Infrared Image Processing for Solar Cell Defect Detection

Armen Petrosyan
Institute for Informatics and Automation Problems, National Academy of Sciences of Armenia
Yerevan, Armenia
e-mail: armenpet@ipia.sci.am

Anna Hovhannisyan
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: anna.l.hovhannisyan@gmail.com

ABSTRACT

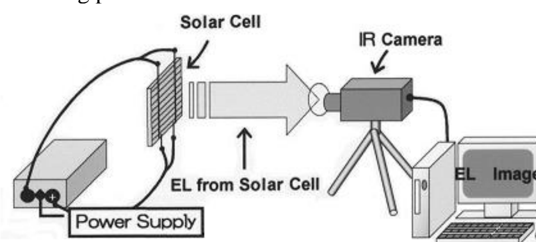
The detection of solar cell defects is highly important in the development, production and use of photovoltaic modules. Solar cell defects can be detected by different methods including electroluminescence, photoluminescence and thermography. In this paper, the algorithm for solar cell defect detection by digital image processing using infrared image acquired from thermal camera is proposed. The algorithm is based on comparison of infrared images of testing cell and the sample of high quality solar cell. For the algorithm realization, NI LabVIEW and NI Vision Development Module have been used. The suggested method of defects detection allows to automatically find the location of the defect with high accuracy.
This is a practical example, which was tested on the production line using solar cell and thermal camera.
The main idea used in the image processing part of the system is particle analysis. It's described in the paper.

Keywords

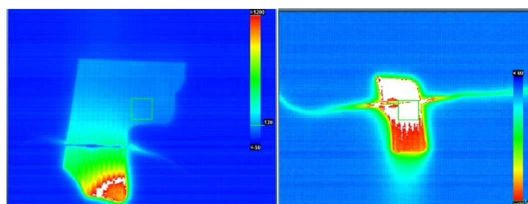
Solar Cell, Electroluminescence, Particle analysis, Digital Image Processing, LabVIEW, Machine Vision.

1. INTRODUCTION

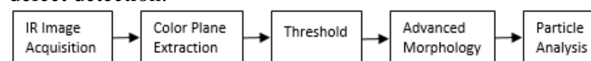
In the 21-st century the technologies allowed people to think about alternative energy resources, which are more clear and natural. One of such kind of energy resources is solar energy, which could be gathered by using solar cells.
The solar cells initially were expensive, but nowadays they are becoming cheaper and can be used instead of traditional energy resources in many countries. The main challenge in this area is the quality of solar cells and its robustness. So, it is important to perform an initial inspection for detecting the defects on solar cells.
There are different methods of solar cell investigation. Module defects such as cracks, soldering and interconnection faults, potential induced degradation, and hot spots can often be characterized by imaging techniques including photoluminescence (PL), electroluminescence (EL), and thermography. These methods are contactless methods for detecting defects on solar cells. Infrared imaging allows anomalies to be seen clearly and can be used to scan installed solar panels during normal operation.
The goal of this paper is to show the methods of solar cell defect detection using different image processing approaches. However, the main principle of achieving good results is to define the defect types in digital (infrared in current case) image, then try to determine those defects using some pre-processing algorithms, then localize their positions and finally distinguish or classify the type of defects.
In collaboration with the Yerevan branch of National Instruments, a functional system was developed and realized, which is detecting the defects on infrared image of solar cell, localizing the position, the area and other characteristics of the defects [1].
In our setup, we use NI SMU 4130 as a power source, which is a very precise programmable power source unit. To change the voltage programmatically and do the measurements it is written a code using LabVIEW programming environment.
The figure below shows a typical hardware setup for solar cell measurements. The solar cell is initially connected to a power source, and for testing purposes the voltage and current of power supply are changing, and the camera is acquiring a bunch of images for different voltages and currents. At the end of acquisition phase, some quantity of IR images is obtained, which can then be transferred to the processing phase.



The features of the research are: a) a new approach of acquiring infrared images by electroluminescence and b) a method (sequence of algorithms) of image processing to find the defects on IR images automatically. Below are some infrared images captured by electroluminescence.



Despite there are several possible image processing solutions for the given task, in this paper it is introduced an approach that converts the infrared image to suitable plane for processing, then using thresholding technique brings it to a binary form and does some morphological filtering on the binary image and at the end uses particle analysis algorithm to get separate particles and determine the characteristics of the defects [2].
The following schema presents the proposed approach for defect detection.



Particle analysis algorithms are one of the most important functions in machine vision, because of their use in varying applications as object classification, segmentation and inspection [3].
Particle analysis can be used to detect connected regions or groupings of pixels in an image and then make selected measurements of those regions. These regions are commonly referred to as *particles*. A particle is a contiguous region of nonzero pixels. Particles can be extracted from a grayscale image by thresholding the image into background and foreground states. Zero valued pixels are in the background state, and all nonzero valued pixels are in the foreground.

Particle analysis consists of a series of processing operations and analysis functions that produce information about particles in an image. Using particle analysis, it is possible to detect and analyze any 2D shape in an image.
Particle analysis consists of three main parts:

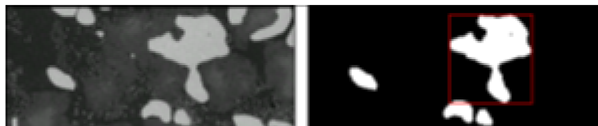
- Image Segmentation – contains information about segmenting images using global grayscale thresholding, global color thresholding, local thresholding, and morphological segmentation.
- Binary Morphology – contains information about structuring elements, connectivity, and primary and advanced morphological transformations.
- Particle Measurements – contains information about characterizing digital particles.

2. THE THEORY OF THE APPROACH

Below are introduced the concepts of particle analysis algorithms, such as thresholding, morphology, and particle measurements [3].

2.1. Concepts and use cases

In applications where there is a significant variance in the shape or orientation of an object, particle analysis is a powerful and flexible way to search for the object. You can use a combination of measurements obtained through particle analysis to define a feature set that uniquely defines the shape of the object [4].
A typical particle analysis process scans through an entire image, detects all the particles in the image, and builds a detailed report on each particle. You can use multiple parameters such as perimeter, angle, area, and center of mass to identify and classify these particles. Using multiple parameters can be faster and more effective than pattern matching in many applications.
Also, by using different sets of parameters, you can uniquely identify a feature in an image. For example, you could use the area of the template particle as a criterion for removing all particles that do not match it within some tolerance. You then can perform a more refined search on the remaining particles using another list of parameter tolerances.
The following figure shows a sample list of parameters that you can obtain in a particle analysis application. The binary image in this example was obtained by thresholding the source image and removing particles that touch the border of the image. You can use these parameters to identify and classify particles.



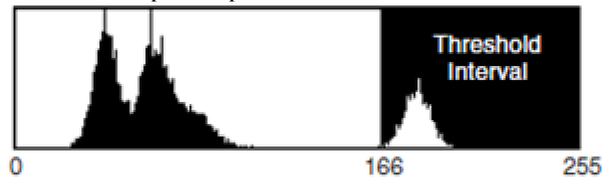
The following table shows the values obtained for the particle enclosed in a rectangle, shown in the figure below.

Particle Measurement	Values
Area	2456
Number of Holes	1
Bounding Rectangle	
Left	127
Top	8
Right	200
Bottom	86
Center of Mass	
X	167.51
Y	37.61
Orientation	82.36º
Dimensions	
Width	73
Height	78

To use particle analysis, first it is necessary to create a binary image using a thresholding process. After that, it will be needed to improve the binary image using morphological transformations to make better measurements on the particles in the image.

2.2. Image Segmentation and Thresholding

Image segmentation is a process of separating objects from the background and each other so that each object can be identified and characterized.
The first step is thresholding, which segments an image into a particle region, which contains the objects under inspection and a background region based on the pixel intensities within the image. The resulting image is a binary image. There are several methods of thresholding, such as global grayscale thresholding, global color thresholding, local thresholding, and morphological segmentation. Global thresholding works best when the inspection images exhibit uniform lighting both within each image and across multiple images. Particles are characterized by an *intensity range*. They are composed of pixels with gray-level values belonging to the given threshold interval. All other pixels are considered to be a part of the background. *Local thresholding*, also known as locally adaptive thresholding, is like global grayscale thresholding as both of them create a binary image by segmenting a grayscale image into a particle region and a background region. Unlike global grayscale thresholding, which categorizes a pixel as part of a particle or the background based on a single threshold value derived from the intensity statistics of the entire image, local thresholding categorizes a pixel based on the intensity statistics of its neighboring pixels.
Thresholding sets all pixels that belong to a range of pixel values, called the threshold interval, to 1 or a user-defined value, and it sets all other pixels in the image to 0. Pixels inside the threshold interval are considered part of a particle. Pixels outside the threshold interval are considered part of the background. The figure below shows the histogram of an image. All pixels in the image with values from 166 to 255 are considered particle pixels.



The threshold interval in a manual threshold has two user-defined parameters: lower threshold and upper threshold. All pixels that have gray-level values equal to or greater than the lower threshold and equal to or smaller than the upper threshold are selected as pixels belonging to particles in the image. In contrast to manual thresholding, automatic thresholding techniques do not require to set the lower and upper threshold values. These techniques are well suited for conditions in which the light intensity varies from image to image.
The well-known automatic thresholding techniques are:
• Clustering, • Entropy, • InterVariance, • Metric, • Moments
Clustering is the only multi-class thresholding method available. Clustering operates on multiple classes so it is possible to create tertiary or higher-level images.
The other four methods - entropy, metric, moments, and interclass variance, are reserved for strictly binary thresholding techniques. The choice of which algorithm to apply depends on the type of image to threshold.

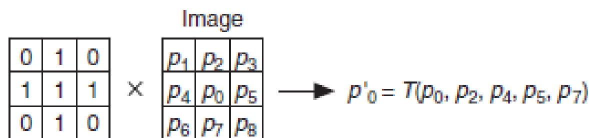
2.3. Binary Morphology

Binary morphological operations extract and alter the structure of particles in a binary image. You can use these operations during your inspection application to improve the

information in a binary image before making particle measurements, such as the area, perimeter, and orientation. A binary image is an image containing particle regions with pixel values of 1 and a background region with pixel values of 0. Binary images are the result of the thresholding process. Because thresholding is a subjective process, the resulting binary image may contain unwanted information, such as noise particles, particles touching the border of images, particles touching each other, and particles with uneven borders. By affecting the shape of particles, morphological functions can remove this unwanted information, thus improving the information in the binary image.

Morphological operators that change the shape of particles process a pixel based on its number of neighbors and the values of those neighbors. A neighbor is a pixel whose value affects the values of nearby pixels during certain image processing functions. Morphological transformations use a 2D binary mask called a structuring element to define the size and effect of the neighborhood on each pixel, controlling the effect of the binary morphological functions on the shape and the boundary of a particle.

Below is an example of morphological transformation using a 3 × 3 structuring element.



Primary morphological operations work on binary images to process each pixel based on its neighborhood. Each pixel is set either to 1 or 0, depending on its neighborhood information and the operation used. These operations always change the overall size and shape of particles in the image.

The primary morphology functions apply to binary images in which particles have been set to 1 and the background is equal to 0. They include three fundamental binary processing functions - erosion, dilation, and hit-miss. The other transformations are combinations of these three functions.

An *erosion* eliminates pixels isolated in the background and erodes the contour of particles according to the template defined by the structuring element.

For a given pixel P_0, the structuring element is centered on P_0. The pixels masked by a coefficient of the structuring element equal to 1 are then referred as P_i.

• *If* the value of one pixel P_i is equal to 0, *then* P_0 is set to 0, *else* P_0 is set to 1.
• *If* AND(P_i) = 1, *then* $P_0 = 1$, *else* $P_0 = 0$.

A *dilation* eliminates tiny holes isolated in particles and expands the particle contours according to the template defined by the structuring element. This function has the opposite effect of an erosion because the dilation is equivalent to eroding the background.

For any given pixel P0, the structuring element is centered on P0. The pixels masked by a coefficient of the structuring element equal to 1 then are referred to as Pi.

• *If* the value of one pixel P_i is equal to 1, *then* P_0 is set to 1, *else* P_0 is set to 0.
• *If* OR(P_i) = 1, *then* $P_0 = 1$, *else* $P_0 = 0$.

The advanced morphology functions are conditional combinations of fundamental transformations, such as binary erosion and dilation.

The *border function* removes particles that touch the border of the image. These particles can be truncated during the digitization of the image, and their elimination then helps to avoid erroneous particle measurements and statistics.

The *hole filling function* fills the holes within particles.

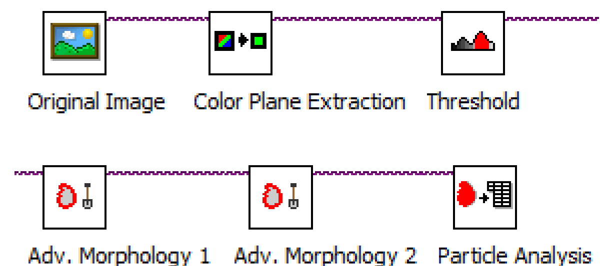
The *lowpass filter* removes small particles according to their widths as specified by a parameter called *filter size*. For a given filter size N, the lowpass filter eliminates particles whose widths are less than or equal to $(N-1)$ pixels. These particles disappear after $(N-1)/2$ erosions.

The *highpass filter* removes large particles according to their widths as specified by a parameter called filter size. For a given filter size N, the highpass filter eliminates particles with widths greater than or equal to N pixels. These particles do not disappear after $(N/2+1)$ erosions.

3. PRACTICAL IMPLEMENTATION

Returning to the real application, there are some image transformation steps, which were not discussed in previous chapter. However, we should consider all steps and highlight the technics, that is used to bring the image to a correct form. So, initially we have a color (infrared) image, acquired from IR camera. It contains thermal information about the surface in camera field of view. Hot parts are colored red, warmer parts with yellow, cool parts with green and cold parts with blue. So, the treated parts are those, which are colored red. It means, that the color detection could be implemented directly, but color operators are time consuming, and in case of real production line, algorithm complexity and speed should be considered as well.

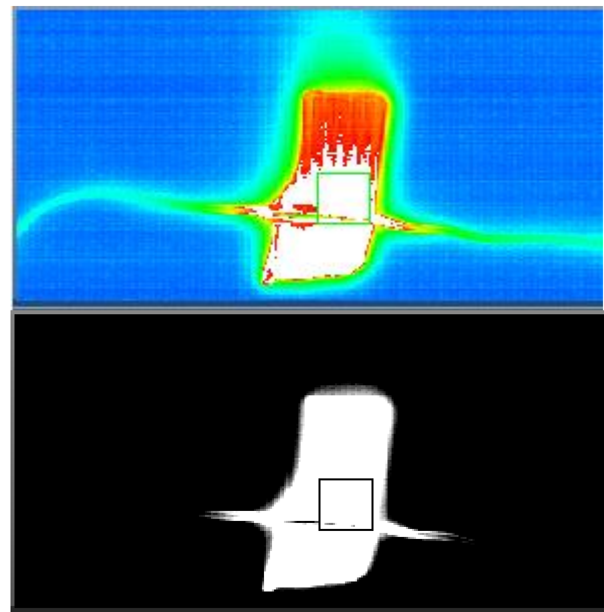
This schema is implemented on the initial color image.



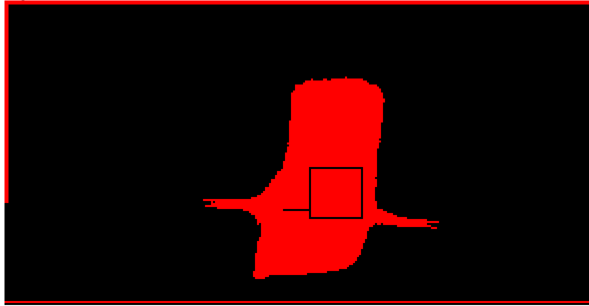
Considering the processing time, as mentioned earlier, we choose the fastest method, and converted the initial color image to the binary one with correct transformation steps.

3.1. Step-by-step description of approach

The first step of the approach is to get the required information from the color image, i.e. red color parts. So, it is obvious, that Red color plane from RGB contains exactly the data needed, and it is feasible to take red plane keeping the image as grayscale for further processing.

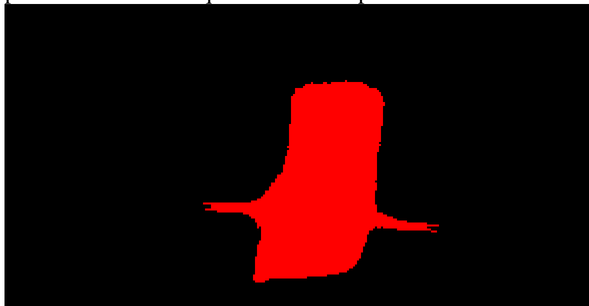


The next step is Thresholding. This is one of the important steps, as by incorrect Thresholding we could lose important information. Easiest method is to set a manual threshold if the light conditions are the same, however, for final system it would be better to develop or use one of the classical automatic methods to make the system more robust to the light conditions.



After thresholding, we get the initial binary image with possible noise and other unnecessary particles, like border lines, which we need to remove to keep only the required data. So, before particle analysis we need to implement some basic or advanced morphology. In the current approach, we used two morphological operations. The first one is *border function*, which removes particles touching image edges.

The second morphological operator is *hole filling function*, which is filling artificial black (0) pixels brought by previous steps. Another morphological function, that can be used is a *lowpass filter*, which can remove small particles (noise) in the binary image. As can be seen from the result, border particles and black square inside the particle were removed.



Now it is feasible to implement a particle analysis on the binary image and get the position, area and other parameters of the particle.

Results ...	1
Center of Mass X	316.55050
Center of Mass Y	240.86982
Max Feret Diameter	237.21720
Area	20456.0000

3.2. Processing time results

The methodology was tested on the Host Machine (usual PC or laptop), as well as on the Embedded hardware. Below are the results on i7 computer. On the Embedded hardware, the result time could be longer by several times.

Average Inspection Time: 2.33 ms
Longest Inspection Time: 3.27 ms (6)
Standard Deviation: 0.46 ms

Step Name	Average	Std-Dev	Shortest	Longest
Color Plane Extraction 1	0.43 ms	0.08 ms	0.34 ms	0.58 ms
Threshold 1	0.09 ms	0.02 ms	0.07 ms	0.12 ms
Adv. Morphology 1	0.16 ms	0.03 ms	0.13 ms	0.22 ms
Adv. Morphology 2	0.78 ms	0.16 ms	0.61 ms	1.11 ms
Particle Analysis 1	0.88 ms	0.17 ms	0.70 ms	1.23 ms

4. ACKNOWLEDGEMENT

We would like to thank the Yerevan branch of National Instruments (head: Aram Salatian) and Armenian National Engineering Laboratories (ANEL) for supporting our research. Also, Armen Petrosyan would like to thank the head of CSP department in IIAP - Hakob Sarukhanyan.

REFERENCES

[1] National Instruments LabVIEW Development System, www.ni.com.

[2] A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall Inc, Englewood Cliffs, 1989.

[3] National Instruments Corporation, “NI Vision Concepts Manual”, 2000-2009.

[4] E.R. Davies, Computer and Machine Vision: Theory, Algorithms, Practicalities, Fourth Edition, ELSEVIER, 2012.