# Generating geometric abstractions

Crystal Qian '17, David Dobkin
Princeton University, Department of Computer Science

## Overview

We developed a fast parametrized algorithm and interactive tool for constructing polygonal representations of images.


## Implementation

### Initialization

We choose an image and the following parameters:

- blur radius
- convolution radius
- edge threshold
- number of vertices


### Grayscale

Pixel $(r, g, b)$ is set to value $(l, l, l)$, where luminosity $l$ is

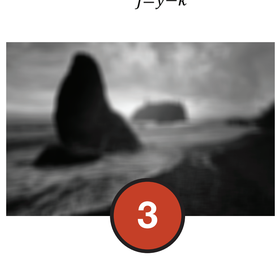
$$l = .21r + .72g + .07b$$


### Box blur

A linear blur of radius $k$ eliminates noise. Each pixel at $(x, y)$ is set to value $v$.

$$h_{x,y} = \sum_{i=x-k}^{x+k} \frac{v_{i,y}}{2k}$$

$$v_{x,y} = \sum_{j=y-k}^{y+k} \frac{h_{x,j}}{2k}$$

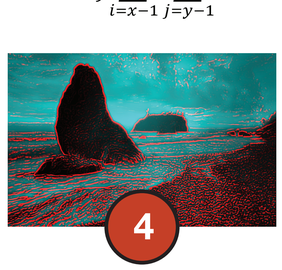



### Edge detection

We convolute the image with a filter and pick vertices by edge density $d$.

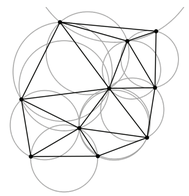
$$\begin{bmatrix} +1 & +1 & +1 \\ +1 & -8 & +1 \\ +1 & +1 & +1 \end{bmatrix}$$

$$d_{x,y} = \frac{1}{9} \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{j=y+1} e_{i,j}$$




### Triangulation

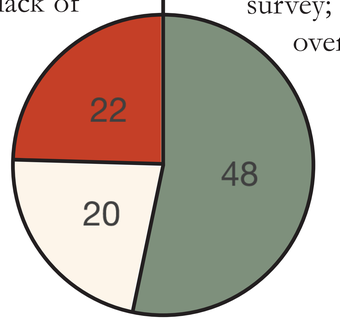
Vertices are connected with a Delaunay triangulation.`




## Previous work

Manual rendering is time-consuming and labor intensive. Automated triangulation algorithms are not necessarily visually pleasing, due to lack of user control or emphasis on artistic value.

We apply research in edge detection and adaptive image approximation to improve "low-poly" generation methods.




## Results

Our tool is a fast and simple way to create visually-pleasing "low-poly" abstractions from images. This was determined by survey; results show 48 positive, 20 neutral and 22 negative overall impressions.

Potential applications:

- object detection (e.g. Snapchat filters)
- computer-generated art