Local Laplacian Filters and some extensions

Zhong Ren[1, a]

[1]School of Computer Software, Tianjin University, Tianjin 300072, China;

[a]renzhongccc@163.com

Keywords: Laplacian pyramid, edge-aware, over-enhancement, haze removing

Abstract. The Laplacian pyramid was always used for image analysis. However, the Laplacian pyramid could also be used for edge-aware image processing. To get successful results, this method spent much less time than most of other methods such as anisotropic diffusion, neighborhood filtering, and specialized wavelet bases. This method was based on the way of iteration and process each pixel again and again. The more times this method computed the image in the iteration, the better the result was. While there was a disadvantage of this method: there existed over enhancement in some areas. To this drawback, My study improved this method, which led to image distortion. Besides, my method made it possible to process video. My research also found that this method could be used to remove haze in the image.

Introduction

It is a general consider that Laplacian pyramid is unable to represent edges well and should not be used for edge-aware operations. It can have bad effects on edge-preserving smoothing and tone mapping. A lot of techniques are invented to solve these problems, e.g. anisotropic diffusion [1], neighborhood filters [2], edge-preserving optimization [3], edge-aware optimization [4]. But to get satisfying results, these techniques spend lots of time because of additional complexity or need of post-processing. Therefore, Edge-aware Image processing with a Laplacian Pyramid is invented, which does not have these disadvantages.

According to the Gaussian pyramid [5], every Laplacian coefficient can be attributed to either a step edge or the detail layer. When using the Laplacian Pyramid to process image, this method uses range compression to lower the workload. For each pyramid coefficient, this method renders an intermediate signal, which consists of clipping high and low values of the original signal.

Given the Input image I, this method first computes its Gaussian pyramid $G[i]$. Then for all pixel(x,y), this method uses Gaussian pyramid to get its value g_0. And then this method decides the sub-region R_0 which needs to apply the remapping function. This method does downsample to each pixel one coefficient a time in different levels of the pyramid. After applying the function which will be explained later, this method create the intermediate output pyramid. After all coefficients of the output pyramid have been computed, this method collapse the output pyramid to get the final result.

My research tries lots of images using the code of this method and discover new areas that this method could be applied to. Besides on this, My study uses Matlab to improve the code, realizing human-computer interaction at the same time.

Method

Remapping Function for Gray-scale Images Here, this method uses a simple threshold σ_r to differentiate the details and edges. If the intensity variations are smaller than σ_r, this method considers them details, and edges while they are larger than σ_r. So, when using the remapping function r, the process has two conditions, when dealing with edges this method uses r_e and r_d when dealing with edges, such that $r(i) = r_d(i)$ if $|i - g_0| \leq r_e(i)$ and $r(i) = r_e(i)$

Otherwise. $r_d(i) = g_0 + sign(i - g_0)\sigma_r f_d(|i - g_0| / \sigma_r)$

where f_d is a smooth function mapping from [0,1] to [0,1] that controls how details are modified. It can easily be seen that the function depends only on the amplitude of the detail $|i-g_0|$ relative to the parameter σ_r.

$$r_e(i) = g_0 + sign(i-g_0)(f_e(|i-g_0|-\sigma_r)+\sigma_r)$$

where f_e is a smooth non-negative function defined over $[0,\infty]$.

Remapping Function for Color Images It is easy to adapt the remapping function for Gray-scale Images to Color images. We just need to know the details within a ball of radius σ_r centered at g_0 and edges as the colors outside it. Therefore, we have

$$r_d(i) = g_0 + unit(i-g_0)\sigma_r f_d(\|i-g_0\|/\sigma_r)$$

$$r_e(i) = g_0 + unit(i-g_0)[f_e(\|i-g_0\|-\sigma_r)+\sigma_r]$$

with $unit(v) = v/\|v\|$ if $v \neq 0$ and 0 otherwise

The application of this method can be concluded in two main aspects, which are detail manipulation and tone manipulation.

Detail Manipulation To modify the details of an image the authors define an S-shaped pointwise function for local manipulation of contrast by using a power curve $f_d(\Delta) = \Delta^\alpha$, where $\alpha > 0$ is a user-defined parameter. Values larger than 1 smooth the details out, while values smaller than 1 increase their contrast. And $f_e(a) = a$. To reduce the noise amplification when increasing the contrast of the details may cause. The process should limit the smallest Δ. So, when $\alpha < 1$, this method computes $f_d(\Delta) = \tau\Delta^\alpha + (1-\tau)\Delta$,where τ is a smooth step function equal to 0 if Δ is less than 1% of the maximum intensity, 1 if it is more than 2%.

Tone Manipulation In tone manipulation, this method defines the function $f_e(a) = \beta a$, where $\beta \geq 0$ is a user-defined parameter. For tone mapping, this method sets the filter with $\alpha \leq 1$ so that details are preserved or enhanced, and $\beta < 1$ so that edges are compressed. Then this method remaps the result to the range of [0,1] by offsetting its values to make its maximum 0, then scaling them. After that, this method multiplies the intensity by color ratios to add RGB channels to the output. In tone manipulation, both β and α control the local and global contrast.[6]

Extensions

My method uses the author's ideas to add details to videos one frame a time. My research first tests α , β, σ_r to find the suitable value for common images. And find that when $\alpha = 0.26$, $\beta = 0$, $\sigma_r = 0.41$, the result is most satisfying.

To solve the over enhancement of the sky, my method uses the user feedback to correct it. The user needs to mark the edge which they do not want it to be enhanced, in the sky of the case . And then input both this image and the original image to my modified program. The program will set $\alpha = 1$ to the user's marking parts which would not change the details. Then by processing the program my research will find the results more pleased. It occurs to me that maybe my research can apply the detail enhancement to image dehazing problems. So my method uses my modified program, changing the parameters and test some images which will gain good results too.

Results

Over enhancement

Fig.1 Comparison of the results of original method and improved method

The top-left is input image. Obvious over enhancement could be found in (top-right) bad image, which makes the image lack fidelity. After modified, (bottom-left) modified image is divided into two areas, the area with black is where my method does not need to process and the other area is where my method needs to process. As a result, (bottom-right) good result looks better than bad result.

Removing haze

Fig.2 Result of removing haze

In my experiment, my code are found that it could remove haze of an image. It can be seen that my method can solve the image rehazing problem to some point by enhancing the details. The result is acceptable, finding that both the four men in the image and the two cars are pretty much easy to see in the result comparing to the original one.

(left)The input image is filled with haze, and it is difficult to distinguish the objects of the image. After image processing with $\sigma_r = 0.4, \alpha = 0.25, \beta = 1$, my method gets a much more clear (right)

image. Thus, this method could be used for remove haze.

Conclusion

My method based on the original ones is very useful in multi-scale image analysis and editing. It is simple to understand and can be easily modified. My research has opened a new way to image dehazing. To some point my study overcomes the misinterpretation of the Local Laplacian Filters. However, the result can be better, which I believe my future work will try to improve it. I could make my method to overcome over enhancement better. My method may make image not good at some edges. And my method to make this code deal with a video could be used in the process of make a movie. I will combine my code and other methods that reduce the fog, and find if I could get better results.

References

[1] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1990, 12(7): 629-639.

[2] Tomasi C, Manduchi R. Bilateral filtering for gray and color images[C]//Computer Vision, 1998. Sixth International Conference on. IEEE, 1998: 839-846.

[3] Farbman Z, Fattal R, Lischinski D, et al. Edge-preserving decompositions for multi-scale tone and detail manipulation[C]//ACM Transactions on Graphics (TOG). ACM, 2008, 27(3): 67.

[4] Fattal R. Edge-avoiding wavelets and their applications[J]. ACM Transactions on Graphics (TOG), 2009, 28(3): 22.

[5] Burt P J, Adelson E H. The Laplacian pyramid as a compact image code[J]. Communications, IEEE Transactions on, 1983, 31(4): 532-540.

[6] Sylvain Paris, Samuel W. Hasinoff, Jan Kautz , 2011 Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid (Proc. SIGGRAPH)