

Self-correcting Symmetry Detection Network

Wonil Chang¹, Hyun Ah Song², Sang-Hoon Oh³, and Soo-Young Lee^{1,2}

¹ Department of Bio and Brain Engineering,

² Department of Electrical Engineering,

KAIST, Daejeon 305-701, Republic of Korea

³ Department of Information Communication Engineering,

Mokwon University, Daejeon 302-729, Republic of Korea

chang.wonil@gmail.com,

{hyunahsong,sylee}@kaist.ac.kr,

shoh@mokwon.ac.kr

Abstract. In this paper, we propose a symmetry axis detection network that can correct asymmetric parts by itself. Our network compares directional blurring of omnidirectional image edges, which plays a significant role in asymmetry detection and correction. The output layer consists of oscillatory neurons, which activates symmetry axes one by one. Given activated symmetry axis, network estimates the difference of image edges and generates a masking filter to cover the asymmetric parts. The network reconstructs ideal mirror-symmetric image with complete symmetry axes by self-correction. Our network models flexible symmetry perception of high-level cognitive function of human brain.

Keywords: Symmetry axis detection, asymmetry reconstruction, asymmetry correction, oscillator network.

1 Introduction

Symmetries are everywhere. We can easily detect them surrounding us, and we perceive them to be natural and intuitive. This is explained by mammals' preference for symmetric patterns in visual information processing [1]. Along with these biological evidences that support importance of symmetry axes in visual information processing, several practical approaches have been developed, which utilizes symmetry axes for visual information processing of objects or faces [2].

Transformational approaches [3] discover underlying structure of symmetries by comparing local pixels of the images [1]. Local region-wise comparisons require excessive search, which takes too much computational cost. As a solution, many symmetry detection methods use spatial filters to reduce the resolution of images, and determine symmetries in hierarchical stages by comparing counterparts of symmetry axes [4]. Filtering reduces computational cost. Apart from those technical approaches, artificial neural networks (ANNs) model biologically plausible algorithms. Especially, Fukushima et al. [5] successfully modelled early visual process of humans for symmetry axis perception.

Those symmetry axes detection models are vulnerable to asymmetric corruptions. If there exists any asymmetric parts in an image, models may not be able to detect underlying symmetry axes. Therefore, correction of asymmetries plays a critical role in symmetry detection process. Human’s ability to perceive dominant symmetries against environmental distractions is also a biological evidence of asymmetry correction in visual processing mechanism.

In this paper, we propose a symmetry axes detection network that can correct asymmetries by itself. We detect symmetry axes by blurring an image with directional filters. Then we correct asymmetric parts with a mask generated by back projection of asymmetric measure along the symmetric axis activated by oscillator network. As a result, underlying symmetry axes are detected and image of ideal symmetry is returned.

Rest of the paper is organized as follows. In section 2, we describe our neural network model for symmetry detection. In section 3, experimental results are presented. We end our paper with conclusion in section 4.

2 Method

Our method uses a recurrent neural network that consists of mainly two parts: symmetry axis detection (part 1) and asymmetry correction (part 2).

The proposed method differs from Fukushima’s or other symmetry axes detection methods in three approaches.

Firstly, we use sliced-cone filter for directional image blurring. Each sliced-cone filter displays unique orientations. The orientation of spatial filter helps not only symmetry detection, but also asymmetry correction.

Secondly, we alternatively activate symmetry axes one at a time with oscillator network. The serialization of multiple symmetry detection enable post-processing respect to individual symmetry axis.

Thirdly, we combined both symmetry detection stage and correction stage in the network. We used the proposed sliced-cone filters for asymmetry detection, and reconstructed perfect symmetries from the original image. As a result, we could significantly improve the robustness of symmetry detection against noise and partial distortions.

Overall network is described in Fig. 1. Symmetry axes are detected in part 1. Part 2 corrects asymmetries in the image and give feedback to part 1. The network recurrently iterates until convergence.

2.1 Part 1 - Symmetry Axes Detection

Edge Extraction (U_G and U_S layer) After input image is presented in layer U_0 , a layer of photoreceptors, it is fed to U_G layer which resembles lateral geniculate nucleus cells (LGNs) [5]. As in [5], U_G layer consists of two cell planes, one for off-center-on-surround cells and the other for on-center-off-surround cells. Each cell plane is responsible for extracting positive and negative contrasts in brightness respectively. The output of U_G layer is denoted by $U_G^{(k)}(m, n)$, which

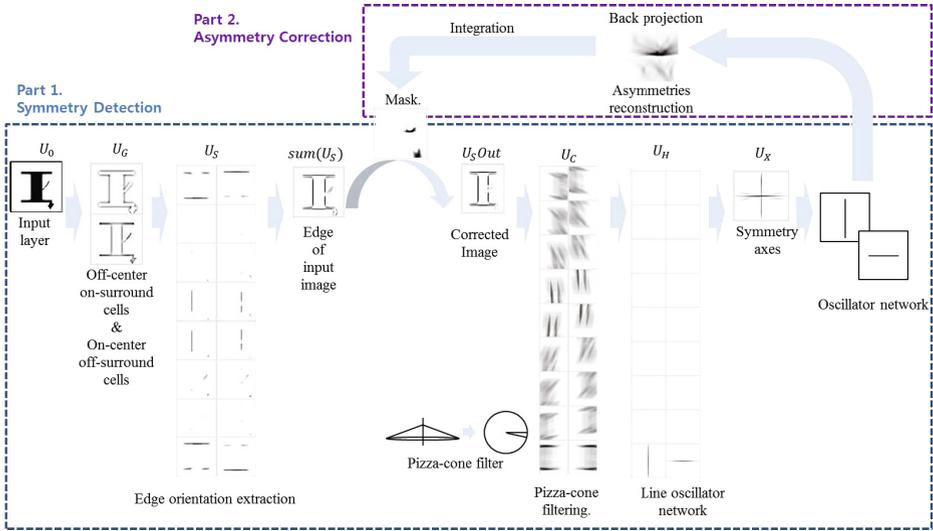


Fig. 1. The architecture of self-correcting symmetry detection network

is in correspondence with $u_C(\mathbf{n}, k)$ of equation (1) in [5]. Here, k is the index for off-center ($k = 1$) and on-center ($k = 2$) cell-plane, and (m, n) is the image coordinate. In U_S layer, we extract oriented edges from U_G layer. The response of U_S layer is denoted as $U_S^{(k)}(m, n)$, which is in correspondence with $u_S(\mathbf{n}, k)$ in equation (2) in [5]. We added additional cell-plane $U_S^{sum}(m, n) = \sum_{k=1}^K U_S^{(k)}(m, n)$, which represent the whole edge extraction result.

Sliced-Cone Filtering (U_C Layer). For symmetry axis detection, Fukushima’s network compares blurred orientatinal edges. In our proposed network, we introduce sliced-cone filters where we assign orientation property for each filter. Since our proposed filter displays orientation properties, we do not need orientational edges of input data like U_S of Fukushima’s. Instead, we blurred extracted edges in layer $U_S^{sum}(m, n)$.

The sliced-cone filter for the k th orientation $a_k = \frac{2\pi k}{K}$ is given by

$$F_{SC}^{(k)}(m, n) = F_S^{(k)}(m, n) \times F_C(m, n), \tag{1}$$

where

$$F_S^{(k)}(m, n) = \begin{cases} 1 & \text{if } |\text{atan2}(n, m) - a_k| \leq \frac{\pi}{K} \text{ and } 0 < \sqrt{m^2 + n^2} < L \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

is the slice filter for orientaiton a_k ,

$$F_C(m, n) = \frac{\varphi(L - \sqrt{m^2 + n^2})}{L} \tag{3}$$

is the cone filter with radius L , and $\varphi()$ is defined by $\varphi(\alpha) = \max(\alpha, 0)$. Sliced-cone filter has a shape of cone filter that is divided into K slices. By cutting cone filter into slices, we assign an orientation angle to each slice of filter for detecting corresponding directional edges. Finally, filter response, $U_C^{(k)}(m, n)$, is computed as

$$U_C^{(k)}(m, n) = \sum_{n'=-L}^L \sum_{m'=-L}^L F_{SC}^{(k)}(m', n') \times U_S^{sum}(m + m', n + n'). \quad (4)$$

Symmetry Axis Detection (U_H and U_X Layer). For symmetry axis detection, we use same principle as [5]. If an image is symmetrical about a given axis, features of the left and right side of the axis should be in correspondence to each other. This means that $U_C^{(kl)}(m, n)$ and $U_C^{(kr)}(m, n)$, the filter responses of sliced-cone filter of orientation a_{kl} and a_{kr} , should be symmetrical about given axis at a_k , where kl and kr are the indices of angle $a_k + \frac{k}{K}i$ and $a_k - \frac{k}{K}i$ for $i = 1, 2, \dots, K$, respectively. In U_H layer, the symmetry measure for axis of orientation a_k at point (m, n) of image, $U_H^{(k)}(m, n)$, is computed by

$$U_H^{(k)}(m, n) = \varphi \left(\gamma(U_C^{(kl)}(m, n) + U_C^{(kr)}(m, n)) - \delta|U_C^{(kl)}(m, n) - U_C^{(kr)}(m, n)| \right). \quad (5)$$

Here, we define $\gamma = \gamma_1 \angle(a_{kl}, a_{kr})$, where $\angle(a_{kl}, a_{kr}) = | \text{mod}(a_{kl} - a_{kr} + \pi/2, \pi) - \pi/2 |$ denotes relative difference between angle a_{kl} and a_{kr} . And γ determines the weight for summation of filter output response of the kl th and kr th orientations. It is maximized when a_{kl} and a_{kr} are perpendicular to a_k , and minimized (becomes zero) when they are parallel to a_k . This is because filter responses $U_C^{(kl)}(m, n)$ and $U_C^{(kr)}(m, n)$ that are parallel to the symmetry axis leave artifacts on the very end of the image which cannot be removed by subtraction part, of (5). δ determines the weight for second term of (5). By subtracting second term from first term, $U_H^{(k)}(m, n)$ is left with only common part of two output responses $U_C^{(kl)}(m, n)$ and $U_C^{(kr)}(m, n)$, which can be interpreted as symmetry axis. The final symmetry axes detected are $U_X(m, n) = \max_k U_H^{(k)}(m, n)$.

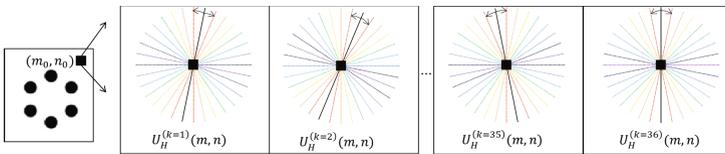


Fig. 2. Symmetry axes detection process. Black lines indicate given symmetry axis candidates with angles of a_k ($k = 1, 2, \dots, 36$). Colored pairs of lines with same colors indicate pairs of directions of sliced-cone filters with angles of a_{kl} and a_{kr} , to test for the symmetricity of given symmetry axis (black line).

Oscillatory Network. Among several symmetry axes detected, we alternatively activate one symmetry axis at a time and reconstruct asymmetry parts with respect to the activated axis. To activate one symmetry axis at a time, we used locally excitatory globally inhibitory oscillator networks (LEGION) [6] and mutual excitatory connectivity based on co-linearity between edge pixels.

Let us imagine two edge pixels e_i and e_j as shown in Fig. 3. We name orientation of edge pixels e_i and e_j , and line $\overline{e_i e_j}$ as θ_i , θ_j , and θ_{ij} , respectively. The distance between two edge pixels is denoted as d_{ij} . The angle between each edge pixel and line $\overline{e_i e_j}$ is defined as:

$$\phi_{ij} = D(\theta_i - \theta_{ij}), \text{ and } \phi_{ji} = D(\theta_j - \theta_{ij}). \tag{6}$$

Here, a function $D(\theta) = \theta - \lfloor \frac{\theta}{\pi} + 0.5 \rfloor \pi$ is a conversion of angle into the range of $[-\frac{\pi}{2}, \frac{\pi}{2})$. Here, $\lfloor x \rfloor$ is the round down value of x .

Given this situation, we detect straight lines using excitatory connection w_{ij} and inhibitory connection v_{ij} (7), where $f_d(d_{ij})$ and $f_l(\phi_{ij}, \phi_{ji})$ denote distance factor and linearity factor respectively. They are computed as in (8) and (9), where b is a constant. The neurons that are close together and appear on a straight line score high value for excitatory connection w_{ij} .

$$w_{ij} = f_d(d_{ij})f_l(\phi_{ij}, \phi_{ji}), \text{ and } v_{ij} = g_{inh}f_d(d_{ij}). \tag{7}$$

$$f_d(d_{ij}) = \begin{cases} \left(\frac{\sigma_d}{d_{ij}}\right)^2 & \text{if } d_{ij} > \sigma_d, \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

$$f_l(\phi_{ij}, \phi_{ji}) = \begin{cases} \cos b(|\phi_{ij}| + |\phi_{ji}|) & \text{if } b(|\phi_{ij}| + |\phi_{ji}|) \leq \frac{\pi}{2}, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The final excitation of neuron is determined by (10). Parameter g_{inh} controls the operation of the network, which acts as a threshold of neuron’s excitation.

$$w_{ij} - v_{ij} = f_d(d_{ij})(f_l(\phi_{ij}, \phi_{ji}) - g_{inh}). \tag{10}$$

Oscillator $X^{(k)}(m, n)$ represents the activity of the symmetry axis with orientation a_k and position (m, n) . It receives $U_H^{(k)}(m, n)$ as external input. The network generates alternate activation of symmetry axes. (See [6] for the details).

2.2 Part 2 - Asymmetry Reconstruction and Correction

The detected symmetry axes are alternatively fed into part 2 one by one, for asymmetry reconstruction and correction.

Asymmetry Reconstruction. Given one symmetry axis activated, we calculate a measure of asymmetry, or back projection coefficient $B_C^{(k)}(m, n)$, for each of neurons on the symmetry axis. $B_C^{(k)}(m, n)$ is defined by

$$B_C^{(k)}(m, n) = \sum_{k'=1}^K A^{(k)}(m, n) \times \varphi \left(U_C^{(k')}(m, n) - U_C^{(k'')}(m, n) \right), \tag{11}$$

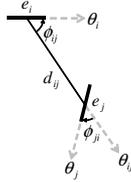


Fig. 3. Two edge pixels

$$A^{(k)}(m, n) = \begin{cases} 1 & \text{if } X^{(k)}(m, n) \geq \Theta_x, \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

where Θ_x is a threshold and $k'' = \text{mod}(2k' - k - 1, K) + 1$.

Given the k th symmetry axis, we compare differences in filter responses of counter angles. This differences is the asymmetry measure, the back projection coefficient $B_C^{(k)}(m, n)$. Asymmetric parts for given symmetry axis is reconstructed by filtering back $B_C^{(k)}(m, n)$ with sliced-cone filters $F_{SC}^{(k)}(m, n)$. The process is described in Fig. 4 (b), and back projection result $B_P(m, n)$ is shown in Fig. 4 (c), and is computed as:

$$B_P(m, n) = \sum_{k=1}^K \sum_{n'=-L}^L \sum_{m'=-L}^L F_{SC}^{(k)}(m', n') \times B_C^{(k)}(m + m', n + n'). \tag{13}$$

Asymmetry Correction. We generate mask image $M(m, n)$ with reconstructed asymmetry part.

$$M(m, n) \leftarrow \psi [M(m, n) + \mu (B_P(m, n) - \max(\eta B_I(m, n), \varepsilon))] \tag{14}$$

where $\psi []$ is defined by $\psi [\alpha] = \max(\min(\alpha, 1), 0)$, and $B_I(m, n) = \sum_k A^{(k)}(m, n) * (F_C(m, n))^2$. As an asymmetric image pixel gets closer to the symmetry axis, it receives stronger back-projection. We introduced $B_I(m, n)$ to compensate for the spatial irregularity of back-projection.

The mask image is assigned to $U_S^{sum}(m, n)$ by

$$U_S^{sum}(m, n) \leftarrow (1 - M(m, n))U_S^{sum}(m, n). \tag{15}$$

It removes asymmetries in $U_S^{sum}(m, n)$ and feed the result to layer $U_C^{(k)}(m, n)$ in part 1, for recurrent feedback of asymmetry correction and symmetry axes detection steps.

3 Experimental Results

In this experiment, we used images provided by Fukushima [5] and generated some corruptions or noises in the images to create asymmetries. The asymmetries

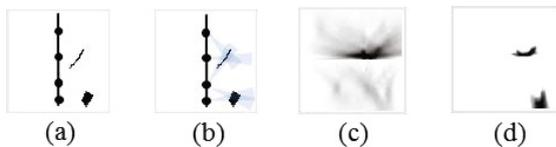


Fig. 4. (a) Asymmetry part with respect to given symmetry axis, (b) back projection of $B_C^{(k)}(m, n)$ by sliced-cone filter $F_{SC}^{(k)}(m, n)$, (c) reconstructed asymmetries, $B_P(m, n)$, (d) generated mask, $M(m, n)$

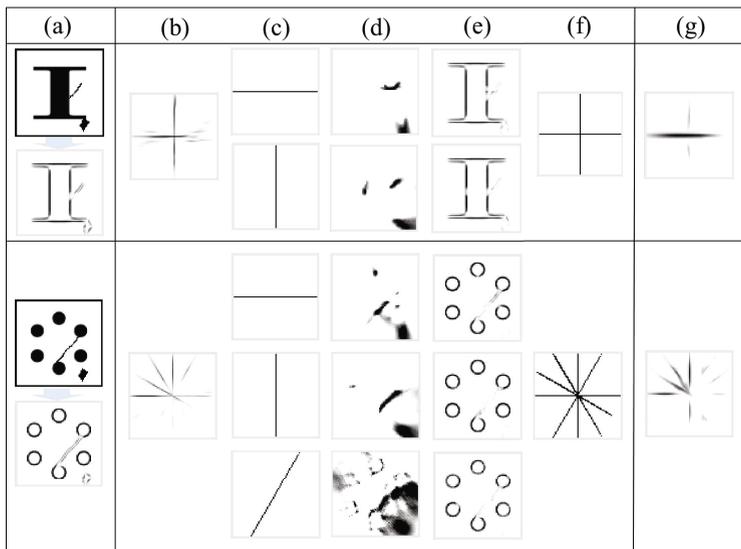


Fig. 5. Experimental results. (a) Input image corrupted with line and blob, and $U_S^{sum}(m, n)$, (b) $U_X(m, n)$, (c) symmetry axis activated one at a time by oscillator network, (d) image mask with respect to active symmetry axis, (e) self-corrected image, (f) accumulation of neural activity in oscillator network, (g) symmetry axes detected by Fukushima's network [5].

include lines and blobs. Examples of corrupted images and its $U_S^{sum}(m, n)$ are shown in Fig. 5 (a). The size of each image is 76×76 . We used the length of filter $L = 50$, and number of orientational angle $K = 36$. For calculation of $U_H^{(k)}(m, n)$ in Eq. (5), we used $\gamma_1 = \frac{1.2}{16}$, and $\delta = 5$. For mask generation in Eq. (14), we used $\mu = 0.003$, $\eta = 0.6$, and $\varepsilon = 1.0$. For fast simulation of oscillatory network, we implemented simplified algorithm of LEGION in [6].

The results of asymmetry corrections and resulting symmetry axes detection are shown in Fig. 5. To verify the effect of asymmetry correction in symmetry axes detection, we compared our result with that of Fukushima's. As shown in Fig. 5 (g), Fukushima's network displays weak activations or sometimes miss

symmetry axes with asymmetric parts. On the other hand, the proposed network successfully corrects asymmetries with mask generated by back-projection (Fig. 5 (d)). Also, our network returns self-corrected image with perfect symmetries (Fig. 5 (e)) and clear symmetry axes (Fig. 5 (f)).

4 Concluding Remarks

This paper proposed the self-correcting symmetry detection network. Several technical contributions were added to conventional ANN-based symmetry detection model for self-reconstruction of image symmetry. The oscillator network in the output layer serializes the extraction of symmetry axes, which enable sequential reconstruction of multiple image symmetries. The back-projection of directional blurring filter plays a significant role in estimating differences from the input image and correcting asymmetries. As shown in experimental results, our network can handle partial occlusion and local distraction by self-correction. It overcomes the limitation of conventional symmetry detection network models that suffer from asymmetry corruption. The network could be extended for practical applications in real environment. As further works, we will test the performance of our network with various types of asymmetry noise.

Acknowledgement. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2011-0029816). Wonil Chang and Hyun Ah Song are co-first authors, who equally contributed to this paper. We thank professor Fukushima for providing code for symmetry detection network [5].

References

1. Treder, M.S.: Behind the Looking-Glass: A Review on Human Symmetry Perception. *Symmetry* 2, 1510–1543 (2010)
2. Wang, W., Gao, Y., Hui, S.C., Leung, M.K.: A Fast and Robust Algorithms for Face Detection and Localization. In: *Proceedings of the 9th International Conference on Neural Information Processing*, pp. 2118–2121. IEEE Press (2002)
3. Palmer, S.E.: *Human and Machine Vision*. Academic Press, USA (1983)
4. Graham, N.V.S.: *Visual Pattern Analyzers*, Oxford, UK (1989)
5. Fukushima, K., Kikuchi, M.: Symmetry Axis Extraction by a Neural Network. *Neurocomputing* 69, 1827–1836 (2006)
6. Wang, D.L., Terman, D.: Image Segmentation Based on Oscillatory Correlation. *Neural Computation* 9, 805–836 (1997)