Smoothing a sequence of line segments

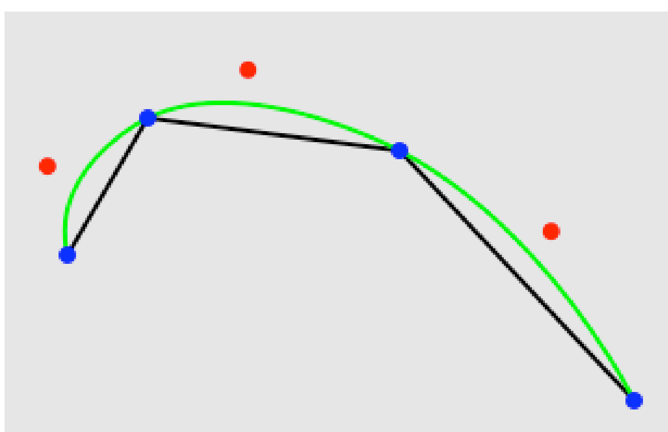
Wagner L. Truppel

July 2, 2007

Suppose we have a set of N points, $\{P_n \,|\, 0 \leq n < N\}$, which have been connected — in sequence — by line segments. How can we smooth the resulting path?

The solution is to replace each segment by a Bezier curve joining its end-points in such a way that adjacent curves are joined smoothly. In order to do so, we need to compute a set of *control points* for each Bezier curve. This document describes the mathematics involved in finding them and solves the problem in its entirety, for two kinds of Bezier curves.

Quadratic Bezier curves



The simplest non-trivial Bezier curve is a parabola, and requires a single control point. We'll attempt to replace each line-segment with one such curve. We'll define the control points so that the quadratic Bezier curve joining points P_n and P_{n+1} is controlled by the point labelled Q_n. Note that there are $N-1$ control points (drawn in red in the picture above).

With the convention above, the two Bezier curves replacing the segments $\overline{P_n P_{n+1}}$ and $\overline{P_{n+1} P_{n+2}}$ are:

$$\left| \begin{array}{rcl} B_n(t) &=& P_n\,(1-t)^2 \;+\; 2Q_n\,t\,(1-t) \;+\; P_{n+1}\,t^2 \\ B_{n+1}(t) &=& P_{n+1}\,(1-t)^2 \;+\; 2Q_{n+1}\,t\,(1-t) \;+\; P_{n+2}\,t^2 \end{array} \right.$$

where $0 \leq t \leq 1$.

Note that $B_n(0) = P_n$, $B_n(1) = B_{n+1}(0) = P_{n+1}$, and $B_{n+1}(1) = P_{n+2}$, that is, the two curves are joined at P_{n+1}. In order for them to join *smoothly*, however, we must require their tangents at that point to be equal to one another. We then need their derivatives:

$$\begin{array}{rcl} B'_n(t) &=& -2P_n\,(1-t) \;+\; 2Q_n\,(1-2t) \;+\; 2P_{n+1}\,t \\ B'_{n+1}(t) &=& -2P_{n+1}\,(1-t) \;+\; 2Q_{n+1}\,(1-2t) \;+\; 2P_{n+2}\,t \end{array}$$

Setting them equal at the point where they join gives us the *smoothness condition*:

$$B'_n(1) = B'_{n+1}(0) \qquad \Rightarrow \qquad Q_n + Q_{n+1} = 2P_{n+1}\,. \qquad (0 \leq n < N-2)$$

In other words, P_{n+1} has to bissect the line joining the two control points in question. Note that the above is a set of $N-2$ conditions, and we have $N-1$ control points to find, which means we need one more condition.

The conditions above allow us to obtain every control point once we know the *first* or the *last* one, but how do we determine that one? We do so by requiring that the Bezier curves have the smallest possible curvatures, which is to say that they're as close as possible to the straight segments they are replacing. We can't, however, require that every curve be of minimal curvature because that would impose too many constraints and we can only afford one more constraint. The solution is to require that the *overall* curvature be minimized, a goal we can attain by computing the overall departure from the line segments, as follows:

$$S\{Q_0, Q_1, \ldots, Q_{N-2}\} = \sum_{n=0}^{N-2} \int_0^1 dt \,\Big|\, B_n(t) - L_n(t) \,\Big|^2,$$

where $L_n(t) = (1-t)\,P_n + t\,P_{n+1}$ is the parametric curve representing the line segment starting at P_n and ending at P_{n+1}, and $|\ldots|^2$ represents the standard Euclidean distance measure.

The above is a functional form on the $N-1$ control points Q_n. We would like now to minimize its value with respect to variations on the values of these control points, but we can't vary them all *independently* because of the $N-2$ smoothness conditions derived previously. This is the prototypical task for the *Method of Lagrange Multipliers*. We can incorporate the conditions derived previously into our functional in such a way that we can then vary the control points independently, as follows.

We assign a Lagrange multiplier λ for each constraint and then redefine our functional:

$$S\{Q_0, Q_1, \ldots, Q_{N-2}\} = \sum_{n=0}^{N-2} \int_0^1 dt \left| B_n(t) - L_n(t) \right|^2 + \sum_{n=0}^{N-3} \lambda_n \left(Q_n + Q_{n+1} - 2P_{n+1} \right).$$

Note that we're really not changing the functional at all, since the constraints amount to a vanishing contribution. We may now minimize the modified functional with respect to independent variations of the control points. First, though, note that

$$B_n(t) - L_n(t) = t\,(1-t)\,(2Q_n - P_n - P_{n+1}).$$

Thus,

$$S\{Q_0, \ldots, Q_{N-2}\} = \int_0^1 dt\, t^2\,(1-t)^2 \sum_{n=0}^{N-2} \left| 2Q_n - P_n - P_{n+1} \right|^2 + \sum_{n=0}^{N-3} \lambda_n \left(Q_n + Q_{n+1} - 2P_{n+1} \right),$$

and the minimization condition takes the form of

$$\begin{aligned}
\frac{\partial S}{\partial Q_0} &= \int_0^1 dt\, 4t^2\,(1-t)^2\,(2Q_0 - P_0 - P_1) + \lambda_0 &&= 0\,, \\
\frac{\partial S}{\partial Q_k} &= \int_0^1 dt\, 4t^2\,(1-t)^2\,(2Q_k - P_k - P_{k+1}) + \lambda_k + \lambda_{k-1} &&= 0\,, \qquad (0 < k < N-2) \\
\frac{\partial S}{\partial Q_{N-2}} &= \int_0^1 dt\, 4t^2\,(1-t)^2\,(2Q_{N-2} - P_{N-2} - P_{N-1}) + \lambda_{N-3} &&= 0\,.
\end{aligned}$$

Using the result

$$I(m,n) \equiv \int_0^1 dt\, t^m\,(1-t)^n = \frac{m!\,n!}{(m+n+1)!}$$

we find

$$\int_0^1 dt\, 4t^2\,(1-t)^2 = 4\,I(2,2) = \frac{2}{15}\,,$$

and

$$\left\{ \begin{aligned}
\lambda_0 &= \tfrac{2}{15}\,(P_0 + P_1 - 2Q_0)\,, \\
\lambda_k + \lambda_{k-1} &= \tfrac{2}{15}\,(P_k + P_{k+1} - 2Q_k)\,, \qquad (0 < k < N-2) \\
\lambda_{N-3} &= \tfrac{2}{15}\,(P_{N-2} + P_{N-1} - 2Q_{N-2})\,.
\end{aligned} \right.$$

The above is a set of $N-1$ conditions. Together with the previously obtained $N-2$ constraints

$$Q_n + Q_{n+1} = 2P_{n+1} \qquad (0 \le n < N-2)$$

we have a total of $2N-3$ conditions on $N-1$ control points and $N-2$ Lagrange multipliers, for a total of $2N-3$ variables. This is a system of equations that should yield a unique solution, which we now proceed to derive.

First, define a new set of Lagrange multipliers, $\alpha_n \equiv (15/2)\,\lambda_n$. Then, our complete set of equations is

$$\begin{cases} \alpha_0 + 2Q_0 = P_0 + P_1\,, & \\ \alpha_{n-1} + \alpha_n + 2Q_n = P_n + P_{n+1}\,, & (0 < n < N-2) \\ \alpha_{N-3} + 2Q_{N-2} = P_{N-2} + P_{N-1}\,, & \\ Q_n + Q_{n+1} = 2P_{n+1}\,. & (0 \le n < N-2) \end{cases}$$

Now define an auxiliary set of variables, φ_n, such that

$$\begin{cases} \varphi_{-1} = 0\,, & \\ \varphi_n = \alpha_n = (15/2)\,\lambda_n\,, & (0 \le n < N-2) \\ \varphi_{N-2} = 0\,. & \end{cases}$$

Then, our set of conditions can be written as

$$\begin{cases} \varphi_{n-1} + \varphi_n + 2Q_n = P_n + P_{n+1}\,, & (0 \le n \le N-2) \\ Q_n + Q_{n+1} = 2P_{n+1}\,. & (0 \le n < N-2) \end{cases}$$

From the second set of conditions, we find:

$$\begin{aligned} Q_{N-3} & & &= 2P_{N-2} - Q_{N-2} \\ Q_{N-4} &= 2P_{N-3} - Q_{N-3} &&= 2P_{N-3} - 2P_{N-2} + Q_{N-2} \\ Q_{N-5} &= 2P_{N-4} - Q_{N-4} &&= 2P_{N-4} - 2P_{N-3} + 2P_{N-2} - Q_{N-2} \\ Q_{N-6} &= 2P_{N-5} - Q_{N-5} &&= 2P_{N-5} - 2P_{N-4} + 2P_{N-3} - 2P_{N-2} + Q_{N-2} \end{aligned}$$

and so on. We see that the sign of Q_{N-2} is positive whenever N minus the index of the control point on the left is an even number, and negative when that difference is an odd number. Thus,

$$Q_n = (-1)^{N-n} Q_{N-2} + \sum_{k=0}^{N-3-n} (-1)^k\, 2P_{n+k+1}\,. \qquad (0 \le n < N-2)$$

From the first set of conditions, we have:

$$\begin{array}{lclcl}
\varphi_0 & & & = & P_1 + P_0 - 2Q_0 \\
\varphi_1 & = & P_1 + P_2 - 2Q_1 - \varphi_0 & = & P_2 - P_0 - 2Q_1 + 2Q_0 \\
\varphi_2 & = & P_2 + P_3 - 2Q_2 - \varphi_1 & = & P_3 + P_0 - 2Q_2 + 2Q_1 - 2Q_0 \\
\varphi_3 & = & P_3 + P_4 - 2Q_3 - \varphi_2 & = & P_4 - P_0 - 2Q_3 + 2Q_2 - 2Q_1 + 2Q_0
\end{array}$$

and so on. This is similar to the previous case, and has the solution

$$\varphi_n = (-1)^n P_0 + P_{n+1} + \sum_{k=0}^{n} (-1)^{n+k+1}\, 2Q_k \,. \qquad (0 \le n \le N-2)$$

Now, since $\varphi_{N-2} = 0$, we demand

$$\varphi_{N-2} = (-1)^N P_0 + P_{N-1} + \sum_{k=0}^{N-2} (-1)^{N+k+1}\, 2Q_k = 0\,,$$

which can be solved for Q_{N-2}:

$$2Q_{N-2} = (-1)^N P_0 + P_{N-1} + \sum_{k=0}^{N-3} (-1)^{N+k+1}\, 2Q_k\,.$$

We finally have a set of $N-1$ equations involving only the $N-1$ control points:

$$\left\{ \begin{array}{rcl}
Q_n & = & (-1)^{N-n} Q_{N-2} + \displaystyle\sum_{k=0}^{N-3-n} (-1)^k\, 2P_{n+k+1} \qquad (0 \le n < N-2) \\
Q_{N-2} & = & \frac{1}{2}\left[(-1)^N P_0 + P_{N-1} \right] + \displaystyle\sum_{n=0}^{N-3} (-1)^{N+n+1}\, Q_n\,.
\end{array} \right.$$

These can be easily solved for Q_{N-2}, resulting in:

$$(N-1)\, Q_{N-2} = \frac{1}{2}\left[(-1)^N P_0 + P_{N-1} \right] + \sum_{n=0}^{N-3} \sum_{k=0}^{N-3-n} (-1)^{N+n+k+1}\, 2P_{n+k+1}\,.$$

The double summation can be simplified and we obtain:

$$\left| \; (N-1)\, Q_{N-2} = \frac{1}{2}\left[(-1)^N P_0 + P_{N-1} \right] + \sum_{k=1}^{N-2} (-1)^{N+k}\, 2k\, P_k\,. \right.$$

Having solved for Q_{N-2}, the remaining Q_n control points can be computed just as easily from

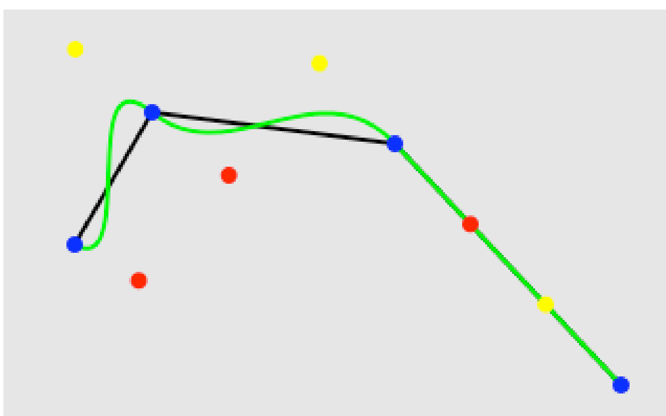
$$Q_n = (-1)^{N-n} Q_{N-2} + \sum_{k=0}^{N-3-n} (-1)^k \, 2P_{n+k+1} \,. \qquad (0 \le n < N-2)$$

Finally, we can also compute the overall departure from the line segments:

$$S = \frac{1}{30} \sum_{n=0}^{N-2} \left| 2Q_n - P_n - P_{n+1} \right|^2 .$$

This concludes the derivation of the control points for the case when the line segments are replaced with *quadratic* Bezier curves.

Cubic Bezier curves



If we decide to replace the line segments with *cubic* Bezier curves, we'll require *two* control points per Bezier curve, defined so that the Bezier curve joining points P_n and P_{n+1} is controlled by the points labelled Q_n and Q'_n (drawn in red and yellow, respectively, in the picture above). Note that there are now $2(N-1)$ control points.

With the convention above, the two Bezier curves replacing the segments $\overline{P_n P_{n+1}}$ and $\overline{P_{n+1} P_{n+2}}$ are:

$$\begin{aligned} B_n(t) &= P_n \,(1-t)^3 + 3Q_n \,t\,(1-t)^2 + 3Q'_n \,t^2\,(1-t) + P_{n+1}\,t^3 \\ B_{n+1}(t) &= P_{n+1} \,(1-t)^3 + 3Q_{n+1} \,t\,(1-t)^2 + 3Q'_{n+1} \,t^2\,(1-t) + P_{n+2}\,t^3 \end{aligned}$$

where $0 \le t \le 1$.

Note once again that $B_n(0) = P_n$, $B_n(1) = B_{n+1}(0) = P_{n+1}$, and $B_{n+1}(1) = P_{n+2}$, that is, the two curves are again joined at P_{n+1}. In order for them to join *smoothly*, however, we once more must require their tangents at that point to be equal to one another. Their derivatives are:

$$\begin{aligned} B'_n(t) &= \quad -3P_n\,(1-t)^2 + \quad 3Q_n\,(1-t)(1-3t) + \quad 3Q'_n\,t\,(2-3t) + 3P_{n+1}\,t^2 \\ B'_{n+1}(t) &= -3P_{n+1}\,(1-t)^2 + 3Q_{n+1}\,(1-t)(1-3t) + 3Q'_{n+1}\,t\,(2-3t) + 3P_{n+2}\,t^2 \end{aligned}$$

Setting them equal at the point where they join gives us the *smoothness condition*:

$$B'_n(1) = B'_{n+1}(0) \qquad \Rightarrow \qquad Q'_n + Q_{n+1} = 2P_{n+1}\,. \qquad (0 \le n < N-2)$$

In other words, P_{n+1} again has to bissect the line joining its two adjacent control points. And, once again, we need more constraints to be able to determine all the control points. This time, however, we have $2(N-1)$ control points and, so far, only $N-2$ conditions, which means we need N more constraints. We can obtain the necessary additional conditions by imposing that *each* Bezier curve is as close as possible to the line segment it replaces. We'll follow a derivation similar to the quadratic case.

First, define a functional of the various control points, for each curve, that measures its deviation from the line segment it replaces:

$$S\{Q_n, Q'_n\} = \int_0^1 dt\,\Big|\,B_n(t) - L_n(t)\,\Big|^2, \qquad (0 \le n < N-1)$$

where $L_n(t) = (1-t)\,P_n + t\,P_{n+1}$ is again the parametric curve representing the line segment starting at P_n and ending at P_{n+1}, and $|\ldots|^2$ represents the standard Euclidean distance measure.

From the definition of the cubic Bezier curve, we find that

$$B_n(t) - L_n(t) = t\,(3Q_n - 2P_n - P_{n+1}) + 3t^2\,(P_n + Q'_n - 2Q_n) + t^3\,(3Q_n - 3Q'_n + P_{n+1} - P_n)$$

and, so,

$$S\{Q_n, Q'_n\} = \int_0^1 dt\,t^2\,\Big|\,(3Q_n - 2P_n - P_{n+1}) + 3t\,(P_n + Q'_n - 2Q_n) + t^2\,(3Q_n - 3Q'_n + P_{n+1} - P_n)\,\Big|^2.$$

This is a functional of both Q_n and Q'_n but — for the first $N-2$ curves — can be turned into a functional of the Q values alone, by making use of the smoothness conditions:

$$\begin{aligned} S\{Q_n, Q_{n+1}\} = \int_0^1 dt\,t^2\,\Big|\,&(3Q_n - 2P_n - P_{n+1}) + 3t\,(P_n + 2P_{n+1} - 2Q_n - Q_{n+1}) + \\ &t^2\,(3Q_n + 3Q_{n+1} - 5P_{n+1} - P_n)\,\Big|^2. \qquad (0 \le n < N-2) \end{aligned}$$

For $n = N-2$, however, we must still use the original form:

$$S\{Q_{N-2}, Q'_{N-2}\} = \int_0^1 dt\, t^2 \Big| (3Q_{N-2} - 2P_{N-2} - P_{N-1}) + \\ 3t\,(P_{N-2} + Q'_{N-2} - 2Q_{N-2}) + t^2\,(3Q_{N-2} - 3Q'_{N-2} + P_{N-1} - P_{N-2}) \Big|^2 .$$

The minimization procedure then yields

$$\frac{\partial S}{\partial Q_k} = \int_0^1 dt\, 6t^2\,(1-t)^2 \Big[(3Q_k - 2P_k - P_{k+1}) + 3t\,(P_k + 2P_{k+1} - 2Q_k - Q_{k+1}) + \\ t^2\,(3Q_k + 3Q_{k+1} - 5P_{k+1} - P_k) \Big] = 0\,, \qquad (0 \le k < N-2)$$

$$\frac{\partial S}{\partial Q_{N-2}} = \int_0^1 dt\, 6t^2\,(1-t)^2 \Big[(3Q_{N-2} - 2P_{N-2} - P_{N-1}) + 3t\,(P_{N-2} + Q'_{N-2} - 2Q_{N-2}) + \\ t^2\,(3Q_{N-2} - 3Q'_{N-2} + P_{N-1} - P_{N-2}) \Big] = 0\,,$$

$$\frac{\partial S}{\partial Q'_{N-2}} = \int_0^1 dt\, 6t^3\,(1-t) \Big[(3Q_{N-2} - 2P_{N-2} - P_{N-1}) + 3t\,(P_{N-2} + Q'_{N-2} - 2Q_{N-2}) + \\ t^2\,(3Q_{N-2} - 3Q'_{N-2} + P_{N-1} - P_{N-2}) \Big] = 0\,.$$

We now have the N additional conditions necessary for the unique determination of all control points.

In terms of the integral

$$I(m,n) \equiv \int_0^1 dt\, t^m\,(1-t)^n = \frac{m!\,n!}{(m+n+1)!}$$

we have

$$I(2,2)\,(3Q_k - 2P_k - P_{k+1}) + 3\,I(3,2)\,(P_k + 2P_{k+1} - 2Q_k - Q_{k+1}) + \\ I(4,2)\,(3Q_k + 3Q_{k+1} - 5P_{k+1} - P_k) = 0\,, \qquad (0 \le k < N-2)$$

$$I(2,2)\,(3Q_{N-2} - 2P_{N-2} - P_{N-1}) + 3\,I(3,2)\,(P_{N-2} + Q'_{N-2} - 2Q_{N-2}) + \\ I(4,2)\,(3Q_{N-2} - 3Q'_{N-2} + P_{N-1} - P_{N-2}) = 0\,,$$

$$I(3,1)\,(3Q_{N-2} - 2P_{N-2} - P_{N-1}) + 3\,I(4,1)\,(P_{N-2} + Q'_{N-2} - 2Q_{N-2}) + \\ I(5,1)\,(3Q_{N-2} - 3Q'_{N-2} + P_{N-1} - P_{N-2}) = 0\,.$$

They simplify to

$$3\,[I(2,2) - 2\,I(3,2) + I(4,2)]\,Q_k + 3\,[I(4,2) - I(3,2)]\,Q_{k+1} +$$
$$[3\,I(3,2) - 2\,I(2,2) - I(4,2)]\,P_k + [6\,I(3,2) - I(2,2) - 5\,I(4,2)]\,P_{k+1} = 0\,,$$
$$(0 \le k < N-2)$$

$$3\,[I(2,2) - 2\,I(3,2) + I(4,2)]\,Q_{N-2} - 3\,[I(4,2) - I(3,2)]\,Q'_{N-2} +$$
$$[3\,I(3,2) - 2\,I(2,2) - I(4,2)]\,P_{N-2} + [I(4,2) - I(2,2)]\,P_{N-1} = 0\,,$$

$$3\,[I(3,1) - 2\,I(4,1) + I(5,1)]\,Q_{N-2} + 3\,[I(4,1) - I(5,1)]\,Q'_{N-2} +$$
$$[3\,I(4,1) - 2\,I(3,1) - I(5,1)]\,P_{N-2} + [I(5,1) - I(3,1)]\,P_{N-1} = 0\,.$$

Defining the constants

$$\begin{aligned}
A &\equiv 3\,[I(2,2) - 2\,I(3,2) + I(4,2)] &&= \frac{1}{35} \\
B &\equiv 3\,[I(4,2) - I(3,2)] &&= -\frac{3}{140} \\
C &\equiv 3\,I(3,2) - 2\,I(2,2) - I(4,2) &&= -\frac{11}{420} \\
D &\equiv 6\,I(3,2) - I(2,2) - 5\,I(4,2) &&= \frac{2}{105} \\
E &\equiv I(4,2) - I(2,2) &&= -\frac{1}{42} \\
F &\equiv 3\,[I(3,1) - 2\,I(4,1) + I(5,1)] &&= \frac{3}{140} \\
G &\equiv 3\,[I(4,1) - I(5,1)] &&= \frac{1}{35} \\
H &\equiv 3\,I(4,1) - 2\,I(3,1) - I(5,1) &&= -\frac{1}{42} \\
I &\equiv I(5,1) - I(3,1) &&= -\frac{11}{420}
\end{aligned}$$

we have our complete set of constraints

$$\begin{cases}
Q'_n + Q_{n+1} - 2P_{n+1} \;=\; 0\,, & (0 \le n < N-2) \\
A\,Q_n + B\,Q_{n+1} + C\,P_n + D\,P_{n+1} \;=\; 0\,, & (0 \le n < N-2) \\
A\,Q_{N-2} - B\,Q'_{N-2} + C\,P_{N-2} + E\,P_{N-1} \;=\; 0\,, & \\
F\,Q_{N-2} + G\,Q'_{N-2} + H\,P_{N-2} + I\,P_{N-1} \;=\; 0\,. &
\end{cases}$$

From the last two we can easily solve for Q_{N-2} and Q'_{N-2}:

$$(AG+BF)\,Q_{N-2} \;=\; -(CG+BH)\,P_{N-2} \;-\; (EG+BI)\,P_{N-1}\,,$$
$$(AG+BF)\,Q'_{N-2} \;=\; \quad (CF-AH)\,P_{N-2} \;+\; (EF-AI)\,P_{N-1}\,,$$

or, more simply,

$$\left|\;\begin{aligned} Q_{N-2} \;&=\; \tfrac{2}{3}\,P_{N-2} \;+\; \tfrac{1}{3}\,P_{N-1}\,, \\ Q'_{N-2} \;&=\; \tfrac{1}{3}\,P_{N-2} \;+\; \tfrac{2}{3}\,P_{N-1}\,. \end{aligned}\right.$$

Using the result for Q_{N-2} and the second equation from the list of constraints,

$$Q_n = \;-\,\frac{B}{A}\,Q_{n+1} - \frac{C}{A}\,P_n - \frac{D}{A}\,P_{n+1}\,, \qquad (0 \le n < N-2)$$

which simplifies to

$$\left|\; Q_n = \frac{1}{12}\Big(9\,Q_{n+1} + 11\,P_n - 8\,P_{n+1}\Big)\,, \qquad (0 \le n < N-2)\right.$$

we can recursively solve (in decreasing order of n) for all Q_n values. Finally, having all the Q_n values, we can use the first set of constraints

$$\left|\; Q'_n = 2P_{n+1} - Q_{n+1}\,, \qquad (0 \le n < N-2)\right.$$

to solve for all the Q'_n values, completing the solution.

We may also, as before, compute the overall departure from the straight-line segments

$$S = \sum_{n=0}^{N-2} \int_0^1 dt\, t^2 \,\Big|\, (3Q_n - 2P_n - P_{n+1}) + 3t\,(P_n + Q'_n - 2Q_n) + t^2\,(3Q_n - 3Q'_n + P_{n+1} - P_n)\,\Big|^2 .$$

This can be ‘simplified’ to:

$$\left|\;\begin{aligned} S \;=\; \sum_{n=0}^{N-2} \Big\{ &\frac{1}{3}\,\Big|\, 3Q_n - 2P_n - P_{n+1}\,\Big|^2 + \frac{9}{5}\,\Big|\, P_n + Q'_n - 2Q_n\,\Big|^2 \\ &+\; \tfrac{1}{7}\,\Big|\, 3Q_n - 3Q'_n + P_{n+1} - P_n\,\Big|^2 + \tfrac{3}{2}\,(3Q_n - 2P_n - P_{n+1}) \cdot (P_n + Q'_n - 2Q_n) \\ &+\; \tfrac{2}{5}\,(3Q_n - 2P_n - P_{n+1}) \cdot (3Q_n - 3Q'_n + P_{n+1} - P_n) \\ &+\; (P_n + Q'_n - 2Q_n) \cdot (3Q_n - 3Q'_n + P_{n+1} - P_n)\Big\}\,. \end{aligned}\right.$$

Summary

Quadratic Bezier curves

$$
\begin{aligned}
B_n(t) &= P_n\,(1-t)^2 + 2Q_n\,t\,(1-t) + P_{n+1}\,t^2 \\
&\qquad (\,0 \le n < N-1, \quad 0 \le t \le 1\,)
\end{aligned}
$$

$$
\begin{aligned}
(N-1)\,Q_{N-2} &= \tfrac{1}{2}\Big[\,(-1)^N P_0 + P_{N-1}\Big] + \sum_{k=1}^{N-2}(-1)^{N+k}\,2k\,P_k \\
Q_n &= (-1)^{N-n}Q_{N-2} + \sum_{k=0}^{N-3-n}(-1)^k\,2P_{n+k+1} \qquad (0 \le n < N-2) \\
S &= \tfrac{1}{30}\sum_{n=0}^{N-2}\Big|\,2Q_n - P_n - P_{n+1}\Big|^2 \qquad\qquad \bar{S} = \frac{S}{N-1}
\end{aligned}
$$

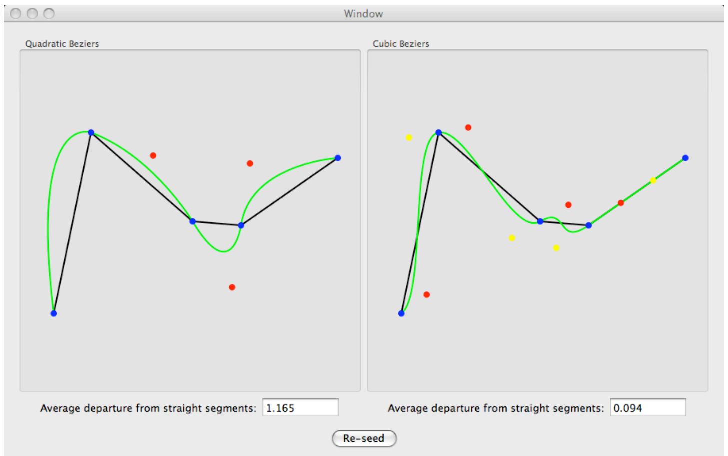
Cubic Bezier curves

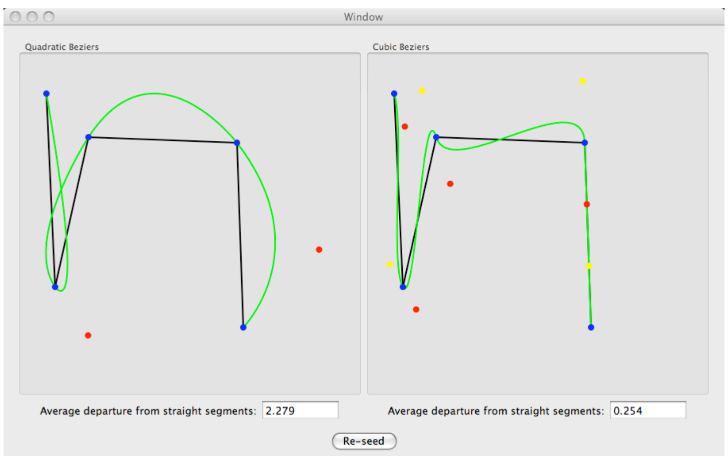
$$
\begin{aligned}
B_n(t) &= P_n\,(1-t)^3 + 3Q_n\,t\,(1-t)^2 + 3Q'_n\,t^2\,(1-t) + P_{n+1}\,t^3 \\
&\qquad (\,0 \le n < N-1, \quad 0 \le t \le 1\,)
\end{aligned}
$$

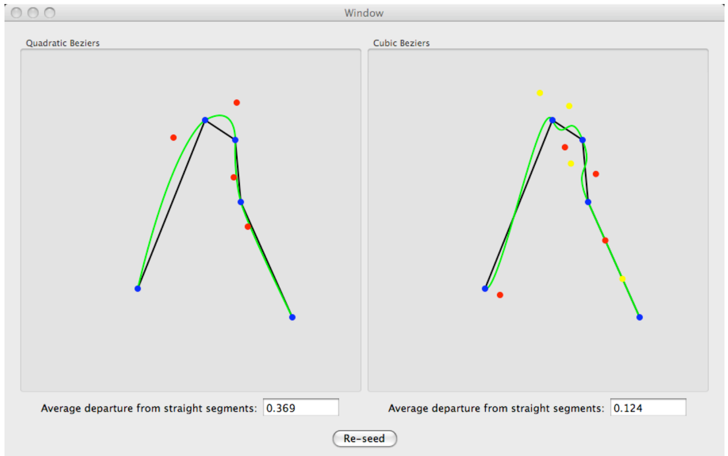
$$
\begin{aligned}
Q'_{N-2} &= \tfrac{1}{3}\,P_{N-2} + \tfrac{2}{3}\,P_{N-1} \\
Q_{N-2} &= \tfrac{2}{3}\,P_{N-2} + \tfrac{1}{3}\,P_{N-1} \\
Q_n &= \tfrac{11}{12}\,P_n + \tfrac{3}{4}\,Q_{n+1} - \tfrac{2}{3}\,P_{n+1} \qquad (0 \le n < N-2) \\
Q'_n &= 2P_{n+1} - Q_{n+1} \qquad (0 \le n < N-2) \\
S &= \sum_{n=0}^{N-2}\Big\{\,\frac{1}{3}\Big|\,3Q_n - 2P_n - P_{n+1}\Big|^2 + \frac{9}{5}\Big|\,P_n + Q'_n - 2Q_n\Big|^2 \\
&+ \tfrac{1}{7}\Big|\,3Q_n - 3Q'_n + P_{n+1} - P_n\Big|^2 + \tfrac{3}{2}\,(3Q_n - 2P_n - P_{n+1})\cdot(P_n + Q'_n - 2Q_n) \\
&+ \tfrac{2}{5}\,(3Q_n - 2P_n - P_{n+1})\cdot(3Q_n - 3Q'_n + P_{n+1} - P_n) \\
&+ (P_n + Q'_n - 2Q_n)\cdot(3Q_n - 3Q'_n + P_{n+1} - P_n)\Big\} \qquad\qquad \bar{S} = \frac{S}{N-1}
\end{aligned}
$$

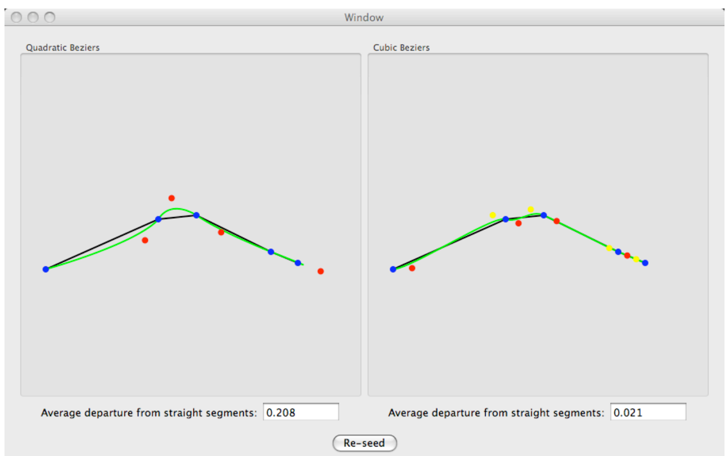
■

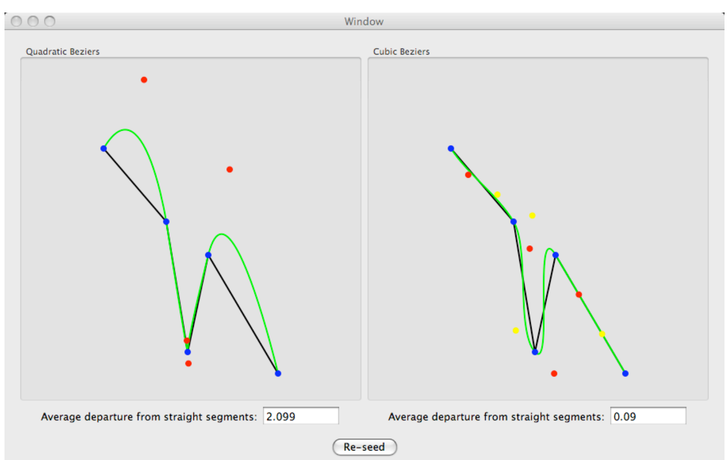
Examples

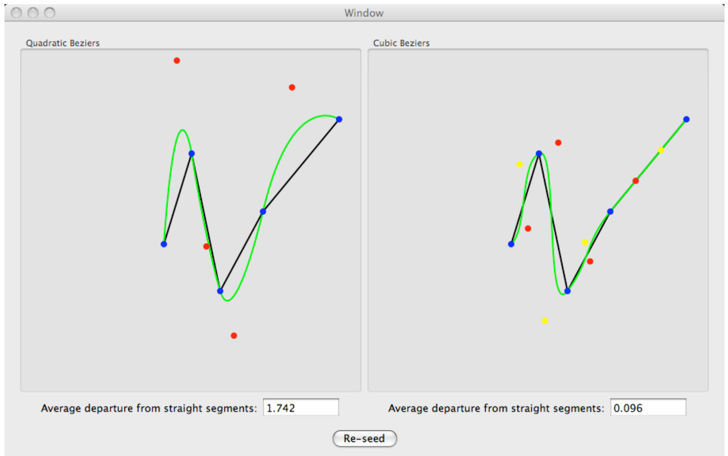

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 1.165
Average departure from straight segments: 0.094
Re-seed

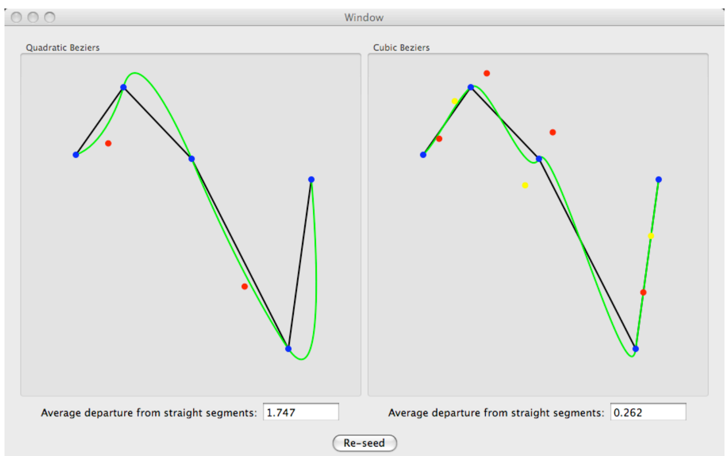

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 2.279
Average departure from straight segments: 0.254
Re-seed

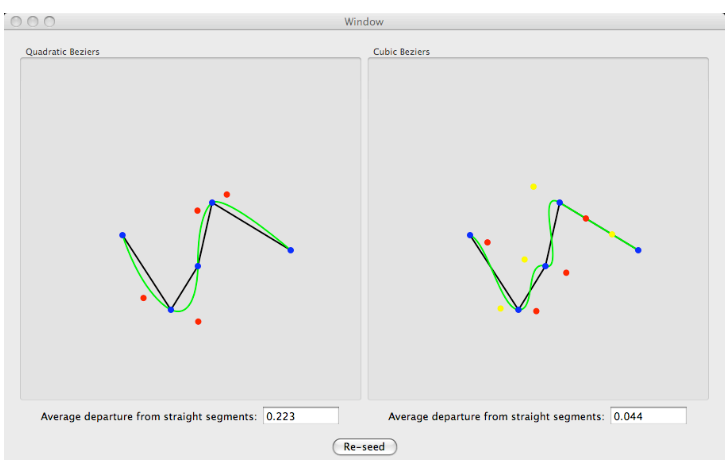

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.369
Average departure from straight segments: 0.124
Re-seed

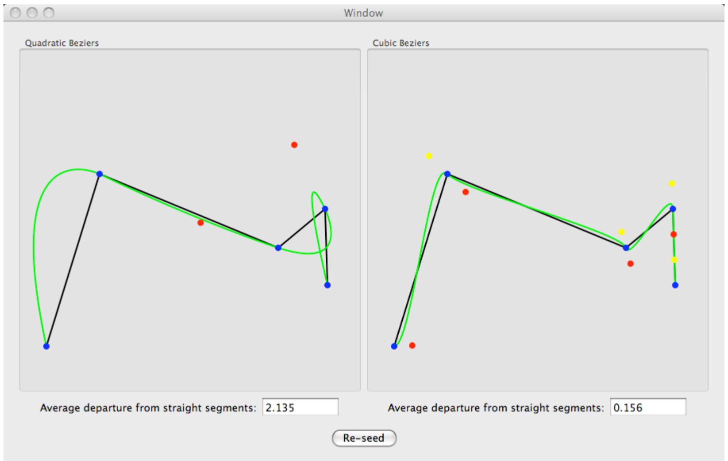

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.208
Average departure from straight segments: 0.021
Re-seed

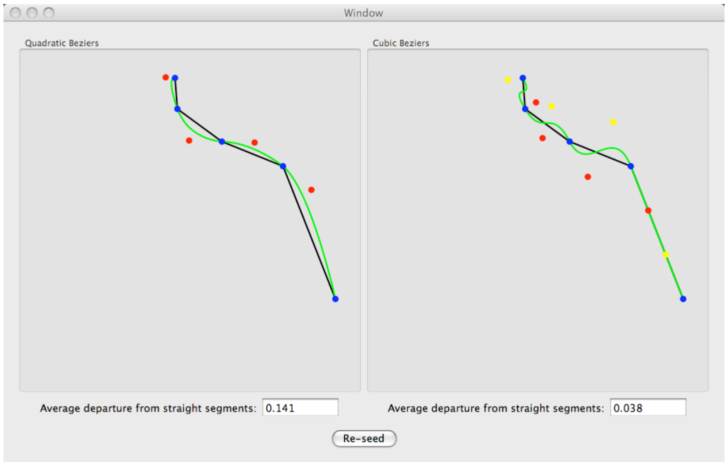

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 2.099
Average departure from straight segments: 0.09
Re-seed

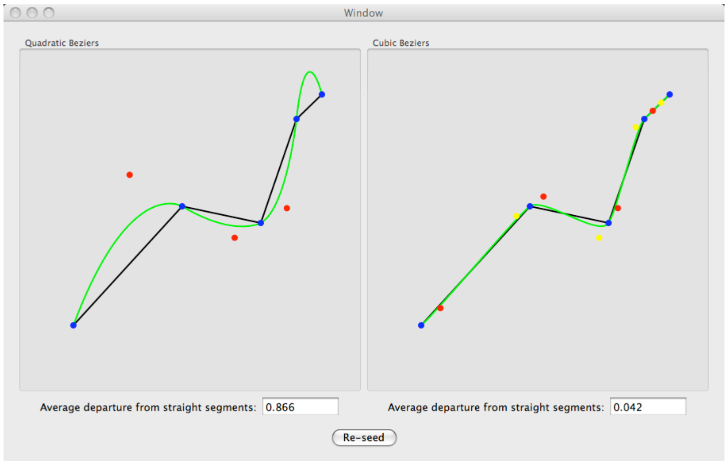

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 1.742
Average departure from straight segments: 0.096
Re-seed

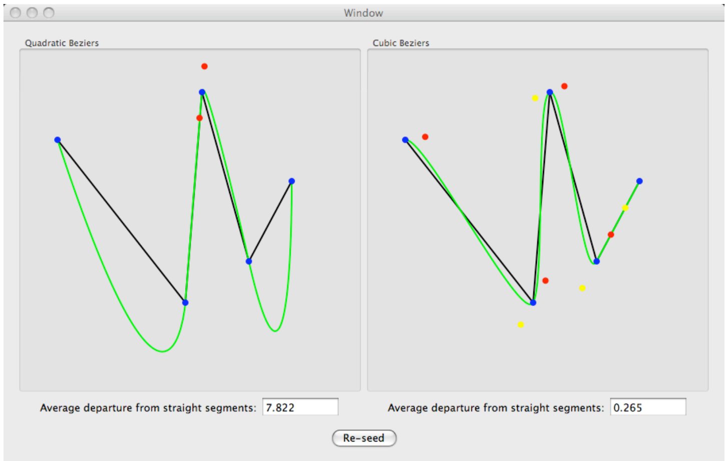

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 1.747
Average departure from straight segments: 0.262
Re-seed

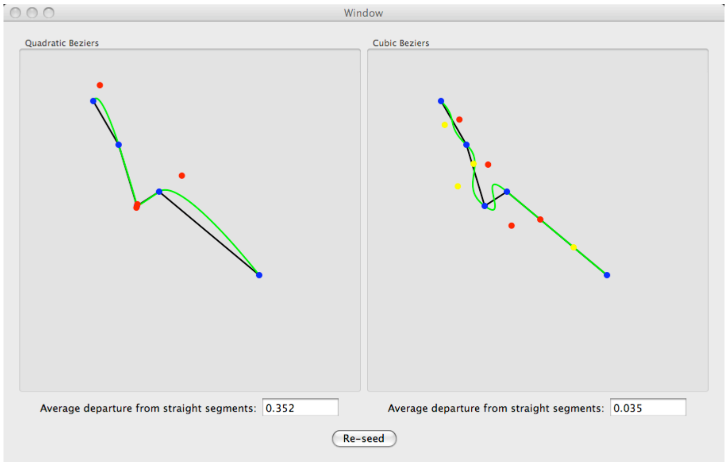

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.223
Average departure from straight segments: 0.044
Re-seed

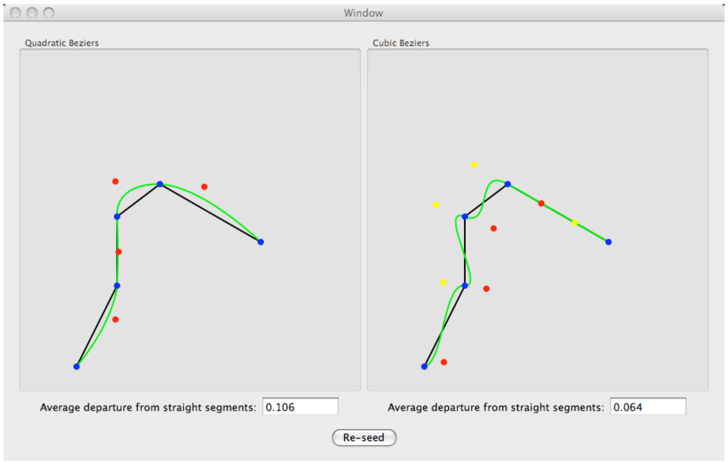

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 2.135
Average departure from straight segments: 0.156
Re-seed

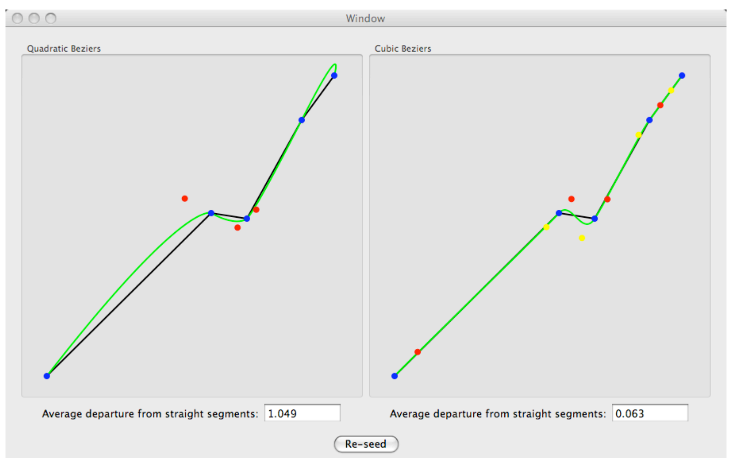

Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.141
Average departure from straight segments: 0.038
Re-seed


Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.866
Average departure from straight segments: 0.042
Re-seed


Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 7.822
Average departure from straight segments: 0.265
Re-seed


Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.352
Average departure from straight segments: 0.035
Re-seed


Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 0.106
Average departure from straight segments: 0.064
Re-seed


Window
Quadratic Beziers
Cubic Beziers
Average departure from straight segments: 1.049
Average departure from straight segments: 0.063
Re-seed