Automatic recognition of electrical and architectural symbols using COSFIRE filters

Student : Luke Agius

May 2013

Supervisor : George Azzopardi

University of London
BSc Creative Computing (Hons)

Submitted as part of the requirements for the award of the Degree in Creative Computing of the University of London.

Acknowledgements

[To be included]

Contents

Table Of Contents iii

List of Figures v

List of Tables vi

Chapter 1. Summary 1
- 1.1 Keywords . . . 1

Chapter 2. Introduction 2
- 2.1 Background . . . 2
- 2.2 Research Questions . . . 4
- 2.3 Deliverables . . . 5
- 2.4 Project Plan . . . 6

Chapter 3. Literature Review 8

Chapter 4. Methods 9
- 4.1 Overview . . . 9
- 4.2 An overview of COSFIRE filters . . . 10
- 4.3 Gabor Filters . . . 11
- 4.4 Configuration of a COSFIRE filter . . . 12
- 4.5 Response of a COSFIRE filter . . . 15
- 4.6 Forming a shape descriptor . . . 16
- 4.7 Classification . . . 17

Chapter 5. Results 19
- 5.1 Dataset description . . . 19
- 5.2 Datasets . . . 19
- 5.3 testing with datasts . . . 20
- 5.4 Subsection for all datasets. . . . 20

Chapter 6. Discussion 21

Chapter 7. Conclusion 22

Chapter A. Appendix **23**

Chapter B. Evaluation **25**

List of Figures

2.1 Example of Symbols 3
2.2 Thesis Project Plan 6

4.1 Example of a prototype pattern 10
4.3 Configuration of a COSFIRE filter using single point approach . . . 12
4.4 Example configuration of a COSFIRE filter 13
4.5 Example of an incorrect configuration due to rho list 14
4.6 An example of a 3-point approach in configuring COSFIRE filters. . 15
4.7 Example of a shape descriptor 17

List of Tables

1. Summary

[To be included]

1.1 Keywords

[To be included]

2. Introduction

2.1 Background

Handheld devices have become ubiquitous due to their flexibility and enhanced productivity. Such devices allow users to write freely on them or to manually sketch symbols. Various applications include the recognition of handwritten letters, digits, signatures, musical notes, electrical and architectural symbols, among others. For instance a musical composer might write down a piece of music on paper or an electrical engineer might also quickly jot down parts of a schematic. Manually converting this hand written/sketched information into a digital representation may take a lot of time.

Such devices enable the composer or the engineer to quickly and efficiently convert their ideas into a correct digital representation. After doing so both the composer and the engineer might be able to quickly incorporate their ideas into a larger, already digitally converted, body of work. For instance, electrical or architectural engineers can quickly convert an image of hand drawn/sketched schematics into a set of symbols and then import them into drafting software such as Auto Cad.

The automatic recognition of symbols in hand-drawn sketches is an important step in such applications due to the fact that each symbol in physical works need to be properly recognised and converted into a digital representation. This results in a more convenient and efficient storage, retrieval and manipulation as compared to conventional.

Symbols are sets of contextually meaningful 2-dimensional (2D) graphical shapes. Symbols can be binary, made out of sets of lines and loops, such as electrical or architectural symbols, or complex such as company logos. Examples of those two types of symbols are shown in Fig. 2.1.

(a) (b)

Figure 2.1: (a) An example of a binary symbol. (b) An example of a complex symbol

From a symbol recognition stand point, documents of interest which mostly hold the most symbols can be grouped into three main groups [3] :-

1. Technical drawings, such as flow diagrams and electrical schematics
2. Maps, including different kinds of maps such as topographical or geographical
3. Others, such as musical scores

Symbols in such documents can contain noise, be deformed or transformed. Symbols can also be isolated or embedded in their surrounding arrangement. They may be in close proximity to each other, touch or partially overlap each other. The main problem with recognising symbols in such documents is not the complexity of features the symbols hold but rather the fact that they are embedded in context that is generally very crowded and that the same symbol may be drawn more than once in the same document with each instance of the symbol varying in terms of noise, scaling, rotation, etc. Therefore it is crucial that the recognition process adheres to intravariant approach.

This introduces a challenge and makes the problem a non-trivial one, thus being a topic of active research with various available state-of-the-art techniques[3]. This will be further elaborated in the literature review section.

One of those state of the art methods are COSFIRE (Combination Of Shifted FIlter REsponses) filters [5]. They are effective for key-point detection and pattern recognition. They are trainable because they can be configured with any given prototypes. They are constructed by a configuration process which automatically analyses the dominant orientations and their mutual spatial arrangement of a given prototype pattern of interest. Gabor filters are used to detect the dominant orientations. The response of a COSFIRE filter is computed as the weighted geometric mean of the involved Gabor filter responses. This means that a response is only achieved when all the concerned contour parts are present [5]. COSFIRE filters can also achieve invariance to rotation, scale, reflection and contrast inversion.

In their work [5], the authors demonstrated that COSFIRE filters can be effectively applied to the detection of vascular bifurcations, recognition of handwritten digits and detection and recognition of traffic signs in complex scenes. In this work, we investigate the effectiveness of the mentioned COSFIRE filters for the recognition of electrical and architectural symbols.

[Should I mention how the thesis is sectioned, like you did in your paper? "The rest of the paper is organised as follows: in Section2 we present......"]

2.2 Research Questions

The following are the questions that we pose in this work:-

1. How effective are COSFIRE filters for the recognition of electrical and architectural symbols?

2. How robust are COSFIRE filters to noisy symbols?

3. How does the results achieved by COSFIRE filters compare to publish result of other state-of-the-art methods?

The effectiveness of COSFIRE filters is evaluated by numerous experiments which are performed on various publicly available data sets [1] [2]. These data sets hold 150 different classes of symbols which are be used to configure COSFIRE filters. Test symbols, are then used to test the configured COSFIRE filters. The test symbols in the mentioned data sets contain both noisy and noise less symbols. Furthermore, the test symbols can also be scaled or/and rotated.

2.3 Deliverables

The deliverables include, a description of the proposed method and a comparison against other state of the art techniques. The three posed research questions will be addressed through the MATLAB implementation of the proposed method. Using this implementation, experiments will be conducted which include an analysis of the results achieved. Performance measurements will be computed in the form of true positives and false positive rates which can be used to derive accuracy, precision and recall rates. Documentation of the MATLAB implementation is provided on how it is built. This also includes a step by step guide on how to re-run the experiments. A discussion is included concerning some aspects of the proposed approach which highlights the differences that distinguish it from other approaches. Finally, conclusions are drawn which will ultimately highlight any limitations of COSFIRE filters.

2.4 Project Plan

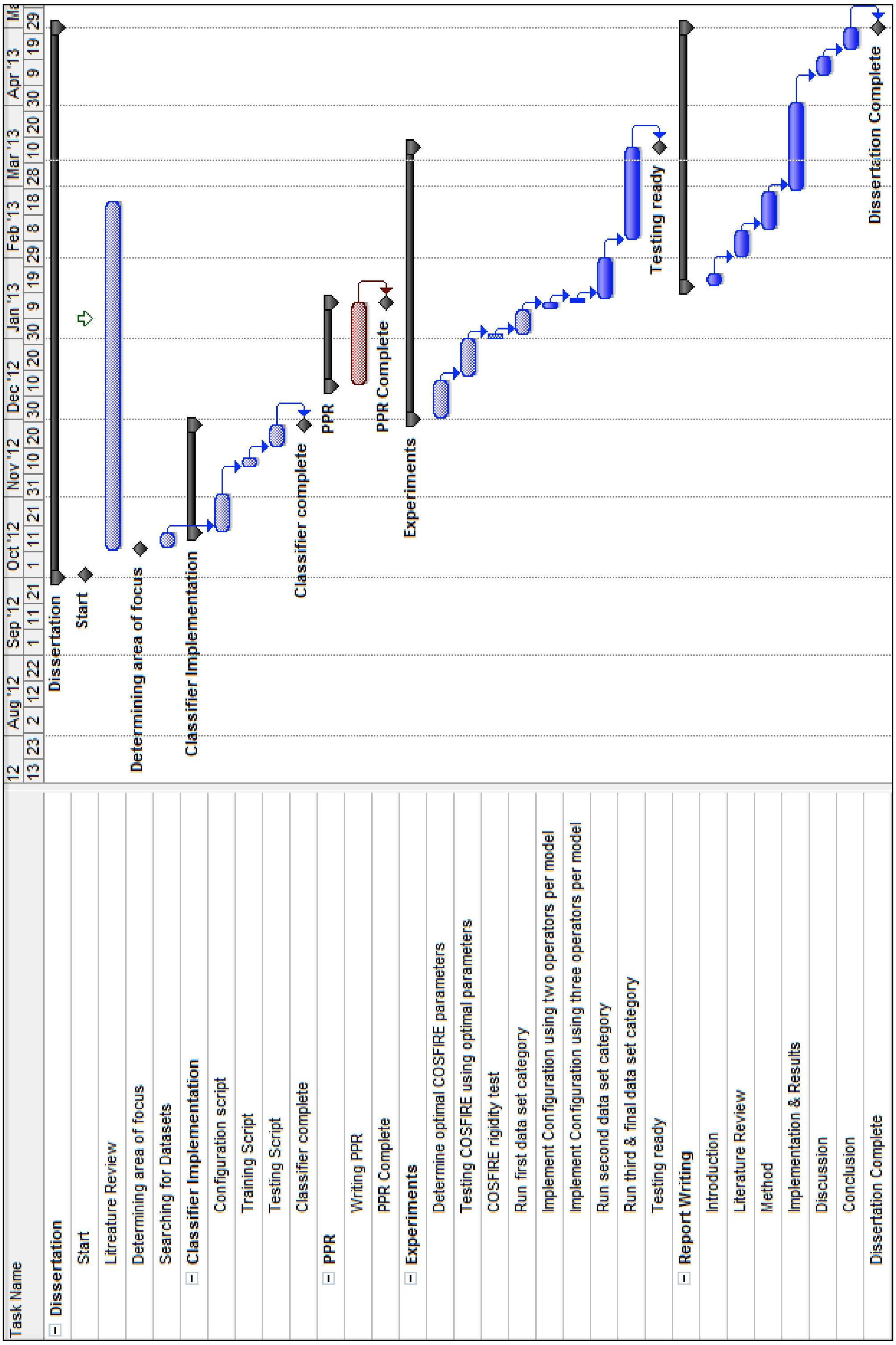


Figure 2.2: Project Plan

Fig. 2.2 shows the project plan of this thesis. The time frame in which work starts and finishes is from October 2012 till March 2013. The first task in the project plan is to start the literature review in order to determine the area of focus. Once the area of focus was determined other tasks, a search started to find publicly available datasets against which the proposed method can be run. In parallel with this the literature review was still underway.

Once adequate data sets are available, a classifier need to be built which includes a configuration phase, a training phase and a testing phase which will be described in detail in Chapter 4 Methods. Once the classifier is completed, the various experiments are executed. Experiments started in December 2012 till March 2013. In parallel with the experiments, work on the final document started. Both the experiments and the final document are to be ready by May 2013.

3. Literature Review

4. Methods

4.1 Overview

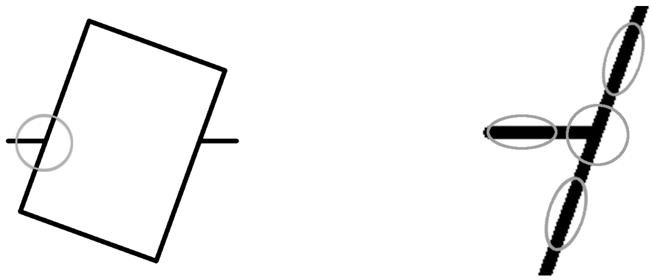
The research method applied in this project is quantitative. This also includes data analysis and hypothesis testing. Through the application of COSFIRE filters and classification techniques on publicly available data sets, performance measurements will be computed in the form of true positives and false positive rates which can be used to derive accuracy, precision and recall rates.

The way we evaluate COSFIRE filters for the recognition of electrical and architectural symbols is as follows. We start by evaluating the proposed method on the data set with the least level of complexity which contains noiseless test images given in the same orientation and scale of their corresponding models. Then we apply the method to the data set with the second lowest complexity and finally we evaluate the COSFIRE filters on the data set with the highest level of complexity. The mentioned order in which we run experiments facilitates the analysis of the performance results and give us insight on further tuning the method.

In Sections 4.2 and 4.3 we describe the main idea behind COSFIRE filters and Gabor filters. In Section 4.4 we explain how we configure COSFIRE filters in a training phase. Then, in Section 4.5 we describe the responses attained from the training phase after which in 4.6 we explain how we form shape descriptors from those responses. Finally, in Section 4.7 we explain how we use the resulting feature vectors to classify test images.

4.2 An overview of COSFIRE filters

The following example gives an overview of ~~the main idea behind~~ COSFIRE filters. COSFIRE filters are configured by first specifying a point of interest. In Fig.4.1 a model symbol is used as an input image from which a vertex is chosen as a prototype ~~pattern,~~ encircled in Fig. 4.1a. This prototype pattern is used to automatically configure a COSFIRE filter that responds to similar spatial arrangements.



(a) (b)

Figure 4.1: (a) A model symbol (of size 512x512), of which one vertex (encircled) is used as a prototype pattern. (b) The enlarged prototype pattern where the dominant orientations around the point of interest are marked by 3 ellipses.

The three ellipses shown in Fig. 4.1b around the encircled vertex illustrate the dominant orientations in the spatial arrangement around the chosen point of interest. These dominant orientations are detected by using symmetric Gabor filters. Upon detecting the dominant orientations, the corresponding Gabor filters' responses are combined by multiplication in order to create COSFIRE filter's response.

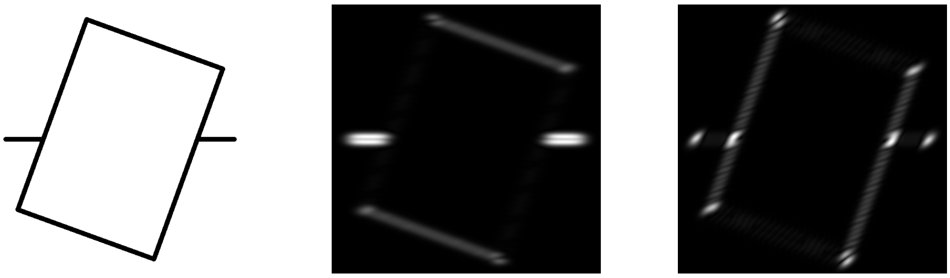
~~Once the COSFIRE filter is configured, it will~~ produce a response only when presented with the same or similar spatial arrangement of lines with ~~specific~~ orientations and thicknesses to the prototype.

4.3 Gabor Filters

~~As mentioned in Section. 4.1~~ the dominant orientations around a specified point of interest are detected by using symmetric two-dimensional (2D) Gabor filters. A 2D Gabor filter is defined as the product of an elliptical Gaussian and a sinusoid plane wave [4].~~When a Gabor filter is applied to an input image we denote its response as follows:~~

$$\cancel{g_{\lambda,\theta}(x,y)} \tag{4.1}$$

The response represents the detection of a contour part within the image of wavelength λ and of orientation θ. In Fig. 4.2 two examples of Gabor filters, of different orientations are being applied to a symbol model image.



(a) (b) (c)

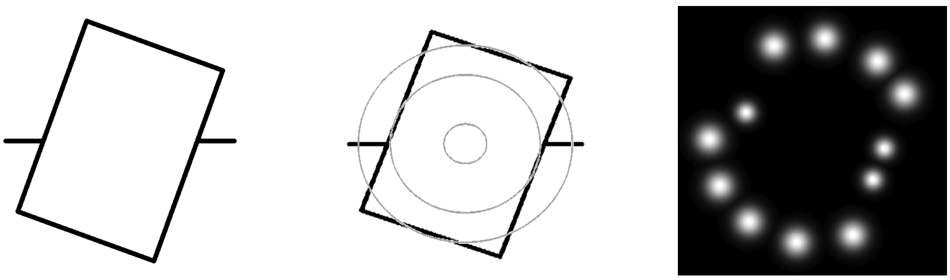
Figure 4.2: (b) The response of a Gabor filter when applied to a (a) model symbol with $\lambda = \cancel{10}$ and $\theta = 90$. (c) The response of a Gabor filter when applied to a (a) model symbol with $\lambda = 10$ and $\theta = 130$. [Is it a good idea to include this figure?]

~~When a set of Gabor responses are available, they can be combined in a Gabor energy filter.~~ When using Gabor filters in the COSFIRE filters, their responses are then held against a threshold. This threshold, $t1 (0 \leq t1 \leq 1)$, is a fraction of the maximum response of $g_{\lambda,\theta}(x,y)$ across all combinations of values (λ, θ) and all possible positions within the image (x, y). The ~~acquired~~ threshold responses are then denoted by $|g_{\lambda,\theta}(x,y)|_{t1}$. The value of the $t1$ threshold depends on the contrast the image contains. This threshold controls the level at which a Gabor filter's response recognises the presence of a contour part.

4.4 Configuration of a COSFIRE filter

In the configuration process of COSFIRE filters, features of interest are chosen by specifying a point or a collection of points within the model symbol. The system will automatically analyse those points and extract information about the dominant orientations and their mutual spatial arrangement according to the list of concentric circles surrounding the point of interest. The number of concentric circles is not intrinsic to the method but rather to the complexity of the given prototype pattern of interest. Ultimately, these are the prototype patterns that the COSFIRE filter will learn to recognise in the testing images. [5].

For the purpose of electrical and architectural symbol recognition, a COSFIRE filter is configured for every model symbol in a given data set. These configurations were approached in two ways. In the first approach, used in the first six experiments, was to select a single static point of interest which was placed in the middle of the model symbol. Fig. 4.3 shows a configuration, using the described single point approach.



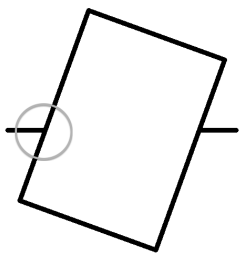
(a) (b) (c)

Figure 4.3: (c) A cosfire filter is created by placing (b) lthe point of interest at the centre of the (a) model symbol.

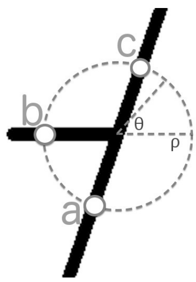
In order for a COSFIRE filter to be configured it needs, as input, the responses of specific Gabor filters. The responses of those Gabor filters are expressed as a set of four parameter values as shown in the following example.

$$GaborResponse_i = (\lambda_i, \theta_i, \rho_i, \phi_i) \tag{4.2}$$

The λ_i parameter represents the width of a contour part, the θ_i parameter represents the orientation of the same contour part. The location is then expressed in polar coordinates using two parameters, (ρ_i, ϕ_i). The parameter values of these contour parts are obtained as follows. The point of interest is encircled by a circle of radius ρ. Along this circle, a bank of Gabor filters is considered. For each position around the circle, all the filters in the filter bank are applied to it to that same position. From all those Gabor filters we only take those who have the maximum response across all filters, that is across all combinations of (λ, θ). These are chosen to be the dominant orientations around the point of interest Fig. 4.3 shows an example of this with a single circle around the point of interest in a symbol model image.



(a)



(b)

Figure 4.4: (a) A chose point of interest is (b) encircled by a circle of radius ϕ. For each point along this circle, all the filters in the filter bank are applied to it. The parts with the greater response such as a,b and c are chosen as the dominant parts around the point of interest.

We then determine the polar coordinates of the chosen points in respect to the center of the COSFIRE filter. The polar coordinates for each point are expressed as (ρ, ϕ). For each location (ρ, ϕ) we then consider all the combinations of (λ, θ) for which the corresponding Gabor filter response, $_{g\lambda,\theta}(x, y)$, is greater than a fraction,$t_2 = 0.75$, of the maximum response across all the combinations of λ and

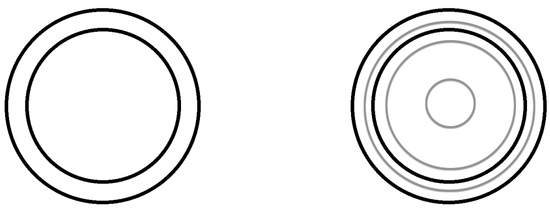
θ. For each value of θ that satisfies the condition above, we consider a single value of λ from the maximum of all responses. For each unique pair of (λ, θ), for location (ρ, ϕ) we then obtain a tuple as shown in the following example.

$$tuple = (\lambda_i, \theta_i, \rho_i, \phi_i) \tag{4.3}$$

More than one tuple can also be formed in the same location. We then express the set of parameter value combinations as $\left\{(\lambda_i, \theta_i, \rho_i, \phi_i) | i = 1...n_f\right\}$. The f subscript in the above example stands for the local prototype pattern around the specified point of interest. The following example shows the same but with real values.

$$S_f = \left\{ \begin{array}{c} \left\{\lambda = 8, \theta = 0, \rho = 0, \phi = 0\right\}, \left\{\lambda = 8, \theta = 0, \rho = 30, \phi = \pi/2\right\} \\ \left\{\lambda = 16, \theta = \pi/2, \rho = 30, \phi = 0\right\}, \left\{\lambda = 16, \theta = \pi/2, \rho = 0, \phi = \pi\right\} \end{array} \right\} \tag{4.4}$$

Although the single point approach produced good results in the first few datasets, a problem is introduced concerning the list of concentric circles used. Using the same list of concentric circles for all the datasets meant that for some model symbols no dominant orientations could be detected.This resulted in test symbols not being properly recognised. An example is shown in Fig. 4.5.



(a) Model Symbol (b) COSFIRE Operator

Figure 4.5: (b) Incorrect Configuration of a COSFIRE for the symbol (a). Here we use the same four concentric circles used in the previous example.

In order to address this issue, an alternative approach is taken in selecting points of interest. The list of concentric circles is reduced to two and three points of interest are placed randomly in the symbol model image. Each of these points hold the same reduced list of concentric circles. Doing so ensures that each model symbol has a unique set of detected contour parts. An example of this approach is shown in Fig. 4.6.

(a) Model Symbol (b) COSFIRE Operator

Figure 4.6: (b) A 3-point Configuration of a COSFIRE filter for the symbol (a). The concentric circles for each point of interest are shown in different colours.

This approach is driven by three parameters, namely NFiltersPerPrototype which determines the number of points of interest placed in the model image, the DistanceBetweenOperators parameter which represents the minimum euclidean distance between each point of interest, and finally the minNumberOfTuplesPerFilter parameter which dictates the minimum number of tuples that should be attained from the point of interest's surrounding spatial arrangement.

4.5 Response of a COSFIRE filter

Should i include the 'Blurring and shifting Gabor filter responses' to the configuration sub section or to the response sub section?

4.6 Forming a shape descriptor

~~After configuring a COSFRE filter for each symbol~~ model image using the three point approach we have available a set of COSFIRE operators three times the size of the total number of symbol model images available. The following example shows an example of applying 3 COSFIRE filters for 150 models resulting in 450 distinct COSFIRE filter operators.

~~**Example 4.6.1.** (150 *models*) x (3 *COSFIRE Filters*) = 450 *Operators*.~~

Once these COSFIRE operators are available a shape descriptor ~~can~~ be built. This descriptor will be used to compare test images from a data set against it for classification purposes. ~~The descriptor is defined as follows.~~ The COSFIRE operators created in the configuration process are applied to each symbol model again. Upon applying all the available operators to a symbol model image, all the operators' responses are in the form of a matrix of values. From each of these responses/matrices produced by the operators, only the maximum value is considered. This means that ultimately a symbol model image is described by a vector of values with a size that is equivalent to the number of COSFIRE filters used.

Ultimately, once this is done for all symbol model images the shape descriptor is defined as a list of vectors for each model symbol, each the size of the number of operators available whose every element is the maximum value from the response of a cosfire operato~~rs~~. The following example outlines the structure of the shape descriptor. Is this example adequate for explanation purposes?

$$\left\{\begin{array}{l} \cancel{Model_1 \left\{ max(Model_1\, Operator_1), \cdots max(Model_1\, Response_k) \right\}} \\ Model_2 \left\{ max(Model_2\, Operator_1), \cdots max(Model_2\, Response_k) \right\} \\ \vdots \\ Model_n \left\{ max(Model_n\, Operator_1), \cdots max(Model_n\, Response_k) \right\} \end{array}\right\} \tag{4.5}$$

Fig. 4.7 illustrates a visualisation of the above example of a shape descriptor, where the number of models images is 17 and 1 operator is applied to each model image. Therefore each model image is represented by a vector of 17 maximum values. The diagonal values in this descriptor have the highest responses. These values correspond to the model image being applied to it's corresponding COSFIRE operator from the configuration process, thus having the highest value.

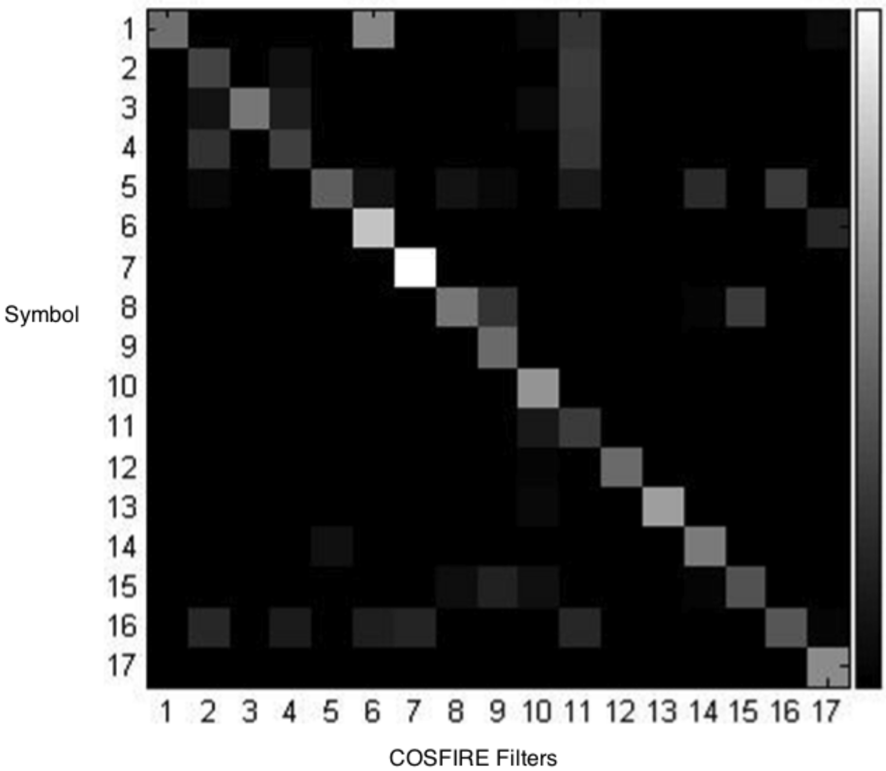


Figure 4.7: A row i represents the feature vector of a symbol model composed of the maximum responses of COSFIRE filters. Element j of a feature vector i is the maximum response of a COSFIRE filter which is configured by model symbol j

4.7 Classification

~~Once that the shape descriptor is built upon the model images,a classification technique can be applied to a specific data set's test images in order to recognise them.~~

The classification process is defined as follows.

Similarly to when creating the shape descriptor, when a test image needs to be classified, all the operators produced in the configuration phase are applied to it. Upon applying all the available operators to the test image, all the operator will produce a response in the form of a matrix of values. From each of these responses produced by the operators, only the maximum value is considered as well. This means that a test image is also described by a vector of values with a size that is equivalent to the number of COSFIRE filters used.

~~It is very crucial for the classification process that both the model and test images are represented in the same way, as a vector of values of the same length. Since both the models in the shape descriptor and the test image share the same form as vectors, they can be classified using a k-nearest neighbour classification technique.~~ This is achieved by computing the Euclidean distance between the test image feature vector to each training feature vector in the shape descriptor. A test vector is then classified to the symbol for which the Euclidean distance is the smallest, therefore the most similar.

5. Results

5.1 Dataset description

5.2 Datasets

Publicly available data sets are used which are provided for GREC contests [1] [2]. These data sets are organised into three categories which have different levels of complexity due to noise, deformation, rotation and scaling.

The first category consists of 24 data sets. Each data set contains a number (varying between 25 and 150) of different electrical and architectural symbols each represented by a single image. These data sets contain 1000 test images of deformed symbols of increasing complexity. For instance, the least complex data set contains images that are only slightly deformed, while the most complex data set consists of images of symbols with the highest deformation. The second category consists of 7 data sets, each comprising 150 different symbols. The first 3 datasets each have a different type of noise applied to their test images. The remaining data sets contain images of geometrically transformed symbols, such as different orientations and scale. The last category contains three data sets. They contain large number of different model symbols and large number of test images that combine different degradation levels and different transformations.

5.3 testing with datasts

5.4 Subsection for all datasets.

6. Discussion

7. Conclusion

A. Appendix

Bibliography

[1] International symbol recognition contest 2011. Available from http://mathieu.delalandre.free.fr/projects/sesyd/symbols/isrc2011.html. Accessed on: 15/11/2012.

[2] Sketched symbols. Available from http://mathieu.delalandre.free.fr/projects/sesyd/symbols/sketches.html. Accessed on: 15/11/2012.

[3] L. Cordella and M.Vento. Symbol recognition in documents: a collection of techniques. *International Journal on Document Analysis and Recognition*, 3(2):73–88, 2000.

[4] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the optical Society of America*, 2(7):1160–1169, 1985.

[5] G.Azzopardi and N.Petkov. Trainable cosfire filters for key point detection and pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):490–503, 2012.

B. Evaluation