Parallel Low Poly Image Conversion

Weichen Ke (weichenk) Zhengjia Huang (zhengjih)

**Introduction**



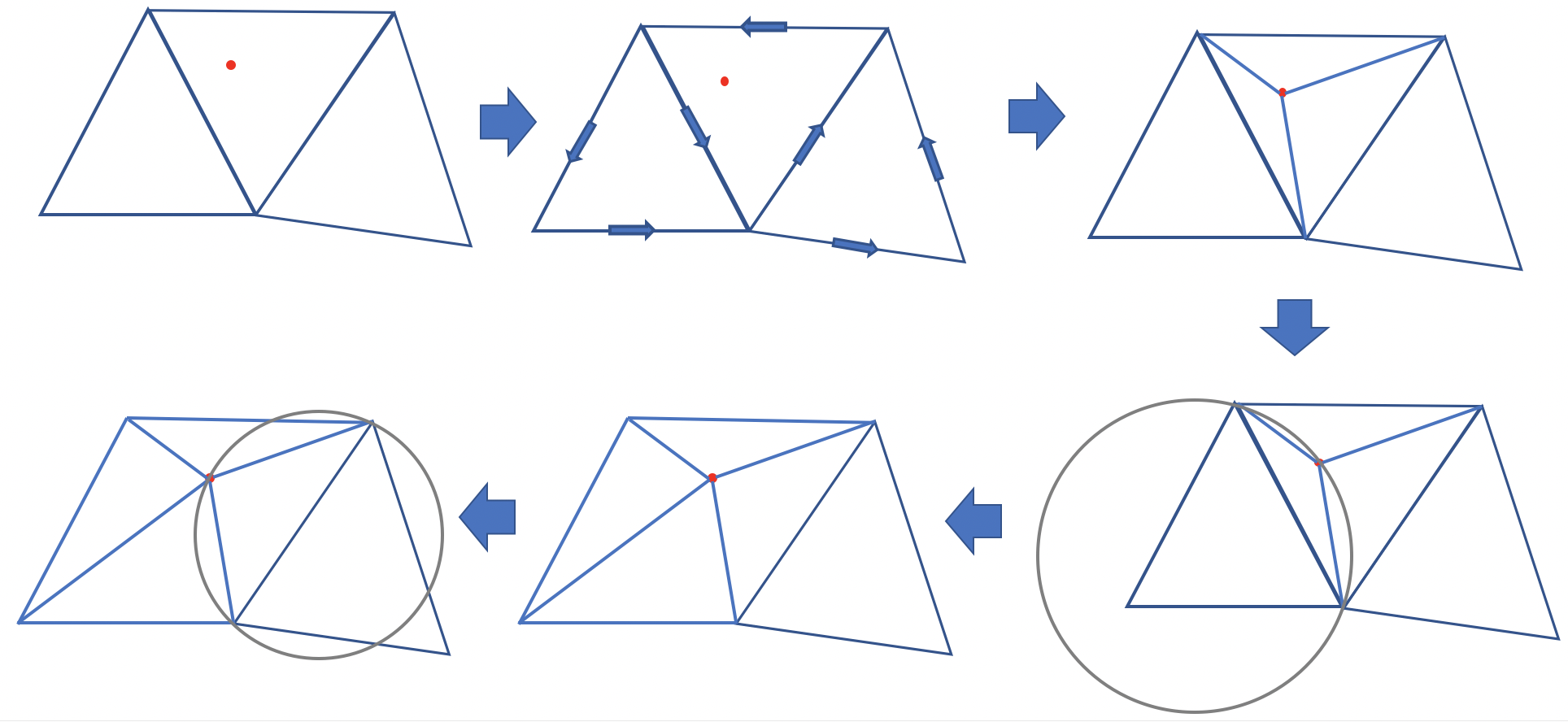
Image from: <https://engageinteractive.co.uk/blog/how-to-create-low-poly-art-in-adobe-illustrator>

**Methods**

Random points generation

We will first use Sobel Edge Detector to find edges in the image. Then we’ll randomly spread points on the image, but with higher possibilities on the edges (texture rich regions).

Triangulation



The key part of this task is triangulation given an image and a bunch of points on the image. We want to use [Delaunay Triangulation](https://en.wikipedia.org/wiki/Delaunay_triangulation) method.

Given a set of points, we want to find a non-overlapping triangulation of them. We first assume we are in the middle of this algorithm so that given a point, there is one and only one triangle that surrounds it. Now follow the steps below and refer to the figure above:

1. Select a new point from the remaining points, as shown in red in the above figure
2. Traverse through all the known triangles, loop the three edges(vertices) of each triangle in counter-clock direction, if the new point falls in the same direction of the three edges, then it is in that triangle.
3. Separate the surrounding triangle into three smaller triangles using the new point.
4. Draw the circumcircles of the new triangles. Notice that we can divide each quadrilateral into triangles in two ways. For aesthetic reason, if the other point of the quadrilateral locates inside the sphere, we switch to the other way of dividing the quadrilateral. If it is outside the sphere, then we keep it.
5. Go back to step 1, end if now more points to be selected.

In order to start the above algorithm, we first add three points out of the image plane and connect them into a huge triangle which contains all the points in the image plane.

Rendering

CUDA…

**Challenges**

* Implicit conflictions. For example, two points inside the same triangle cannot be examined at the same time by two processes. And when switching the division of a quadrilateral, there cannot be other processes that is examining points in the quadrilateral.
* Communication between processors. If we divide the image into several blocks and assign blocks to processors, we need to figure out a correct and efficient way for the processors to communicate.
* Workload balancing. For aesthetic reason, the points on the image are not evenly distributed. In regions with rich texture, we have more points and in regions with less texture, we have fewer points.