Literature Review 1

Primary paper:

κ-Curves: Interpolation at Local Maximum Curvature

BibTeX:

```
@article{Yan:2017:KCI:3072959.3073692,
  author = {Yan, Zhipei and Schiller, Stephen and Wilensky, Gregg and Carr, Nathan and Schaefer, Scott},
  title = {K-curves: Interpolation at Local Maximum Curvature},
  journal = {ACM Trans. Graph.},
  issue_date = {July 2017},
  volume = {36},
  number = {4},
  month = jul,
  year = {2017},
  issn = {0730-0301},
  pages = {129:1--129:7},
  articleno = {129},
  numpages = {7},
  url = {http://doi.acm.org/10.1145/3072959.3073692},
  doi = {10.1145/3072959.3073692},
  acmid = {3073692},
  publisher = {ACM},
  address = {New York, NY, USA},
  keywords = {curvature continuity, interpolatory curves, monotonic curvature},
}
```

Secondary paper:

Constructing G1 Quadratic Bézier Curves with Arbitrary Endpoint Tangent Vectors

BibTeX:

```
@inproceedings{Gu2009ConstructingGQ,
   title={Constructing G1 quadratic B{\'e}zier curves with arbitrary endpoint tangent vectors},
   author={He-Jin Gu and Jun-Hai Yong and Jean-Claude Paul and Fuhua Cheng},
   booktitle={CAD/Graphics},
   year={2009}
}
```

Summary:

Gu et al. [Gu et al. 2009] construct the G^1 quadratic Bézier curves with two segments that can fit all the directions of the endpoint tangent vectors and proof it. The reason why authors want to proof previous statement is that people often ignore the single segment may not fit that. At first, Gu et al. list the necessary and sufficient conditions for the single quadratic Bézier curve, in order to show that single one may be not enough to satisfy all end points directions. On the second step, authors build two segments to proof that will be better to fit endpoint constrains and use $C_1(s)$ and $C_2(s)$ to represent two quadratics that has $P_{0,1}$, $P_{0,2}$ and $P_{2,1}$, $P_{2,2}$ endpoints respectively. In order to compose those two quadratics, they made $P_{0,2}$ equal to $P_{2,1}$. In addition, there are two control points $P_{1,1}$ and $P_{1,2}$ which $P_{1,1} = P_{0,1} + rV_0$ and $P_{1,2} = P_{2,2} – rV_1$ (V_0, V_1 are endpoint tangent vectors, r is a specific parameter) and these two points are the intersection points of two lines, one of line is in V_0 direction and has $P_{0,1}$, and the other one is in V_1 direction and has $P_{2,2}$. After several transformations and substitutions, Gu et al. have a quadratic equation in terms of r: $(P_{1,2} - P_{1,1})^2 – 4r^2 = 0$. Finally, authors get an equation in terms of r, $\mathrm{r} = \frac{\|P_{2,2} - P_{0,1}\|}{2(cos\alpha + cos\beta)}$ (r only be the positive value, α is the angle between $(P_{2,2} - P_{0,1})$ to V_0, β is the angle between $(P_{2,2} - P_{0,1})$ to V_1). At the final part, authors show some example of the two G^1 quadratic Bézier curves. There are two shapes of the composite Bézier curve: the “C”-shape (V_0 and V_1 have the same direction) and the “S”-shape (V_0 and V_1 have the opposite direction). These shape is determined by the directions of two endpoints and the value of r. According to previous, Gu et al. successfully construct and proof the G^1 quadratic Bézier curves that meet the endpoint constrains.

Yan et al. [Yan et al. 2017] construct the piecewise quadratic curves that have control points only at the positon where the absolute value of the curvature is the maximum. The reason why authors want to create this kind of curves is that it can provide users with the directly control of the curve which has certain geometric properties. And this curve system can be implemented in the advertising illustration software. Yan et al. have the premise that the salient geometric features in this K-curve only appear at those control points because the maximum curvature is one of the geometric features of curves. Based on this prerequisite, at the first step, authors use three control points $c_{i,0}$, $c_{i,1}$, $c_{i,2}$ to represent the quadratic curve that the point p_i is interpolated at the point of maximum curvature and $c_i(t_i) = p_i$. They specify the curve by $c_{i,1}$ and parameter t_i. At the second step, they do smoothness that connect those piecewise curves together. They make the connected curves are almost G^2 and have G^1 curves at the point of junction where convexity changed and have opposite signs. Yan et al. uses λ_i as a specific parameter to produce a G^2 curve. At the final step, they optimize and combine what have done previously. For the first iteration, the initial value of λ_i is $\frac{1}{2}$ and $c_{i,1}$ = p_i. After estimate the λ_i, they use an equation to get updated $c_{i,0}$, $c_{i,2}$ and then compute t_i so that have the position of p_i, which is the position of $c_{i,1}$. They do these on several iterations until all the maximum convexity converge. The last but not the least, the end-points, Yan et al. bound $c_{n,2}$ to p_{n+1}. Finally, they create the steerable curves which has the maximum curvature at the control points. This method has good and fast performance because that the control points are coincide with a large number of the maximum curvature points just after one iteration.

Relationship:

Yan et al. cited Gu et al. paper and introduce in brief. After read their papers, I find there are something in common. At first, I think Yan et al. using Gu et al. conclusion as a basis. Because the proof of the G^1 curve can have arbitrary endpoint vectors, so Yan et al. are not worry about the connect point where the curvature changed. From this viewpoint, although Yan et al. develop the curves different from others' methods, they still need fundamental knowledge to support the basic operations of curve. Secondly, from the obvious part, they both use three control points to represent the piecewise quadratic curves. Yan et al. have $c_{i,0}$, $c_{i,1}$, $c_{i,2}$ and $c_{i,1}$ is the main control point. This is similar with Gu et al. construction. Moreover, they come up to use the piecewise curve to find the method and solution. Because the curve in pieces is easier to control and represent so that they can find out the simple equation and good parameter to calculate. For building various curves, they all choose to give the endpoints and then use the specific parameter to determine the shape that they want to get.

Although Yan et al. and Gu et al. have common things in some aspects, in my opinion, I also find some differences between them. Firstly, Yan et al. do no use Bézier form to represent the curve because a cubic parametric curve has three maximum in every segment. It is difficult to use to get just one maximum curvature point. However, Gu et al. proof the single Bézier curve may not fit constrains by using the definition of that curve. Secondly, Yan et al. focus on making the joint points enforce the G^2 continuity as they have the connected points between different curvature curves, but Gu et al. just aim to control all the edges of $C_1(s)$ and $C_2(s)$ have the same length by using different values of r when they connect two segments of curves. Thirdly, because G^2 curve have features: continuity, tangency and curvature while G^1 curve have features: continuity and tangency, so when they try to compute in terms of other value, Yan et al. choose to start from the curvature equation and Gu et al. begin at the its tangency. Fourthly, in Yan et al. method, every two curves that have the same curvature create "C" shape, while in Gu et al. those two segments have opposite tangencies.