Extraction of Level Sets from 2D Images

David Eberly
Geometric Tools, LLC
http://www.geometrictools.com/

Created: September 4, 2000
Last Modified: March 2, 2008

Contents

1 Introduction

For the purposes of this algorithm, a *2D image* is assumed to be a $B_x \times B_y$ array of integer values. The pixel locations are (x, y) where $0 \le x < B_x$ and $0 \le y < B_y$. No restriction is assumed on the pixel values I_{xy} other than they are integer-valued.

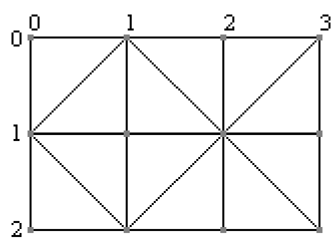
A continuous formulation of the image is required. For *domain square* with corners (x_0, y_0), $(x_0 + 1, y_0)$, $(x_0, y_0 + 1)$, and $(x_0 + 1, y_0 + 1)$, let $F(x, y)$ be a continuous function that represents the image on the square. It is not necessary that F match the pixel values at the four corners, but the two algorithms discussed in this document do have that property.

If L is a value in the range of F, then the *level set of F for L* is the set of points (x, y) that satisfy $F(x, y) = L$. Generally one expects the level sets to consist of curves, but isolated points are possible. For example, if $F(x, y) = x^2 + y^2$, the level set $F(x, y) = L > 0$ consists of a single circle, but the level set $F(x, y) = 0$ is a single point. The algorithms in this document construct both curves and points. A level set for the entire image is constructed from the level sets for the functions on the domain squares.

2 Extraction Using Linear Interpolation

The continuous formulation on domain squares is based on decomposing each square into two triangles and using linear interpolation on each triangle. Figure 2.1 shows a typical decomposition called a *symmetric triangulation*.

Figure 2.1 Symmetric triangulation of a 4×3 image.



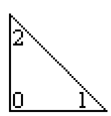
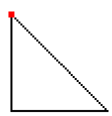
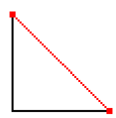
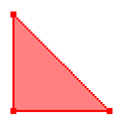
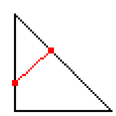
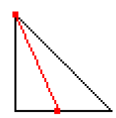
The triangulation of the upper left domain square is said to be *even*. Its two neighbors are said to be *odd*. Let the upper left corner of the domain square be (x_0, y_0). The parity of the triangulation depends on the parity of the components. It is even whenever $\mathrm{Par}(x_0) = \mathrm{Par}(y_0)$, otherwise it is odd. The linear interpolant $F(x, y)$ is defined for $x_0 \le x \le x_0 + 1$ and $y_0 \le y \le y_0 + 1$ as follows. Define $F_{00} = F(x_0, y_0)$, $F_{10} = F(x_0 + 1, y_0)$,

$F_{01} = F(x_0, y_0 + 1)$, and $F_{11} = F_(x_0 + 1, y_0 + 1)$. Define $dx = x - x_0$ and $dy = y - y_0$; then

$$F(x,y) = \left\{ \begin{array}{ll} F_{00} + (F_{10} - F_{00})dx + (F_{01} - F_{00})dy, & \mathrm{Par}(x_0) = \mathrm{Par}(y_0) \text{ and } dx + dy \le 1 \\ F_{10} + F_{01} - F_{11} + (F_{11} - F_{01})dx + (F_{11} - F_{10})dy, & \mathrm{Par}(x_0) = \mathrm{Par}(y_0) \text{ and } dx + dy \ge 1 \\ F_{00} + (F_{10} - F_{00})dx + (F_{11} - F_{10})dy, & \mathrm{Par}(x_0) \ne \mathrm{Par}(y_0) \text{ and } dy \le dx \\ F_{00} + (F_{11} - F_{01})dx + (F_{01} - F_{00})dy, & \mathrm{Par}(x_0) \ne \mathrm{Par}(y_0) \text{ and } dy \ge dx \end{array} \right\}.$$

Given a level value L and a triangle in the decomposition, each vertex value F satisfies $F < L$, $F = L$, or $F > L$, for a total of 27 possibilities for the triangle. Each possibility corresponds to a triangle that has no level set points, a single level set point, a single level set line segment, or the function is zero over the entire triangle. The 27 possibilities can be partitioned into 6 topologically distinct cases, each case having at most 6 permutations. Table 2.1 shows the cases and permutations.

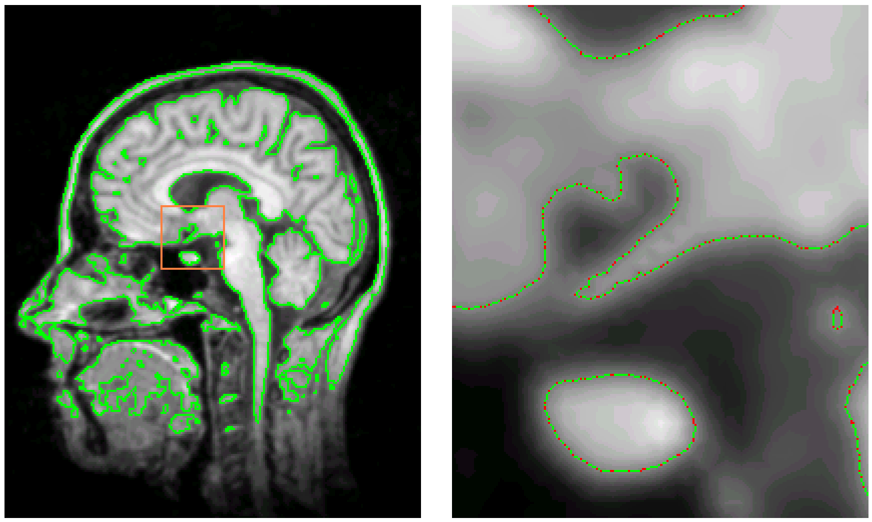
Table 2.1 *Cases and permutations for level sets of F on a triangle.*

case	*0*	*1*	*2*	*3*	*4*	*5*
permute						
0	$+++$	$++0$	$+00$	000	$++-$	$+-0$
1	$---$	$+0+$	$0+0$		$+-+$	$+0-$
2		$0++$	$00+$		$-++$	$0+-$
3		$--0$	-00		$--+$	$-+0$
4		$-0-$	$0-0$		$-+-$	$-0+$
5		$0--$	$00-$		$+--$	$0-+$

The vertex ordering is shown in the figure for case 0 and is the same ordering for the figures of the remaining cases. Permutation 0 refers to that ordering. For example, case 5 and permutation 0 indicates that vertex 0 has positive value, vertex 1 has negative value, and vertex 2 has zero value. In case 0, the permutation is irrelevant since in either case there is no contribution to the level set. For cases 1, 2, 4, and 5, permuations 3 through 5 are effectively the same as permutations 0 through 2 for purposes of extracting level sets. The code makes this reduction. The total number of distinct possibilities is therefore 14.

Figure 2.2 shows a gray scale image with integers in the range $[0, 1023]$. The left image shows the original image superimposed with the vertex locations for level set 512. The right image shows a subimage superimposed with level sets. The edges from the level set extraction are drawn in green. The vertices are drawn in red.

Figure 2.2 Image and subimage with superimposed level sets from linear interpolation.



3 Extraction Using Bilinear Interpolation

The continuous formulation on domain squares is based on using each square as is and bilinearly interpolating. The linear interpolant $F(x, y)$ is defined for $x_0 \le x \le x_0 + 1$ and $y_0 \le y \le y_0 + 1$ as follows. Define $F_{00} = F(x_0, y_0)$, $F_{10} = F(x_0 + 1, y_0)$, $F_{01} = F(x_0, y_0 + 1)$, and $F_{11} = F_(x_0 + 1, y_0 + 1)$. Define $dx = x - x_0$ and $dy = y - y_0$; then

$$F(x, y) = F_{00}(1 - dx)(1 - dy) + F_{01}(1 - dx)dy + F_{10}dx(1 - dy) + F_{11}dxdy.$$

Given a level value L and a square in the decomposition, each vertex value F satisfies $F < L$, $F = L$, or $F > L$, for a total of 81 possibilities for the square. These possibilities can be partitioned into topologically distinct cases, each case having at most 8 permutations. Tables 3.1, 3.2, and 3.3 show the cases and permutations.

Table 3.1 *Some cases and permutations for level sets of F on a square.*

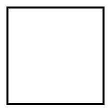
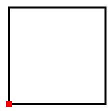
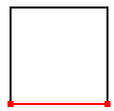
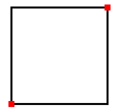
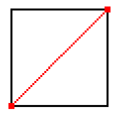
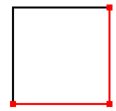
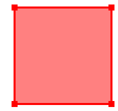
case	*0*	*1*	*2*	*3*	*4*	*5*
permute				 		
0	+ + ++	0 + ++	00 + +	0 + 0+	000+	0000
1	− − −−	+0 + +	+00+	+0 + 0	00 + 0	
2		+ + 0+	+ + 00	0 − 0−	0 + 00	
3		+ + +0	0 + +0	−0 − 0	+000	
4		0 − −−	00 − −	0 + 0−	000−	
5		−0 − −	−00−	+0 − 0	00 − 0	
6		− − 0−	− − 00	0 − 0+	0 − 00	
7		− − −0	0 − −0	−0 + 0	−000	

Table 3.2 *Some cases and permutations for level sets of F on a square.*

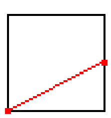
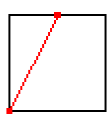
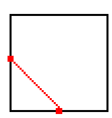
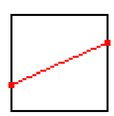
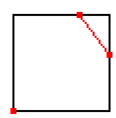
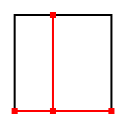
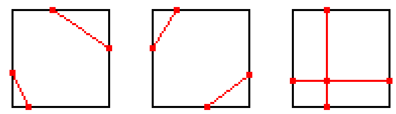
case	*6*	*7*	*8*	*9*	*10*	*11*
permute						
0	0 + −−	0 + +−	+ − −−	+ + −−	0 + −+	00 + −
1	−0 + −	−0 + +	− + −−	− + +−	+0 + −	−00+
2	− − 0+	+ − 0+	− − +−	− − ++	− + 0+	+ − 00
3	+ − −0	+ + −0	− − −+	+ − −+	+ − +0	0 + −0
4	0 − ++	0 − −+	− + ++		0 − +−	00 − +
5	+0 − +	+0 − −	+ − ++		−0 − +	+00−
6	+ + 0−	− + 0−	+ + −+		+ − 0−	− + 00
7	− + +0	− − +0	+ + +−		− + −0	0 − +0

Table 3.3 *Last case and permutations for level sets of F on a square. The three cases depend on the relative values of the x-coordinates of the points.*

case	*12*
permute	
0	$+-+-$
1	$-+-+$

The vertex ordering is shown in the figure for case 0 and is the same ordering for the figures of the remaining cases. Permutation 0 refers to that ordering. For example, case 10 and permutation 0 indicates that vertex 0 has zero value, vertex 1 has positive value, vertex 2 has negative value, and vertex 3 has positive value. As in the linear interpolation, some permutations are effectively the same as others. The total number of distinct cases is 41.

Figure 3.1 shows a gray scale image with integers in the range $[0, 1023]$. The left image shows the original image superimposed with the vertex locations for level set 512. The right image shows a subimage superimposed with level sets. The edges from the level set extraction are drawn in green. The vertices are drawn in red.

Figure 3.1 Image and subimage with superimposed level sets from bilinear interpolation.