step function,  $D(\cdot)$  to increase temporal coherence in. The parameter  $\tau$  in Equation 4 controls the amount of center surround difference required for cell activation, and  $\phi_e$  controls the sharpness of the activation fall off. In the following, edge detection defined  $S_{\sigma_r} \equiv S(\hat{x}, \sigma_e)$  and  $S_{\sigma_r} \equiv S(\hat{x}, \sqrt{1.6} \cdot \sigma_e)$  with blur function  $S(\cdot)$  given in Equation 5.

$$D(\hat{x}, \sigma_e, \tau, \phi_e) = \begin{cases} 1 & \text{if}(S_{\sigma_e} - \tau \cdot S_{\sigma_r}) > 0 \\ 1 + \tanh(\phi_e \cdot (S_{\sigma_e} - \tau \cdot S_{\sigma_r})) & \text{otherwise} \end{cases} \dots \dots \dots (4)$$

$$S(\hat{x}, \sigma_e) = \frac{1}{2\pi\sigma_e^2} \int f(x)e^{-\frac{1}{2}(\frac{\|\hat{x}-x\|}{\sigma_e})^2} dx \dots \dots \dots (5)$$

They have used soft luminance quantization on bilateral filtered image and then add dog filtered image and quantized image together. But it was found that too much shading appeared for soft luminance quantization (Figure 2(b)). So we just add the bilateral filtered image with DoG image and found that result is better than previous(Figure 1(a)). An anim like look appeared. The following figure shows the differences.

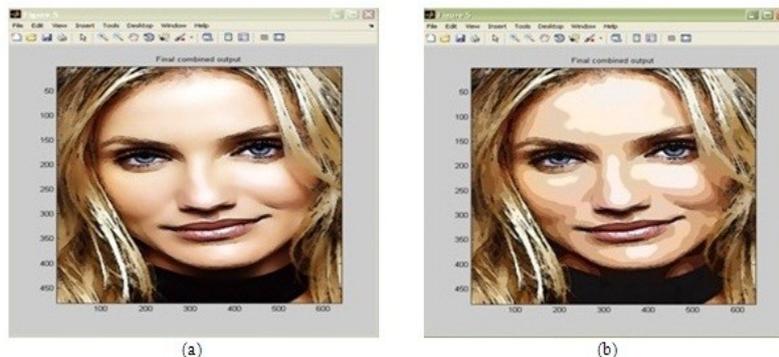


Figure 1. Difference between non-quantized(a) and quantized(b) abstraction

**Image Abstraction by Structure Adaptive Filtering**

This work presents a framework of automatic non photorealistic image processing technique that creates stylistic illustrations from color images, videos and 3D renderings. To smooth low-contrast regions while preserving edges, they present a new fast separated implementation of the bilateral filter. Their approach works by filtering in direction of the gradient and then filtering the intermediate resulting perpendicular direction. When applied iteratively, their approach does not suffer from horizontal or vertical artifacts and creates smooth output at curved boundaries. To extract salient important edges they first apply a one-dimensional difference-of-Gaussians filtering direction of the gradient and then apply smoothing along a flow field which they derive from the smoothed structure tensor. Their method creates smooth coherent output for line and curve segments. They calculate Eigen vectors by local structure estimation before diffusion. [7].

Experimenting the result of above two process, it was found that structure adaptive filtering is good for abstraction purpose but takes more time than winmoller’s video abstraction process. By the experimental analysis and result we propose real time video abstraction by Winne möller et al. [2006] is batter procedure for

better abstraction and less calculating time. So we have applied this abstraction procedure for making animation video.

**IV. Fastening approaches for making animated video**

Normally bilateral filtering takes time than any other filter because it is a non-linear process. Iteration over bilateral filter takes too much time. It takes around three hours to make a video which duration is only thirty seconds in Matlab 7. But there is no good procedure of making abstracted image instead of bilateral filter. There are several approaches have been taken for fastening the bilateral filter. Several methods have been used on bilateral filter equation for fastening animation purpose. Among them trigonometric filter is comparatively efficient and faster.

They offered a raised cosine method into the range kernel. At first they consider the bilateral filter equation described in [1] in an unbiased form that is  $w(x, y) = w(x-y)$  and where range filter is symmetric and depends on the difference of intensity  $\phi(f(x) - f(y))$  [13]. The equation is like follow:

$$\tilde{f} = \eta^{-1} \int w(y) \phi(f(x-y) - f(x)) f(x-y) dy \dots \dots (1)$$

Where,

$$\eta = \int w(y) \phi(f(x-y) - f(x)) dy \dots \dots \dots (2)$$

The presence of term  $(f(x-y)-f(x))$  in (1), makes the filter non-linear. In the absence of this term, that is when  $\phi(s)$  is constant the filter is simply given by averaging.

$$\overline{f(x)} = \int w(y) f(x-y) \dots \dots (3)$$

If the equation (1) can be expressed in terms of (3) then fast averaging algorithm can be leveraged. This can be done if the range kernel is like follow:

$$\phi(s) = \cos(\gamma s) \quad (-T \leq s \leq T) \dots \dots (4)$$

By plugging (4) into (1)

$$\begin{aligned} & \cos(\gamma f(x)) \int w(y) \cos(\gamma f(x-y)) f(x-y) dy \\ & + \sin(\gamma f(x)) \int w(y) \sin(\gamma f(x-y)) f(x-y) dy \end{aligned}$$

Similarly the following can be written:

$$\begin{aligned} & \cos(\gamma f(x)) \int w(y) \cos(\gamma f(x-y)) dy \\ & + \sin(\gamma f(x)) \int w(y) \sin(\gamma f(x-y)) dy \end{aligned}$$

This trick allows to express (1) in terms of linear convolution filters applied to point wise transforms of the image. The aforementioned idea can be present as the following:

$$\phi(s) = \sum_{|n| \leq N} c_n \exp(i n \gamma s) \dots \dots \dots (5)$$

Equation (5) expressed as raised cosine of using binomial theorem.

$$\phi(s) = \sum_{n=0}^N 2^{-N} \binom{N}{n} \exp(i(2n - N)\gamma s)$$

Now using the addition and multiplication property, we can write the following;

$$\phi(f(x-y) - f(x)) = \sum_{|n| \leq N} d_n(x) \exp(i n \gamma f(x-y))$$

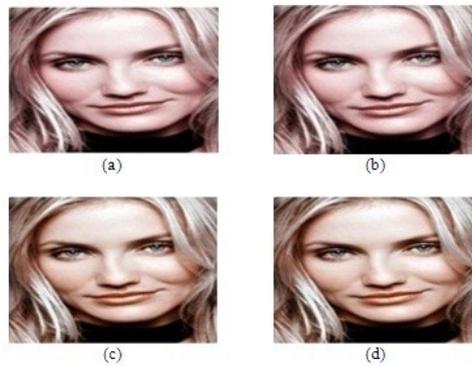
Where,  $d_n(x) = c_n \exp(i(-j n \gamma f(x))$ , plugging this to (1),

$$\tilde{f}(x) = \frac{\sum_{|n| \leq N} d_n(x) \overline{g_n}(x)}{\sum_{|n| \leq N} d_n(x) h_n(x)} \dots \dots \dots (6)$$

$h_n(x)$  and  $g_n(x)$  refers to auxiliary image and N is the degree of kernel.

**Changing Color-Space**

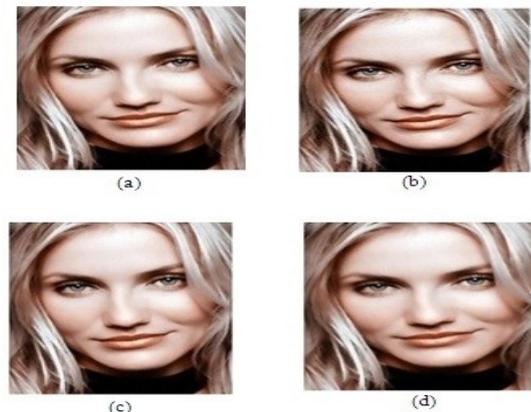
In fast O(1) and shift-able process, the filter directly applied to image of RGB colorspace. The filter allows the three color bands appropriately, and measuring photometric distances between pixels in the combined space this combined distance can be made to correspond closely to perceived dissimilarity in the CIE-Lab color space. This space is based on a large body of psychological data concerning color matching experiments performed by human observers. In this space small Euclidian distances correlate strongly with the perception of color discrepancy as experienced by an “average” Color normal human observer. Thus, in a sense, bilateral filtering performed in the CIE-Lab colorspace is the most natural type of filtering for color images: only perceptually similar colors are averaged together, and only perceptually important edges are preserved. So after changing the color space from rgb to lab we apply the filter to lab image and then re-convert it to Rgb image. In figure2 one can see the effects of changing colorspace on fast O (1) and Shiftable Filter.



**Figure 2.** Comparative result of Fast O(1) kernel and shiftable kernel , (a) Fast O(1) (before changing color space), (b) Shiftable bilateral (before changing color space), (c) O(1) (after changing color space) and (d) Shiftable bilateral (after changing color space)

**Scaling Window Size**

One of the most important parameter is W which is domain of spatial kernel or half size of Gaussian bilateral filter window. The window size has to be  $2 \cdot \sigma_d + 1$ . We have to make the window size odd because if any signal length is odd then it’s easy to calculate the middle point otherwise we will get a complex number after convolution which is tough to handle. The concept of Gaussian function is the bell shape will go increasingly to exact middle point, if the window size is even then he cannot find zero point. So it has to be odd. In fast O (1) and shiftable filter author check the  $\sigma_d$  is odd or even if even than take the window size  $w = \sigma_d + 1$  and if it is odd than  $w = \sigma_d$ . But window size should be greater than 3 time faster than  $\sigma_d$ . Otherwise we will not get the smooth image for abstraction. The effect of window have shown in figure 3.



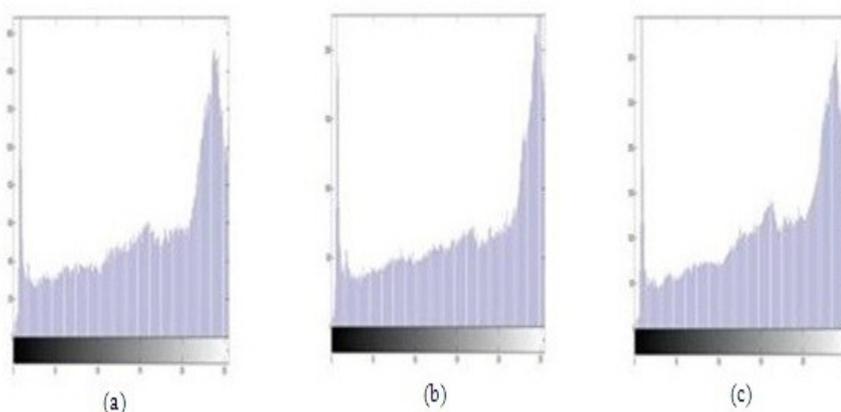
**Figure 3.** Comparative result of Fast O(1) kernel and shiftable kernel , (a) Fast O(1) (before scaling window size), (b) Shiftable bilateral (before scaling window size), (c) O(1) (after scaling window size) and (d) Shiftable bilateral (after scaling window size)

**After Adding Dog Edge**

After adding dog edge with both fast bigo and shiftable bilateral filter we can see the result in figure 4. But Image gets little bit blurred by shiftable kernel (figure 4(a)) so we have worked with fast O(1) filter for our cartoonizing purpose (figure 4(b)). Fast O(1) filter output is also very similar to winmoller output by quality and histogram analysis (figure 6). But we add some properties of shiftable kernel with the bigo 1 trigometric kernel.



**Figure 4.** Comparative result of Fast O(1) bilateral filtered image(a),winmoller abstracted image (b), shiftable bilateral filtered image (c)



**Figure 5.** Comparison by histogram of Fast O(1) bilateral filtered image(a),winmoller abstracted image (b), shiftable bilateral filtered image (c)

**Algorithm: Animation Making From Real Video**

Input: video\_frame,  $\sigma_d, \sigma_r, \max\_gradient, \min\_edge \text{ strength}, \sigma_{dog}, \phi_{dog}$

Return: Animated Image.

1. Set double(frame)/255;
2. Smoothing:  
 $B = \text{fast\_bilateral}(frame, \sigma_d, \sigma_r)$ 
  - I. If  $\sigma_d$  is odd then set  $w = \sigma_d$  else  $w = \sigma_d + 1$
  - II. Scale  $w, w = w * 3$ ;
  - III. Set  $T = \text{maxFilter}(frame, w)$ ;
  - IV. Set  $\gamma = \frac{\pi}{2T}$  and  $\rho = \gamma \sigma_r$
  - V. If  $\sigma_r > \gamma^{-2}$ , pick any large  $N$ . else set  $N = (\gamma \sigma_r)^{-2}$
  - VI. For  $0 \leq n \leq N$  set coefficient, spatial filter  $gn(x)$  and range filter  $hn(x)$ .
  - VII. Filter main frame with the  $gn(x)$  and  $hn(x)$ .
- Return filtered image
3. Get the Dog edge  
 $\text{DoG} = \text{dogFilter}(frame, \sigma_{dog}, \phi_{dog})$ 
  - I. Set  $\sigma_e = \sigma_{dog}$
  - II. Set  $\sigma_r = \sqrt{1.6} \sigma_{dog}$
  - III. Set  $S_{\sigma_e}$  and  $S_{\sigma_r}$  gaussian kernel.
  - IV.  $\tau = 0.98$ ;
  - V. if  $(S_{\sigma_e} - \tau * S_{\sigma_r}) > 0$  DoG=1
- Else DoG=  $1 + \tanh(\phi_{dog} * (S_{\sigma_e} - \tau * S_{\sigma_r}))$
- return, DoG;
4. Final=Add B from step 2 and DoG from step 4

## V. Future Work

Bilateral filter is the best filter for cartoonizing. Mean shift take more time than bilateral procedure. We have worked with the time consuming fact of this procedure to make my video processing faster. As we say in previous section that it has a true animation look but a few sequences blurred much and has a dim look. Again color goes more flat for using Trigonometric bilateral. The abstraction depends mainly on initial parameters. Some probabilistic method has been used in various image-processing techniques to set accurate parameters like particle filters, Hidden Markov Model. In future we want to work with those procedures to make the abstraction more exclusive.

## VI. Conclusion

Animation is an attractive and fascinating subject in this century. This research is based on this fascinating subject. We have proposed some concept and methodology for making cartoon animation from real video and make video with a batter procedure among themselves.

## References

### Journal Papers:

- [1]. C. Tomasi, R. Manduchi. Bilateral filtering for gray and color images. Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India, 1998.
- [2]. Danny Barash and Dorin Comaniciu. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Video Computing*, 22(1):73–81, 2004.
- [3]. H.Winnemöller, S.C.Olsen, and B.Gooch, “Real-time video abstraction,” *ACM Siggraph*, pp.1221–1226,2006.
- [4]. D.Comaniciu, P. Meer, Mean shift analysis and applications,Proceedings of the 1999 IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, p.1197.
- [5]. F. Durand and J.Dorsey, “Fast bilateral filtering for the display of high-dynamic- range images,” *ACM Siggraph*,vol. 21, pp. 257–266,2002.
- [6]. Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. Realtime video abstraction. To Appear in Proceedings of SIGGRAPH2006.
- [7]. Jan EricKyprianidis, JürgenDöllnerImage Abstraction by Structure Adaptive Filtering EGUKTheoryandPracticeofComputerGraphics(2008)
- [8]. Dorin Commenciu, Peter Mer Mean Shift: A robust approach toward feature space analysis, IEEE 2002
- [9]. HenryKang, Member, IEEE,SeungyongLee, Member, IEEE,andCharlesK.Chui, Fellow, IEEE, Flow-BasedImageAbstraction IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL.15, NO.1,JANUARY/FEBRUARY2009
- [10]. F.Porikli,“Constant time $O(1)$ bilateral filtering,”in Proc.IEEECnf.Comput. Vis.PatternRecog.,2008,pp.1–8.
- [11]. Q.Yang,K.-H.Tan,andN.Ahuja,“Real-time  $O(1)$ bilateral filtering,”in Proc.IEEECnf.Comput. Vis.PatternRecog.,2009,pp.557–564.
- [12]. B.Weiss,“Fast median and bilateral filtering,”in Proc.ACMSIG-GRAPH,2006,vol.25,pp.519–526
- [13]. K.N.Chaudhury,Daniel Sage,and M.Unser,“Fast  $O(1)$  bilateral filtering using trigonometric range kernels,”IEEE Transactions on Image Processing ,vol.20,no.12,pp.3376-3382,2011.
- [14]. G.Wyszeckiand W.S.Styles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*.Wiley, NewYork, NY,1982.