

# Simple Computer Vision Algorithm Production Using OpenCV for “Virtual Ecosystem” Project

Andrew Willis<sup>1</sup>, Kemal Hasan<sup>2</sup>

<sup>1,2</sup>Department of Film & Animation, Universitas Multimedia Nusantara, Tangerang, Indonesia

[andrew.willis@umn.ac.id](mailto:andrew.willis@umn.ac.id)

[kemal.hasan@umn.ac.id](mailto:kemal.hasan@umn.ac.id)

Accepted on October 9, 2019

Approved on June 18, 2020

**Abstract**—The main purposes of this research is to produce a Computer Vision (CV) algorithm using various library in python programming language (mainly OpenCV, Numpy, and ZBar) to automate extraction and process information from an analog drawing into a digital image to be used in the “Virtual Ecosystem” project. The Computer Vision step will include detecting data within the analog drawing using QR-code, determine drawing area, replace white background with certain threshold with transparency, and finally save the digital image following required ratio.

**Index Terms**—Computer Vision, Numpy, OpenCV, Python

## I. INTRODUCTION

Computer vision technology has been implemented in various industries nowadays. Starting from simply logging you in to your smartphone [1], detecting disease through your eyes [2], to autonomous driving. It has been an exciting technology since its inception in 1970 [3] that helps automate tasks that the human visual system can do through high level understanding from digital images. However, the implementation of computer vision in new media art is still in its nascent stage.

Computer vision techniques in arts commonly used to detect motion, detecting presences, detection through brightness thresholding, simple object tracking, and basic interactions such as the one used by suicide box (object tracking), Videoplace (basic interaction), and cheese (detecting presences) [4].

Therefore, in this research, the authors will discuss the implementation of elementary computer vision technology in new media art installation within “Virtual Ecosystem” project using python as main programming language with additional OpenCV, Zbar, and Numpy library. The author will focus mainly on detecting presence technique to find certain attributes within a given raw image to extract an analog drawing.

## II. RESESARCH PURPOSES

The aim of this research is to produce a simple computer vision algorithm that is able to gain an analog creature drawing, extract information contained within the analog data, and process the image data to be used by the game AI within the “Virtual Ecosystem” project.

The author also wishes this paper to encourage more implementation of computer vision in practical art by artists alike either by art students, academicians, or general artists from various backgrounds.

## III. GAME REQUIREMENT

The CV algorithm produced in this research is one of many algorithms written for the “Virtual Ecosystem” project. Therefore, the image data output processed by the algorithm will have to meet certain requirements of the next algorithm (next referred to as the game) of the project in order to work.

First, the game has various creature types with its own animation and AI behaviour where each image data sent to the game has to contain a creature information code. This information has to be passed from the analog drawing template through the CV algorithm to the game without any human intervention to minimize human error.

Next, the image data sent to the game has to contain only the colored part of the drawing area without the border and the white background. The white background with a certain threshold has to be replaced with an alpha background to give a cut-out illusion in the game.

Finally, the image alignment and direction produced by the CV algorithm has to be consistent on every image data. Any deviation or error won't be able to be fixed by the game itself.

#### IV. SYSTEM FLOWCHART

The following flowchart shows the general idea on how the system works and how data is processed on each stage.

First, the analog drawing using the specialized drawing sheet discussed on section 6 will be scanned using Fujitsu SV600 scanner which then creates a digital image representation of the analog drawing in the system.

Next, the computer vision algorithm discussed on this paper will process the digital image and extract the creature data and creature drawing within the drawing.

Finally, if the CV algorithm able to detect both creature data type and drawing area, it will extract the creature image and send the resulting image to the game networked drive to be used by the game.

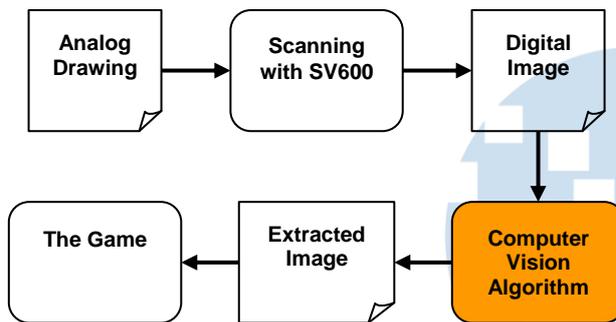


Fig. 1. System Flowchart

#### V. IMAGE DATA ACQUISITION

Various ways of image data acquisition (scanning) have been tested to find the most efficient technique with minimal human intervention possible in this research.

Initial technique involves using a webcam and a tower platform where a webcam is installed on top of custom-made simple tower structure made of aluminium with a large wooden platform as the base.



Fig. 2. Image Scanner Using Tower Platform and Webcam Installed on The Top

Using this technique, the system is able to capture analog drawing. However, the setup produces inconsistent lighting on image data due to its reliance on the surrounding light environment where the same

image scanned in different places will yield different readability results.

In order to achieve stable consistent images but maintain a rapid scanning process, the authors utilize Fujitsu SV600 scanner. Its non-cover and self-lighting method provides high quality image data with consistent readability result and rapid scanning process. Additionally, the system provides an automatic skew correction algorithm which helps prepare the image data before processed by the CV algorithm.



Fig. 3. Fujitsu SV600 Scanner Produce Consistent Raw Image Regardless of The Ambient Light

#### VI. ANALOG DRAWING TEMPLATE (CONTROLLED ENVIRONMENT)

Due to its simplistic approach, the CV algorithm requires a certain number of controlled environments to ensure smooth operation, easy image readability analysis and minimize human intervention. This is predominantly done using a rectangular shape [5] template which act as a certain marker in the analog drawing.

Initially, using a webcam shown on the previous chapter, the authors utilized no template on scanning process and only a red bracket marker on the computer viewfinder and this method relied on the operator to fit the image inside the bracket.

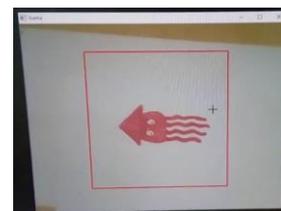


Fig. 4. Red Bracket Viewfinder with Non-Template Drawing Being Scanned

This method, although very simple to use, requires lots of human intervention which make scanning time longer due to image fitting and adjustment. This method also didn't provide creature type information as per the game requirement.

Therefore, the authors decided to use an easily recognizable template using basic shape to frame a

certain area of interest as shown below where the drawing area has been framed using basic rectangle following quad detection found commonly in ARTag [6].

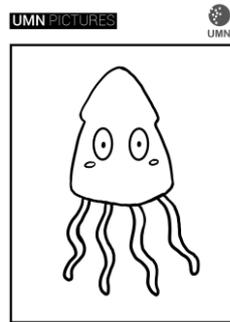


Fig. 5. Basic Shape Frame The Drawing Area with Creature Outline Inside

The quad frame creates a controlled environment for the CV algorithm to identify the drawing area. Human intervention has also been limited to scanning only as the detection of the drawing area and the skew correction will be done by the CV algorithm which in turn accelerates the entire scanning process.

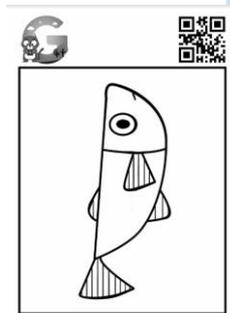


Fig. 6. Drawing Template with QR-code Embedded Top Right

The template final requirement will need the template to carry information for the game to process their creature type. This has been done using QR-code embedded on the top right of the drawing template [7]. The QR-code contains a numeric information where each number represents a type recognized by the game.



Fig. 7. Analog Drawing Sample Retrieved Using SV600 Scanner

## VII. IMAGE CORRECTION

Image scanned using various methods mentioned in this research often still possess imperfections in it such as unaligned image with small rotation, defective printing, warped image, etc. These imperfections need to be corrected before proceeding to the next step to prevent error or misreading from the image data.

The first image scanning method (using a webcam but no template) requires little to no image correction due to its simplistic scanning mechanism. However, the second method (using a webcam with a template) requires rotation correction since the traditional webcam has no automatic rotation correction algorithm. Using threshold and minAreaRect function in open CV [8], the author is able to determine image rotation deviation and make corrections before proceeding to the next step.



Fig. 8. Slanted Scanned Image

Another common imperfection from the scanned image data is the printing imperfection due to using non-specialized QR-code printers. QR-code itself is characterized with a finder pattern on three or four corners of QR-code [9] which if not printed properly like in this case a fading barcode, it will make the barcode harder to read and make QR-code detection inconsistent. In order to address this issue, the author crops the QR-code area (located on top right) then

apply gaussian blur [10] to fill in the fading barcode. This method increases the QR-code readability greatly by filling the faded QR-code.



Fig. 9. Fading QR-Code on Analog Drawing and Blurred QR-code

In the previous chapter, the drawing area needs to be characterized with the largest rectangular frame within the drawing template. However, the said rectangular frame has a thickness which we need to clear. Therefore, the CV algorithm will reduce the image size cropping the thickness of the rectangular frame.



Fig. 10. Scanned Image with Rectangular Frame and Processed Image with Rectangular Frame Cropped

## VIII. TEMPLATE DETECTION AND IMAGE PROCESSING

In the previous chapter, the author discusses a controlled environment using a drawing template to simplify the image detection process. In this chapter, image scanned and corrected will be processed further to meet the game requirements.

First, the CV algorithm has to detect the creature type contained in the template. This can be done by reading the QR-code located on the top right of the drawing template. The author uses z-bar library to read the encoded type using numerical value where each value represents the creature type where the game will assign it to each respective creature behavior.

Next, to detect the drawing area the CV algorithm will list all contours found in the image data using findContours function to detect area of interest. By calculating each contour area and poly count, the author is able to determine the drawing area which characterized by possessing 4 poly count (hence the rectangular frame) and with the largest area of any contour.



Fig. 11. Contours Found in The Image (Visualized by Red Enclosure)

Finally, in order to meet with the game requirement where the image white background has to be replaced to alpha (transparent), the CV algorithm extract white area from the original image using threshold function then using its alpha mapping, the author merge it back to replace the white background with transparent background leaving the non-white pixel to be intact while the white background (with certain threshold) to be replaced with alpha.

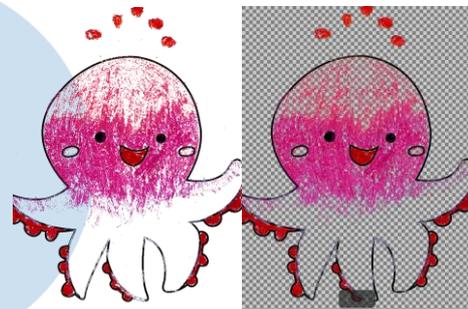


Fig. 12. Original Processed Image, White Map Extracted, and Final Image with Alpha Merged (Alpha Represented with Checkerboard)

The image data then will be saved to a network drive with a name syntax <entry number>\_<type>.png. The .png used to carry the alpha information to the game.

## IX. CONCLUSION

Producing a simple offline computer vision algorithm without employing machine learning technology requires a combination of mathematical solution with lots of controlled environment to work with. Starting from the paper has to be designed to be easily processed (controlled environment), QR-code to carry creature type information, and some manual configuration adjustment.

## REFERENCES

- [1] B. Amos, B. Ludwiczuk,†M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications", Carnegie Melon University, June 2016.
- [2] Y. Cho, S. Lee, S. Woo,, "The Kirsch-Laplacian edge detection algorithm for predicting iris-based disease", IEEE 21st International Conference on Computer Supported Cooperative Work in Design, 2017.
- [3] R. Szeliski, "Computer Vision: Algorithms and Applications", Springer, pp. 10, 2011.

- 
- [4] Levin, G. "Computer Vision for Artists and Designers: Pedagogic Tools and Techniques for Novice Programmers". Journal of Artificial Intelligence and Society, Vol. 20.4. Springer Verlag, 2006.
  - [5] Y. H. Jen, Z. Taha, E. T. Hong, C. J. Yeong, "Design and Development of a Marker-Based Augmented Reality System using OpenCV and OpenGL", University of Malaya, December 2009.
  - [6] M. Fiala, "ARTag, a Fiducial Marker System Using Digital Technique", pp.3, 2005.
  - [7] D. Hopkins, "QR Codes in Education", Create Space Publishing, 2013.
  - [8] D. L. Baggio, S. Emami, D. M. Escriva, K. Levgen, J. Saragih, R. Shilkrot, "Mastering OpenCV 3", Packt, 2017.
  - [9] B. Furht, "Handbook of Augmented Reality", Springer, pp. 341, 2011.
  - [10] R. Laganière, "OpenCV Computer Vision Application Programming Cookbook Second Edition", Packt Publishing, 2013.

