Abstract

Subject : Abstract representation of spacial data for machine vision

In the context of machine vision, spacial data comes the form of two dimensional images taken by a digital image sensor. An abstract represention of two dimentional image data involves representing the contents of the image or color data into concise and reduced form for quick, easy and flexible inference without sacrificing on its signature. A signature of an image would be its most reduced form which retains any meaningful data that can be extracted from the image.

For the purpose of this study i am considering grayscale image profiles instead of color profiles for reducing the complexity of the proposed algorithm / pseudo code.

Few terms which will be used from time to time.

1. Pixel

Two dimensional images are stored in memory as arrays. So, if the image has 640 columns and 480 rows , the data would be stored in an array of size 640*480, which equals to 307200. Each item in the array represents a position in the 2D image.

2. Blob

Group of pixels which have the same color and are connected.

3. Boundary of the blob

The pixels around the circumference of a blob.

4. Seed of the blob

The leftmost and topmost pixel in the boundary of a blob.

5. Resolution

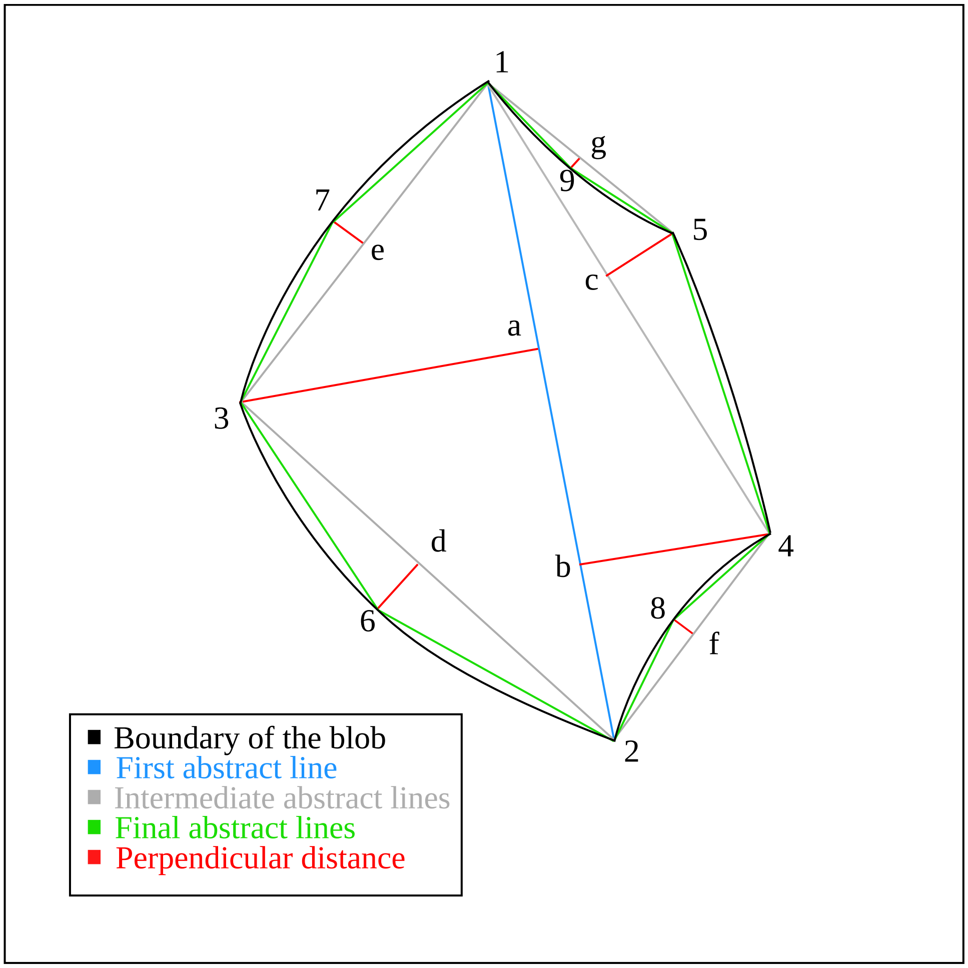
The least amount of change in direction that is considered as perceived.

A 2D image can be considered as a collection of blobs. The first stage of abstraction involves representing blobs in a reduced form. This is called the boundary of the blob ie an ordered collection of pixels around the circumference of the blob.

The second stage involves finding two pixels from the boundary which is farthest from each other. These two points are the first two abstract points of the boundary. The line joining these two pixels is called the first abstract line.

The third stage involves travelling around the boundary and finding a pixel which is farthest from the first abstract line. If the distance of the pixel is greater than the resolution, the pixel is added to the list of abstract pixels. Now, the boundary is represented by three abstract pixels and we have two more abstract lines formed by connecting the third abstract pixel to the first and second abstract pixels. Now we have three abstract pixels and three abstract lines connecting the three abstract pixels. Similarly, traverse around the boundary of the blob and find the pixel which has the largest perpendicular distance. If the perpendicular distance is greater than the resolution, add the pixel to list of abstract pixels.

The process is repeated by travelling around the boundary and finding more abstract pixels till no more abstract pixels can be found for the resolution. Now we have an abstract representation of the blob.



For example in the figure, pixels 1 and 2 are the farthest pixels in the boundary and becomes the 1st and 2nd abstract pixels. The blue line joining pixels 1 and 2 is the first abstract line. To find the next abstract pixel traverse around the boundary and find the pixel which has the largest perpendicular distance to the first abstract line. The pixel 3 has the largest perpendicular and becomes the 3rd abstract pixel. The line joining pixel 3 to pixel 1 and 2 becomes the 2nd and 3rd abstract line. To find the next abstract pixel traverse around the boundary and find the pixel which has the largest perpendicular distance to its corresponding abstract line. The process is continued to obtain more abstract pixels till the perpendicular distance becomes less than the threshold or resolution. Now we have an abstract representation of the boundary.