ROTATION OF FOVEATED IMAGE IN THE WAVELET DOMAIN

Hang Yu *Vu-Thanh Nguyen* *Ee-Chien Chang*

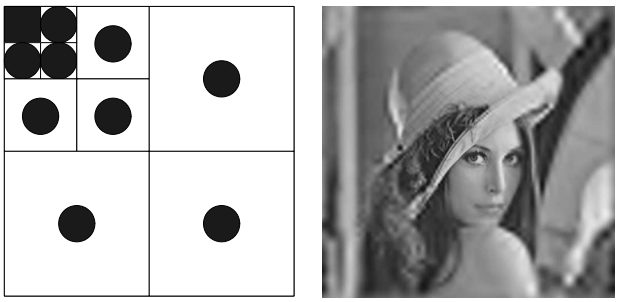
School of Computing,
National University of Singapore
{yuhang, nguyenvu, changec}@comp.nus.edu.sg

ABSTRACT

An advantage of wavelet transform is its efficiency in representing natural images, that is, a natural image can be accurately represented by only a small number of retained wavelet coefficients. It is interesting to know whether some common image operations, e.g. rotation, can be performed very fast in wavelet domain. Preferably, the running time should depend only on the number of retained coefficients, not the size of the original image. However, it is not clear how this can be achieved. In this paper, we consider rotation, and images with a special structure: foveated images. Wavelet coefficients of a foveated image vanish outside an arrangement of circles. We exploit this structure to derive algorithms that accurately approximate rotation. The running time is $\Theta(m)$ where m is the number of retained coefficients. Experiments show the accuracy of the approximation.

1. INTRODUCTION

Wavelet transform is known to be efficient in representing natural images, that is, a natural image can be accurately represented by only a small number of retained wavelet coefficients. Since the amount of data being processed is reduced, it brings forth the possibility of speeding up performance by operating only on the retained coefficients. Some works have exploited this characteristic for specific task, e.g. feature extraction [1]. Preferably, the running time should depend only on the number of retained coefficients. However, for some common image operations, e.g. rotation, it is not easy to achieve the speed-up. This is partly due to the "shift sensitive" characteristics of wavelet transform, i.e. a small shift in a signal generates unpredictable changes in its discrete wavelet transform coefficients. The shift insensitive complex wavelets could be employed to handle the translation [2] but it is not clear how they can be applied to handle slightly more complicated image operations like rotation and shearing. Also note that in general, even if the original image has many zero coefficients, the operated image could have few or none zero coefficients. Therefore, it is impossible to have both fast and exact algorithm.



(a) (b)

Fig. 1. (a) the mask when the fovea is at the center; (b) foveated image whose mask has radius of 30 and locates at center.

In this paper, we focus on rotation, and a particular type of image: foveated image. The resolution of a foveated image is highest at a point (fovea) and falling off away from the fovea. It is corresponding to our biological visual system, which has a space-variant nature where the resolution is high in the center (fovea) but decreasing towards the peripheral. As the biological visual system is highly effective, the space-variant nature of foveated image has inspired the design of many vision and imaging systems [3, 4]. An approximate foveated image can be obtained from the original image by multiplying the discrete wavelet transform (DWT) of the image by a predetermined 0-1 mask followed by the inverse discrete wavelet transform (IDWT) [4]. The mask indicates which coefficients are to be retained. Fig. 1(a) shows the mask when the fovea is at the center and Fig. 1(b) shows a foveated image (512x512) whose mask has radius of 30 and locates at the image's center. Observe that only a small number of coefficients are retained in the foveated image. More accurate approximation can be achieved by using a smoother mask, rather than the 0-1 mask [4].

We exploit the special structure of foveated image to derive fast algorithms that directly process foveated image in the wavelet domain. The running time is proportional to the

0-7803-8554-3/04/$20.00 ©2004 IEEE.

number of coefficients retained. That is, we obtain a speed-up from $\Theta(n^2)$ to $\Theta(m)$ where n^2 is the number of pixels, and m is the number of coefficients retained. We propose two algorithms. The second algorithm is more accurate but the first algorithm is faster by a constant factor. Thus, theoretically, both algorithms run in $\Theta(m)$ time. However, it is instructive to present both algorithms.

2. THE PROBLEM AND PROPOSED METHODS

2.1. The problem

Consider a foveated image I_1 and its rotated version I_2. The input of our problem is W_1, the wavelet coefficients of I_1, and the output is W_2, an approximation of the wavelet coefficients of I_2.

A direct method requires three steps: (i) reconstructing the foveated image I_1 from W_1, i.e. applying IDWT on W_1, (ii) rotating the image I_1 to obtain I_2 and (iii) applying DWT on I_2 to get W_2.

Assuming that the image size is $n \times n$ pixels, each of the above steps requires a running time in the order of $\Theta(n^2)$, which is considerable when the image is large. This is especially so when higher order of interpolation is used during rotation. Recall that the W_1 can be represented by small number of coefficients. Let m the number of coefficients retained during the foveation. In Fig. 1, m is the number of 1's in the mask. We want to design an algorithm that depends only on m.

Our algorithms are based on the observation that for a foveated image, the radius of the circles in the mask are the same across every level. Furthermore, the centers of the circles all correspond to the same location, which is the fovea, in the spatial domain. Consequently, we can focus the operation on the area around the fovea and significantly reduce the running time.

2.2. Algorithm 1

Consider the sub-bands in the wavelet transform. We call the i-th level LH, HL and HH high frequency sub-band h_i (see Fig. 2(a)), and l_i the LL sub-band of the i-th level. Thus l_3 can be reconstructed from l_1, h_1 and h_2. For a foveated image, its non-zero coefficients are concentrated in the arrangement of circles of radius r shown in Fig. 1(a). We write $h_{i,r}$ as the coefficients retained in the sub-bands h_i. Thus, $h_{i,r}$ contains coefficients in 3 circles of radius r, where each circle corresponds to the LH, HL and HH sub-band.

Fig. 3 shows the steps of Algorithm 1 when the number of levels is 3. It is easy to generalize to any number of levels. In this figure, $mask(r_0)$ is an operation that applies a mask of radius r_0 in the sub-band. For example, after l_3 is applied a $mask(2r)$, coefficients at a distant greater than $2r$ from the fovea will be removed (or equivalently, set to zeros). The operation "rotate" is the usual image rotation. We are not concerned with the interpolation method used during rotation. In our experiment, we employ bicubic interpolation.

At level 1, the inputs of Algorithm 1 are l_1 and h_1 of the foveated image. We apply IDWT on l_1 and h_1 to get l_2. Since the energy of l_2 is concentrated in the circle of radius $2r$, to achieve speed-up, a circular mask of radius $2r$ is applied to obtain $l_{2,2r}$. Next, $l_{2,2r}$ is rotated to produce a rotated K_2, follow by DWT to give the sub-bands L_1 and H_1. Similarly, the energy of each of these sub-bands is concentrated in a circle of radius r. Hence, we apply a mask of radius r on H_1 to produce $H_{1,r}$. At level 2, similar process is repeated, except that the input is $l_{2,r}$. Note that masking is performed a few times in the above steps. Although masking reduces accuracy, it is necessary to make sure that the data size is small. After all, the values of the discarded coefficients are small. This recursive process is carried out until reaching the final level.

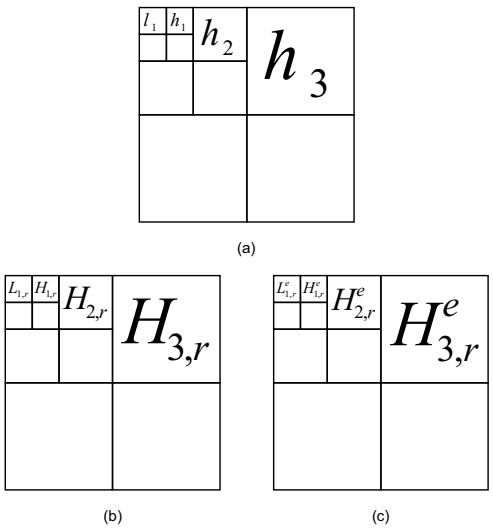


Fig. 2. Wavelet transforms of: (a) original foveated image; (b) foveated image rotated by Algorithm 1; (c) foveated image rotated by Algorithm 2.

Note that each of DWT, mask(), and rotations can be done in $\Theta(r^2)$ time and there are only N levels. Thus the total time require is $\Theta(Nr^2)$, which is $\Theta(m)$.

2.3. Algorithm 2

Algorithm 2 is a modified version of Algorithm 1. The main observation is the following. Let $K_{i,r}$ be the coefficients obtained by masking K_i with the circle of radius r. Ideally, $K_{i,r}$ should be the same as L_i. However, this is not the case due to the combined effects of rotation and wavelet transformation. L_i is more accurate because it is computed in

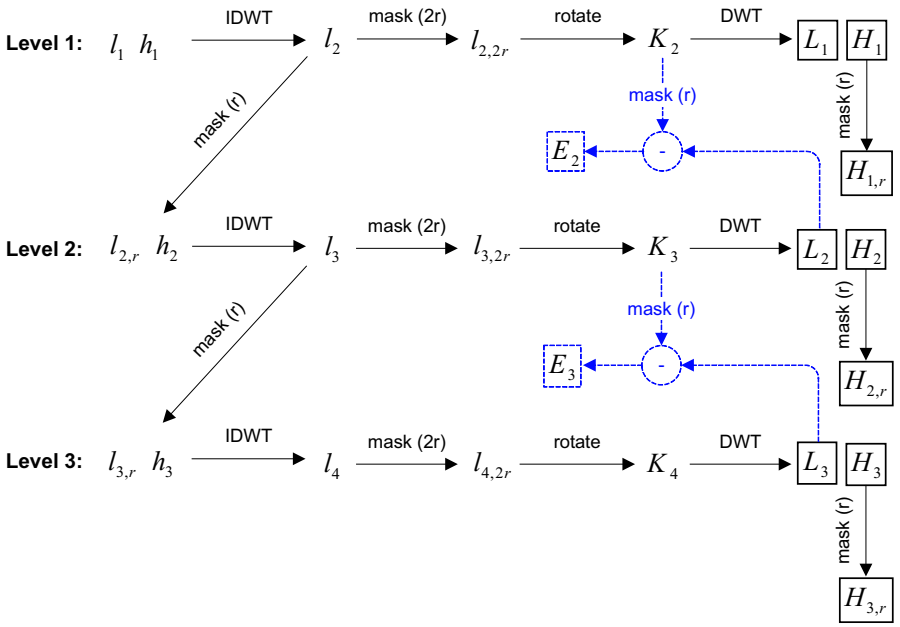


Fig. 3. Algorithm 1 to rotate foveated image directly in wavelet domain (dotted lines are for Algorithm 2).

the higher level. Therefore, if we retain L_{i-1}, H_{i-1} which are calculated from K_i, in the final output, the LL sub-band at level i will be the imprecise K_i.

In Algorithm 2, we introduce a step after L_i is obtained. We first compute the error $E_i = L_i - K_{i,r}$. Next, DWT is applied on E_i. The wavelet transform of E_i are then added to L_1, H_1, H_2, ..., H_{i-1}. To ensure $\Theta(m)$ computation, we approximate the DWT by restricting the coefficients in each sub-band to be in the circle of radius r.

3. EXPERIMENTAL RESULTS

We implement our proposed algorithms using several different images: "Lena" (512x512), "Mandrill" (512x512), "Cameraman" (256x256), "Peppers" (256x256) and "Barbara" (256x256). The performance of accuracy achieved is measured by Normalized Mean Square Error (NMSE). The wavelet filter used is the biorthogonal 7/11.

Fig. 4 shows the rotated image by different methods when the degree of rotation is 45, and the radius of the mask is 15. To clearly show the differences between images, we present their zoom-in versions whose focuses are around the fovea. We denote J_1 the image obtained by directly rotating the foveated image I_1 in the space domain, which is showed in Fig. 4(a). We denote J_2 the image obtained by a straightforward algorithm: applying DWT on J_1, $mask$ on the wavelet coefficients and IDWT on the wavelet coefficients in the mask. The zoom-in of J_2 is shown in Fig. 4(b). We denote A_1 and A_2 the images obtained by our proposed Algorithm 1 and Algorithm 2, respectively. The zoom-in of A_1 and A_2 are shown in Fig. 4(c) and Fig. 4(d). It is obvious that the image A_2 is more accurate than the image A_1 and comparable to J_1.

To compare the differences among A_1, A_2, J_1 and J_2, we use Normalized Mean Square Error (NMSE), which is the mean square error normalized by the energy of J_1. Fig. 5(a) shows the NMSE as the rotation degree increases and Fig. 5(b) shows the NMSE as the radius of the mask increases. Because of the rotation and wavelet transform, there may exist spike values on the boundaries of rotated images, which lead to inaccurate NMSE values. Therefore, we ignore the image boundary by comparing only the region within the inscribed circle of the image boundaries. The displayed NMSE values are the average results of 5 different images which we mentioned above.

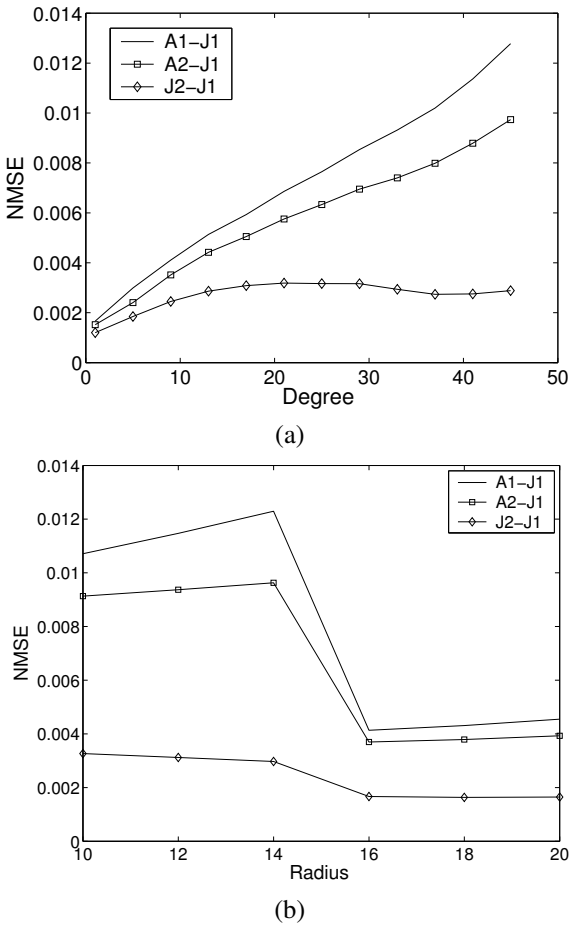
We can see that the NMSE increases as the rotation degree increases, and decreases as the radius of the mask increases. It is because the larger the rotating degree is, the larger the number of non-zero coefficients are created in wavelet domain. It means that while the radius is unchanged, the number of non-zero coefficients being discarded (by our algorithms) increases and consequently NMSE increases. Similarly, when the radius increases, more number of coefficients are retained and we obtain more accurate results. The NMSE difference showed in Fig. 5 also confirms what we have visually concluded from Fig. 4 about the accuracy of the proposed algorithms. Note that the graph of $(J_2 - J_1)$ indicates the best approximation ones can achieve.

(a) (b) (c) (d)

Fig. 4. Rotated images: (a) J_1 obtained by rotating the foveated image Fig. 1(b); (b) J_2 obtained by applying DWT on J_1 and IDWT on the coefficients in the mask; (c) A_1 obtained by Algorithm 1; (d) A_2 obtained by Algorithm 2.

4. CONCLUSION

Although wavelet transforms have been studied for more than a decade, there are few researches on direct and efficient manipulation in the wavelet domain. In this paper, we propose two algorithms that directly rotate foveated images in the wavelet-based compressed domain. The running time of our algorithms depends only on the number of retained coefficients, and this is considerably faster than the straightforward algorithm. While the proposed approximation algorithms are more efficient, experimental result shows that the approximation is also accurate. Here is a possible deployment of our algorithms: Suppose a method can efficiently extracts features of foveated images directly from its wavelet coefficients, and we want to extend it to rotated images. This extension can be done by first applying our algorithm, follows by the original efficient extraction method. In the future, it would be interesting to explore similar techniques for other operations.



(a)

(b)

Fig. 5. Performance ratios: (a) NMSE as the rotating degree increases; (b) NMSE as the mask's radius increases.

5. REFERENCES

[1] Chang S.F., Smith J., "Extracting multi-dimensional signal features for content-based visual query," *SPIE Visual Comm. and Image Processing,* Taipei, May 1995.

[2] Fernandes F.C.A., van Spaendonck R.L.C., Burrus C.S. "A new framework for complex wavelet transforms," *IEEE Trans. on Signal Processing,* 51(7):1825-1837, July 2003.

[3] Burt B.J., "Smart sensing within a pyramid vision machine," *Proc. of the IEEE*, 76(8):1006-1015, 1988.

[4] Ee-Chien Chang, Stephane Mallat and Chee Yap, "Wavelet foveation," *J. Applied and Computational Harmonic Analysis,* 9(3):312-335, October 2000.