Translating an electrical circuit schematic into a SIMULINK model

Andreas Boltres
Karlsruhe Institute of Technology
Kaiserstr. 12, 76131 Karlsruhe, Germany
andreas.boltres@student.kit.edu

Cem Gülşan
Hamburg University of Technology
Am Schwarzenberg-Campus 1, 21073 Hamburg, Germany
cem.guelsan@tuhh.de

Christian Alexander Vecsei
RWTH Aachen
Templergraben 55, 52062 Aachen, Germany
christian.vecsei@rwth-aachen.de

Thomas Frei
University of Applied Sciences and Arts Northwestern Switzerland
Bahnhofstrasse 6, 5210 Windisch, Switzerland
thomas.frei1@students.fhnw.ch

Abstract

We present a functionality developed within the computing environment MATLAB, which analyses schematics of electrical circuits and transforms them into an electrical circuit model of MATLAB's SIMULINK environment.

1. Introduction

1.1. Motivation

Students and engineers often use schematics while studying or developing. A schematic is a drawing of an electric circuit which should contain all information about the circuit but should also be as simple as possible.

With increasing size, electrical circuits often become more complicated and confuse less experienced engineers, rendering the estimation of the circuit's behaviour hard or impossible.

A program that confidently identifies the components of a schematic with their respective connections and translates them into a user-friendly digital model can substantially accelerate the understanding of the given circuit. Analyzing, modifying and storing a schematic becomes easy once it is properly digitized.

Figure 1. Example circuits

1.2. Program overview

The aim of this student project is, as stated in the abstract, to create a functionality developed within the computing environment MATLAB, analyzing schematics of electrical circuits and transforming them into an electrical circuit model within MATLAB's Simulink environment. There, it can be used to facilitate calculations or behaviour analyses. The process of obtaining the Simulink model from the given image can be divided into three phases which will be briefly described in the following sentences and in detail in the following sections.

Figure 2. Top: example screenshot of a circuit. Bottom: example photography of a circuit

Initially, the program expects a digital image to be selected by the user. This image can be either a photography of a schematic on a sheet of paper or the image of a digitally created schematic. If the image is a photography, the program will first detect the boundaries of the sheet of paper in order to then rectify the image such that the sheet of paper lies entirely in the image plane. This implies that the entire paper is clearly included in the photography. After rectification, the photo can be treated in the same way digitally created schematic pictures are to be treated.

Figure 3. ROI selection within a rectified photography

The user selects a region of interest (ROI) within the provided (rectified) image; the following analysis will be limited to that specific ROI. The first step of the image analysis algorithm is **Optical Character Recognition** (OCR), as every electrical element comes with electrical properties textually encoded next to them in the image. Recognized text is stored and deleted from the image. Hereby OCR is running in a loop: After each OCR iteration, the program asks the user to confirm the removal of all text within the selected ROI. If this is not the case, he is asked to manually select new regions of interest so that the OCR routine can obtain and remove the remaining text.

Following up is the **retrieval of geometric primitives** within the images which will be further explained in subsection 3.1. Obtained through the binarized and thinned image, these primitives are used to retrieve the scale of the electrical components of the schematic in the image.

Using the knowledge of the components' scale, the **object detection** is done using a sliding window to retrieve the location and orientation of electrical elements in the image. Following up is the **detection of wires** based on the image after deletion of the detected elements. See subsections 3.2 and 3.3 for more detailled descriptions.

Finally, after having detected elements and the connections in between, the **output generation** model uses the objectified results of the image analysis and creates and fills a Simulink model. This model will then be shown and saved for further usage. Refer to section 4 for further imformation.

1.3. LTSpice

TODO 1

TODO auch auf normierung eingehen

Figure 4. Examples of LTSpice-conform circuit elements

TODO 2

2. Preprocessing

Before being able to obtain the electrical circuit elements from the provided image, several steps have to be taken in order to facilitate the information retrieval.

2.1. Rectification

When dealing with photographies, the depicted schematic may be distorted due to the surface not being aligned to the image plane. This has to be taken care of in order to use the object recognition approach presented in this work. Therefore, asserting that the entire surface and its corner points are included in the image, a rectification step is taken to bring the schematic onto the image plane.

TODO explain rectification

Figure 5. Pre/Post-rectification img

2.2. Optical Character Recognition

In depictions of non-distorted or rectified schematics, MATLAB's OCR functionality can be used to retrieve text and its position within the image. After retrieval, the corresponding text regions in the image are erased, simplifying the following recognition of electrical elements and connections.

Figure 6. Some ocr img

TODO OCR explanation

2.3. Binarization, Morphology

After all text has been recognized and erased from the image, a binarized and thinned-out version of the image is prepared to further facilitate the following retrieval of lines, circles, corners and element template matchings. Both operations are done using pre-defined functions within MATLAB.

Figure 7. Left: image region before binarization and line thinning. Right: the same image region after binarization and line thinning.

3. Image Analysis

The goal of the functions within the image analysis package is to retrieve all objects from the pre-processed image. This includes the electrical components themselves as well as the wires connecting them. The approach used for the detection of the electrical components is a sliding-window approach, implying that a robust estimation of the scale electrical components is essential in order to find the objects quickly. This in turn calls for the usage of geometric primitives in order to determine that scale.

3.1. Geometric primitive retrieval

For lines, a parametrized hough transformation is used. The results are rearranged into a data structure including length and angle of all lines, using the given start- and endpoints of the lines.

For circles and corners, which are unused in the current version of the program, predefined MATLAB functionality is used to detect the harris features of the image and the centers and radii of circles are found using a circular hough transform.

In order to retrieve the scale of the elements, skewed lines are used because in rectified schematics conforming to LTSpice, they are used in resistors only. They can be used to retrieve their dimensions, when assuming that every schematic has at least one resistor and every resistor is conforming to the LTSpice regulations. Then, the greater one

Figure 8. Top: pre-processed image region. Bottom: retrieved geometric primitives: circles in blue, lines parallel to the axes in red, skewed lines in green and corners as green crosses.

of the coordinate differences of full-length lines (of which one is highlighted in Image 9) along the axes corresponds to the shorter side of the resistor whereas four times the coordinate difference along the other axis corresponds to the longer side of the resistor.

Figure 9. Resistor template with highlighted full-length line and resistor dimension explanations

Using this heuristic, the size of a resistor can be estimated within tolerance of a few pixels. Furthermore, including the stretched half-length lines which lie at the end of resistors, the algorithm becomes more robust.

3.2. Object Detection

The general approach used for object detection is a sliding-window approach. The structure of object detection goes as follows:

- Specific pre-processing for object detection
- Sliding window: Generation of an error-image (difference from template to reference)
- Get position of minimum error/maximum correlation
- Ranking and selecting candidates
- Return found elements

Preprocessing for Object Detection

Before the sliding window can be applied, the image needs to undergo specific preprocessing, which is an integral part of object detection. The template (i.e. a resistor) and the reference image (i.e. the circuit) are first binarized and then dilated. The dilating factor is an essential parameter and influences how tolerant the algorithm is of dissimilarities between the template image and actual representations of the template in the reference image. Increasing the dilation factor increases the change to find high-correlation areas but also decreases accuracy and precision. In figure 10 and figure 11 we can see a template and a reference image being (heavily) dilated.

Figure 10. Dilated template

Sliding window and error-image

The preprocessed template is now sliding over the reference image, while a normalized mean-squared error (or difference value) is calculated at each position. The error values form the error image. Figure 12 and13 each show a dilated reference image on the left and the corresponding error visualization on the right. Hereby, figure 12 shows much lower dilation values than figure 13. Observing the error patterns, it is clear that the highly dilated image yields better results, but also more maxima where none should be.

Figure 11. Dilated reference image

Figure 12. Slightly dilated reference image and resulting error image

Figure 13. Heavily dilated reference image and resulting error image

Get Maximum Correlation

Using the error image (or inverted: the correlation image), the algorithm tries to find significant maxima in the image. Noise and similarities between objects (e.g. grounds and resistors, or capacitors and wires) make the search for maxima difficult. The algorithm first supresses all values below a certain dynamic threshold and then binarizes again, using another dynamic threshold. Occasionally, the final binarization fails to produce good results and higher dilation values must be used.

Figure 14 to 16 show the different attempts to identify a capacitor, which is particularly hard to detect with certainty due to its very simple geometric shape. In figure 14, the target maximum is not significant enough and its value is too close to the rest of the error-image. The final binarization results in an all positive image. In figure 15, other areas have significant - although faulty - maxima aswell, resulting in a faulty final binarization. Figure 16 shows a successful suppression of minima and a successful binarization, with an isolated, significant dot as result.

The algorithm detects those isolated dots and decides whether it has found an object at concerned position, whether it needs to use different parameters or whether there is no element of this kind to be found.

Figure 14. Faulty binarization - main maximum too insignificant

Ranking and Selecting Candidates

As mentioned before, the algorithm is prone to confuse different elements with each other, provided they share geometric similarities. When dilating template and reference image, this confusion gets worse as the tolerance for non-perfect matches increases. Frequently, multiple elements are detected (with varying tolerance) in the same area in the reference image. To counteract this, the algorithm assigns a score to each found element relating to its certainty. After the algorithm searched for all elements, it checks whether the found elements are out of bounds (the edges of the reference image are prone to faulty maxima) and compares their store against each other.

Figure 15. Faulty binarization false maxima detected

Figure 16. Successful binarization

Figure 17 shows all found candidates for this particular example. Attached to each found element is a confidence score, with lower scores being better. When two or more elements occupy the same area, the element with the lower score gets priority and the others are deleted. The result of this selection process can be seen in figure 18.

Return Found Elements

The found and selected elements are stored and returned to the main image analysis function for further processing. In the image, the areas of the found elements are erased to enable the following wire detection.

3.3. Wire Detection

After having detected the electrical elements and after having erased them from the image, cables and connections are the only entities left in the image. Therefore, a second hough transformation application retrieves the endpoints of the connections between elements.

Figure 17. All candidate elements with scores

Figure 18. Selected candidates (final result)

Figure 19. Top: circuit image with detected elements. Bottom: The same circuit without the elelctrical elements and the recognized cables.

After the wire detection, all detection results are passed onto the output generation module.

4. Output Generation

In this module, the results of the image analysis routines are reframed and sent to SIMULINK in order to build an electrical circuit model. After its creation, the user can interact with the model which is also saved to enable further usage.

4.1. SIMSCAPE Electrical Model

TODO Chris describe Model

Figure 20. some model pic

4.2. Component Generation and Assembly

TODO Chris describe the reshaping of our data so that you can use the SIMSCAPE Api

Figure 21. some result model pic

5. Evaluation

Extensive evaluation is a key technique to determine the feasibility of the approach and its parts presented in this work. Therefore, every functionality should be tested qualitatively as well as quantitatively. For this purpose, a bunch of electrical circuit schematics have been created; the algorithm's parts will be tested on each of these images.

5.1. Rectification

TODO

5.2. OCR

TODO

5.3. Scale Retrieval

In order to evaluate the performance of the scale retrieval routine, the dimensions of the resistors of each image has been measured manually. Thereafter, each image was sent through the program until the resistor dimensions were retrieved. In table 1, the dimensions of resistors in each image (column *Measured*) can be found alongside the dimensions retrieved by the algorithm for each picture (column *Calculated*) and a maximum deviance of the resistor size for each dimension. A manual measurement tolerance of one pixel is included in this maximum deviance metric.

File	h/v	Measured	Calculated	Deviance
01.png	horiz.	180×91	180×88	$< 1 \times 4$
02.png	vert.	91×180	90×194	$< 2 \times 5$
03.png	horiz.	180×90	176×87	$< 5 \times 4$
04.png	horiz.	140×71	140×69	$< 1 \times 3$
05.png	vert.	46×92	46×96	$< 1 \times 5$
06.png	vert.	46×92	46×96	$< 1 \times 5$
07.png	horiz.	206×103	208×102	$< 3 \times 2$
08.png	vert.	108×215	106×216	$< 3 \times 2$
09.png	vert.	85×169	83×168	$< 3 \times 2$
10.png	vert.	127×255	126×256	$< 2 \times 2$
11.png	horiz.	216×108	216×107	$< 1 \times 2$
12.png	vert.	109×216	108×212	$< 2 \times 5$
13.png	horiz.	199×100	200×98	$< 2 \times 3$
14.png	vert.	94×185	91×184	$< 4 \times 2$
15.png	horiz.	265×133	264×130	$< 2 \times 4$
16.png	horiz.	215×108	216×106	$< 2 \times 3$
17.png	vert.	98×195	94×192	$< 5 \times 4$
18.png	vert.	108×215	108×224	$< 1 \times 10$
19.png	vert.	118×235	114×232	$< 5 \times 4$
20.png	horiz.	245×122	248×120	$< 4 \times 3$

Table 1. Resistor size measurements and estimations (in px)

A maximum deviance of 5 pixels is considered satisfactory for an effective usage of the object detection algorithm. With the exception of one case, the scale retrieval algorithm could retrieve the resistor size in all images it got called on. This corresponds to an accuracy of 95%.

5.4. Object Detection

TODO

5.5. Wire Detection

TODO

Contributions

Name	Intro	Research	Presentation	Spirit
Andreas Boltres	25%	25%	25%	250%
Cem Gülşan	25%	25%	25%	250%
Christian Alexander Vecsei	25%	25%	25%	250%
Thomas Frei	25%	25%	25%	250%
	100%	100%	100%	1000%

Table 2. List of members and their contributions to the project.