Simulink Component Recognition Using Image Processing

Ramya R, Anand Kumar S, Krinish N K, Suraj V

ABSTRACT: In early stages of engineering design, pen-and-paper sketches are often used to quickly convey concepts and ideas. Free-form drawing is often preferable to using computer interfaces due to its ease of use, fluidity and lack of constraints. The objective of this project is to create a trainable sketched Simulink component recognizer and classifying the individual Simulink components from the input block diagram. The recognized components will be placed on the new Simulink model window after which operations can be performed over them. Noise from the input image is removed by Median filter, the segmentation process is done by K-means clustering algorithm and recognition of individual Simulink components from the input block diagram is done by Euclidean distance. The project aims to devise an efficient way to segment a control system block diagram into individual components for recognition.

Keywords: MATLAB, Median filter, K-means clustering algorithm

INTRODUCTION
This project presents a new computational model for automatically interpreting hand-drawn sketches of schematic diagrams. Our model employs a multi-level parsing and recognition architecture. Our approach allows user to continuously sketch without having to identify when one symbol ends and another begins. Additionally, it does not restrict the number of strokes in a symbol, or the order in which they are drawn. Hence, it eliminates many of the unnatural constraints imposed by existing sketch understanding systems, such as limitations to single-stroke objects, or the need for user involvement in separating different symbols.

EXISTING METHOD:
Our approach differs from earlier techniques in that it acts selectively in the early stages to identify small sets of easily recognizable “markers” symbolizing system components. These markers anchor a spatial analysis which parses the interpreted strokes into distinct clusters, each representing a single symbol. Finally, a symbol recognizer, informed by clustering and domain-specific knowledge, is used to find the best interpretation of these strokes. We have argued that this approach has the advantage of quickly guiding the recognizer in a direct manner while preventing unfruitful explorations.

This work emphasizes that techniques aimed at uncovering the underlying structure of a sketch, such as preliminary recognition and stroke clustering, can have significant benefits in improving computational efficiency and recognition accuracy. The computational cost of the resulting system is low enough to be suitable for interactive sketch understanding.

PROPOSED METHOD:
To demonstrate our techniques, we have built SimuSketch, asketch-based interface for Matlab’s Simulink package, and VibroSketch, asketch-based interface for analysing vibratory mechanical systems. In both systems, users can construct functional engineering models by simply sketching them on a computer screen. Users can then interactively manipulate the sketched system's parameters and run simulations.

PROJECT FLOW:

- Input Image
- Noise removal by median filter
- Segmentation by Enhanced k-means
- Recognition using Euclidean Distance
- Model Generation

Fig 1.1 Flow of Project

The input image is scanned by the program. Noises are removed with the help of median filter. Objects on the image are segmented using K-means clustering algorithm.

- Ramya R, Anand Kumar S, Krinish N K, Suraj V
- Assistant Professor, Department of Electronics and Instrumentation Engineering, Sri Ramakrishna Engineering College, India
- Department of Electronics and Instrumentation Engineering, Sri Ramakrishna Engineering College, India
The separate objects are recognised as their respective Simulink models. The new model is generated using the recognised components.

DIGITAL IMAGE PROCESSING
The Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modelled in the form of multidimensional systems.

GOALS OF DIGITAL IMAGE PROCESSING
The goal of digital processing is to improve the visualization of pathology by optimizing these physical parameters. Processing parameters need to be chosen correctly in order to overcome the inverse relationship between contrast and latitude while producing images that retain a conventional appearance. UMF is a simple technique for improving image quality. This technique, however, suffers from serious drawbacks, such as the suppression of pathologic lesions or artifacts that may simulate pathology. Manufacturers have developed different approaches in order to overcome problems and artifacts derived from this technique.

APPLICATIONS
Digital camera images
Digital cameras generally include dedicated digital image processing chips to convert the raw data from the image sensor into a colour-corrected image in a standard image file format. Images from digital cameras often receive further processing to improve their quality, a distinct advantage that digital cameras have over film cameras. The digital image processing typically is executed by special software programs that can manipulate the images in many ways. Many digital cameras also enable viewing of histograms of images, as an aid for the photographer to understand the rendered brightness range of each shot more readily.

Film
West world (1973) was the first feature film to use digital image processing to pixel late the android’s point of view.

Intelligent transportation systems
Digital image processing has wide applications in intelligent transportation systems, such as automatic number plate recognition and traffic sign recognition.

COMPONENT RECOGNITION
There have been numerous effortstocreate experimental sketch understandingsystems. This section begins with discussion of work focused on sketch parsing and recognition, and then surveys existing sketch based applications.

SYMBOL RECOGNITION
Graph based methods have been one of the most prominent approaches to object representation and matching, and have recently been applied to hand drawn pattern recognition problems. With these methods, sketched symbols are first decomposed into basic geometric primitives, such as lines and arcs, which are then assembled into a graph structure that encodes both the intrinsic attributes of the primitives and the geometric relationships between them. Pattern detection is then formulated as a graph-sub graph isomorphism problem, a problem extensively studied by computer vision practitioners. This approach has been used to recognize machine drawn symbols, symbols drawn using templates, and precise hand-drawn symbols. This method provides automatic training, although the drawing order must be consistent across the training examples. These sorts of graph-based approaches are sensitive to segmentation errors, and graph matching can be expensive. This makes our approach less sensitive to segmentation errors and drawing variations. As an alternative to graphical methods, there is a recognizer that uses some special geometric properties of particular shapes. Astishard-coded, this recognizer is not easily extended to new symbols. Each shape is described by four geometric features calculated from the set of special polygons defined by the convex hull of the shape. Because this recognizer works from the convex hull properties, it cannot distinguish between different shapes with the same convex hull.

SKETCH INTERPRETATION SYSTEM
A few sketch-based interfaces have been developed for interpreting electrical circuit sketches. Its uses hard-coded recognizers that are fixed drawing order. Also, the system avoids issues of parsing by requiring the user to paste symbols. Gates must be drawn in either one or two strokes. It describes a trainable recognizer for electrical circuits. Symbols are classified by comparing symbol’s atribute graph to that of a probabilistic model of the learned symbol. In addition to electric circuits, recent years have seen the development of experimental sketch-based interfaces for a variety of other disciplines. The program’s task is to determine what the geometry of the sketch should have been to make the sketched device behave as intended. To do this, the program employs anovel behavioural representation, called qualitative configuration space (qcspace), that captures the behaviour suggested by the sketch while abstracting away the particular geometry used to suggest that behaviour. The program is concerned only with the high-level interpretation of the sketch, and it does not consider the low-level issues of parsing and symbol recognition.
PARSING AND RECOGNITION ARCHITECTURE

This approach has a number of distinct advantages. First, by focusing on marker symbolearlyon, it avoids unfinitive explorations and quickly directs the analysis in the right direction. Second, the approach provides a platform for encoding contextual knowledge. For example, at the conclusion of the clustering step, the system can narrow down the set of possible interpretations for each symbol. This both increases recognition accuracy and reduces recognition cost. Finally, once the initial analysis is complete, our system can use domain knowledge to identify and correct errors that may have occurred during parsing and recognition. While there are many different domains that a make use of the mark-group-recognize architecture outlined above, this thesis demonstrates its utility in two domains. The first involves network diagrams in which a set of symbols (nodes) are connected by a set of arrows. The second involves vibratory mechanical systems which typically contain objectssuch as masses, dampers, springs, external forces, and grounds.

FEATURE EXTRACTION

After resampling, a feature vector is computed forming the input to the neural network. Unlike the speed information used in the previous recognizer, this recognizer uses information related to the inverse curvature of the sampled stroke. Their inverse curvature is represented as the cosine of the angles between lines segments connecting consecutive points. Although the cosine is not precisely the inverse curvature, it is closely related to it and is suitable for our purposes. Hence, the same techniques used for identifying the key points in these speed-based recognizer could be used in this recognizer. Based on the key points, we could then use the same geometric test described in Section 4.2 to decide whether the stroke is a stroke or a segment. We have tested this idea, and found that the performance of the hard-coded geometric teststohesame, regardless of whether the key points are determined basing on the speed profile or the curvature profile.

Average

The average brightness of a region is defined as the sample mean of the pixel brightness within that region. The average, ma, of the brightness over the Λ pixels within a region (9) is given by:

\[ \text{Avg} = \sum a[m, n] \]

Alternatively, we can use a formulation based upon the (unnormalized) brightness histogram, h(a) = Λ*p(a), with discrete brightness values a. This gives:

\[ m = \sum a \cdot h[a] \]

The average brightness, ma, is an estimate of the mean brightness, μa, of the underlying brightness probability distribution.

Standard deviation and Variance

The unbiased estimate of the standard deviation of the brightness within a Region with Λ pixels is called the sample standard deviation and is given by: Often, we want some information about the precision of the mean we obtained. We can obtain this by determining the standard deviation of the sample mean. The standard deviation of the mean is related to the standard deviation of the distribution by:

\[ \sigma_{\text{mean}} = \frac{1}{\sqrt{N}} \sigma \]

Where N is the number of observations in the sample used to estimate the mean. This can easily be proven with (see basic properties of the variance):

Covariance

For the covariance approximation we use the first-order Taylor expansion in matrix form

\[ h(Y) = h(Y) + V h(Y) (Y - Y) \]

Where,

\[ V = [\& \ldots 21] \]

Taking the covariance of both sides yields the following

Well known approximation

\[ \text{Cov}(B) = \text{Cov}(h(Y)) = \text{Var}(h(Y)) = V h(Y) \text{Vh}(Y) \]

STROKECLUSTERING

This section describes an algorithm to locate the distinct symbols in the Simulink domain. Note that this step is concerned only with stroke-clustering and not recognition. Recognition is deferred until later after additional sources of information, such as context, have been considered. The arrow analysis identifies the arrows in the sketch but leaves the rest of the strokes uninterpreted. The goal in this step is to group the strokes into different clusters such that each cluster contains a distinct Simulink object that can be subsequently recognized. The key idea behind strokeclustering is that strokes are deemed to belong to the same symbol only when they are spatially proximate. The challenge is to reliably determining when two open strokes should be considered close together.

Here, we rely on arrowstohelpmakethis determination. Each network diagram connects a source object to an object of interest. Hence, different clusters can be identified by grouping together all the strokes that are near each other.

SYMBOLRECOGNITION

This chapter presents an overview of the three symbol recognizers we have developed, together with a comparative
evaluation of the advantages and disadvantages of each. A detailed presentation of each of the recognizers is contained in the following chapters. A short description of the first two recognizers was presented previously.

**Image-Based Recognizer**

The development of the first recognizer was inspired by techniques from image processing. Symbols are internally represented as quantized bitmap images; we call "templates." Each symbol is centered within its template, and is uniformly scaled to fill it, thus making the approach insensitive to uniform scaling. One distinct advantage of this recognizer is its tolerance to rotational and translational invariance. We call these new definitions from single prototype examples. An unknown template is matched to a definition template within an ensemble of four different classifiers. These classifiers are extensions of the following methods:

1. Hausdorff distance
2. Modified Hausdorff distance
3. Tanimoto coefficient
4. Yule coefficient

These classifiers were originally intended for matching precise bitmaps. These rankings are combined, and the definition with the best combined score is selected. In practice, the combined performance is better than that of any of the individual classifiers. To achieve rotation invariance, the recognizer uses an novel polar coordinate analysis that avoids expensive computations in the drawing coordinates. The recognizer is versatile in that we use both analytical and digitized recognition.

**K-means Clustering Algorithm of Segmentation**

The K-means is another simple algorithm of segmenting or classifying images into k different clusters based on feature, attribute or intensity value. It is computationally efficient and does not require the specification of many parameters as compared to other methods of segmentation. Unlike local thresholding, which can only group into two main classes while K-means Algorithm can group into k different classes and that is part of the reason why we chose as segmentation method for this work. The classification is done by minimizing the sum of the squares of distances between data and the corresponding clustering centroid. Type of distance calculation compatible with K-means Algorithm includes Mahalanobis and Euclidean distance etc.

**LIMITATION AND FUTURE WORK**

Obviously, this work is only a small step toward achieving truly natural and practical pen-based computer interaction and there are many issues that remain unsolved. Some of these issues are not directly addressed by this study. For instance, our techniques are developed primarily for 2D sketches. Hence, it is not clear whether they can be extended to sketches of 3D scenes. Likewise, our techniques are most suitable for interpreting "schematic" sketches, and are not particularly useful for sketches that are "artistic" in nature. Also, although it would be interesting to study, this work does not investigate how well our techniques can be employed in small handheld devices, such as personal digital assistants.

**REFERENCES**


