REPRESENTATION OF IMAGE BY LOCAL SYMMETRY DECOMPOSITION

[1]Soo-Chang Pei(貝蘇章) [2]Hsin-Ying Tasi (蔡馨瑩) [3]Chiou-Shann Fuh (傅楸善)

[1] Digital Image Processing Laboratory, Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan
E-mail: cvul35jc@gmail.com

ABSTRACT

We present a new technique for extracting local features from images, based on detecting and representing local symmetries. We introduce the even/odd symmetries. Then, we generalize the even/odd decomposition by concentrating the energy on either the even or the odd part by optimally placing the center of symmetry. Local symmetry intervals are thus located. An image is segmented into adjacent, variable-width, and variable-height where strong symmetry characteristics exist. Finally, we detect the even/odd symmetry in the natural image, and compute the Peak Signal-to-Noise Ratio (PSNR) and compression ratio between the original and compressed image.

Keywords — Symmetry, 2-D segmentation, Even/Odd representation, Image compression

1. INTRODUCTION

First, in image processing, image classification and labeling are very important. Applications can be divided into anomaly detection [1], the event classification [2], and others [3]. Under normal circumstances, the method consists of the following steps, such as the random nature of the measurement, the extraction function, looking for patterns, and correlations of the measurements. [4, 5, 6]. In general, these methods only indirectly or do not at all involve exploiting inherent symmetry properties that may be present in many signals, such as local periodic waveform, changes in brightness in a specific direction along natural images. In addition, the use of the inherent symmetry of the signal can be improved signal coding efficiency analysis. The image matching using the local symmetry features can be referred in [7]. Their main contribution is a simple method for densely characterizing local symmetries across image and scale space, and defining a corresponding local symmetry feature detector and descriptor.

Second, even/odd decomposition consistent with the simplest decomposition is proposed for the 1-D signal manipulation [8]. In fact, due to the resulting intuitive geometrical interpretation and the parity preservation of the Fourier transform, even/odd decomposition is omnipresent in signal processing. The compression of the even/odd decomposition in 1-D signal, they prove it by showing how individually compressing those intervals achieves better performance than compressing the whole signal using a linear transformation of the data [9]. Fourier representations have been conceived and lead to sparse representation of the information, hence improving compression. This has turned out also useful for tasks such as denoising [10].

We present a new method and represent the basis of an image on its local symmetry around the point of the even/odd decomposition effectively. Thus, an image is segmented into variable height and width blocks in which strong symmetrical characteristics exist. At the end, we detect the symmetry of blocks in the natural pictures, and identify the block whether is even or odd, and compute the Peak Signal-to-Noise Ratio (PSNR) and compression ratio between original image and compressed image.

The rest of the paper is organized as follows. We start to introduce the related work in Section 2. In Section 3, we derive a method for finding the globally optimum symmetry point for a 2-D continuous-time, finite support signal, in the process extending the definition of even/odd decomposition to arbitrary points. This analysis is extended to digital sequences in Section 4. Then, Section 5 presents the algorithm to segment a discrete segmental block into multiple near-symmetric sub-blocks, using the theoretic approach described in the previous sections to remove overlapping segmental blocks. Experimental results show that the PSNR and compression ratio between original image and compressed image, and simultaneously we detect the blocks which have the even or odd symmetry on the natural image in Section 6. Conclusions are finally drawn in Section 7.

2. RELATED WORK

Researchers in computer vision have made significant progress in representing and detecting symmetries in images and other types of data (Liu et al. [11] present an excellent survey). However, there has been relatively little work on using local symmetries as explicit features for matching tasks. In this case, they use various symmetries, rather than repetitions, as cues, and use these to define both a feature detector 1 and a descriptor. Schindler et al. [12] also detect and match repetitive patterns on building facades, and Wu et al. [13] detect large repeating elements of facade images for use as robust support regions for computing features. In contrast, we do not make hard decisions about symmetric structures, but instead compute many local features based on a softer definition of a symmetry score. Loy and Zelinsky also propose a feature detector based on radial symmetries [14], using a fast radial symmetry transform that accumulates votes for symmetric regions from gradient information. In an earlier approach, Reisfeld et al. [15] used a similar voting scheme to detect interest points using radial symmetries via a generalized symmetry transform.

Kovesi observed that local bilateral symmetry in image intensity relates to the response of filters of different phases [16], and later extended this concept of "phase-congruency" for detecting features such as edges and corners [17]. Di Gesu et al. proposed the discrete symmetry transform [18], based on axial moments related to the medial axis transform. Finally, we present a new technique for extracting local features from images, based on detecting and representing local symmetries. Therefore, we generalize the even or odd decomposition by concentrating the energy on either the even or the odd part by optimally placing the center of symmetry.

3. GLOBALLY OPTIMAL CONTINUOS SYMMETRY

We consider the spatial coordinate (x,y), finite-energy of a 2-D signal $f(x,y)$. In short, we assume without loss of generality that the 2-D signal has finite support, for example, with support $[-L_x,L_x]$ and $[-L_y,L_y]$. The parity decomposition of a 2-D signal states that $f(x,y)$ can be expressed as the sum of its even and odd parts, respectively $f_e(x,y)$ and $f_o(x,y)$, given by:

$$f_e(x,y)=\frac{f(x,y)+f(-x,-y)}{2}$$
$$f_o(x,y)=\frac{f(x,y)-f(-x,-y)}{2} \qquad (1)$$

The energy E is defined as the squared Euclidean norm of the signal $f(x,y)$ and we can see in Eq. (2)

$$\begin{aligned}E&=\int_{-L_x}^{L_y}\int_{-L_x}^{L_y}|f(x,y)|^2dxdy\\&=\int_{-L_x}^{L_y}\int_{-L_x}^{L_y}|f_e(x,y)+f_o(x,y)|^2dxdy\\&=\int_{-L_x}^{L_y}\int_{-L_x}^{L_y}|f_e(x,y)|^2dxdy+\int_{-L_x}^{L_y}\int_{-L_x}^{L_y}|f_o(x,y)|^2dxdy\\&=E_e+E_o \qquad (2)\end{aligned}$$

Because the even and odd parts are orthogonal, Eq. (3) is set up. An example of decomposition in a 1-D signal is shown in Fig. 1.

$$\int_{-L_y}^{L_y}\int_{-L_x}^{L_x}f_e(x,y)f_o(x,y)dxdy=0 \qquad (3)$$

The new representation does not increase the support needed to reconstruct the original 2-D signal. In fact, the parity property of the even and odd parts implies that even if they both have the same support of the original 2-D signal. For instance, their part is informative and thus sufficient to describe the entire 2-D signal. If the original 2-D signal $f(x,y)$ has even or odd symmetry, almost all of its energy will be carried by its even or odd part energy.

Under the circumstances, the parity decomposition as defined in Eq. (1) has no effect even if the center of symmetry of the original 2-D signal is not in the origin. The reason is that the parity decomposition as defined above only considers the parity characteristics with the origin $(x,y)=0$. Thus, let us extend the definition of parity decomposition to allow an arbitrary symmetry point $x=x_0$ and $y=y_0$. In this case, we have:

$$f_e(x;x_0,y;y_0)=\frac{f(x,y)+f(2x_0-x,2y_0-y)}{2}$$
$$f_o(x;x_0,y;y_0)=\frac{f(x,y)-f(2x_0-x,2y_0-y)}{2}$$
$$f(x,y)=f_e(x,y)+f_o(x,y) \qquad (4)$$

and we fall back to the standard case by setting $(x_0,y_0)=(0,0)$. Still assuming that the original 2-D signal $f(x,y)$ has support $[-L_x,L_x]$ and $[-L_y,L_y]$, after even/odd decomposition according to (x_0,y_0) the non-informative support on both $f_e(x,y)$ and $f_o(x,y)$ is no more L_x and L_y but $L_x+2|x_0|$ and $L_y+2|y_0|$. However, the informative support is still L_x and L_y because the added support simply mirrors part of the 2-D signal confined in the informative support. As a result, the price to pay for searching for the best of the energies E_e and E_o is just increased complexity using the same support L_x and L_y . Note that Eq. (2) for the computation of energies is still valid, provided that the extremes of integration are changed to reflect the increased support of the even and odd 2-D signals.

We find the global optimal symmetry point (x_0,y_0) , that represents the position of the space for which there is a maximum energy between the even and odd parts. Since the energies E_e and E_o are function of the parameter (x_0, y_0), but still constrained by Eq. (2) to have sum E, both possess the same extrema points. We find that the globally optimal symmetry point (x_0, y_0) corresponds to the extrema of either E_e or E_o.

In order to find (x_0, y_0), let us concentrate on the extrema points of the energy of the even part. Its definition is written as follows :

$$E_e(x_0,y_0)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|f_e(x,y)\right|^2 dxdy$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|\frac{f(x,y)+f(2x_0-x,2y_0-y)}{2}\right|^2 dxdy$$

$$=\frac{1}{4}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|f(x,y)\right|^2+\left|f(2x_0-x,2y_0-y)\right|^2 dxdy$$

$$+\frac{1}{4}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)f(2x_0-x,2y_0-y)dxdy$$

$$=\frac{1}{2}E+\frac{1}{2}f*f(x_0,y_0)$$

$$E_o(x_0,y_0)=\frac{1}{2}E-\frac{1}{2}f*f(2x_0,2y_0) \qquad (5)$$

Thus, candidate extreme points can be determined by convolving the signal with the derivative of its conjugate, finding the zero-crossing points and then dividing by 2. From Eq. (5), we find a local minimum for E_e corresponding to a local maximum for E_o and vice-versa. To find the global minimum between E_e and E_o it is insufficient to look for the global minimum for E_e but one must consider as well its global maximum (i.e. the global minimum of E_o). Point (x_0, y_0) is then declared to correspond to the location that leads to the global minimum between E_e and E_o.

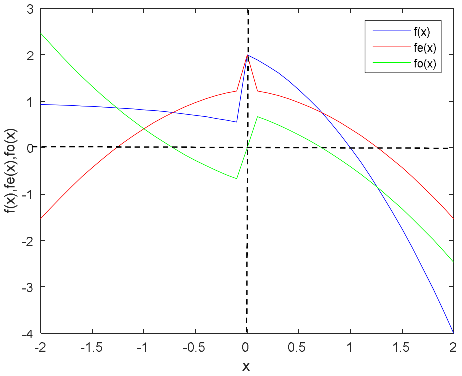


Fig. 1: An example of standard decomposition in a 1-D signal with $L_x = 2$. In this figure, the blue line means a signal $f(x)$, and the red line means an even signal $f_e(x)$, and the green line means an odd signal $f_o(x)$.

Similarly, we find that if $f_e(x,y)$ and $f_o(x,y)$ are orthogonal and they also satisfy Eq. (4), then it must satisfy Eq. (3).

4. GLOBALLY OPTIMAL DISCRETE SYMMETRY

In the 2-D discrete cases, if we only find the symmetrical types of f_e and f_o, we can use auto-convolution of $f(n,m)$ to find the symmetry candidate points. Otherwise, we consider all of points in $f(n,m)$ as the symmetry candidate points.

$$f_e(n;n_0,m;m_0)=\frac{f(n,m)+f(2n_0-n,2m_0-m)}{2}$$

$$f_o(n;n_0,m;m_0)=\frac{f(n,m)-f(2n_0-n,2m_0-m)}{2}$$

$$f(n,m)=f_e(n;n_0,m;m_0)+f(n;n_0,m;m_0) \qquad (6)$$

Next, searching for the optimal symmetry point consists in choosing the position that lets either E_e or E_o be the global minimum. where $[1\ ;\ 2n_0]$ is the non-null support of the product in the summation. The energy of the odd part E_o has the same expression provided the sign of the summation in the last row is changed. Using the definition of convolution for energy sequences yields:

$$E_e(n_0,m_0)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\left|f_e(n;n_0,m;m_0)\right|^2$$

$$=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\left|\frac{f(n,m)+f(2n_0-n,2m_0-m)}{2}\right|^2$$

$$=\frac{1}{4}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\Big[\left|f(n,m)\right|^2+\left|f(2n_0-n,2m_0-m)\right|^2$$

$$+2f(n,m)f(2n_0-n,2m_0-m)\Big]$$

$$=\frac{1}{2}E+\frac{1}{2}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}f(n,m)f(2n_0-n,2m_0-m)$$

$$=\frac{1}{2}E+\frac{1}{2}f*f(2n_0,2m_0)$$

$$E_o(n_0,m_0)=\frac{1}{2}E-\frac{1}{2}f*f(2n_0,2m_0)$$

$$E=E_e(n_0,m_0)+E_o(n_0,m_0) \qquad (7)$$

Thus, we need to compute the auto-convolution of $f(n,m)$ .The optimal candidate locations can thus be derived from:

$$[2n_0,2m_0]=\underset{n,m}{\arg\max}\left|(f*f)[n,m]\right| \qquad (8)$$

Finally, the convolution can be described by the form of Fast Fourier Transform (FFT), so we can simplify the equation of Eq. (8). If the size of the image f is $N\times M$ pixels, then the size of auto-convolution of f is $(2N-1)\times(2M-1)$ pixels. Therefore, we can do the $(2N-1)\times(2M-1)$ pixels of FFT of f and multiply by the conjugate of itself, then we use Inverse Fast Fourier Transform (IFFT) to obtain the result which is equal to Eq. (9). The fast algorithm of optimal candidate locations can be derived from :

$$f * f = IFFT(FFT(f) \times FFT^{*}(f)) \qquad (9)$$

In addition, we find that (n_0, m_0) is not necessarily an integer , may be a multiple of 0.5. So, in Section 5, we can choose the size of blocks ($N \times M$) that both N and M can be even or odd. Finally, we can select any size of block, such as 10×10 , 11×11 , 10×11 , and so on.

5. LOCALLY OPTIMAL DISCRETE SYMMETRY

In this section we use the approach proposed in Section 3 to identify, for an arbitrary sub-block, maximal support blocks that exhibit near perfect even or odd local symmetries. The procedure is derived from the theory [9]. It takes three steps, as shown by the flow diagram of Fig. 2.

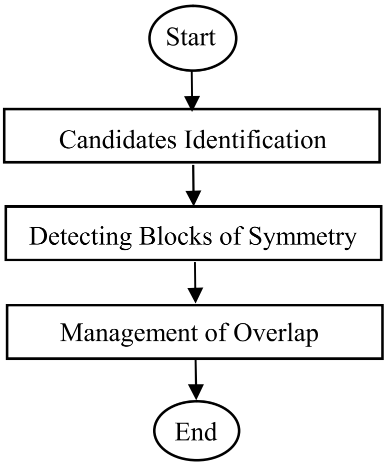


Fig. 2: Block diagram describing the whole process.

5.1. Candidates Identification

In order to locate possible even or odd symmetry points, we compute the auto-convolution of $f(n,m)$, and the position of peak is equal to candidate points. Detailed discussion may be referred in Section 4. In Figs. 3 and 4, we compute auto-correlation of the threshold image, and take the position of local extrema into candidate points, which are stored in a list P , say an i-th location P_i .

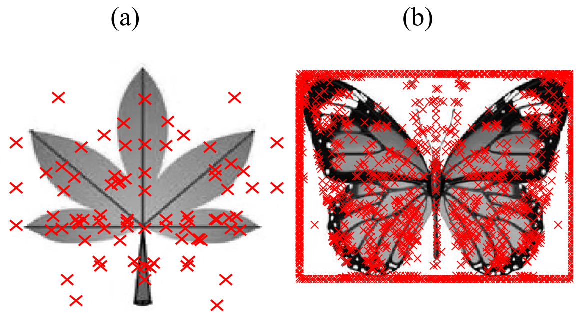


Fig. 3: We use auto-convolution in Fig. 4, and find the candidate points. The red mark means the candidate points.

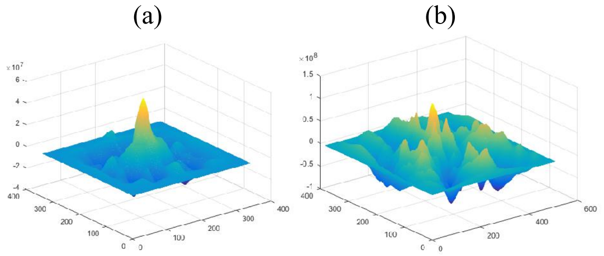


Fig. 4: It is shown respectively auto-convolution of Fig. 3(a) and 3(b). In order to find the candidate points, we can use the location of the extreme point of auto-convolution and divide by 2.

5.2 Detecting Blocks of Symmetry

Three parameters must be set for this process: the minimum width support $w_{\min}$, the minimum height support $h_{\min}$, and the even or odd energy ratio threshold p_t ; p_t is usually between 0.5 and 1. When p_t is close to 1, then it means the block has the strong even or odd symmetry characteristic. So, we usually choose p_t between 0.95 and 1.

In section 5.1, we find the candidate points. At an i-th location P_i , a segmental block consisting of a sub-block $g\left(n,m\right)$ with the minimum width support $w_{\min}$ and the minimum height support $h_{\min}$ centered around is selected P_i . Then, the even or odd decomposition of $g\left(n,m\right)$ with respect to P_i is carried out, including mean removal, computing E_g , E_{ge} and E_{go} , which represent the energies of $g\left(n,m\right)$ and of its even and odd decomposition respectively.

To classify $g\left(n,m\right)$ into an even or odd symmetric segment, we check whether the energy ratio $E_{ge} / E_g \geq p_t$ and $E_{go} / E_g \geq p_t$. If neither condition is verified, P_i is removed from the list of the candidate locations. Finally, we gradually increase the width support of the sub-block $g\left(n,m\right)$, and consider the height support on $h_{\min}$, and the maximal width support is found that repeating the process until the energy ratio smaller than p_t . Otherwise, if we increase the height support of the sub-block $g\left(n,m\right)$, and consider the width support on $w_{\min}$, and then the maximal height support is found that repeating the process until the energy ratio smaller than p_t .

Because we execute this algorithm twice, sometimes the blocks overlap at the same symmetry point. Therefore, we compare the number of pixels of each block, and we save the block which has height in pixels, and remove other blocks at the same symmetry point. Detailed discussion is in Fig. 5.

This operation allows to track the local symmetry and the large support of the block, and the result of Detecting Blocks of Symmetry is in Fig. 6. The whole of Detecting Blocks of Symmetry of diagram has three steps. First, if the energy ratio of minimum blocks satisfies p_t, then we keep the candidate points. Otherwise, we remove the candidate points. Second, given the symmetry points, we use even or odd energy ratio whether is satisfied by p_t to find the maximal blocks and simultaneously identify the symmetrical type. Third, we find the blocks overlap at the same point, so we remove all blocks except one block which has height in pixels. Detailed discussion is shown by the flow diagram of Fig. 7.

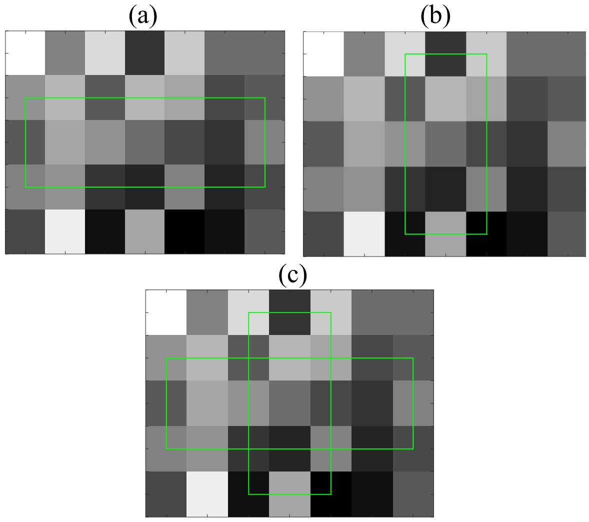


Fig. 5: From Fig. 5(a), fixed height support and adjusted width support. In Fig. 5(b), fixed width support and adjusted height support. In Fig. 5(c), the blocks overlap at the same point, so we choose the height number of pixels, and preserve Fig. 5(a).

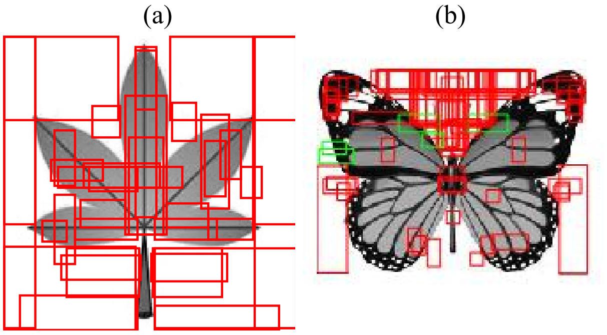
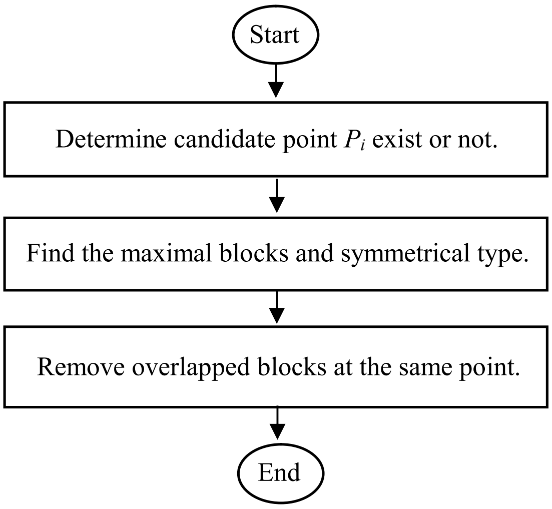


Fig. 6: The background is the intensity of the image. From Fig. 6(a), it is a feature of leaf, and form Fig. 6(b), it is a feature of butterfly. In order to decompose even and odd symmetry, we take the original image into the threshold image, and then try to find the large support block on the threshold image. The algorithm has been used in $w_{\min} = 6$, $h_{\min} = 6$ and $p_t = 0.95$. The red block means even symmetry and the green block means odd symmetry.


Start

Determine candidate point P_i exist or not.

Find the maximal blocks and symmetrical type.

Remove overlapped blocks at the same point.

End

Fig. 7: Block diagram describing Detecting Blocks of Symmetry.

5.3 Management of Overlap

The next step concerns which blocks to keep and how to remove their overlaps. In particular, it is necessary to define a priority criterion to help choosing blocks. Possible criteria can be (i) the largest segment support and (ii) the highest energy ratio. To achieve the two criteria, we make a new algorithm to remove their overlaps. First, we introduce Find Neighbor, and then Manage Overlaps.

5.3.1 Find Neighbor

To efficiently manage overlaps, we take all the blocks into different groups which do not overlap each other without considering only one block in the group, and each block of the same group has at least one or more overlapping by itself. For example, if the block A, B, and C overlap each other, we take them into Group 1, and if the Block D, E overlap each other, we take them into Group 2, and so on. The result is in Fig. 8.

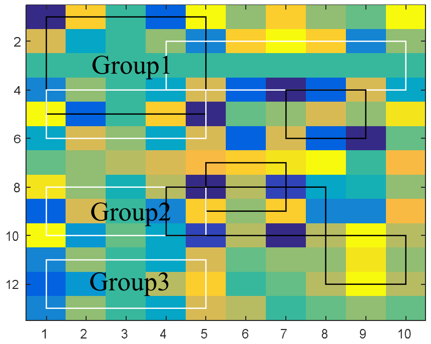


Fig. 8: The background is the color image that describes the different number. The black blocks represent even symmetry, and the white blocks represent odd symmetry. In the image, we can find the three groups. The first group has four blocks, and the second group has four blocks, and the third group has only one block.

5.3.2 Manage Overlaps

Next, all the blocks are classified into the different groups. And then, we find the largest block in the same group, and reduce the rest of the block. Sometimes, the overlapped blocks have the same size, so we retain blocks that have higher ratio and reduce the size of blocks that have lower energy ratio. This process is interactive and it stops until all the overlapped blocks are the smallest size. We will use the same way to delete overlapped blocks until all the blocks do not overlap. Finally, the process of managing overlap of Fig. 8 is shown in Fig. 9, and the final experiment of Fig. 6 is shown in Fig. 10, and the program of diagram block is written in Figs. 11(a) and 11(b).

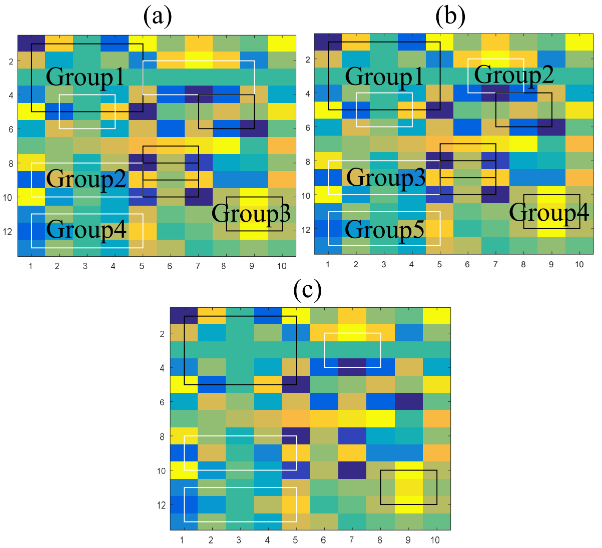


Fig. 9: From Fig. 9(a), we reduce the size of blocks of Fig. 8, and simultaneously do the method in Section 5.3.1. In Fig. 9(b). We still reduce the size of blocks until the size of overlapped blocks (except the maximum size of block) is minimum and simultaneously do the method in Section 5.3.1. From Fig. 9(c), we delete the overlapped the minimum size of blocks, and all of the blocks do not overlap each other.

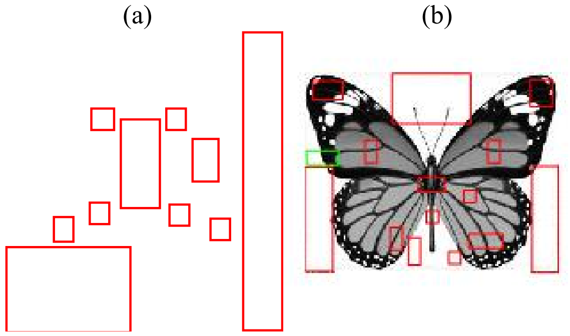


Fig. 10: The algorithm has been used in $w_{\min} = 6$, $h_{\min} = 6$ and $p_t = 0.95$. We can see that there are non-overlapped blocks in Fig. 6, such that it can help us compress image. The red blocks mean even symmetry and the green blocks mean odd symmetry.

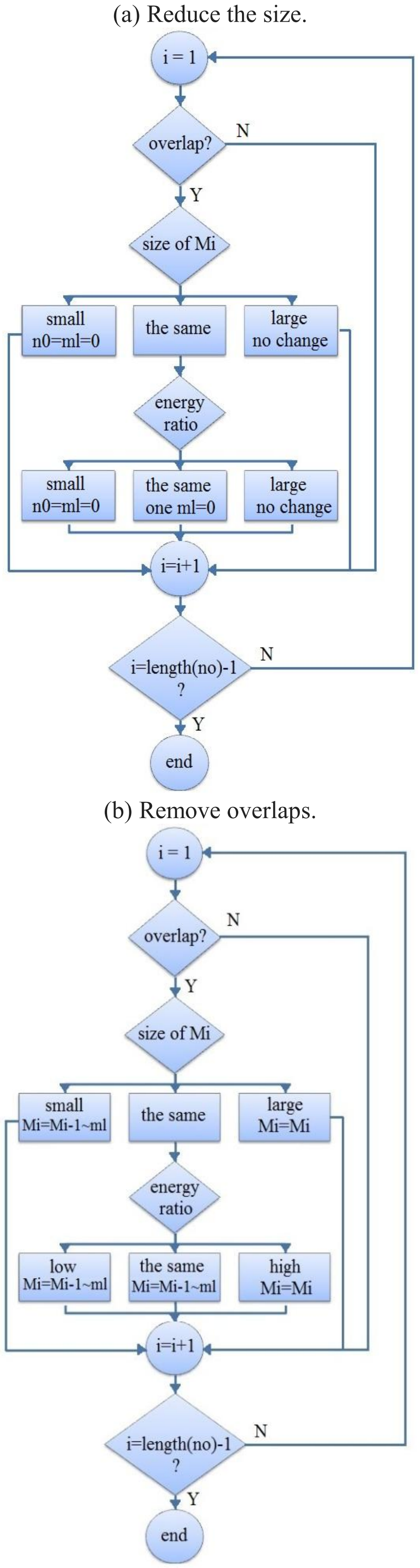


Fig. 11: First, we reduce the size of overlapped blocks in the same group from Fig. 11(a). Second, we remove the smallest size of overlapped blocks in the same group from Fig. 11(b).

6. EXPERIMENTAL RESULT

6.1 Compute PSNR

The mathematical equation of PSNR is $10\log\left(I_{\max}{}^{2}/e^{2}\right)$. In this equation, $I_{\max}$ is the maximal intensity of the image, and e is the error between the compressed image and the original image. In order to simplify the computation of the error (e), we only compute the error between the compressed blocks and the original blocks. For example, we assume that the compressed image is $g\left(n,m\right)$, and the original image is $f\left(n,m\right)$. Next we segment the compressed image into non-overlapped block $g_i\left(n,m\right)$, and $g\left(n,m\right)=g_1\left(n,m\right)+...+g_K\left(n,m\right)$, we do the same thing in the original image. The total of errors can be simplified by the sum of errors of the non-overlapped blocks, because $g_i\left(n,m\right)\times f_j\left(n,m\right)=0$. The PSNR of the experiment of Figs. 10 is shown in Table 7.

Table 7: PSNR of Figs. 10.

Fig. 10(a)	Fig. 10(b)
61.2289	50.4769

6.2 Compute Compression Ratio

To know compression ratio, we know the symmetry of each block, so we can remove the redundant pixels of each block. For example, if the total number of pixels $(m\times n)$ is odd, we removed the $(m\times n-1)/2$ redundant pixels and preserve a center pixel of symmetry point. In contrast, if the total number of pixels $(m\times n)$ is even, we removed the $(m\times n)/2$ redundant pixels. Finally, we assume that the size of the i_{th} block is $m_i\times n_i$ and there are K blocks. We sort out the number (r_i) of redundant pixels of the i_{th} block and simultaneously calculate the total number (R) of removed pixels in Eq.10.

$$r_i=\begin{cases}\dfrac{m\times n-1}{2}, m\times n \ is\ odd.\\ \dfrac{m\times n}{2} \quad , m\times n\ is\ even\end{cases}$$

$$R=r_1+r_2+...+r_K \tag{10}$$

In addition, we need to consider the overhead bits, the overhead bits are above the location (x,y) of the center symmetry point of the i_{th} block and the information of the height (h_i) and width (w_i) of the i_{th} block. There are four information in each block, and each data type is float about 32 bits, so we calculate that the overhead bits of each block are 128 bits. Finally, we assume that the size of the original image is $N\times M$ and one pixel represent 8 bits, so the mathematical equation of the compression ratio is descripted in Eq. 11.

$$compression\_ratio=\frac{(N\times M-R)\times 8+K\times 128}{N\times M} \tag{11}$$

The compression ratio of the experiment of Fig. 10 is shown in Table 8.

Table 8: Compression ratio of Fig. 10.

Fig. 10(a)	Fig. 10(b)
0.64318	0.72417

6.3 The Application of Natural Images

We show that the overlapped image and non-overlapped image with the even and odd symmetry of the blocks, and use even and odd energy ratio to identify even or odd. Finally, we compute the PSNR and compression ratio in the natural pictures. Detail discussion is in Fig. 12, Table 9 and Table 10.

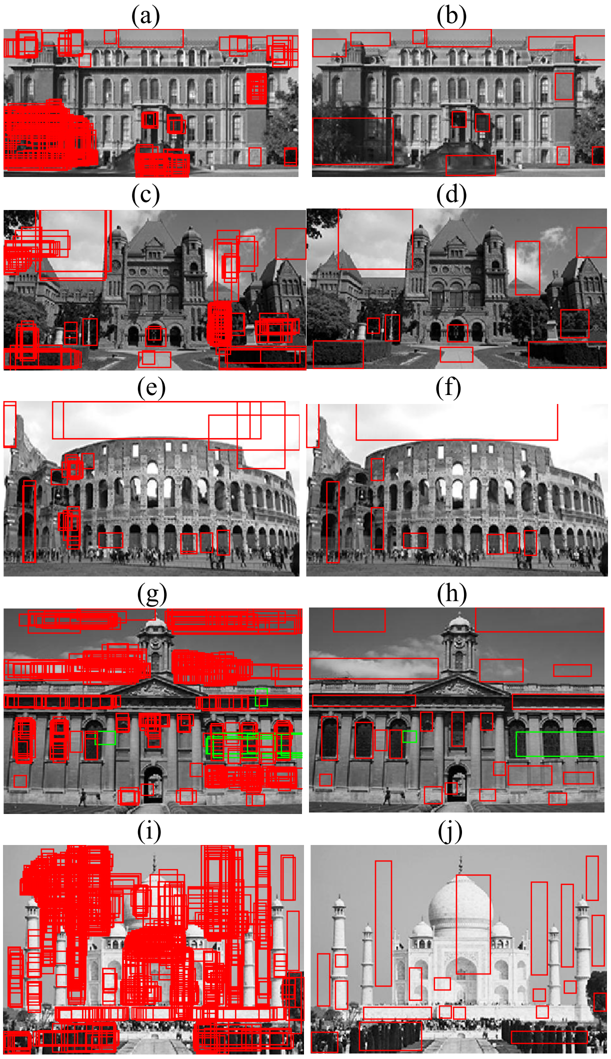


Fig. 12: The background is an original image. In order to decompose even and odd symmetry, we take the original image into the threshold image, and then try to find the large support block on the threshold image. The algorithm has been used in $w_{\min}=6$, $h_{\min}=6$ and $p_t=0.95$. The red blocks mean even symmetry (f_e), and the green blocks mean odd symmetry (f_o).

Table 9: PSNR of Fig. 12.

Fig. 12(b)	Fig. 12(d)	Fig. 12(f)	Fig. 12(h)	Fig. 12(j)
73.5603	63.4818	43.7470	71.1539	54.0965

Table 10: Compression ratio of Fig. 12.

Fig. 12(b)	Fig. 12(d)	Fig. 12(f)	Fig. 12(h)	Fig. 12(j)
0.7713	0.7492	0.7936	0.67749	0.7819

7. CONCLUSION

In this paper, we present a method to identify blocks in an image denoted by inherent local symmetry. First, we use the auto-convolution for finding the symmetry point of an image. Then, we use energy ratio to determine the symmetrical blocks, and manage overlapped blocks to compress easily. Finally, we detect the natural image with the even or odd symmetry, and simultaneously compute both the PSNR and compression ratio between the original image and compressed image.

Current work focuses on detecting an image with symmetrical blocks. In addition, the extension of the algorithm to add more symmetry types and use different shapes of the blocks and improve the threshold to get more the odd symmetry. It is the subject of currently undergoing studies.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 15, 2009.

[2] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. Smeaton, and G. Quenot, “Trecvid 2014 –an overview of the goals, tasks, data, evaluation mechanisms and metrics,” *Proc. of TRECVID*, NIST, USA, 2014.

[3] E. Hjelmas and B. K. Low, “Face detection: A survey,” *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.

[5] X. Zhou, X. Zhuang, S. Yan, S.-F. Chang, M. HasegawaJohnson, and T. Huang, “Sift-bag kernel for video event analysis,” *Proc. of the 16th ACM Int. Conf. on Multimedia*, pp. 229–238, 2008.

[6] G. Medioni, I. Cohen, F. Brmond, S. Hongeng, and R. Nevatia, “Event detection and analysis from video streams.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 873–889, 2001.

[7] Hauagge, D. C., & Snavely, N, “Image matching using local symmetry features,” *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pp. 206-213, 2007.

[8] A. Oppenheim, A. Schafer, and C. Jones, *Discrete-Time Signal Processing.*, Prentice Hall, 1999.

[9] Gnutti, A., Guerrini, F., & Leonardi, R, “Representation of signals by local symmetry decomposition,” *Proceedings of Signal Processing Conference (EUSIPCO)*, pp. 983-987, 2015.

[10] S. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[11] Y. Liu, H. Hel-Or, C. Kaplan, and L. V. Gool, Computational Symmetry in Computer Vision and Computer Graphics, *Foundations and Trends in Computer Graphics and Vision*, 2009.

[12] Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., & Dellaert, F, “Detecting and matching repeated patterns for automatic geo-tagging in urban environments,” *Proceedings of Computer Vision and Pattern Recognition*, pp. 1-7, 2008.

[13] Wu, C., Frahm, J. M., & Pollefeys, M., “Detecting large repetitive structures with salient boundaries,” *Proceedings of European conference on computer vision*, pp. 142-155, 2010.

[14] Loy, Gareth, and Alexander Zelinsky, “Fast radial symmetry for detecting points of interest,” *IEEE Transactions on pattern analysis and machine intelligence*, 2003.

[15] Reisfeld, D., Wolfson, H., & Yeshurun, Y., Context-free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, pp. 119-130, 1995.

[16] Kovesi, P., Symmetry and asymmetry from local phase. *Proceedings of Tenth Australian joint conference on artificial intelligence*, pp. 2-4, 1997.

[17] P. Kovesi, “Image features from phase congruency,” *Videre: Journal of computer vision research*, pp. 1-26, 1999.

[18] V. Di Ges ` u, V. Di Gesu, and C. Valenti. The Discrete Symmetry Transform in Computer Vision. *Technical report*, Universita di Palermo, 1995.