PAPER • OPEN ACCESS

Industrial robot in fine art: can an industrial robot draw a binary image?

To cite this article: S Timoftei 2018 IOP Conf. Ser. Mater. Sci. Eng. 444 052027

View the article online for updates and enhancements.
**Industrial robot in fine art: can an industrial robot draw a binary image?**

**S Timoftei**
Machine Building, Engineering Design and Robotics Department, Technical University of Cluj-Napoca, 103-105 Muncii Avenue, Cluj-Napoca, Romania

E-mail: s.timoftei@yahoo.com

**Abstract.** This research paper investigates, through a case study, which should be the basic requirements such as an industrial robot to make drawings based on data provided from image processing. An algorithm for image processing and data sorting to be further used by the robot for lines drawing is presented in this paper. Using LabVIEW, image is converted into a binary format. Researches from this paper indicate that, due to the specific requirements needed to program linear and circular robot motions, as well as due to the specific procedure to define robot tool centre point (TCP), information embedded in the binary format necessitate transformation into points with relative coordinates to a reference frame of the drawing plane such as a robot to be able to make drawings. A conversion approach from the binary matrix to the robot drawing matrix is proposed in the paper, together with the necessary settings of the robot motions.

1. **Introduction**

Each of our daily activity is an algorithm even if it is about going from home to work, to prepare the breakfast, just simply looking for a store, or why not looking after clothes. In addition, in computer science are used algorithms to implement the science in this domain. An algorithm is nothing more than a set of steps that must be followed to accomplish a certain task. [1]

Doing research for a scientific paper implies a certain algorithm that must be followed to get a good paper. To publish the paper in a journal another algorithm is used. Big companies or organizations are using complex algorithms to achieve their goals, like offering services or protecting the people. Maybe we do not realize, but each of our activity is an algorithm formed by further steps that helps use to accomplish proposed tasks.

Existing algorithms on your researching domain can help you to finish your work faster or it can be a point of start for what you can do further. Software updates are the best example of algorithm improvements, talking about modifying the actual structure of the algorithm or just going further to the existing one.

Khan Academy, which has science as domain of interest, is using efficient algorithms to analyze data at a high level or to offer the possibility to choose from further decisions that are possible, the best one. Each algorithm has its own difficulty, they are not equal. [1]

The desire of perfection can be also found here, when it is about algorithms, because it is wanted to be certain, to be fast and to give just the only right solution, now. Starting from here, algorithms can be classified by how the problem is solved and how efficient it was created. To get a perfect algorithm,
it must take time. Another classification of the algorithms is about their type, in this way they can be defined as recursive, randomize, selection or sorting, graph algorithms. [1]

It is not enough just to have the best version of the algorithm and give a result instantly, also it is necessary that the result to be efficient. Their efficiency depends on the language programming and the computer on which it is running, things that gives a time for the algorithm to be used and get the results. In computer science, researchers use a technique, “asymptotic analysis” [1] that offers the possibility to compare algorithms in an independent way to check the hardware or the language programming and have a conclusion if one of them is better that another one. [1]

2. Image processing

This common term “image processing” is daily used by people from different domains that must do research or must take a decision based on an image of the studied subject. [2]

One of the most common and researched domains is medicine, where doctors and researchers (e.g. engineers) want to create those “tools” that can help to study in detail different healthy problems, like brain, lungs, heart issues and so on. Image processing is not only to determine a human illness, but also to do different studies about how human react in a certain situation, or why an illness is so common, like depression.

Any image that is of interest is subjected to a “quantitative analysis” that further can help to generate 3D “parametric maps” based on a set of calculations. Tendency of development in automation is increasing more and more, because of this, algorithms used for image processing are becoming more and more sophisticated. [2][7]

An image is seen as a function of 2 variables:\[ f(x, y) \] where \( f \) - the amplitude (e.g. brightness) of the image, \( (x, y) \) - coordinate position. [3] Image processing is a process, or a method used to transform a digital image, its characteristics presented before, into a better version of it, using different operations or to extract from it different data or information that can be used further. Moreover, image processing is to see things that cannot be seen, to distinguish or to measure an object in a certain image. [4][8]

Technology has evolved so much as the studies have arrived at the level of studying this process by manipulating multi-dimensional signals being not enough to use common digital circuits, just realizing a parallel network of computers. Because of this, this action can be divided in categories like [5]:

- Image Processing (image in -> image out)
- Image Analysis (image in -> measurements out)
- Image Understanding (image in -> high-level description out)

To use this process and to get good results it is necessary to have an image obtained by an “optical scanner or by digital photography” [4], to apply further techniques to analyse and manipulate an image, like “data compression and image enhancement and spotting patterns” [4], resulting a report that gives information about the initial image, information that it was wanted to be obtained. [4]

Image processing is a manipulation technique of images with the help of computer, being available two types of processes like digital and analogy image processing. A digital image processing is done with the help of a computer and needs to pass through three important stages, “using digital technique are Pre-processing, enhancement and display, information extraction.” [4] (see figure 1).

Rob Lind explained in its chapter [5] how open source software can be used to analyse an image in image processing. For this he took as example open source software, ImageJ that is using Java, thing that is giving it certain flexibility. The flexibility of the software is given by the source architecture, being very used with academic aims. Java language programming offers the possibility to be used on different operating systems, so the executable code can be easily used by different researchers from different places of the world. [5]
3. Study case on an industrial robot

In this section of the paper, it will be presented an algorithm realized on a program based on the researches done until now. What must be mentioned is that the results obtained by processing the image, must be transferred to an industrial robot with the aim to get the same result as the input image. In these conditions, the results offered by image processing process with the software used and the safety conditions of working of the robot are very important and must be compatible.

3.1. Binary image

As its name says, a binary image (figure 2) has two values, 0 and 1. Considering that an image is formed by pixels, it is known that their values are between 0 and 255 (which means the intensity of each pixel), where 0 is white and 255 is black. The value of each pixel has its own intensity, from where results its own color. Each pixel is part of the background or part of the objects that are found in the image. To get a binary image, over the initial image further techniques are applied or methods are used to get the desired results.

![Figure 2. Example of binary image. [Google]](image-url)
3.2. Binary information using LabVIEW
LabVIEW is a programming platform created by National Instruments, but in this case, it is about a visual programming-language. This was created to realize applications that can be tested, controlled or to measure different phenomena that offers a rapid access to hardware. [6] In figure 3 it can be noticed that its interface is user friendly and with its help, an application of binary image conversion that was realised will be presented in the next paragraphs.

![LabVIEW interface](image)

Figure 3 LabVIEW interface. [authors contribution]

In the next images it can be seen a general view of the front panel (figure 4) where all kind of buttons, windows or slides are available and what is it in their background, how they are formed to look like this and have their own function (the block diagram, figure 5). This example is of the application realized to process a coloured digital image into a binary image using LabVIEW. It must be mentioned that the background is represented by “0” and the object from the image is represented by “1”.

![Front panel in LabVIEW](image)

Figure 4. Front panel in LabVIEW. [authors contribution]
3.2.1. Greyscale image
As it was mentioned in section 2, first time an image must be read, it must be known the path from where that image is saved or from where it will be taken. From the block diagram (figure 5 and figure 6), the part of the diagram that is used to read the image is presented in figure 5. With the help of a File Path Control (1) that will make the connection with the software LabVIEW. It is not enough just to know the path of the file, it also must be known the dimension of the image that has to be positive (3). With the help of a comparison function, once the image is converted in a string, the image can be compared to 0.

The next step of the Flat sequence structure (figure 6) represents the beginning for image conversion. Once the image is converted to a greyscale form (4) using IMAQ Create a temporary file for greyscale form of the image is created in the memory and IMAQ ReadFile is used to read the greyscale file, that further will be displayed (5) (figure 4, left). To have more information about the converted image, a histogram, using IMAQ Histogram function (6), is created and displayed (figure 4, right).

The next step of the process is to get the binary form of the image. With the help of an IMAQ Create function (7), a temporary location in RAM memory was located for the binary image. To have a better control over the pixels, the mean value of the intensity will be calculated. This value will be a threshold for the steps that will follow. All the information will be saved in a *.txt file, thing that is happening in that part marked with red from figure 6.

![Figure 5. Reading the image.](authors contribution)

![Figure 6. Conversion to greyscale image.](authors contribution)
3.2.2. Binary image

In this research paper an image is converted to a binary form and the results are used to realize an application with an industrial robot. Figure 7 represents that part of Flat Sequence Structure where the greyscale form of the image is transformed to a binary image (figure 8). A detail of the binary form of the image can be seen in figure 9. In this case 0 represents black and 1 represents white.

Conversion of the image to a binary form was realized with the help of a function, called IMAQ Threshold (9). With the help of the mean value of the pixels and the histogram report (figure 4, right), each pixel of the binary image will be compared with the threshold (7) and depending of its value, it will be converted in 0 or 1.

In this study is used a manual threshold (marked with red in figure 4) and as can be seen in the figure, the threshold is set to 120. In the Block Diagram, the threshold was created with the help of a case structure block. With the help of Bin Image function (11), the image from figure 4 (centre), will be obtained.

Figure 7. Binary image sequence. [authors contribution]

Figure 8. Binary image. [authors contribution]

Figure 9. Detail of binary image resulted (from figure 8). [authors contribution]
3.2.3. Report for the binary image

The final of the image processing process using LabVIEW is represented by saving the data obtained, but in this case, the action will be done in two forms. The final image, the binary image (figure 10 and figure 11) is saved as a *.jpg file. A file with data that can be used further is needed. Once the image is converted and the results are good, a confirmation is needed (figure 8 - OK).

What must be mentioned is that the image has a binary form, but it is represented as a matrix. This way of arrangement does not help to go further, so this matrix must be transformed. For this, it will be converted to a 2D array using IMAQ ImageToArray function (12) and the results will be saved in a document (13). To have the digital image of the result (14) (from figure 10) a conversion in a binary form of the image to a *.bmp image using IMAQ ArrayToImage (15) was done. The last step it will be to erase all temporary files (16) and the file will be closed (17).

The program and all this information about LabVIEW function have been taken from National Instruments – LabVIEW Tutorial [6].

Figure 10. Binary report. [authors contribution]
3.3. ABB architecture

Programming an industrial robot is to manipulate each axis of the robot, to control a point. Depending on the position of that point, with reference to a certain frame, any robot is getting a configuration. In the controller memory further TCPs can be defined, but only one can be used at once. The TCP can be stationary or moveable, which means to move in the same time with the manipulator (welding gun, gripper). [10] [11] [12]

TCP is that point according to which the robot is moving to make a trajectory. This point is characterized by a frame, with the origin on the tool or not, being unique for each tool. To define this point, a data must be created and defined. For ABB architecture, “tooldata” is defined using 3 points method, which means that the proposed TCP will touch the same point (e.g. the top of a mandrel), using 3 different configurations. For a better precision, TCP definition will be done using from 4 to 9 points with different configurations (figure 12). [11] [12]

The position of “tooldata” is defined according to wrist coordinate system (x, y and z) in mm, its orientation being expressed in wrist coordinate system, quaternion (q1, q2, q3 and q4). Position and orientation for this type of data, is marked with bold in equation (1). (see figure 13) [13]

\[
PERS \ \text{tooldata tool0 := [robhold, [[x, y, z], [q1, q2, q3, q4]], [mass,[cog x, cog y, cog z], [qo1, qo2, qo3, qo4], Ix, Iy, Iz]]};
\]  

(1)
Moving on a certain trajectory, the TCP must know exactly where are the points that are part of that one to reach them. Once TCP is defined, and its position is known on the tool, to realize the trajectory, the last one must be created having a reference. This reference means a frame in the working space that can be user or object frame. The last two frames represent the working area of the robot, which is established by the operator (see figure 14).

Equation (2) [12] is a general form of a line from RAPID code, used by ABB in linear interpolation [12], that highlights the importance of a user or object frame. “ToPoint” represent the coordinates of the point (its form is presented in equation (3) [12]) that will be reached, depending on a certain frame. “Tool” (by default, tool0 with TCP in the origin of the wrist coordinate system) is the name of the “tooldata” that was created and defined. “WObj” (by default, wobj0) means the working object (a user or an object frame) according to which the position of “ToPoint” is set.

\[
\text{CONST robtarget start} := \begin{bmatrix} x, y, z, \\ q_1, q_2, q_3, q_4, \\ \text{cfg1, cfg2, cfg3, cfg4}, \\ 9E9, 9E9, 9E9, 9E9 \end{bmatrix};
\]  

(3)

As an example, in equation (4) [12], a linear interpolation is created from the initial point to point “start”, using “MoveL” instruction, point that was saved related to “WObj := fixture” as work frame, with the help of a “tooldata grip3”. The presence of “\" means that setting the working frame is optional.

\[
\text{MoveL \"Start }, v2000, z40, \text{\textit{grip3}} \ WObj := \text{\textit{fixture}};
\]  

(4)

A well-defined and precise TCP helps you get better results. The presence of this section, about ABB technology and what TCP is and which is its importance, is the main condition in implementing an industrial robot in fine arts, precisely drawing. As human wants to know exactly, at a portrait for example, where are the lines of the eyes, of the mouth, where starts a wrinkle and where it ends, also the robot must know exactly where to draw a line, even if it is straight or not. As already was mentioned, when we move a robot from one position to another, precisely we move the TCP from one point to another one, each of them having their own coordinates in relation to the chosen frame.

In the last paragraphs, was presented the way how an industrial robot is working concerning the coordinates systems that exist and those that are necessary to be created. Based on this, an algorithm must be realized to make a relation from binary form to point coordinates. An example of this kind of algorithm will be presented in the next section 3.4.
3.4. Algorithm proposed to draw using binary data

Getting a binary image as result, an algorithm must be established to convert the binary values in point coordinates, “P(x, y, z)” where “P” is the name of the point. For this the image resolution must be known and the pixels number from the image too. Considering the result obtained (see figure 15, detail of figure 7) and knowing that the robot is moving according to point coordinates (according to section 3.3), it must trace a line between the middle of two binary values, even if are the same or not. To do this, a frame with the origin in the corner of the binary image must be set.

With the help of the following equations, from (5) to (10), the center of each binary value can be found, the point coordinates being presented in equation (11). The importance of points coordinates values is highlighted by the motion instructions used by the robot, presented in equations (1), (2), (3) and (4).

\[
\begin{align*}
\mathbf{n} \cdot \mathbf{u}_x & \rightarrow \frac{h}{d_o} \cdot [0.5 + (n-1)] = \frac{h}{d_o} \cdot [n-0.5] \\
\mathbf{u}_x & = \frac{h}{1/d_o} \cdot (n-0.5)/n \\
1 \leq n & \leq d_o
\end{align*}
\]
where:
\( n \) – the number of the pixel (horizontal)
\( u_x \) – the width of pixel \( "n" \) (horizontal)
\( h \) – page width
\( d_o \) – number of pixels horizontally

\[
m \cdot u_x \rightarrow H/d_o \cdot [0,5 + (m-1)] = H/d_o \cdot [m-0.5]
\]
\[
u_y = H \cdot 1/d_o \cdot (m-0.5)/m
\]
\[
1 \leq m \leq d_v
\]

where:
\( m \) – the number of the pixel (vertical)
\( u_y \) – the height of pixel \( "m" \) (vertical)
\( H \) – page height
\( d_v \) – number of pixels vertically

\[
P_{(n,m) \text{ robot}} = P_{(n,m)} (u_x, u_y) = P_{(n,m)} (h/d_o \cdot (n-0.5)/n, H/d_o \cdot (m-0.5)/m)
\]

For drawing that image, just the “1” binary value is needed, because of this, before starting to find out the point coordinates, the binary matrix will be scanned to see where those binary values are of “1”. Once one value is found, equation (11) helps to get the point coordinates and save them in a file, to be easily accessed by the robot controller.

Based on the image from figure 15, a case can be found, which means further binary values of “1” (figure 15, highlighted with red), positioned in such a way that they are not representing the border of the feature that will be drawn. In this case, it must be detected the relation of a pixel with its neighbours. Taking into account that a pixel has 8 neighbours, each of them has a certain connection with it horizontally, vertically or diagonally. Because of this, 8-connected method is used (figure 16).

**Figure 16.** 8-Connected Neighbourhoods. [15]

### 4. Conclusions

The paper “Industrial robot in fine art: can an industrial robot to draw a binary image?” is about a study concerning the collaboration between the results obtained by an image processing process, having as a result a binary image and an industrial robot.

In section 3.2, it was about binary images, formed by 0 and 1. The original image that is wanted to be processed and finally drawn by a robot is transformed until it is formed just by numbers like 0 and 1. The result is similar with the original image, but is not proper to program an industrial robot.

Based on the information written at section 3.3, about ABB technology and general industrial robot knowledge, things about the research from this paper become more and more interesting. It is known that to do a job, a robot must follow a certain path, but it must know exactly its own configuration for
each step that is doing. Having as source of information a binary image, the robot cannot know the path that must be followed.

In conclusion, a method of conversion has been done. With the help of an algorithm to transform a binary image into an image formed just by points, whose coordinates were specified, was done. What was more interesting at this research is that a program used for measurements, control and test in engineering, can also be used with other aims, in this case to process an image.

Going further, more studies can be done on the algorithm of image processing realized in LabVIEW and continue this research in the same way. Another study can be done starting from here and finding another algorithm or method such a way the robot to recognize a binary image and to know where the limit between background and object is, respectively 0 and 1 and draw the image that is wanted.

5. References
[1] Algorithms https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/v/what-are-algorithms accessed at 10.05.2018
[2] Image processing https://www.sciencedirect.com/topics/neuroscience/image-processing accessed at 10.05.2018
[3] World Heritage Encyclopedia, Image processing, article ID WHEBN0000076332 http://self.gutenberg.org/articles/image_processing accessed at 10.05.2018
[4] https://www.engineersgarage.com/articles/image-processing-tutorial-applications accessed at 10.05.2018
[5] Lind R 2012 Open Source Software in Life Science Research Open source software for image processing and analysis: picture this with ImageJ (a volume in Woodhead Publishing Series in Biomedicine) chapter 5 pp 131-149;
[6] https://www.sciencedirect.com/science/article/pii/B9781907568978500059
[7] J. R. Parker 2011 Algorithms for image processing and Computer Vision - Second Edition, Wiley Publishing, Inc.
[8] R. C. Gonzales, R. E. Woods 2008 Digital Image processing - Third Edition, Pearson International Edition
[9] ABB IRB 1600 Data Sheet - http://new.abb.com/products/robotics/industrial-robots/irb-1600 accessed at 12.05.2018
[10] Tool Centre Point - ABB http://developercenter.robotstudio.com/BlobProxy/manuals/IRC5FlexPendantOpManual/doc208.html accessed at 12.05.2018
[11] Tool centre Point – ABB, accessed at 12.05.2018 http://developercenter.robotstudio.com/BlobProxy/manuals/IRC5FlexPendantOpManual/doc9.9.html
[12] RAPID reference manual, RAPID overview, Controller software IRC5, RobotWare 5.0, Page 157
[13] Technical reference manual, Controller software IRC5, RobotWare 5.0, Page 1010
[14] RAPID reference manual, RAPID reference part 1, Instructions A-Z, RobotWare-OS 4.0, Page 157
[15] 8-Connected Neighbourhoods http://radio.feld.cvut.cz/matlab/toolbox/images/binary6.html, accessed at 12.07.2018