

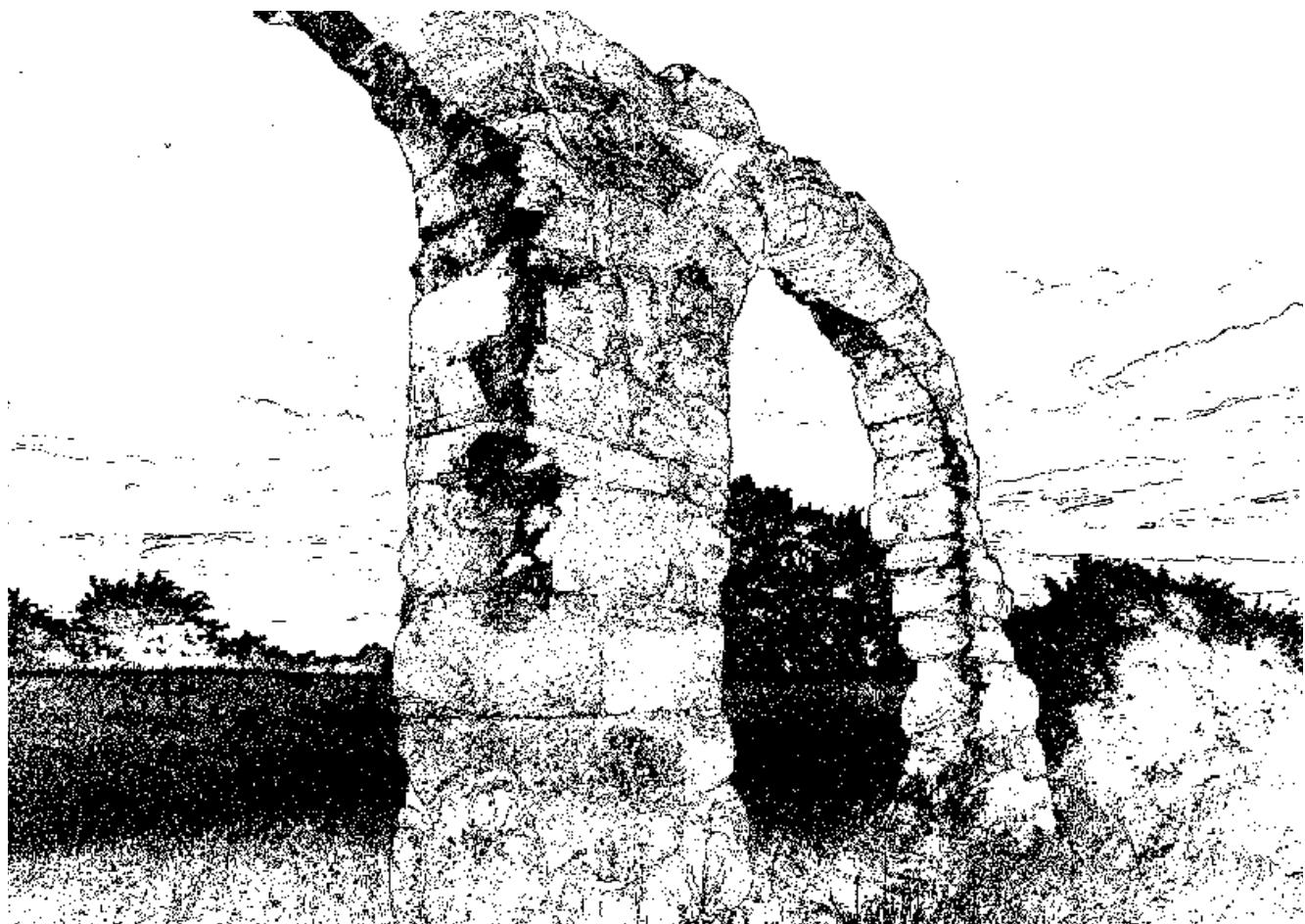
---

# Jarrett's Edge Detection

## Image Manipulation Software

Jarrett Philips - December 7th, 2015

---



## Overview

The objective of edge detection is to determine if there is a great enough difference between two adjacent colors in an image. And if so, that they must belong to different entities, and thus, warrant an edge or outline. Edge detection parses an image, and identifies lines and details in the photo, and is applicable in many photo manipulation applications, such as image enhancement, correction, or purely for aesthetic. Keep in mind that some picture may turn out better than others, as the algorithm can be tricked by shadows and light. The higher contrast a photo has, the better it will turn out.

I have tested both ready and writing both JPEG and PNG images, other formats may or may not be supported.

Please be patient with it, especially on laptops, large images can take some time to process.

## Use

Jarrett's Edge Detection is a command line based program. In order to run it, you must first compile, then run it. Please note that *Image Magick* is required by the *CImp* library to run. Installation should be easy and strait forward. For Mac OS, first install *Homebrew*, to ease in installation. This can be done by entering the following line in the terminal:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Then to install *Image Magick*, simple enter:

```
brew install imagemagick
```

That's it! All the needed libraries are loaded (aside from *CImp*, but that is packaged with the project). Now, direct the terminal to the directory containing the source code, and insure the image is in the same directory. To compile, enter:

```
g++ main.cpp Scanner.cpp Writer.cpp -O2 -lm -lpthread -I/usr/X11R6/include  
-L/usr/X11R6/lib -lm -lpthread -lX11
```

Once compiled, you can run the program. Enter:

```
./a.out nameOfImage.extension nameOfDestination.extension distance
```

The distance is the amount of difference between two pixels that will constitute an edge. It is an integer in value. A larger number will make a much "whiter" image, with less lines. There

is no perfect setting and it will be entirely dependent on the given image. A lower distance will reveal finer details, but can also become more noisy or blotchy. For the provided example, the terminal input will be:

```
./a.out croatia.png croatiaProcessed.png 4
```

But you're probably lazy like me, so for your convenience, I have pre-compiled the project. and you can simple run it with:

```
./JarrettsEdgeDetection nameOfImage.jpg nameOfDestination.jpg distance
```

It will output the new file into the same directory as the rest of the files.

