MY **BRIGHT** FUTURE

DSU Dongseo University 동서대학교

Mobile Game Programming:

Hermite Spline

jintaeks@dongseo.ac.kr
Division of Digital Contents, DongSeo University.
February 2019

Type of functions

Explicit function

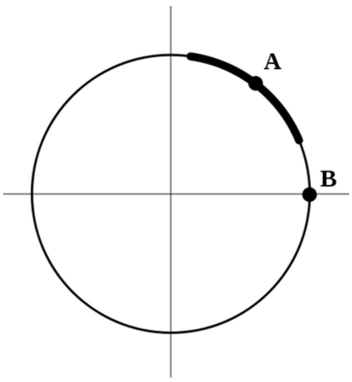
y=$\sqrt[2]{R^2 - x^2}$ or y=-$\sqrt[2]{R^2 - x^2}$

Implicit function

$x^2+y^2=R^2$

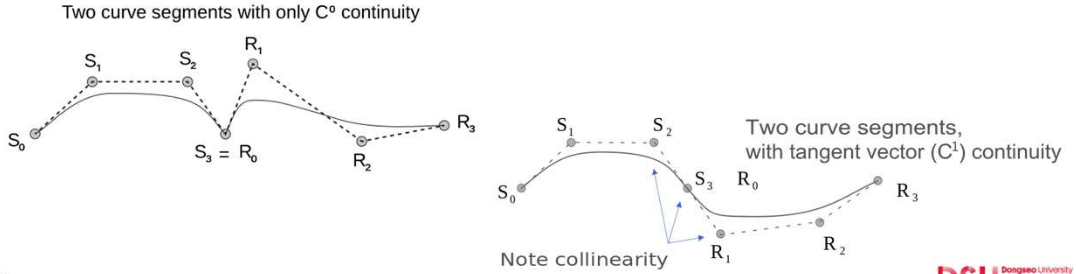
Parametric representation

x=Rcosθ, y=Rsinθ (0≤ θ < 2π)



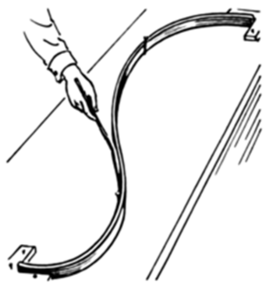
Order of continuity

- ✓ The various order of parametric continuity can be described as follows:[5]
- ✓ C^{-1}: curves include discontinuities
- ✓ C^0: curves are joined
- ✓ C^1: first derivatives are continuous
- ✓ C^2: first and second derivatives are continuous
- ✓ C^n: first through nth derivatives are continuous



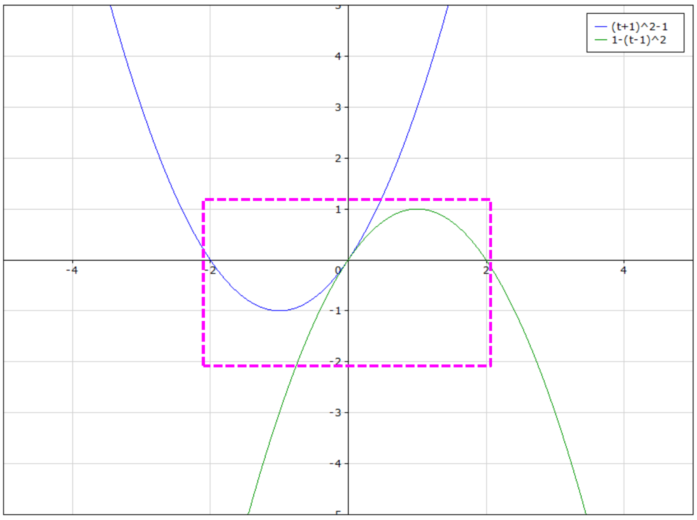
Spline

- ✓ A **spline** is a numeric function that is piecewise-defined by polynomial functions.
  - – A flexible device which can be bent to the desired shape is known as a flat spline.
- ✓ In interpolation problems, *spline interpolation* is often preferred to polynomial interpolation because it yields similar results to interpolating with higher degree polynomials while avoiding instability.
- ✓ The most commonly used splines are **cubic splines.**



Ex) Quadratic spline

$$S(t) = \begin{cases} (t+1)^2 - 1 & -2 \le t < 0 \\ 1 - (t-1)^2 & 0 \le t \le 2 \end{cases}$$



DSU Dongseo University 동서대학교

Ex) Cubic spline

$$S(t) = \begin{cases} t^3 & t \ge 0 \\ -t^3 & t < 0 \end{cases}$$

DSU Dongseo University 동서대학교

Definition of spline

- ✓ A spline is a piecewise-polynomial real function

  S:[a,b] → R
- ✓ on an interval [a,b] composed of k subintervals $[t_{i-1},t_i]$ with

  $a=t_0<t_1< \dots <t_{k-1}<t_k=b$.
- ✓ The restriction of *S* to an interval *i* is a polynomial

  $P_i:[t_{i-1},t_i] \rightarrow R$,
- ✓ so that

  $S(t)=P_1(t),\ t_0<= t <= t_1,$

  $S(t)=P_2(t),\ t_1<= t <= t_2,$

  …

  $S(t)=P_k(t),\ t_{k-1}<= t <= t_k,$
- ✓ The highest order of the polynomials $P_i(t)$ is said to be the **order of the spline** *S*.

Cubic spline

- ✓ A **cubic spline** is a spline where each piece is a third-degree polynomial.
- ✓ **A cubic Hermite spline** Specified by its values and first derivatives at the end points of the corresponding domain interval.

Unit interval (0,1)

- ✓ On the interval (0,1), given a starting point p_0 at t = 0 and an ending point p_1 at t = 1 with *starting tangent m_0 at t = 0* and *ending tangent m_1 at t = 1*, the polynomial can be defined by

$$\boldsymbol{p}(t) = (2t^3 - 3t^2 + 1)\boldsymbol{p}_0 + (t^3 - 2t^2 + t)\boldsymbol{m}_0 + (-2t^3 + 3t^2)\boldsymbol{p}_1 + (t^3 - t^2)\boldsymbol{m}_1$$

- ✓ where t ∈ [0,1].

- ✓ a point Q(t) on a curve at parameter t can be defined a cubic equations for each value:

$$Q(t)=\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}=\begin{bmatrix} a_x t^3 + b_x t^2 + c_x t + d_y \\ a_y t^3 + b_y t^2 + c_y t + d_y \end{bmatrix}$$

- ✓ We can define **coefficient matrix C** and **parameter matrix T**, then Q(t) can be defined like this:

$$T=\begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$C=\begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \end{bmatrix}$$

$$Q(t)=\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}=C \cdot T$$

✓ We will decompose coefficient matrix **C** with 2×4 **Geometry Matrix G** and 4×4 **Base Matrix M**.

C=G·M

$$Q(t)=\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}=\boldsymbol{C}\cdot T = G\cdot M\cdot T$$

$$=\begin{bmatrix} g_{1x} & g_{2x} & g_{3x} & g_{4x} \\ g_{1y} & g_{2y} & g_{3y} & g_{4y} \end{bmatrix}\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}\begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Geometry matrix G

✓ has an information about control data.

✓ in case of Hermite spline, it has two points and two tangent vectors for each point.

- P_1, P_2, R_1, R_2

Base matrix M

✓ a coefficient matrix for the blending functions.

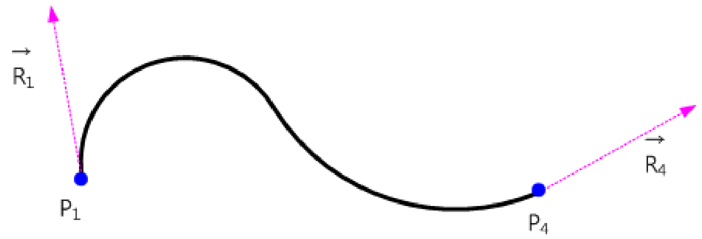
✓ blending function means **M·T**.

✓ the summation of all blending functions is 1.

- it means that spline curve calculates the weighted average for control points.

Hermite Spline

- ✓ Specified by its values and first derivatives at the end points of the corresponding domain interval



Geometry matrix G

$$G=\begin{bmatrix} P_{1x} & P_{4x} & R_{1x} & R_{4x} \\ P_{1y} & P_{4y} & R_{1y} & R_{4y} \end{bmatrix}$$

Find Base matrix M

- ✓ Four conditions to find Base matrix M.
- ✓ Begin point is (P_{1x}, P_{1y}), and End point is (P_{4x}, P_{4y}).

$$Q(0)=\begin{bmatrix} P_{1x} \\ P_{1y} \end{bmatrix}$$

$$Q(1)=\begin{bmatrix} P_{4x} \\ P_{4y} \end{bmatrix}$$

- ✓ Derivative at Begin point is (R_{1x}, R_{1y}), and derivative at End point is (R_{4x}, R_{4y}).

$$Q'(0)=\begin{bmatrix} R_{1x} \\ R_{1y} \end{bmatrix}$$

$$Q'(1)=\begin{bmatrix} R_{4x} \\ R_{4y} \end{bmatrix}$$

for two end points

✓ Q(t=0)= $\begin{bmatrix} P_{1x} \\ P_{1y} \end{bmatrix} = G \cdot M \cdot T$

$$= \begin{bmatrix} P_{1x} & P_{4x} & R_{1x} & R_{4x} \\ P_{1y} & P_{4y} & R_{1y} & R_{4y} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$= G \cdot M \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

✓ Q(t=1)= $\begin{bmatrix} P_{4x} \\ P_{4y} \end{bmatrix} = G \cdot M \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

15

for two tangent vector

✓ Q′(t=0)= $\begin{bmatrix} R_{1x} \\ R_{1y} \end{bmatrix} = G \cdot M \cdot T'$

$$= \begin{bmatrix} P_{1x} & P_{4x} & R_{1x} & R_{4x} \\ P_{1y} & P_{4y} & R_{1y} & R_{4y} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} 3t^2 \\ 2t \\ 1 \\ 0 \end{bmatrix}$$

$$= G \cdot M \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

✓ Q′(t=1)= $\begin{bmatrix} R_{4x} \\ R_{4y} \end{bmatrix} = G \cdot M \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix}$

16

Base matrix M

✓ $\begin{bmatrix} P_{1x} & P_{4x} & R_{1x} & R_{4x} \\ P_{1y} & P_{4y} & R_{1y} & R_{4y} \end{bmatrix} = G = G \cdot M \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} = G \cdot M \cdot M^{-1}$

$$\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} = M^{-1}$$

✓ $M = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$

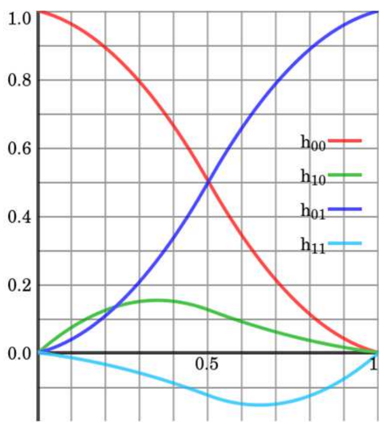
$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$\mathbf{A}^{-1} = \frac{\begin{bmatrix} ei-fh & hc-ib & bf-ce \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix}}{|\mathbf{A}|} = \frac{\begin{bmatrix} ei-fh & hc-ib & bf-ce \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix}}{aei+bfg+cdh-gec-hfa-idb}$$

Blending function

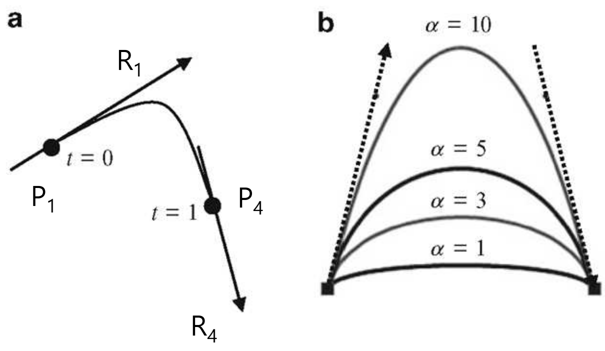
✓ $Q(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = G \cdot M \cdot T = G \cdot B$

$$\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t^1 \\ 0 \end{bmatrix}$$



$$= (2t^3 - 3t^2 + 1) \begin{bmatrix} P_{1x} \\ P_{1y} \end{bmatrix} + (-2t^3 + 3t^2) \begin{bmatrix} P_{4x} \\ P_{4y} \end{bmatrix} + (t^3 - 2t^2 + 1) \begin{bmatrix} R_{1x} \\ R_{1y} \end{bmatrix} + (t^3 - t^2) \begin{bmatrix} R_{4x} \\ R_{4y} \end{bmatrix}.$$

Meaning of tangent vector R



Implement Hermite Spline

```
class KHermiteCurve
{
public: // equation m_a*(t)^3 + m_b*(t)^2 + m_c*t + m_d*1.
    float   m_a;
    float   m_b;
    float   m_c;
    float   m_d;

    /// constructor.
    /// @param  p1: begin point
    /// @param  p2: end point
    /// @param  v1: tangent vector at begin point
    /// @param  v2: tangent vector at end point
    KHermiteCurve(float p1, float p4, float r1, float r4)
    {
        Construct(p1, p4, r1, r4);
    }
```

```
/// set coefficient of Hermite curve
inline void Construct(float p1, float p4, float r1, float r2)
{
    m_a =  2.0f*p1 - 2.0f*p4 + r1 + r4;
    m_b = -3.0f*p1 + 3.0f*p4 - 2.0f*r1 – r4;
    m_c = r1;
    m_d = p1;
}
```

$$\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t^1 \\ 0 \end{bmatrix}$$

$$(2t^3-3t^2+1)\begin{bmatrix} P_{1x} \\ P_{1y} \end{bmatrix} + (-2t^3+3t^2)\begin{bmatrix} P_{4x} \\ P_{4y} \end{bmatrix} + (t^3-2t^2+1)\begin{bmatrix} R_{1x} \\ R_{1y} \end{bmatrix} + (t^3-t^2)\begin{bmatrix} R_{4x} \\ R_{4y} \end{bmatrix}.$$


```
/// calculate first derivative on parameter u.
/// can be used to get tangent vector at u.
/// derivative of equation t^3 + t^2 + t + 1
inline float CalculateDxDu(float u) const
{
    return 3.0f*m_a*u*u + 2.0f*m_b*u + m_c;
}

/// get value at parameter u.
inline float CalculateX(float u) const
{
    float uu, uuu;
    uu = u * u;
    uuu = uu * u;
    return m_a*uuu + m_b*uu + m_c*u + m_d;
}
```

KHermiteSpline2

```
class KHermiteSpline2
{
private:
    KHermiteCurve    m_aHermiteCurve[2];

public:
    /// constructor.
    KHermiteSpline2(){}
    KHermiteSpline2( const KVector& p0, const KVector& p1
                    , const KVector& dp0, const KVector& dp1 )
    {
        Construct( p0, p1, dp0, dp1 );
    }
```



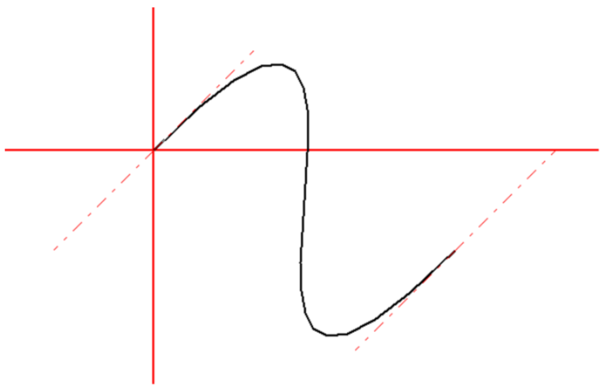
```
    void Construct( const KVector& p0, const KVector& p1
                  , const KVector& dp0, const KVector& dp1 )
    {
        m_aHermiteCurve[ 0 ].Construct( p0.x, p1.x, dp0.x, dp1.x );
        m_aHermiteCurve[ 1 ].Construct( p0.y, p1.y, dp0.y, dp1.y );
    }
```

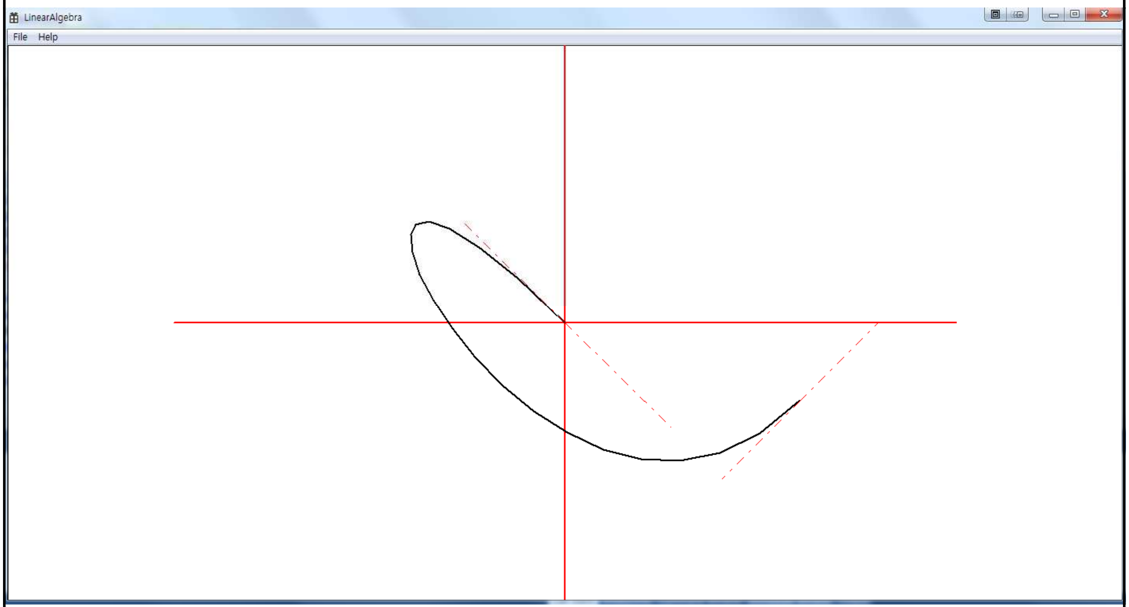
```
const KVector GetPosition( float u_ ) const
{
    return KVector( m_aHermiteCurve[ 0 ].CalculateX( u_ )
                    , m_aHermiteCurve[ 1 ].CalculateX( u_ ), 0.f );
}

const KVector GetTangent( float u_ ) const
{
    return KVector( m_aHermiteCurve[ 0 ].CalculateDxDu( u_ )
                    , m_aHermiteCurve[ 1 ].CalculateDxDu( u_ ), 0.f );
}
```

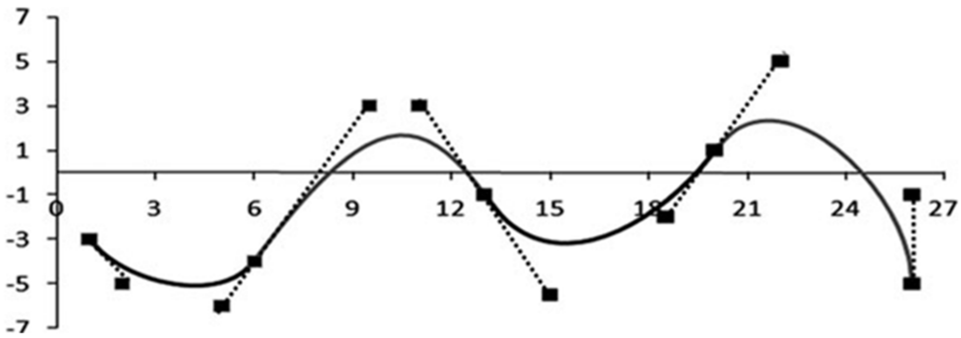

Draw a tangent line at u

```
const float u = 0.5f;
KVector position = spline.GetPosition(u);
KVector tangent = spline.GetTangent(u);
DrawLine(hdc, position, position + tangent * 100.f);
```





Path: Combine Splines



Line in 3D space

- ✓ Given two 3D points $\mathbf{P}_1$ and $\mathbf{P}_2$, we can define the line that passes through these points parametrically as

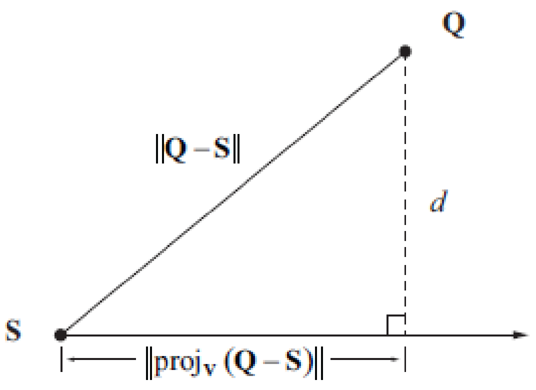
$$\mathbf{P}(t) = (1 - t)\mathbf{P}_1 + t\mathbf{P}_2$$

- ✓ A *ray* is a line having a single endpoint **S** and extending to infinity in a given direction **V**. Rays are typically expressed by the parametric equation

$$\mathbf{P}(t) = \mathbf{S} + t\mathbf{V}$$

- ✓ Note that this equation is equivalent to previous equation if we let $\mathbf{S} = \mathbf{P}_1$ and $\mathbf{V} = \mathbf{P}_2 - \mathbf{P}_1$.

Distance between a point and a line



$$d^2 = (\mathbf{Q}-\mathbf{S})^2 - [\mathrm{proj}_{\mathbf{V}}(\mathbf{Q}-\mathbf{S})]^2$$
$$= (\mathbf{Q}-\mathbf{S})^2 - \left[\frac{(\mathbf{Q}-\mathbf{S})\cdot\mathbf{V}}{V^2}\mathbf{V}\right]^2.$$

$$d = \sqrt{(\mathbf{Q}-\mathbf{S})^2 - \frac{[(\mathbf{Q}-\mathbf{S})\cdot\mathbf{V}]^2}{V^2}}.$$

Spline path tracing demo

MY **BRIGHT** FUTURE

동서대학교

DSU Dongseo University 동서대학교