Fit a Convex Quadrilateral by a Rectangle

David Eberly, Geometric Tools, Redmond WA 98052
https://www.geometrictools.com/

Created: January 24, 2023

Contents

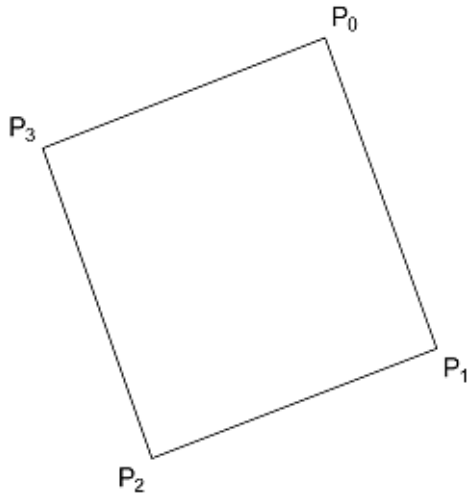
1 Introduction

The algorithm described in this document is about fitting a rectangle to a convex quadrilateral that is *nearly* a rectangle. However, the algorithm can be used even when the convex quadrilateral is not nearly a rectangle.

2 The Algorithm

Let the convex quadrilateral have vertices $\boldsymbol{P}_i$ for $0 \le i < 4$. The upper right vertex is $\boldsymbol{P}_0$, the lower right vertex is $\boldsymbol{P}_1$, the lower left vertex is $\boldsymbol{P}_2$ and the upper left vertex is $\boldsymbol{P}_3$. Figure 1 illustrates the ordering.

Figure 1. A convex quadrilateral with ordered vertices.



The center point of the fitted rectangle is chosen to be the average of the vertices,

$$\boldsymbol{C} = \frac{1}{4}\,(\boldsymbol{P}_0 + \boldsymbol{P}_1 + \boldsymbol{P}_2 + \boldsymbol{P}_3) \tag{1}$$

The rectangle axes are

$$\boldsymbol{U}_0(\theta) = (\cos\theta, \sin\theta), \;\; \boldsymbol{U}_1(\theta) = (-\sin\theta, \cos\theta) \tag{2}$$

An angle θ and extents $e_0 > 0$ and $e_1 > 0$ must be computed for a best-fit rectangle. The rectangle corners are $\boldsymbol{C} \pm e_0\boldsymbol{U}_0(\theta) \pm e_1\boldsymbol{U}_1(\theta)$. Define the point differences

$$\boldsymbol{Q}_i = \boldsymbol{P}_i - \boldsymbol{C}, \;\; 0 \le i < 4 \tag{3}$$

The purported right edge of the rectangle lies on the line with direction $\boldsymbol{U}_0(\theta)$ and containing the point $\boldsymbol{C} + e_0\boldsymbol{U}_0(\theta)$. The signed distances from $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$ to the line are $s_0 - e_0$ and $s_1 - e_0$, respectively, where $s_i(\theta) = \boldsymbol{U}_0(\theta) \cdot \boldsymbol{Q}_i$ for $i = 0, 1$. The purported left edge of the rectangle lies on the line with direction $\boldsymbol{U}_0(\theta)$ and containing the point $\boldsymbol{C} - e_0\boldsymbol{U}_0(\theta)$. The signed distances from $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$ to the line are $s_2 - e_0$ and $s_3 - e_0$, respectively, where $s_i(\theta) = \boldsymbol{U}_0(\theta) \cdot \boldsymbol{Q}_i$ for $i = 2, 3$.

The purported top edge of the rectangle lies on the line with direction $\boldsymbol{U}_1(\theta)$ and containing the point $\boldsymbol{C} + e_1\boldsymbol{U}_1(\theta)$. The signed distances from $\boldsymbol{P}_0$ and $\boldsymbol{P}_3$ to the line are $t_0 - e_1$ and $t_3 - e_1$, respectively, where $t_i(\theta) = \boldsymbol{U}_1(\theta) \cdot \boldsymbol{Q}_i$ for $i = 0, 3$. The purported bottom edge of the rectangle lies on the line with direction $\boldsymbol{U}_1(\theta)$ and containing the point $\boldsymbol{C} - e_1\boldsymbol{U}_1(\theta)$. The signed distances from $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ to the line are $t_1 - e_1$ and $t_2 - e_0$, respectively, where $t_i(\theta) = \boldsymbol{U}_1(\theta) \cdot \boldsymbol{Q}_i$ for $i = 1, 2$.

In summary, the signed distances depend on

$$s_i(\theta) = \boldsymbol{U}_0(\theta) \cdot \boldsymbol{Q}_i, \;\; t_i(\theta) = \boldsymbol{U}_1(\theta) \cdot \boldsymbol{Q}_i, \;\; 0 \le i < 4 \tag{4}$$

The signed distances depend also on the extents e_j. The fitting algorithm uses a least-squares formulation with error function

$$\begin{aligned} f(e_0, e_1, \theta) &= (s_0 - e_0)^2 + (s_1 - e_0)^2 + (s_2 + e_0)^2 + (s_3 + e_0)^2 + \\ &\quad (t_0 - e_1)^2 + (t_1 + e_1)^2 + (t_2 + e_1)^2 + (t_3 - e_1)^2 \\ &= 4e_0^2 - 2((s_0 + s_1) - (s_2 + s_3)) + 4e_1^2 - 2((t_0 + t_3) - (t_1 + t_2)) \end{aligned} \tag{5}$$

The idea is to force the points jointly to be close to the rectangle edges. The global minimum occurs at a location where the gradient of f is the zero vector.

The first-order partial derivative of f with respect to e_0 is

$$\frac{\partial f}{\partial e_0} = 8e_0 - 2((s_0 + s_1) - (s_2 + s_3)) \tag{6}$$

The partial derivative is zero when $e_0 = ((s_0 + s_1) - (s_2 + s_3))/4$.

The first-order partial derivative of f with respect to e_1 is

$$\frac{\partial f}{\partial e_0} = 8e_1 - 2((t_0 + t_3) - (t_1 + t_2)) \tag{7}$$

The partial derivative is zero when $e_1 = ((t_0 + t_3) - (t_1 + t_2))/4$.

Because s_i and t_i depend only on θ, the function $f(e_0, e_1, e_2)$ reduces to a function of θ only.

$$g(\theta) = \frac{1}{4}\left(\boldsymbol{s}(\theta)^{\mathsf{T}} A \boldsymbol{s}(\theta) + \boldsymbol{t}(\theta)^{\mathsf{T}} B \boldsymbol{t}(\theta)\right) \tag{8}$$

where

$$A = \begin{bmatrix} 3 & -1 & +1 & +1 \\ -1 & 3 & +1 & +1 \\ +1 & +1 & 3 & -1 \\ +1 & +1 & -1 & 3 \end{bmatrix}, \;\; B = \begin{bmatrix} 3 & +1 & +1 & -1 \\ +1 & 3 & -1 & +1 \\ +1 & -1 & 3 & +1 \\ -1 & +1 & +1 & 3 \end{bmatrix} \tag{9}$$

and where $\boldsymbol{s}(\theta)$ is the 4×1 representation of the 4-tuple (s_0, s_1, s_2, s_3) and $\boldsymbol{t}(\theta)$ is the 4×1 representation of the 4-tuple (t_0, t_1, t_2, t_3).

The derivatives of the rectangle axis directions are

$$\boldsymbol{U}_0'(\theta) = (-\sin\theta, \cos\theta) = \boldsymbol{U}_1(\theta), \;\; \boldsymbol{U}_1'(\theta) = (-\cos\theta, -\sin\theta) = -\boldsymbol{U}_1(\theta) \tag{10}$$

The derivatives of the s_i and t_i are

$$s_i'(\theta) = \boldsymbol{U}_0'(\theta)\cdot\boldsymbol{Q}_i = \boldsymbol{U}_1(\theta)\cdot\boldsymbol{Q}_i = t_i(\theta),\;\; t_i'(\theta) = \boldsymbol{U}_1'(\theta)\cdot\boldsymbol{Q}_i = -\boldsymbol{U}_0(\theta)\cdot\boldsymbol{Q}_i = -s_i(\theta),\;\; 0 \le i < 4 \tag{11}$$

The global minimum of $g(\theta)$ occurs when its first derivative is zero. That derivative is

$$g'(\theta) = \frac{1}{4}\left(\boldsymbol{s}(\theta)^{\mathsf{T}}A\boldsymbol{s}'(\theta) + \boldsymbol{t}(\theta)^{\mathsf{T}}B\boldsymbol{t}'(\theta)\right) = \frac{1}{4}\left(\boldsymbol{s}(\theta)^{\mathsf{T}}A\boldsymbol{t}(\theta) - \boldsymbol{t}(\theta)^{\mathsf{T}}B\boldsymbol{s}(\theta)\right) = \frac{1}{4}\,\boldsymbol{s}(\theta)^{\mathsf{T}}(A-B)\boldsymbol{t}(\theta) \tag{12}$$

where

$$A - B = 2\begin{bmatrix} 0 & -1 & 0 & +1 \\ -1 & 0 & +1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{13}$$

Some algebraic and trigonometric manipulation reduces equation (12) to

$$-k_0\cos(2\theta) + k_1\sin(2\theta) = 0 \tag{14}$$

where

$$\begin{aligned} k_0 &= (x_1 - x_3)(y_0 - y_2) + (x_0 - x_2)(y_1 - y_3) \\ k_1 &= (x_1 - x_3)(x_0 - x_2) - (y_0 - y_2)(y_1 - y_3) \end{aligned} \tag{15}$$

Equation (14) implies

$$(\sin(2\theta), \cos(2\theta)) = \pm\frac{(k_0, k_1)}{\sqrt{k_0^2 + k_1^2}} \tag{16}$$

The second-derivative test can be used to determine which sign corresponds to the global minimum. Notice that

$$(\sin(2\theta \pm \pi), \cos(2\theta \pm \pi)) = (-\sin(2\theta), -\cos(2\theta)) \tag{17}$$

From the geometry and algebra of choosing rectangle axes, it is sufficient to restrict $\theta \in [-\pi/2, \pi/2)$. The function $g(\theta)$ has a global minimum and a global maximum on this interval. Moreover, if $\theta_{\max}$ is the location of the global maximum, then $\theta_{\min} = \theta_{\max} \pm \pi/2$ are locations for the global minimum.

The second derivative of $g(\theta)$ is

$$\begin{aligned} g''(\theta) &= \tfrac{1}{4}\left(\boldsymbol{s}(\theta)^{\mathsf{T}}(A-B)\boldsymbol{t}'(\theta) + \boldsymbol{s}'(\theta)^{\mathsf{T}}B\boldsymbol{t}(\theta)\right) \\ &= \tfrac{1}{4}\left(-\boldsymbol{s}(\theta)^{\mathsf{T}}(A-B)\boldsymbol{s}(\theta) + \boldsymbol{t}(\theta)^{\mathsf{T}}(A-B)\boldsymbol{t}(\theta)\right) \\ &= ((s_0 - s_2)(s_1 - s_3) - (t_0 - t_2)(t_1 - t_3)) \end{aligned} \tag{18}$$

Equation (16) can be solved for 2θ using the `atan2` function. If $g''(\theta) > 0$, θ is the location of the minimum. Otherwise, $\theta + \pi/2$ is the location of the minimum. This location might be larger than π, but when used to evaluate $\boldsymbol{U}_0$ and $\boldsymbol{U}_1$, the sine and cosine terms produce the desired axes for the rectangle.

In the event θ is the location of the global maximum of $g(\theta)$, the location of the global minimum is $\theta + \pi/2$. The rectangle axis directions are

$$\boldsymbol{U}_0(\theta + \pi/2) = -\boldsymbol{U}_1(\theta),\;\; \boldsymbol{U}_1(\theta + \pi/2) = -\boldsymbol{U}_0(\theta) \tag{19}$$

We can use the already computed $\boldsymbol{U}_0(\theta)$ and $\boldsymbol{U}_1(\theta)$ instead of evaluating trigonometric functions at $\theta+\pi/2$.

If θ is the location of the global minimum, then the extents are

$$e_0(\theta) = ((s_0(\theta) - s_2(\theta)) + (s_1(\theta) - s_3(\theta))/4, \;\; e_1(\theta) = ((t_0(\theta) - t_2(\theta)) - (t_1(\theta) - t_3(\theta))/4 \tag{20}$$

If instead $\theta+\pi/2$ is the location of the global minimum, we can avoid evaluating trigonometric functions by using

$$s_i(\theta + \pi/2) = t_i(\theta), \;\; t_i(\theta + \pi/2) = -s_i(\theta) \tag{21}$$

to produce

$$e_0(\theta+\pi/2) = ((t_0(\theta) - t_2(\theta)) - (t_1(\theta) - t_3(\theta))/4, \;\; e_1(\theta+\pi/2) = -((s_0(\theta) - s_2(\theta)) + (s_1(\theta) - s_3(\theta))/4 \tag{22}$$

3 Pseudocode

Pseudocode for the algorithm is shown in Listing 1.

Listing 1. Pseudocode for fitting a rectangle to a convex quadrilateral that is nearly a rectangle. The P[i] had x-component x[i] and y-component y[i]. The rectangle has center C, unit-length and perpendicular axes U[0] and U[1], and extents e[0] and e[1]. If U[0] = (x,y), then U[1] = (-y,x).

```
Rectangle2 FitRectangleToConvexQuadrilateral(Vector2<Real> P[4])
{
    Rectangle2 rectangle;
    rectangle.C = (P[0] + P[1] + P[2] + P[3]) / 4;
    Vector2<Real> Q[4];
    for (int i = 0; i < 4; ++i)
    {
        Q[i] = P[i] - rectangle.C;
    }
    Vector2<Real> Q0mQ2 = Q[0] - Q[2];
    Vector2<Real> Q1mQ3 = Q[1] - Q[3];
    Real k0 = Q1mQ3[0] * Q0mQ2[1] + Q0mQ2[0] * Q1mQ3[1];
    Real k1 = Q1mQ3[0] * Q0mQ2[0] - Q1mQ3[1] * Q0mQ2[1];
    Real theta = atan2(k0, k1) / 2, cosTheta = cos(theta), sinTheta = sin(theta);
    rectangle.U[0] = { cosTheta, sinTheta };
    rectangle.U[1] = { -sinTheta, cosTheta };
    Real s0ms2 = Dot(rectangle.U[0], Q0mQ2);
    Real s1ms3 = Dot(rectangle.U[0], Q1mQ3);
    Real t0mt2 = Dot(rectangle.U[1], Q0mQ2);
    Real t1mt3 = Dot(rectangle.U[1], Q1mQ3);
    Real gder2test = s0ms2 * s1ms3 - t0mt2 * t1mt3;
    if (gder2test < 0)
    {
        // The angle θ is the location of the global maximum. The minimum is at θ + π/2.
        rectangle.U[0] = { -sinTheta, cosTheta };
        rectangle.U[1] = { -cosTheta, -sinTheta };
        Real oldt0mt2 = t0mt2, oldt1mt3 = t1mt3;
        t0mt2 = -s0ms2;
        t1mt3 = -s1ms3;
        s0ms2 = oldt0mt2;
        s1ms3 = oldt1mt3;
    }
    rectangle.e[0] = (s0ms2 + s1ms3) / 4;
    rectangle.e[1] = (t0mt2 - t1mt3) / 4;
    return rectangle;
}
```