

DETECTING A PATTERN IN A VIDEO STREAM AFTER DENOISING AND COLOR FADING PREPROCESSING

Andrei Danescu¹

ABSTRACT

This paper presents an algorithm for identifying a pattern in an image, by using the “cross-correlation” technique. In addition, it shows the difficulties of implementing the algorithm in a video stream context and the solutions applied to remedy these shortcomings. The modified “template matching” algorithm manages to process the image in real-time by using downscaling for restricting the search area.

Filtering noise, in real time, has applications in speech and image processing. Considerable interest has arisen in recent years regarding filtering in the wavelet transform domain. This technique has been effective in noise removal with minimum side effects on important features such as image details and edges. In this paper, the effectiveness of both soft and hard thresholding for desired detail levels has been demonstrated. Python implementation is proposed due to its simplicity.

This article presents an image conversion method from color space to grayscale, using Fourier transform on luminance and chrominance channels. The method keeps the image chromatic contrast, which could be lost in case of a simple luminance channel extraction. Frequency domain naturally offers the contrast values on all channels, in all spatial scales. Consequently, there are just arithmetic operations necessary to factorize chromatic differences in final intensity of the image, so the image processing speed is raised up considerably.

KEYWORDS: pattern matching, template matching, Fourier transform, luminance, color to grayscale, wavelet, Haar.

1. INTRODUCTION

A common problem in the image processing field is matching a certain pattern [1][2][3] in an image or a section of the image, called interest region. The proposed optimizations should be able to combine with existing methods [8]. Another common problem is pattern identification in multiple frames or images in order to perform alignment or merge operations [4][5][6]. This problem is very similar to the identification of a specific digital signal in the processing signals field, by using, for example, a “matched filter”.

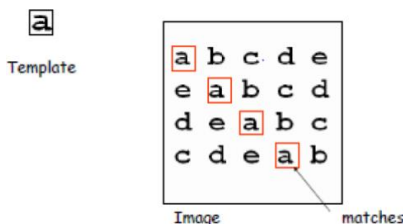


Figure 1. Matching a template in a target image

¹ Engineer, andrei.danescu@cti.pub.ro, "Politehnica" University of Bucharest, 060042 Bucharest, Romania

A frequently used solution for solving this problem is the “template matching”. This requires a pixel by pixel comparison of segments of the image against a pattern that contains the desired model.

A wavelet is a wave-like oscillation with amplitude that begins at zero, increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one might see recorded by a seismograph or heart monitor. Generally, wavelets are purposefully crafted to have specific properties that make them useful for signal processing.

Also, wavelets are a mathematical tool for hierarchically decomposing functions. They allow a function to be described in terms of a coarse overall shape, plus details that range from broad to narrow. Regardless of whether the function of interest is an image, a curve, or a surface, wavelets offer an elegant technique for representing the levels of detail present. This article is intended to provide people working in computer graphics with some intuition for what wavelets are, as well as to present one field of applications. In the first part, we discuss the simple case of Haar wavelets in one and two dimensions. In the second part, we will present a technique and also an algorithm for removing noise.

Most fading methods obtain good results for a specified data set, but are not only very difficult to implement, but also provide little flexibility in parameter control, or rely on very slow algorithms. Among these algorithms there are some which treat the fading problem as a minimization problem, algorithms that try to keep chromatic differences in fixed neighborhoods and algorithms which try to realize a global nonlinear mapping optimization of color image to grayscale or algorithms which try to code the color information inside the spatial information of the luminance channel [2].

The method presented in the following sections succeeds to offer good results, through contrast conservation in all characteristics steps present in transformed image using frequency domain [1] and real-time execution (assuming that the Fourier transform is also computed on the GPU).

2. TEMPLATE MATCHING

To achieve the pattern detection in an image, the template has to be discretely moved u steps towards x and y direction, then, the comparison is made on a rectangle with the size of the image template for each position (u, v). Following the comparison, a metric is computed to determine the best matches for the entire image. In case of a video stream, the calculation is performed at each frame (or once a number of frames to increase performance).

A common method to measure the similarity between image regions and the pattern is the correlation or the “cross-correlation”:

$$c = \sum_{x,y} f(x,y) * t(x,y) \quad (1)$$

To calculate the correlation response in each pixel (x, y) the following formula is used:

$$c(x,y) = \sum_{k=-W}^W \sum_{l=-H}^H f(x+k, y+l) * t(k,l) \quad (2)$$

where t is the template image with the size $[2W, 2H]$, and f is the image where the search is being made. The final result is a pixel for which $c(x, y)$ is maximum, indicating the left upper corner of the region that matched the pattern.

In that case, the correlation has a few drawbacks which make it impractical in detecting the patterns:

If the brightness/contrast of the image or pattern varies, the correlation may fail. For example, the correlation between the template and the corresponding interest area of the image may be lower than the correlation between the template and a much brighter area.

The range of values of $c(x, y)$ depends on the size of the searched pattern.

The correlation is not invariant to image intensity changes (e.g. stronger illumination).

A solution to this problem is the normalization of the correlation. This operation is performed at each step, by lowering the average and dividing by standard deviation.

So, the improved formula is:

$$c = \frac{\sum_{x,y} [f(x, y) - \bar{f}] * [t(x, y) - \bar{t}]}{(\sum_{x,y} [f(x, y) - \bar{f}]^2 * \sum_{x,y} [t(x, y) - \bar{t}]^2)^{1/2}} \quad (3)$$

Where $\bar{f}$ and $\bar{t}$ represent the pixel average of that pattern image.

The normalized correlation formula obtains results with a maximum value of 1. Generally speaking, it is required to suppress the non-maximal values by using a filter that finds the pixel with the maximum value from the current window and sets the other pixels values at 0. After calculating each pixels value, the maximum location has to be searched in the results matrix. In this case, the maximum corresponds to the brightest point of the image.

3. PATTERN DETECTION IMPROVEMENTS

Detecting patterns by normalized “cross-correlation” is an expensive operation in terms of computation, especially if multiple templates are used (rotation tolerance, scaling etc.). To speed up the detection process, the following technique can be applied:

Downsample both the image and the pattern by a factor of 2, 4, 8 or even more, depending on the image and pattern sizes, making sure not to lose too much information from the pattern.

Apply the pattern matching algorithm for the downsampled image-pattern pair and obtain an approximate pattern position in the searched image.

Determine a search window in the original image (full resolution), by specifying a pixel tolerance.

The algorithm is applied again by using the original template image and pattern, limiting the search at the previous step window.

4. HAAR WAVELETS

We set the stage here by first presenting the simplest form of wavelets, the Haar basis. We cover one-dimensional wavelet transforms and basis functions, and show how these tools

can be used to compress the representation of a piecewise-constant function. Then we discuss two-dimensional generalizations of the Haar basis, and demonstrate how to apply these wavelets to noise reduction.

In preparation for image denoising, we need to generalize Haar wavelets to two dimensions. First, we will consider how to perform a wavelet decomposition of the pixel values in a two-dimensional image. We then describe the scaling functions and wavelets that form a two-dimensional wavelet basis.

To obtain the standard decomposition of an image, we first apply the one-dimensional wavelet transform to each row of pixel values. This operation gives us an average value along with detail coefficients for each row. Next, we treat these transformed rows as if they were themselves an image and apply the one-dimensional transform to each column. The resulting values are all detail coefficients except for a single overall average coefficient.

5. WAVELET THRESHOLDING

Wavelet thresholding (first proposed by Donoho) is a signal estimation technique that exploits the capabilities of wavelet transform for signal denoising. It removes noise by killing coefficients that are insignificant relative to some threshold. It is proven that small coefficients of the wavelet keep the noise and small detail information while the big coefficients keep the details and important data of the image. Researchers have developed various techniques for choosing denoising parameters and so far there is no “best” universal threshold determination technique. In this article we analyze two thresholding techniques called hard thresholding and soft thresholding.

6. NOISE REMOVAL USING WAVELETS

As a support for this paper stands a python implementation of the Haar wavelet transform using the previous described algorithm. This was the easy part. The most difficult part is to determine the threshold value. For that I applied the following strategy. I have walk through the entire image with a 30x30 pixel window to determine the most uniform area of the image. I have computed the standard deviation for each window separately and I have chosen the one with the minimum standard deviation value. This area represents the information source for threshold calculation. I have marked this area in each wavelet level. The window gets smaller as the depth of the wavelet increases. Also only the window in the bottom right corner is square, for each depth level, the rest of them are not.

Starting from this point, I implemented 2 methods for the threshold calculation.

In the first one I consider the threshold value for each section equal with the standard deviation of the window in that section (the window is previously identified with the described method).

In the second method, the mean value of the window is considered as threshold for that section of the wavelet.

The third method is proposed by [16] and it is dedicated for salt and pepper noise. They say that one must apply a different strategy for removing salt and pepper noise.

7. FADING IN FOURIER DOMAIN

The first step is RGB color space to CIE Lab transformation. CIE space is suitable for this method because it offers a clear separation of luminance and chrominance channels, allowing for a better control over contrast differences.

The second step requires applying the Fourier transform on each channel (L, a and b) and applying the following formulas for chromatic contrast introduction in grayscale values of G1 image, in Fourier domain. G1 is obtained using inverse Fourier transform, as following:

$$\tilde{G}_1 = IF(\tilde{E})$$

$$\tilde{E} = H(\hat{L}, \hat{a}, \hat{b})$$

H calculates grayscale intensity modified in Fourier domain, E, using Fourier transform values applied over luminance and each of two chrominance channels. H is calculated on each frequency using the next relation:

$$H(\hat{L}, \hat{a}, \hat{b}) = (1 - \theta) \cdot \hat{L} + \theta \cdot (\phi \cdot \hat{a} + (1 - \phi) \cdot \hat{b})$$

where Θ controls chromatic contrast level of influence in the final result, and Φ is a coefficient that determines the relative contribution of chrominance channels a and b. In the above equation, the Fourier values and coefficients are dependent on frequency.

8. IMPROVEMENTS

Automatic parameter control

We can calculate Θ using the difference between RGB channels and luminance, thus obtaining the relative loss of conversion from one chromatic space into other. This difference can be represented as follows:

$$\theta = \frac{|\bar{R}| + |\bar{G}| + |\bar{B}| - |\bar{L}|}{|\bar{R}| + |\bar{G}| + |\bar{B}|}$$

Where $||$ represent the spectral complex values, and $\bar{R}$, $\bar{G}$, $\bar{B}$ are the Fourier transformed of RGB channels.

Similarly, the coefficient Φ can be calculated as the relative proportion between the two chrominance channels:

$$\phi = \frac{\hat{a}}{|\hat{a}| + |\hat{b}|}$$

The two coefficients can be calculated automatically at each frequency. Experimentally, good results were obtained using the average coefficients calculated at each frequency. In this manner, the results satisfy global consistency (i.e. pixels with the same color are mapping in the same gray value, due to linearity of formula which calculates H).

Grayscale values normalization

For a better visualization of the result, the final image can be normalized for using all values between 0-255, whereas CIE Lab space does not have such a large range.

Application of a homomorphic filter

The result of filtering with a homomorphic filter offers a better contrast and a uniform distribution of image luminosity. The method requires the separation of lighting component from the reflectance component, the aim being suppressing the contribution of light (low frequency) in favor of reflectance, which offers more relevant chromatic information.

The necessary steps for filter application are the following:

1. **Image logarithmation – necessary to transform the relation between lightning and reflectance from multiplication to addition, for linear filter application in frequency domain.**
2. **Low frequencies removing – high-pass filter**
3. **Decreasing the result from logarithmated image – removing the lightning low frequency component.**
4. **Exponential item application and obtaining the final image.**

9. RESULTS

The obtained results were very good, the pattern detection using improved algorithm running at the original frame-rate of the video camera (i.e. with no tracking):

- Simple video stream 1280x720 => 13 FPS
- Video stream with the pattern detecting by using the trivial algorithm => 8-9 FPS
- Video stream with the pattern detecting by using the improved algorithm => 12-13 FPS

This multiresolution template matching method can be used alongside an optimization using partial sums, described in [7], to provide real time results for traffic [10] or surveillance systems[12], like [9], which deduce moving objects, create a template from them and follows them through the video. Optimizations relative to clustering can be obtained by using filters and edge detection, in order to accurately take out the moving tracked object from the video [11].

Wavelets are a useful tool for image processing. From the results presented I can say that wavelet thresholding is a good method to remove noise. For salt and pepper noise one must apply different strategy and the results are not the expected ones.

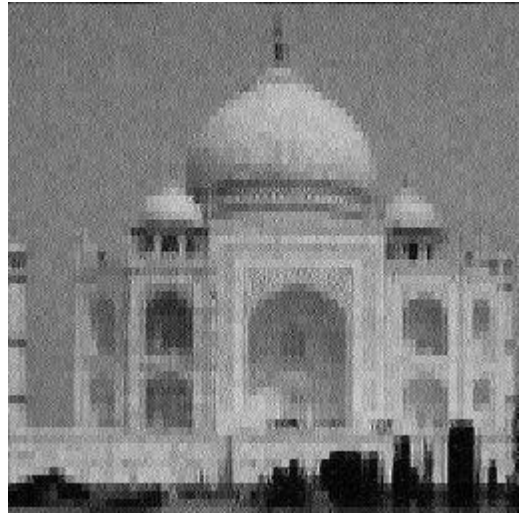
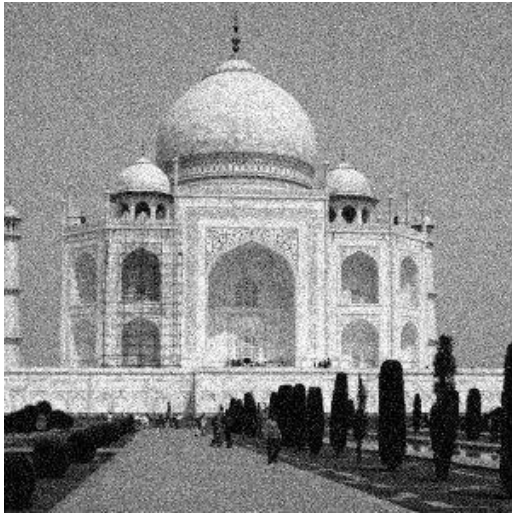


Figure 2. Original Image and result of method 1 applied on original

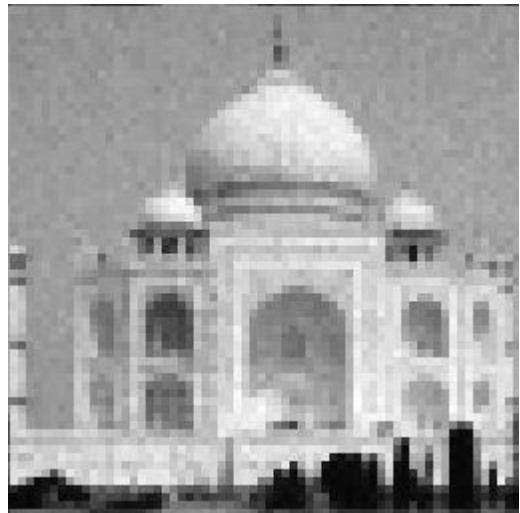
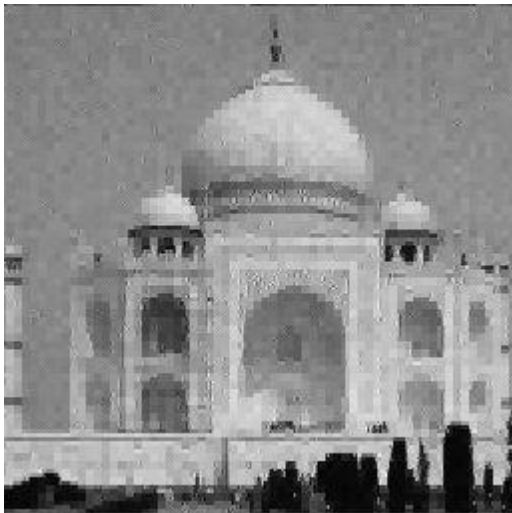


Figure 3. Result of method 2 and 3 applied on original

The results show that different close-by colors mapped on different grayscale components, calculations having a low computational impact. The purpose of the algorithm was to make all elements in the color image distinguishable in the grayscale equivalent, not being interested in the perceptual similarity [3][4]; even thou, the transformations manage to preserve the global consistency, making the image naturally looking.

These results can be used as preprocessing step for binarization [5][6], in order to preserve more information.



Figure 4. Original, Luminance, Frequency fading

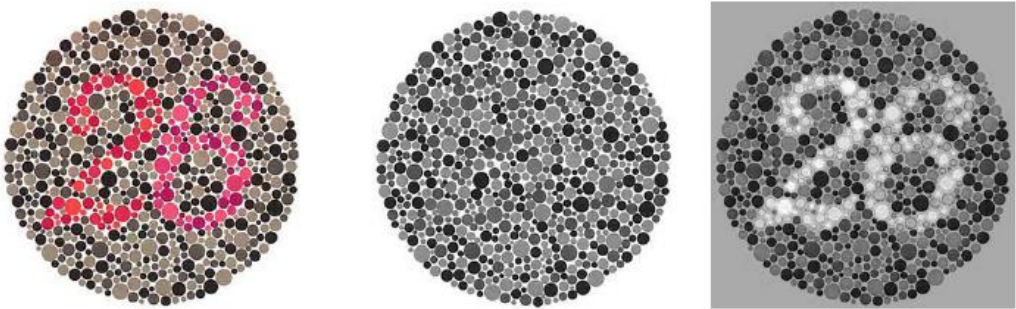


Figure 5. Original, Luminance, Frequency fading

10. ACKNOWLEDGEMENTS

The authors would like to thank Mihai Dobre for his support and assistance with this paper.

11. REFERENCES

- [1] Costin-Anton Boiangiu, Bogdan Raducanu. "Line Detection Techniques for Automatic Content Conversion Systems". WSEAS Transactions on Information Science, Applications, Issue 7, Volume 5, pp. 1200 – 1209, July 2008
- [2] Costin-Anton Boiangiu, Bogdan Raducanu. "Robust Line Detection Methods". Proceedings of the 9th WSEAS International Conference on Automation and Information, WSEAS Press, pp. 464 – 467, Bucharest, Romania, June 24-26, 2008
- [3] Costin-Anton Boiangiu, Bogdan Raducanu. "Effects of Data Filtering Techniques in Line Detection". Annals of DAAAM for 2008, Proceedings of the 19th International DAAAM Symposium, pp. 0125–0126, Trnava, Slovakia, October 22-25, 2008
- [4] JP Lewis, "Fast normalized cross-correlation", Vision interface, volume 10, issue 1, pp. 120-123
- [5] Manolache Florentina Cristina, Sofron Angela, Stanciu Adelina, Costin-Anton Boiangiu – "Study on the Optimizations of the Hough Transform for Image Line Detection", The Proceedings of Journal ISOM Vol. 7 No. 1, pp. 141-155, ISSN 1843-4711, 2013
- [6] N. Amamoto and A. Fujii, "Detecting Obstructions and Tracking Moving Objects by Image Processing Technique", Electronics and Comm. Japan, Part 3, vol. 82, no. 11, pp. 28-37, 1999.
- [7] B. Gloyer, H.K. Aghajan, K.Y. Siu and T. Kailath, "Video-Based Freeway Monitoring System Using Recursive Vehicle Tracking," Proc. SPIE Symp. Electronic Imaging: Image and Video Processing, 1995

- [8] S. Jabri, Z. Duric, H. Wechsler and A. Rosenfeld, "Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information," Proc. Int'l Conf. Pattern Recognition, pp. 627-630, 2000.
- [9] N. Rota and M. Thonnat, "Video Sequence Interpretation for Visual Surveillance," Proc. IEEE Workshop Visual Surveillance (VS '00), pp. 325-332, 2000.
- [10] Ye Zhao, Zakiya Tamimi, "Spectral Image Decolorization", Advances in Visual Computing, Lecture Notes in Computer Science Volume 6454, 2010, pp 747-756
- [11] Ricardo L. de Queiroz, Karen M. Braun, "Color to Gray and Back: Color Embedding Into Textured Gray Images", IEEE Transactions on Image Processing, 15(6), pp.1464–1470, 2006
- [12] Costin-Anton Boiangiu, Alexandru Victor Ștefănescu, "Target validation and image calibration in scanning systems", in Proceedings of the 1st International Conference on Image Processing and Pattern Recognition (IPPR '13), Budapest, Hungary, December 10-12, 2013, pp. 72-78
- [13] M. Čadík, Perceptual Evaluation of Color-to-Grayscale Image Conversions, Computer Graphics Forum, Vol. 27, 2007, pp. 1745-1754
- [14] Costin-Anton Boiangiu, Alexandra Olteanu, Alexandru Victor Ștefănescu, Daniel Rosner, Alexandru Ionut Egner (2010). „Local Thresholding Image Binarization using Variable-Window Standard Deviation Response” (2010), Annals of DAAAM for 2010, Proceedings of the 21st International DAAAM Symposium, 20-23 October 2010, Zadar, Croatia, pp. 133-134
- [15] Eric J. Stollnitz, Tony D. DeRose, David H. Salesin, "Wavelets for Computer Graphics: A Primer"
- [16] Rohtash Dhiman and Sandeep Kumar, "An Improved Threshold Estimation Technique for Image Denoising Using Wavelet Thresholding Techniques"
- [17] Rami Cohen, "Signal Denoising Using Wavelets", Project Report 2012, Department of Electrical Engineering, Technion, Israel Institute of Technology