Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 5
Lopo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)

A NEURAL SYSTEM DESIGN FOR CDF OPERATIONS

Wanwu Guo* Anthony Watson

School of Computer and Information Science
Edith Cowan University
2 Bradford Street, Mount Lawley
Western Australia 6050, Australia

ABSTRACT

Conjugate directional filtering (CDF) is a method proposed by us recently. By using CDF, two direction-filtered results in conjugate directions can be merged into one image that shows the maximum linear features in the two conjugate directions. However, we only reported the CDF concepts and operations in our previous paper, without considerations in how to implement this method. In this paper, we report the progressive result of our CDF study, which shows that a neural system can be used to implement the CDF operations. Two neural network designs are proposed based on whether a program that conducts directional filtering should be included. Trial work on a synthetic model processed using this neural system is also reported.

1. INTRODUCTION

Guo et al [1] proposed a method named the conjugate directional filtering (CDF) to combine two directional-filtered results in conjugate directions into one image that exhibits the maximum information of the linear features in the two conjugate directions. Several CDF operations were also defined to carry out this method. However, there was no consideration in how to implement this method given in that paper [1]. Our further study shows that neural networks can be used to implement the CDF method. When the directional filtered data are made available by a separate program, a CDF interface based on neural network design can be used to carry out and coordinate all the CDF operations. Alternatively, a program that conducts directional filtering can also be included in a CDF package that uses an integrated and neural network-based interface to carry out all the processing. In this paper, we first present the concepts and general structures of the neural networks that are used to carry out the CDF operations. A synthetic model is then used to demonstrate the results from some CDF operations by using this neural system.

2. CONCEPTS OF CDF OPERATIONS

Assuming f_0 to be the original data file, f_1 and f_2 to be the directional-filtered data files in the two conjugate directions, some of the CDF operations are defined as [1]:

CDF(add1) = $f_1 + f_2$; (1)
CDF(add2) = abs($f_1 + f_2$); (2)
CDF(add3) = abs(f_1) + abs(f_2); (3)
CDF(max1) = max(f_1, f_2); (4)
CDF(max2) = max[abs(f_1), abs(f_2)]; (5)
CDF(origin) = origin[CDF(max2)]; (6)
CDF(ampl) = sqrt(f_1* f_1 + f_2* f_2). (7)

Operation (6) means to allocate the original sign to the absolute value resulted from operation CDF(max2). In addition, more CDF operations can be defined, particularly the following two as:

CDF(norm1) = norm1[CDF(*)]; (8)
CDF(norm2) = norm2[CDF(*)]; (9)

Operation (8) means to normalise all negative values to zero or a positive value V_1 and all positive values to another unique positive value V_2 ($>V_1$). With operation (9), all positive values are normalised to a unique positive value whereas all negative values are normalised to a certain negative value. In practice, depending on the structure of the original data, some of these operations may produce the same outcome, which will be seen in our synthetic model.

All these CDF operations can be summarised as:

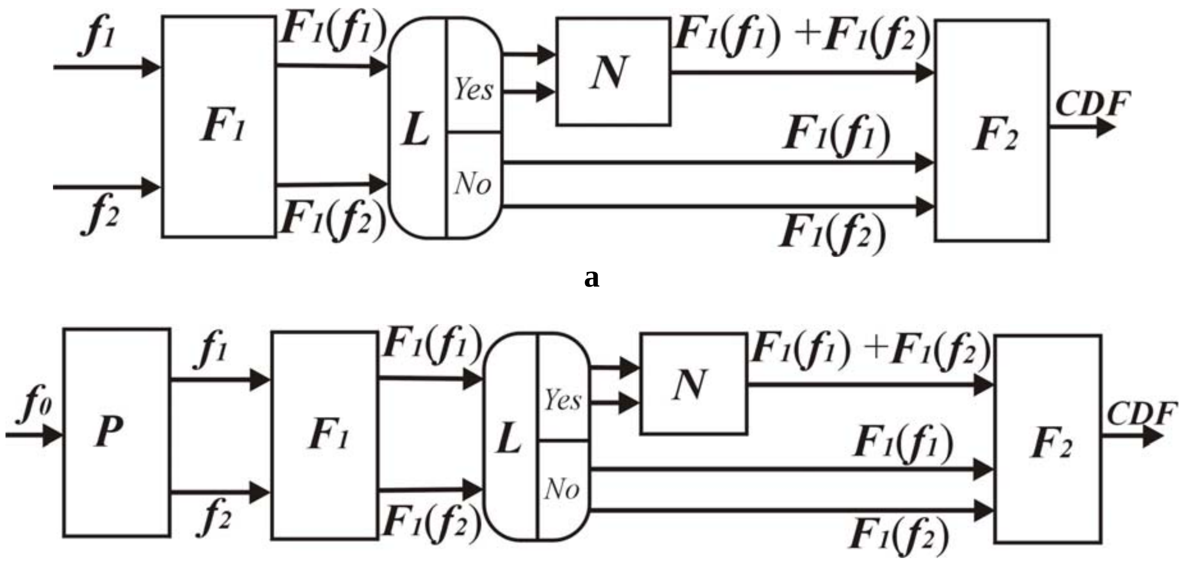
CDF = $F_2[F_1(f_1), F_1(f_2)]$. (10)

3. GENERAL STRUCTURE OF THE CDF NEURAL SYSTEMS

We first assume that directional filtering has been applied to an original file (f_0) in two conjugate directions and the corresponding files are f_1 and f_2. The corresponding structure of a CDF neural system is illustrated in Figure 1a. This system consists of four parts: initial transfer $\boldsymbol{F_1}$, logical operator $\boldsymbol{L}$, neuron $\boldsymbol{N}$, and final transfer $\boldsymbol{F_2}$.

When the two data files of directional filtering in conjugate directions are firstly input into the system, they are manipulated by the same transfer function $\boldsymbol{F_1}$. The pair of outputs, $F_1(f_1)$ and $F_1(f_2)$, is then used as the input to the logical operator $\boldsymbol{L}$ that determines whether a neural operation is needed before the final transfer function is called. If a no decision is made, the pair is directly sent to the final transfer unit $\boldsymbol{F_2}$ for processing. If a yes decision is made, the pair is directed to the neural unit for neural processing. Theoretically, a neural network can also have any selective weights and biases [2]. Given the purpose of the network for handling the CDF operations that do not need offsets and weighting effects, the neuron unit in this system is designed to have a zero bias and a constant weight of 1.0. The output $N = F_1(f_1) + F_1(f_2)$ from the neuron $\boldsymbol{N}$ is then sent to the final transfer unit $\boldsymbol{F_2}$. Data into the final transfer unit $\boldsymbol{F_2}$ are further manipulated there, and the output is CDF = $F_2[F_1(f_1), F_1(f_2)]$, which is as same as formula (10). Initial and final transfer functions ($\boldsymbol{F_1}$ and $\boldsymbol{F_2}$) corresponding to the nine CDF operations are listed in Table 1.

This system can also include an additional unit P as a major program to carry out the directional filtering before CDF operations. In this case, the extended structure of the proposed neural system is shown in Figure 1b.



a

b

Figure 1. Schematic diagrams of the CDF neural systems

Table 1. Features of the neural system corresponding to the CDF operations

CDF operation	$\boldsymbol{F_1}$	$\boldsymbol{L}$	$\boldsymbol{N}$	$\boldsymbol{F_2}$	**CDF output**
CDF (add1)	1	Yes	$f_1 + f_2$	1	$f_1 + f_2$
CDF(add2)	1	Yes	$f_1 + f_2$	Abs	abs($f_1 + f_2$)
CDF(add3)	abs	Yes	abs(f_1) + abs(f_2)	1	abs(f_1) + abs(f_2)
CDF(max1)	1	No	-	max	max(f_1, f_2)
CDF(max2)	abs	No	-	max	max[abs(f_1), abs(f_2)]
CDF(origin)	abs	No	-	sign*max	origin[CDF(max2)]
CDF(ampl)	square	Yes	f_1* f_1 + f_2* f_2	sqrt	sqrt(f_1* f_1 + f_2* f_2)
CDF(norm1)	1/abs/square	Yes/No	Yes/No	norm1	norm1[CDF(*)]
CDF(norm2)	1/abs/square	Yes/No	Yes/No	norm2	norm2[CDF(*)]

4. A SYNTHETIC EXAMPLE

We first use a synthetic example to demonstrate the general operation of this neural system for CDF operations on the digital data of this example. The CDF digital data are then presented as individual images. This means that for this synthetic example, the CDF operations are not carried out directly on the original pixel-based image. For the same model, image presentation using digital data will differ from that using pixel-based image, which will be discussed in a separate paper.

The synthetic model consists of three centered squares and a cross sign in the middle of the squares (Fig. 2a), but the central cross cannot be seen because the contrast between the cross and the inner square is too weak. After applying directional filtering in both the horizontal and vertical directions, the two CDF data files are input into the neural system similar to the one illustrated in Figure 1a. Given different combinations of system parameters, the corresponding CDF images are showed in Figures 2b-2f. Because the data structure of this model is symmetric, some CDF operations produce the same image.

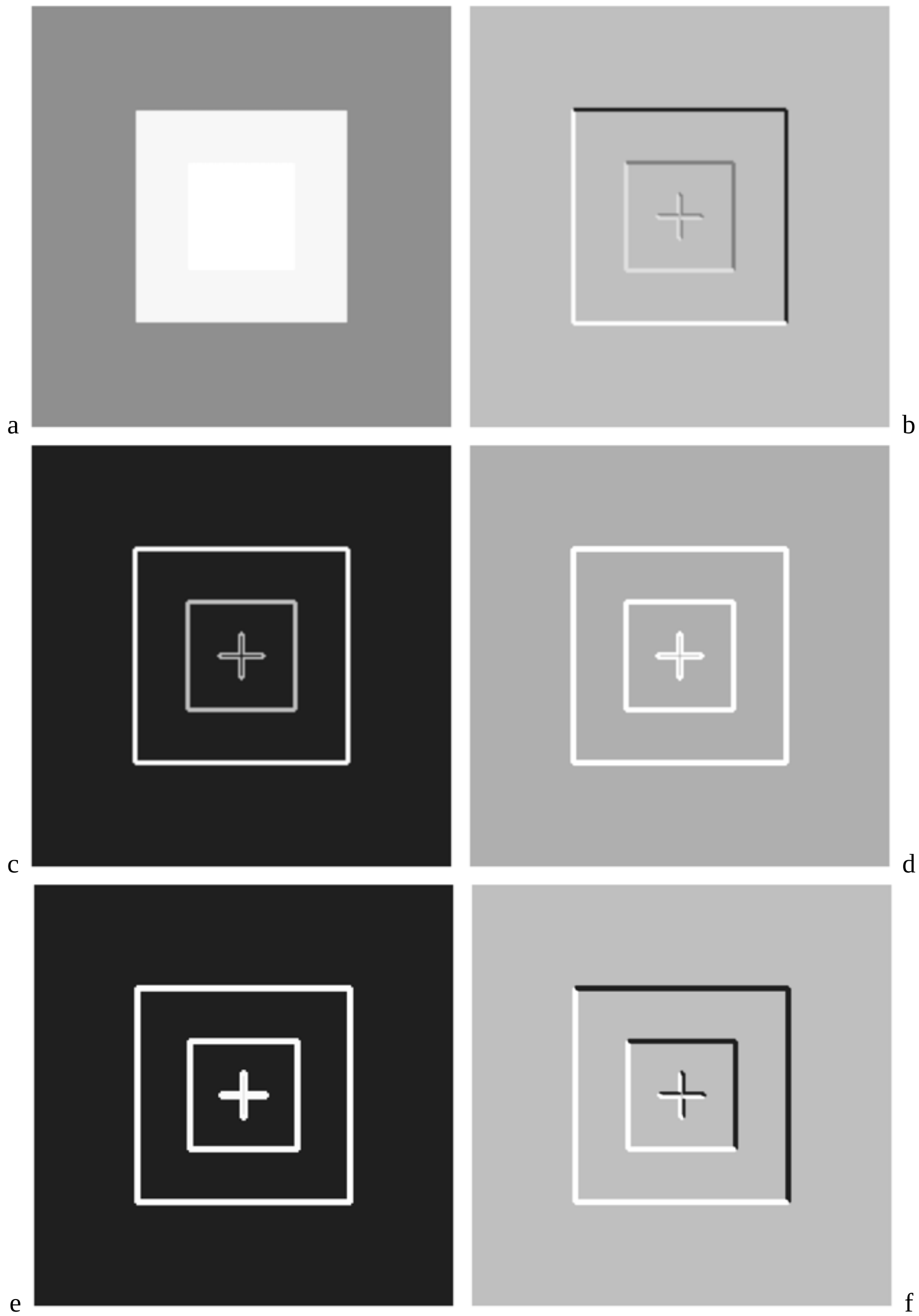


Figure 2. Synthetic model (a), images of CDF(add1) (b), CDF(add2) (c), CDF(norn1) (d-e), and CDF(norm2) (f).

5. DISCUSSION AND CONCLUSION

Figure 2b shows the image of CDF(add1). The three centered squares and the cross are clearly outlined by this CDF operation. It is interesting to notice the following three features also showed on this image. First, because the directional filtering is applied from left to right in horizontal and from bottom to top in vertical, the left and bottom edges of all the objects have positive values and are showed as lighter colors; the right and top edges of all the objects have negative values and are showed as darker colors. Visually, this image looks stereoscopic, or has 3D effect, being illuminated from southwest to northeast. Second, after applying directional filtering, mathematically all the areas except the boundaries of all the three squares should have the same value of zero because each square has the same value. In a grayscale image, this means that the backgrounds of all the three squares should be black. As a result, the right and top edges of all the objects that have negative values should be merged into the black background. By using this neural system, one can set up a specific set of grayscale values to replace the original CDF values. For example, this neural system can set the smallest CDF value to a positive value V_1 (< 255) and the greatest CDF value to another positive value V_2 ($V_1 < V_2 < 256$) in grayscale. Other CDF values between the smallest and greatest can be then replaced into V_1 and V_2 based on linear, logarithmic, or other scales. Figure 2b is the result of a linear replacement of the CDF(add1). Thirdly, since the symmetry of the original data, the image of CDF(add1) is the same as that of CDF(origin). A similar case to Figure 1b is Figure 2f that is the image of CDF(norm2). The major difference on Figure 1f from Figure 1b is that in this case all positive values are set to 255 and all negative values are set to 0. However, the 3D effect is retained on both images.

The result of CDF(add2) is presented in Figure 2c. Note that in this case the background is set to 0 and the maximum value is set to 255, which represents the boundaries of the second square. The inner square and middle cross are outlined in different grayscale values other than 0 and 255. Again, this image also represents CDF(add3), CDF(max2), and CDF(ampl) because of the symmetry of the original model.

The result of CDF(norm1) is showed in two images using different grayscales (Figs. 2d-2e). Although in both cases, the non-zero values are all set to 255, in Figure 1d the zeros (background) are set to a non-zero value but less than 255 whereas in Figure 1e the background is set to 0. Note there is no 3D effect in Figures 2c-2e.

The synthetic example presented above shows that the neural system is very useful in handling CDF operations. The adjustable parameters in the neural system allow a CDF image to be not only produced, but also presented in different colour or greyscale schemes, which gives users more choices to look into the same image from different angles for different specific purposes.

6. ACKNOWLEDGEMENT

The Faculty of Communication, Health and Science of the Edith Cowan University is thanked for supporting this research project. We are grateful to the constructive comments from anonymous referees.

REFERENCES

[1] W. Guo, D. Li, & A. Watson, Conjugated linear feature enhancement by conjugate directional filtering, IASTED international conference on visualization, imaging and image processing, Marbella, Spain, 2001, 583-586.

[2] J.C. Principe, N.R. Euliano and W.C. Lefebvre, Neural and adaptive systems (New York, NY: John Wiley and Sons, 1999).