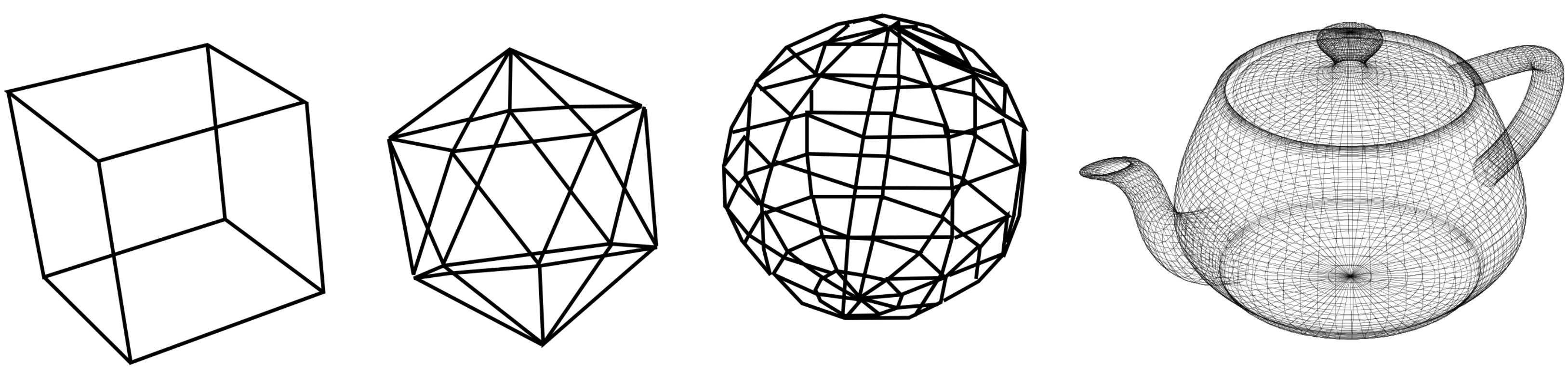
VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY
UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

COMPUTER GRAPHICS

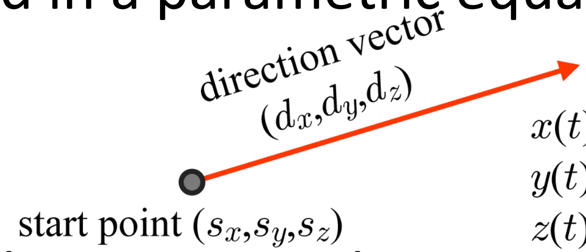
Lecture 2: Parametric Curve and Surface

Lecturer: Dr. NGUYEN Hoang Ha



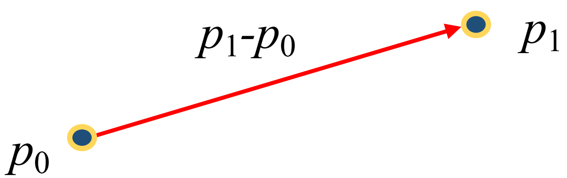
Reference: JungHyun Han. 2011. 3D Graphics for Game Programming (1st ed.), chapter 6

- A ray defined by the start point and the direction vector is represented in a parametric equation.



$$x(t) = s_x + t d_x$$
$$y(t) = s_y + t d_y$$
$$z(t) = s_z + t d_z$$

- Consider a line segment between two end points, p_0 and p_1. Vector p_1-p_0 corresponds to the direction vector → line segment can be represented as:

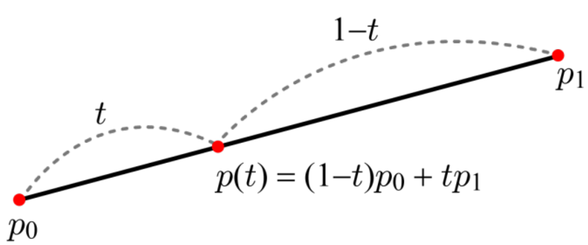


$$p(t) = p_0 + t(p_1 - p_0)$$
$$= (1-t)p_0 + tp_1$$

$$p(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} (1-t)x_0 + tx_1 \\ (1-t)y_0 + ty_1 \\ (1-t)z_0 + tz_1 \end{pmatrix}$$

[illegible] ([illegible])

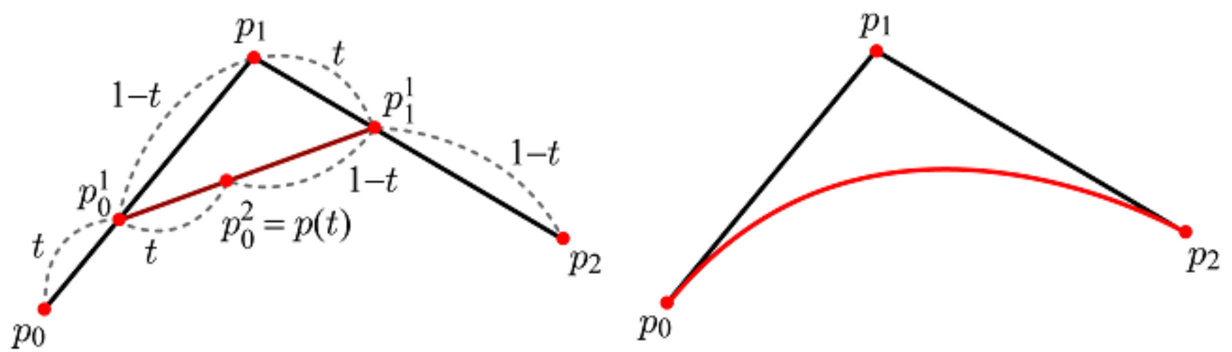
- A line segment connecting two end points is represented as a *linear interpolation* of the points.
- The line segment may be considered as being divided into two parts by $p(t)$, and the weight for an end point is proportional to the length of the part "on the opposite side," i.e., the weights for p_0 and p_1 are $(1\text{-}t)$ and t, respectively.



$$p(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} (1-t)x_0 + tx_1 \\ (1-t)y_0 + ty_1 \\ (1-t)z_0 + tz_1 \end{pmatrix}$$

□□□□□□□□□B□□□□□C□□□□□

- De Casteljau algorithm = recursive linear interpolations for defining a curve. The quadratic Bézier curve interpolates the end points, p_0 and p_2, and is pulled toward p_1, but does not interpolate it.



$$p_0 \xrightarrow{1-t} p_0^1, \quad p_1 \xrightarrow{t} p_0^1 = (1-t)p_0 + tp_1$$

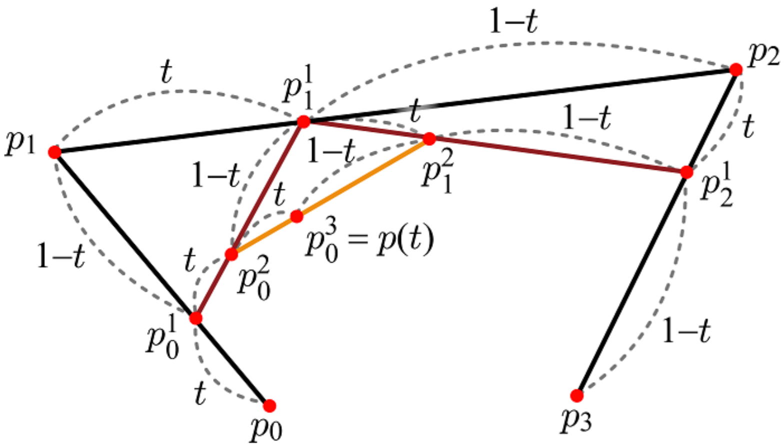
$$p_1 \xrightarrow{1-t} p_1^1, \quad p_2 \xrightarrow{t} p_1^1 = (1-t)p_1 + tp_2$$

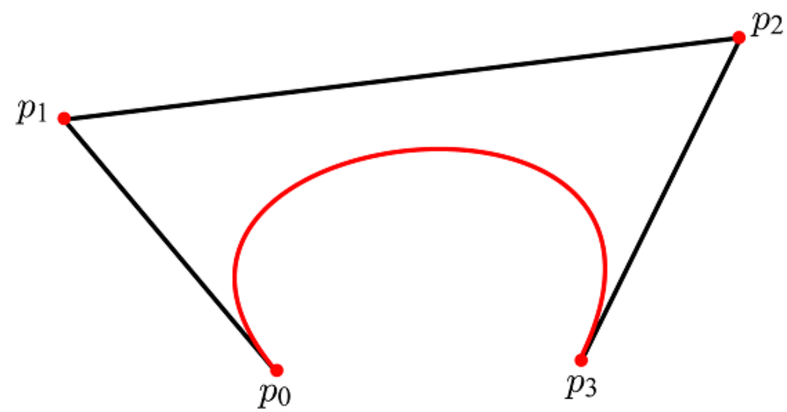
$$p_0^1 \xrightarrow{1-t} p_0^2, \quad p_1^1 \xrightarrow{t} p_0^2 = (1-t)p_0^1 + tp_1^1$$

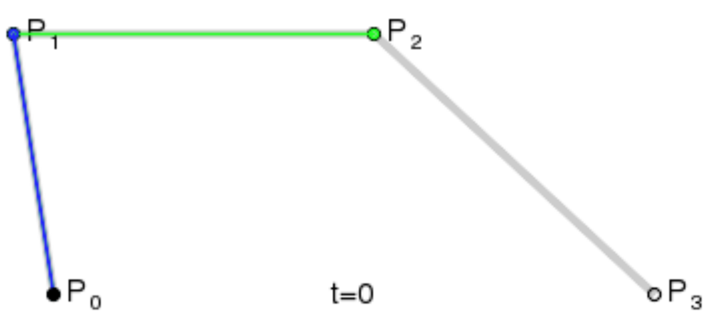
$$= (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$$

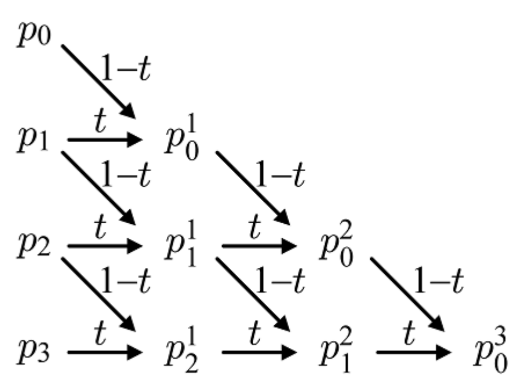
C□□□□B□□□□□C□□□□□

- The cubic Bézier curve interpolates the end points, p_0 and p_3, and is pulled toward p_1 and p_2.





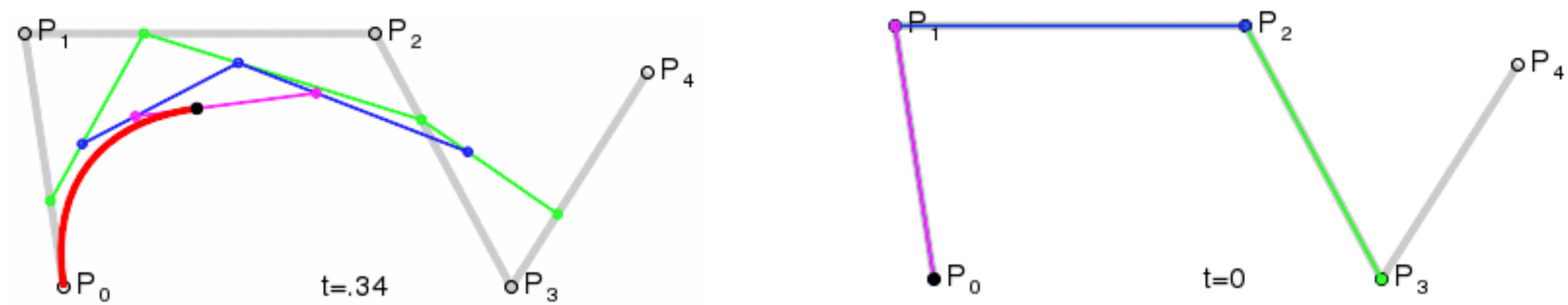




$$p(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t) p_2 + t^3 p_3$$

□□□□□□B□□□□□C□□□□

- The de Casteljau algorithm can be applied for a higher-degree Bézier curve. For example, a quartic (degree-4) Bézier curve can be constructed using five points.



[from Wikipedia]

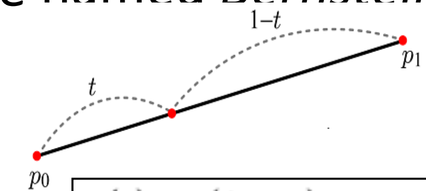
- Such higher-degree curves often reveal undesired wiggles. Worse still, they do not bring significant advantages. In contrast, quadratic curves have little flexibility. Therefore, cubic Bézier curves are most popularly used in the graphics field.

B□□□□C□□□□C□□□□□□□□□□□□□□□□B□□□□□□□□□□□□□□□

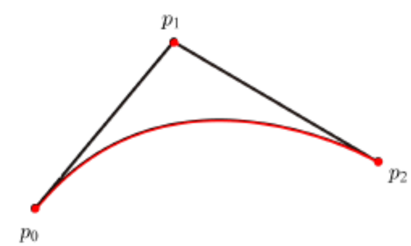
- The points p_is are called *control points.* A degree-*n* Bézier curve requires (*n*+1) control points. The coefficients associated with the control point are polynomials of parameter *t*, and are named *Bernstein polynomials*.

$$B_i^n(t) = {}_nC_i t^i (1-t)^{n-i}$$

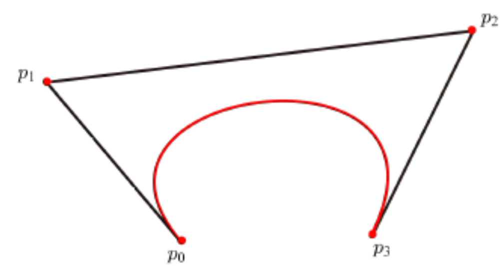
$$p(t) = \sum_{i=0}^{n} B_i^n(t) p_i$$



$$p(t) = (1-t)p_0 + tp_1$$



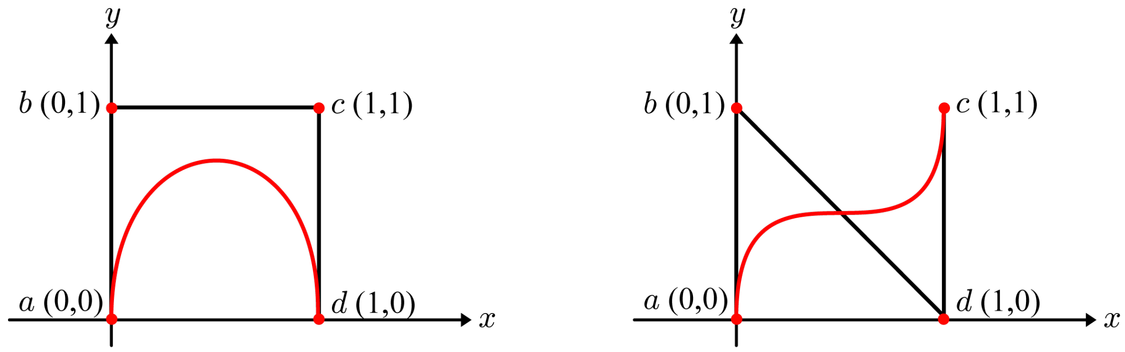
$$p(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$$



$$p(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3$$

B[illegible]C[illegible]C[illegible]

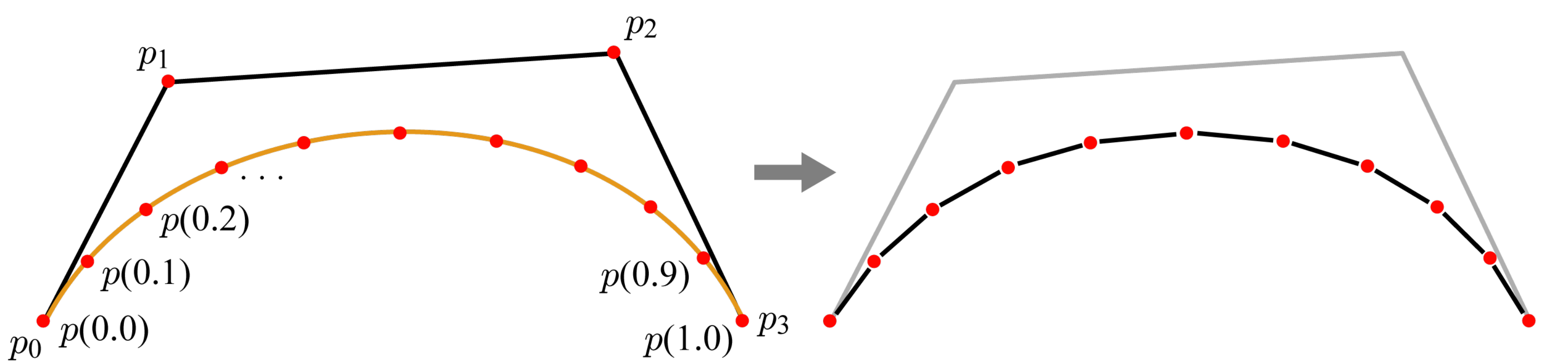
- Different orders of the control points produce different curves.



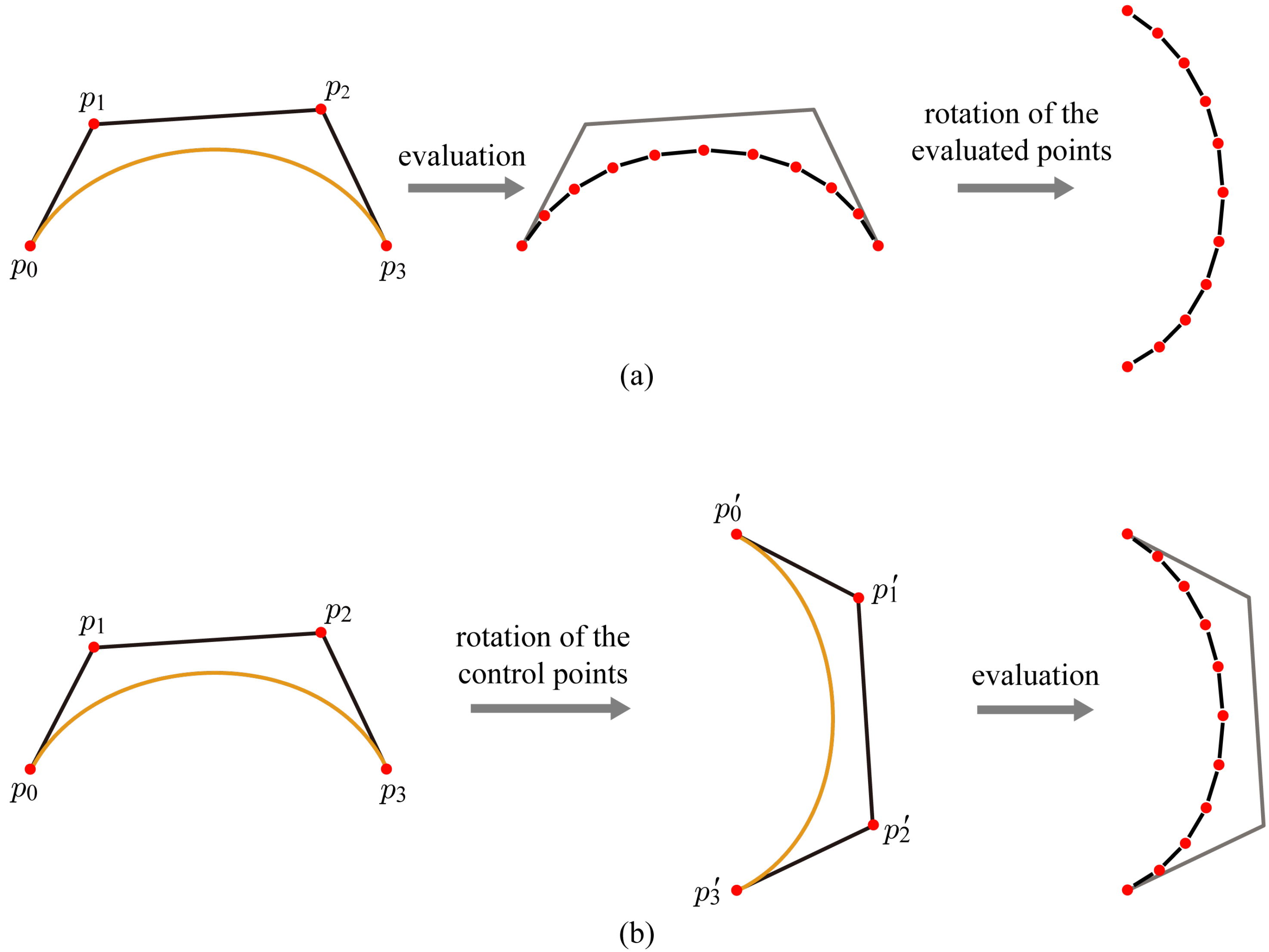
(a) $a \rightarrow b \rightarrow c \rightarrow d$ (b) $a \rightarrow b \rightarrow d \rightarrow c$

B□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□

- The typical method to display a Bézier curve is to approximate it using a series of line segments. This process is often called *tessellation*. It evaluates the curve at a fixed set of parameter values, and joins the evaluated points with straight lines.



B[illegible] C[illegible] A[illegible] [illegible]



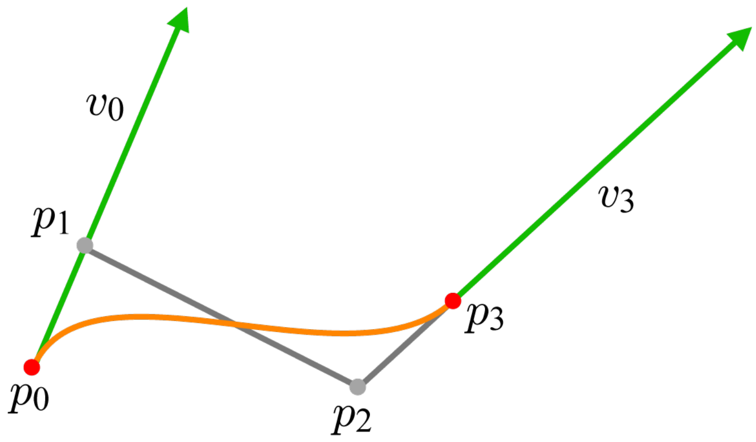
p_1
p_2
p_0
p_3
evaluation
rotation of the evaluated points
(a)
p_1
p_2
p_0
p_3
rotation of the control points
p_0'
p_1'
p_2'
p_3'
evaluation
(b)

H[illegible]C[illegible]

$$\begin{aligned}\dot{p}(t) &= \tfrac{d}{dt}[(1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3] \\ &= -3(1-t)^2 p_0 + [3(1-t)^2 - 6t(1-t)]p_1 + [6t(1-t) - 3t^2]p_2 + 3t^2 p_3\end{aligned}$$

$$v_0 = \dot{p}(0) = 3(p_1 - p_0) \Rightarrow p_1 = p_0 + \frac{1}{3}v_0$$

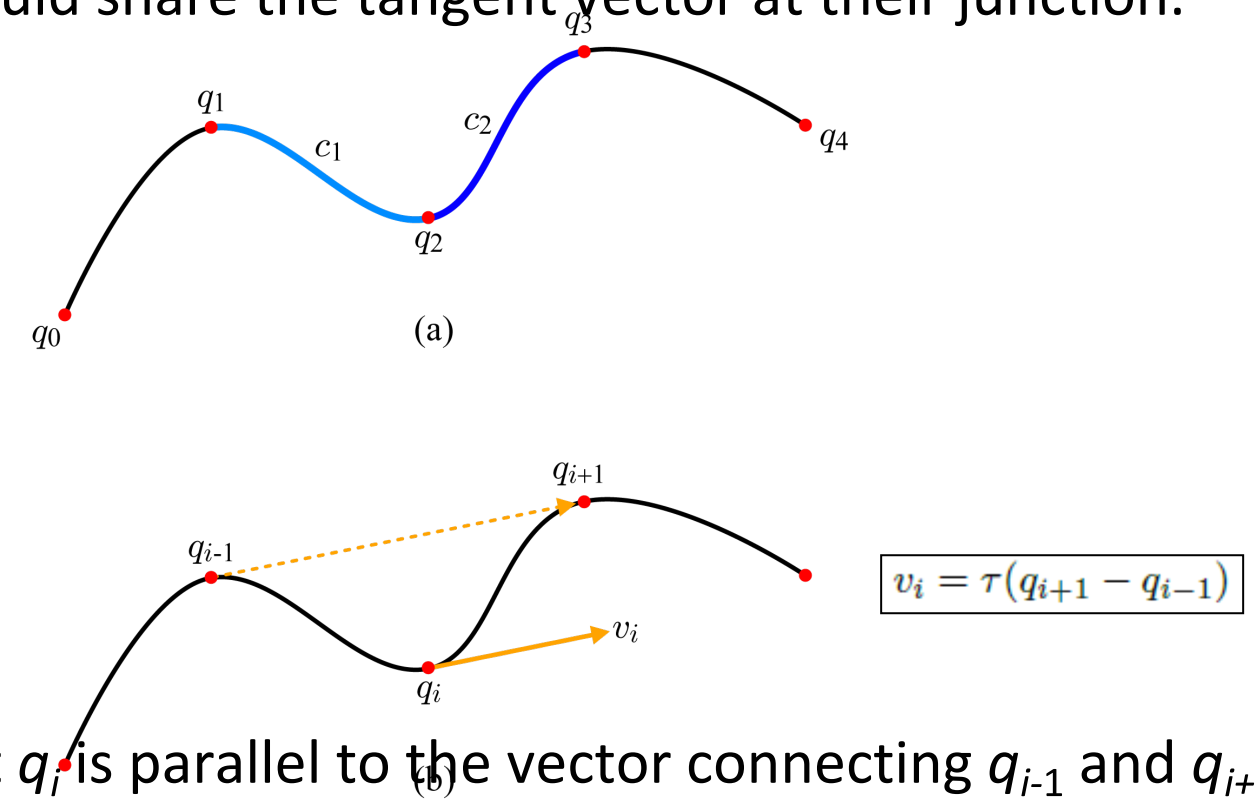
$$v_3 = \dot{p}(1) = 3(p_3 - p_2) \Rightarrow p_2 = p_3 - \frac{1}{3}v_3$$



$$\begin{aligned}p(t) &= (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3 \\ &= (1-t)^3 p_0 + 3t(1-t)^2(p_0 + \tfrac{1}{3}v_0) + 3t^2(1-t)(p_3 - \tfrac{1}{3}v_3) + t^3 p_3 \\ &= (1 - 3t^2 + 2t^3)p_0 + t(1-t)^2 v_0 + (3t^2 - 2t^3)p_3 - t^2(1-t)v_3\end{aligned}$$

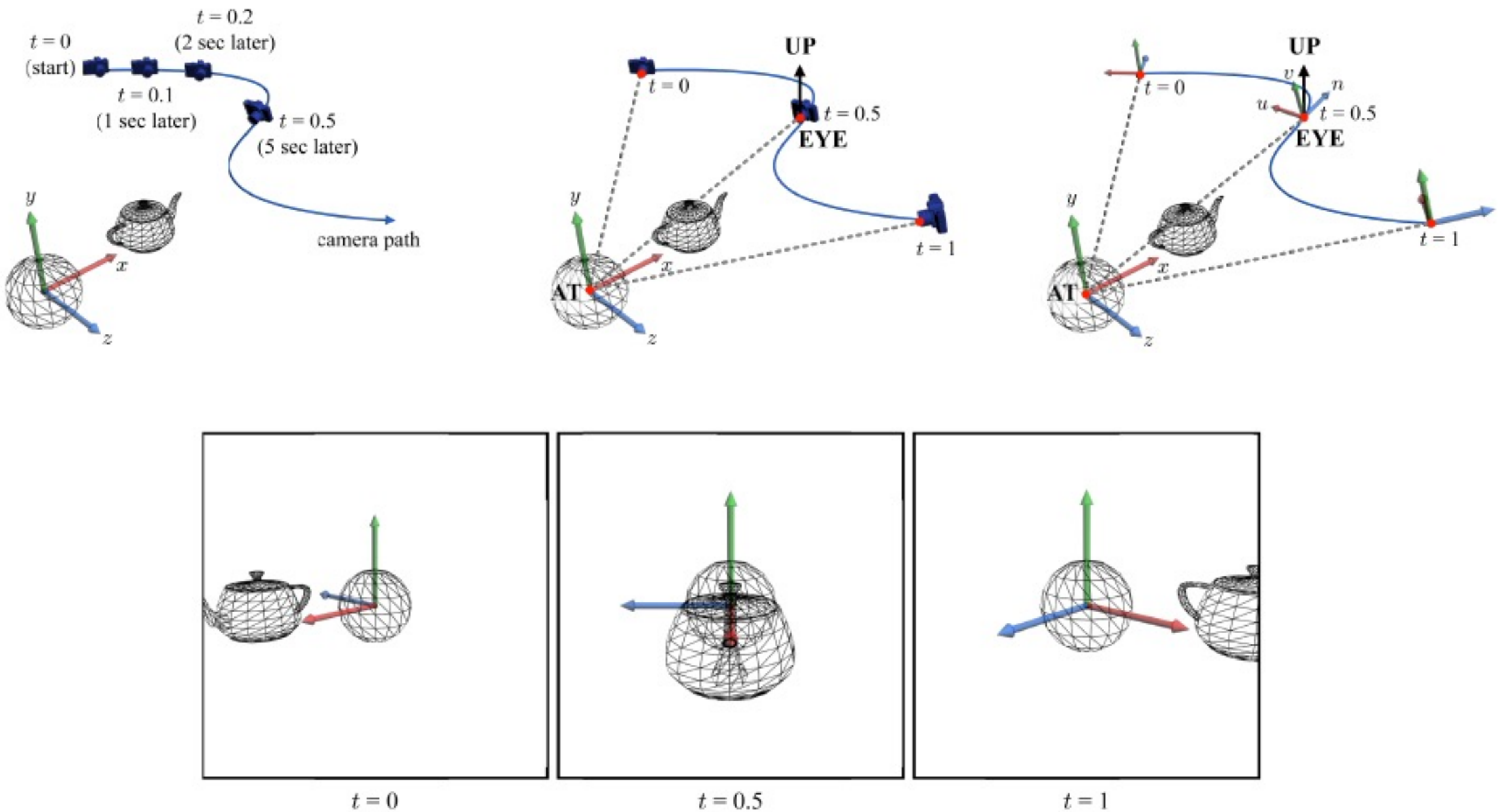
C[illegible]-[illegible] [illegible]

- A spline (piecewise curve) composed of cubic Hermite curves passes through the given points q_is. Two adjacent Hermite curves should share the tangent vector at their junction.



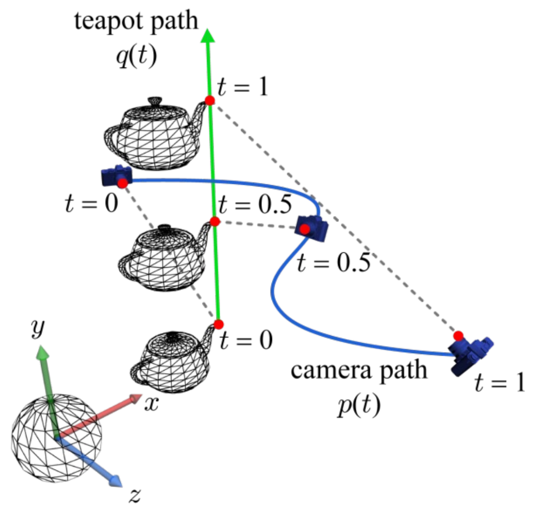
- Tangent at q_i is parallel to the vector connecting $q_{i\text{-}1}$ and q_{i+1}.

A

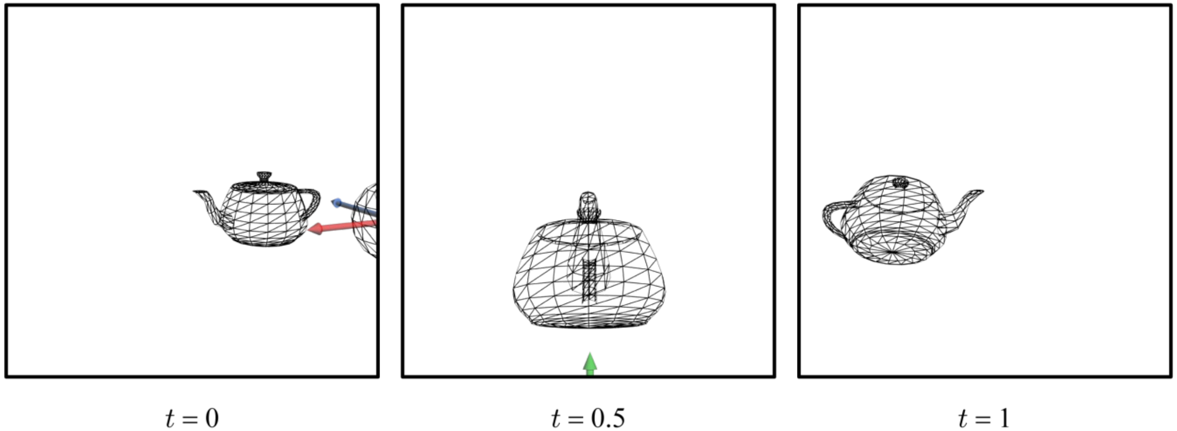


t = 0
(start)
t = 0.1
(1 sec later)
t = 0.2
(2 sec later)
t = 0.5
(5 sec later)
camera path
y
x
z
UP
t = 0
t = 0.5
EYE
AT
t = 1
v
n
u
t = 0
t = 0.5
t = 1

A ( )



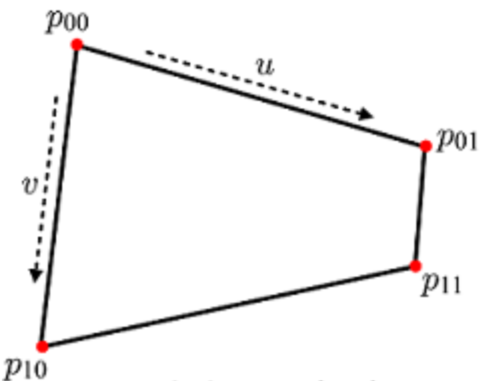
(a)



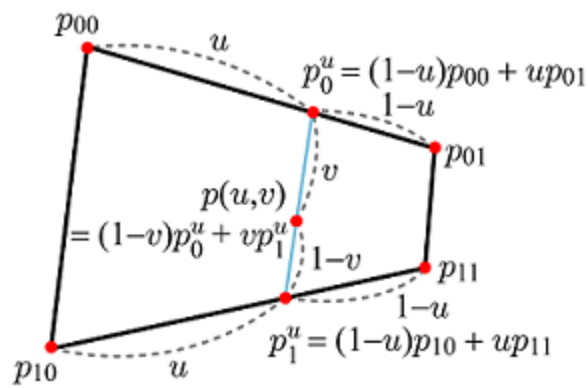
(b)

B□□□□□□□□□□□□□

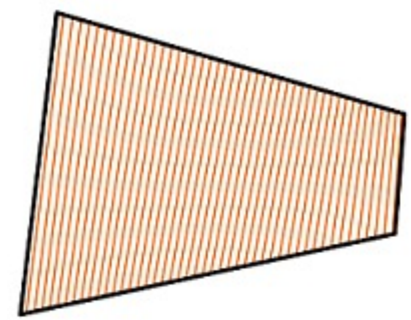
- Combination of linear interpolations using four control points



2×2 control points



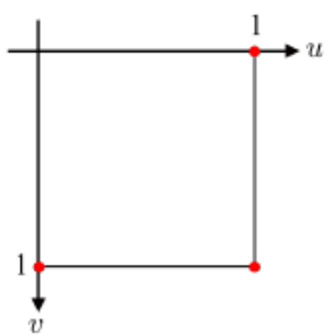
Bilinear interpolation
(u first, and then v)



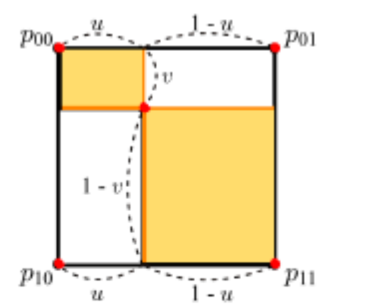
A collection of line segments

- Matrix representation where the center matrix corresponds to the control point net

$$
\begin{aligned}
p(u,v) &= \begin{pmatrix} 1-v & v \end{pmatrix} \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \begin{pmatrix} 1-u \\ u \end{pmatrix} \\
&= \begin{pmatrix} 1-v & v \end{pmatrix} \begin{pmatrix} (1-u)p_{00} + up_{01} \\ (1-u)p_{10} + up_{11} \end{pmatrix} \\
&= (1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11}
\end{aligned}
$$



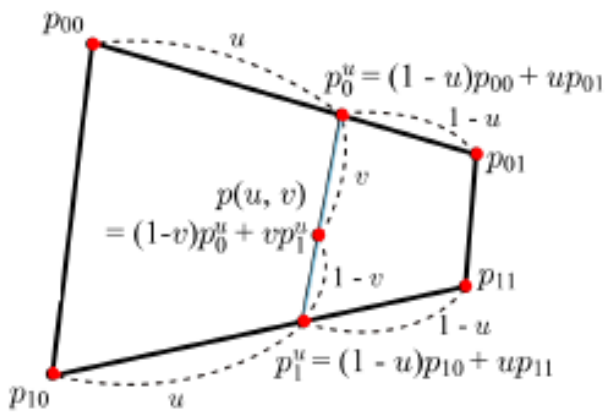
parametric domain

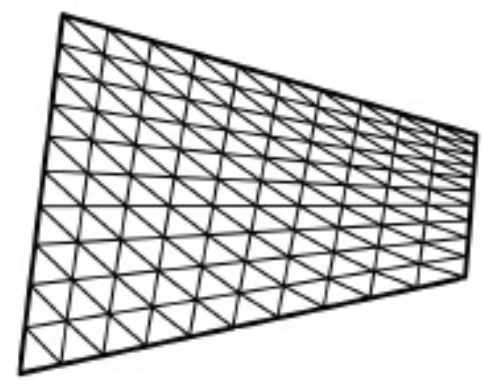


weights for the control points

B[illegible]([illegible])

- Tessellation with nested for loops



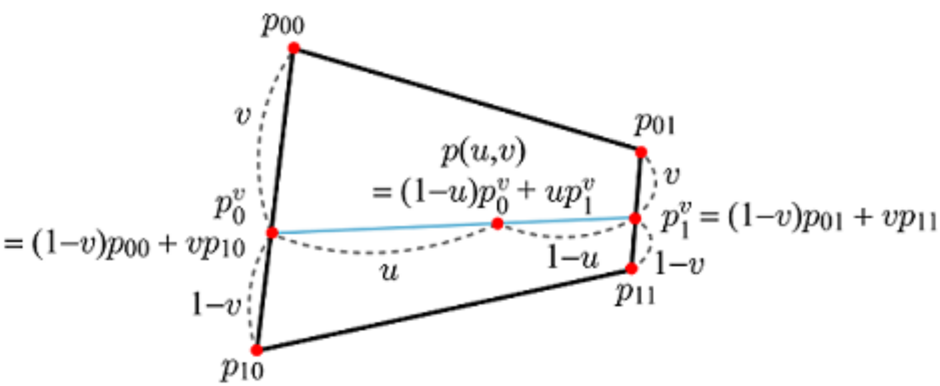


$$p(u,v) = (1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11}$$

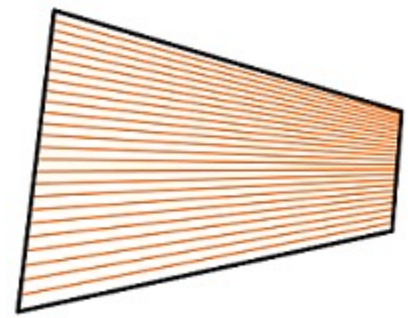
```
foreach u in the range [0,1]
  foreach v in the range [0,1]
    Evaluate the patch using (u,v) to obtain (x,y,z)
  endforeach
endforeach
```

B[illegible] ([illegible])

- Let's reverse the order of *u* and *v* in linear interpolations.



Bilinear interpolation
(v first, and then u)

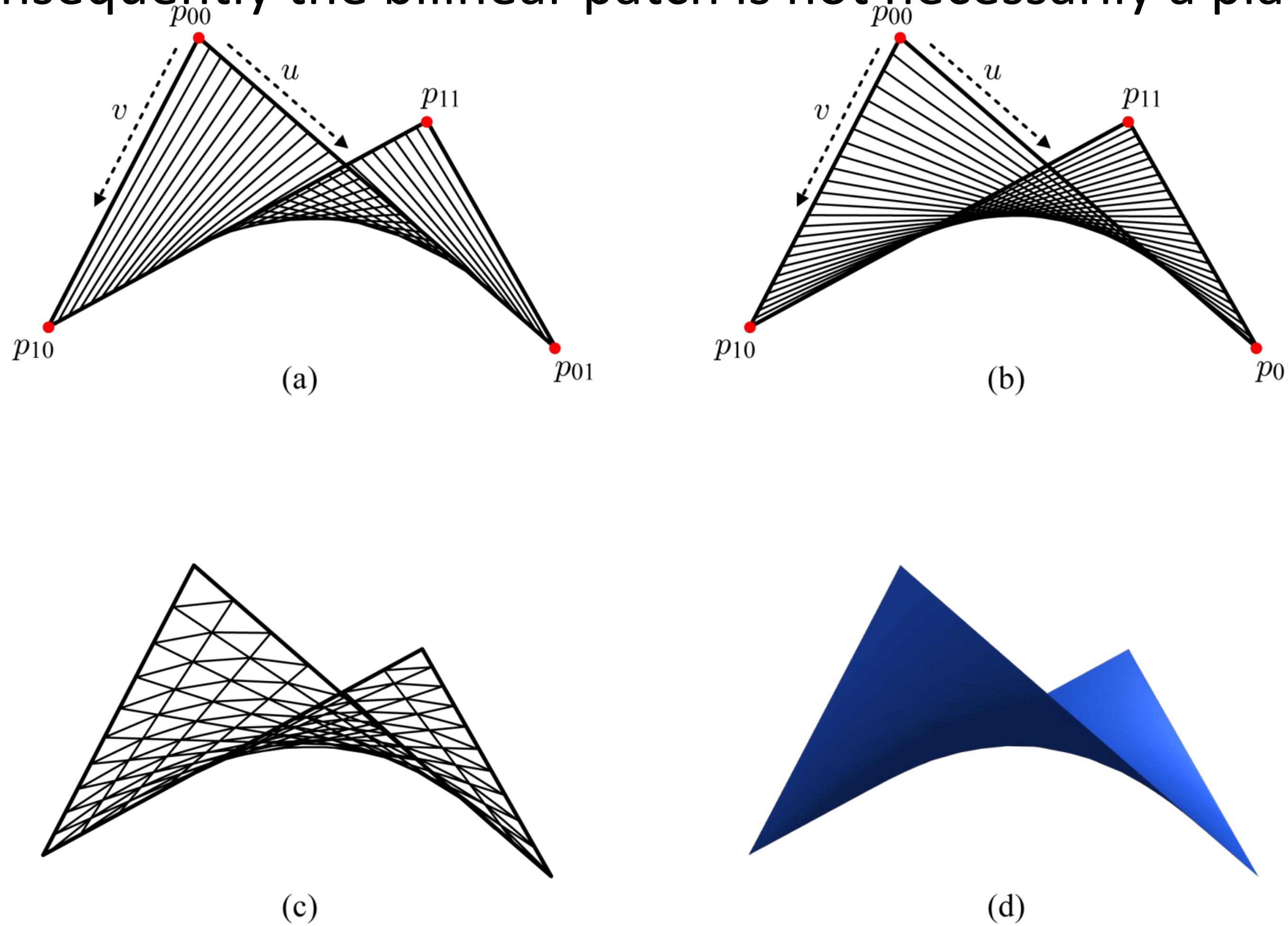


Another collection of
line segments

$$
\begin{aligned}
p(u,v) &= \begin{pmatrix} 1-v & v \end{pmatrix} \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \begin{pmatrix} 1-u \\ u \end{pmatrix} \\
&= \begin{pmatrix} (1-v)p_{00} + vp_{10} & (1-v)p_{01} + vp_{11} \end{pmatrix} \begin{pmatrix} 1-u \\ u \end{pmatrix} \\
&= (1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11}
\end{aligned}
$$

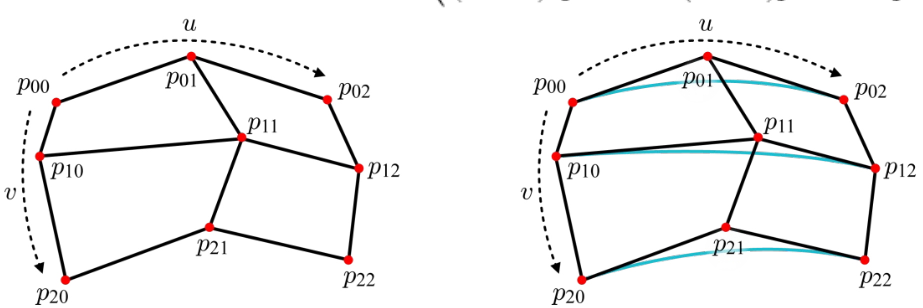
B[illegible] ([illegible])

- The control points are not necessarily in a plane, and consequently the bilinear patch is not necessarily a plane.

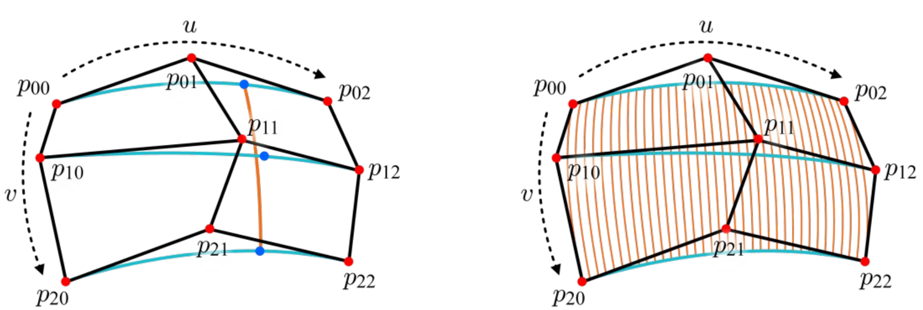


$$p(u,v) = \begin{pmatrix}(1-v)^2 & 2v(1-v) & v^2\end{pmatrix} \begin{pmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{pmatrix} \begin{pmatrix} (1-u)^2 \\ 2u(1-u) \\ u^2 \end{pmatrix}$$

$$= \begin{pmatrix}(1-v)^2 & 2v(1-v) & v^2\end{pmatrix} \begin{pmatrix} (1-u)^2 p_{00} + 2u(1-u)p_{01} + u^2 p_{02} \\ (1-u)^2 p_{10} + 2u(1-u)p_{11} + u^2 p_{12} \\ (1-u)^2 p_{20} + 2u(1-u)p_{21} + u^2 p_{22} \end{pmatrix}$$



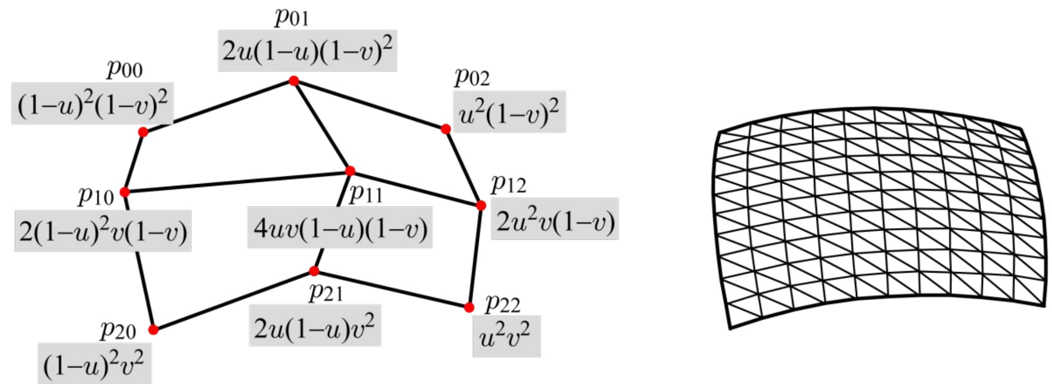
(a) 3×3 control points (b) Three quadratic Bézier curves (in u)



(c) A quadratic Bézier curve (in v) (d) A biquadratic Bézier patch as a collection of Bézier curves (each defined in u)

B[illegible] B[illegible] ([illegible])

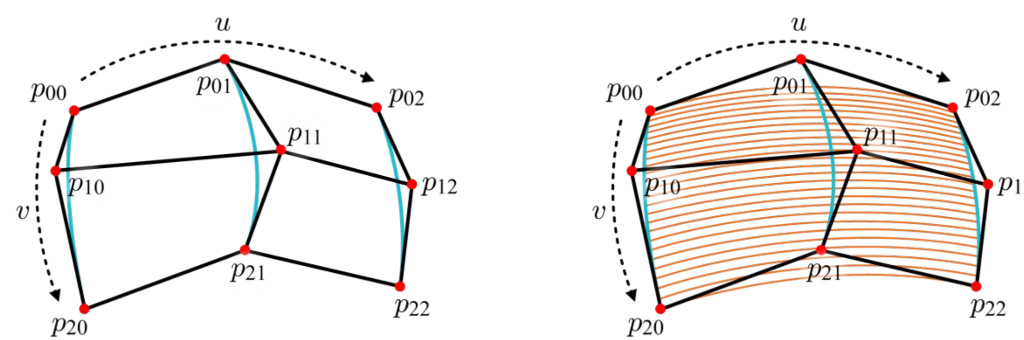
$$p(u,v) = (1-u)^2(1-v)^2 p_{00} + 2u(1-u)(1-v)^2 p_{01} + u^2(1-v)^2 p_{02} +$$
$$2(1-u)^2 v(1-v) p_{10} + 4uv(1-u)(1-v) p_{11} + 2u^2 v(1-v) p_{12} +$$
$$(1-u)^2 v^2 p_{20} + 2u(1-u)v^2 p_{21} + u^2 v^2 p_{22}$$



(e) Weights for the control points

(f) Tessellation

$$\begin{pmatrix} (1-v)^2 & 2v(1-v) & v^2 \end{pmatrix} \begin{pmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{pmatrix} \begin{pmatrix} (1-u)^2 \\ 2u(1-u) \\ u^2 \end{pmatrix}$$

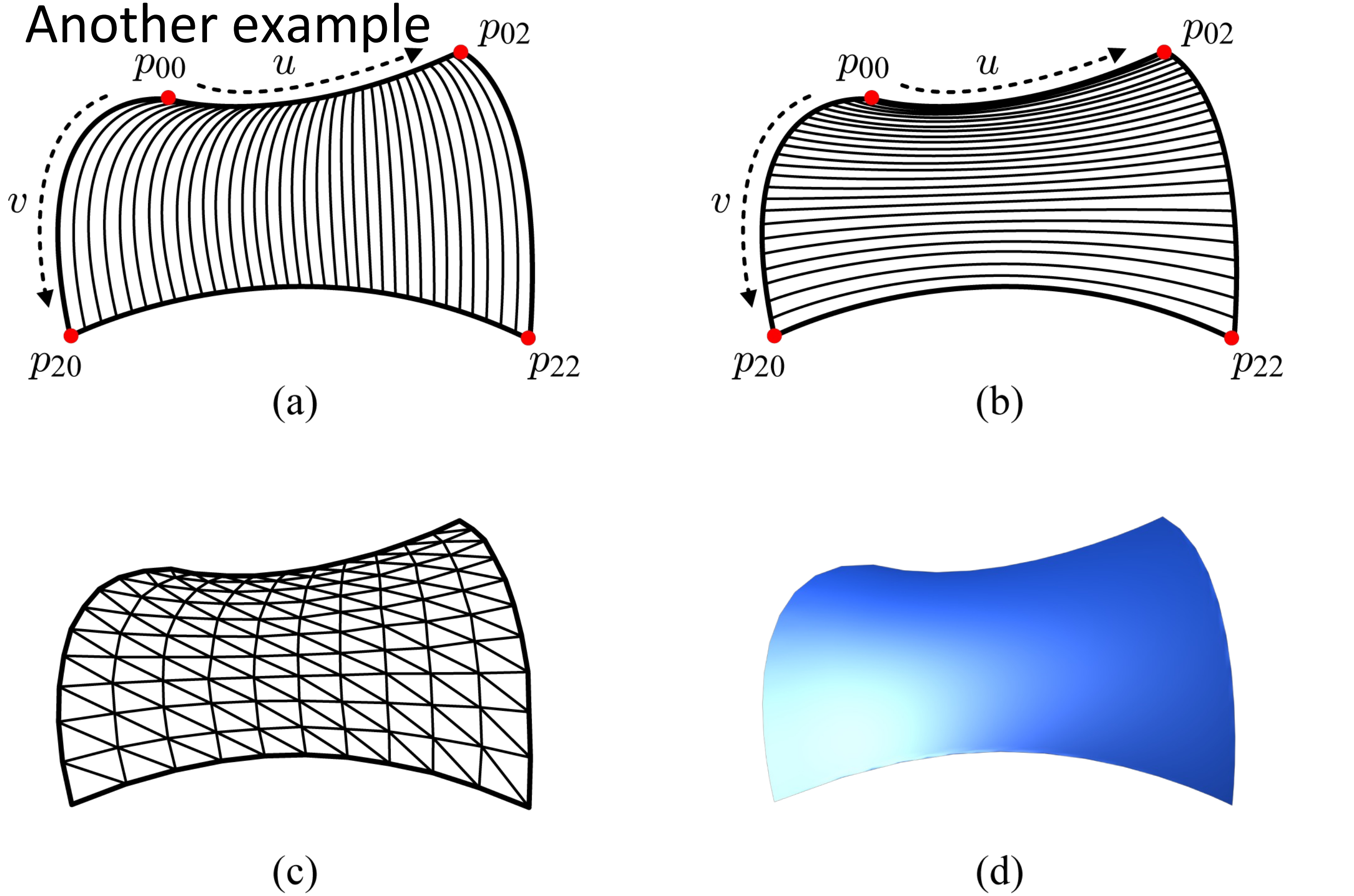


(g) Three quadratic Bézier curves (in v)

(h) A biquadratic Bézier patch as a collection of Bézier curves (each defined in u)

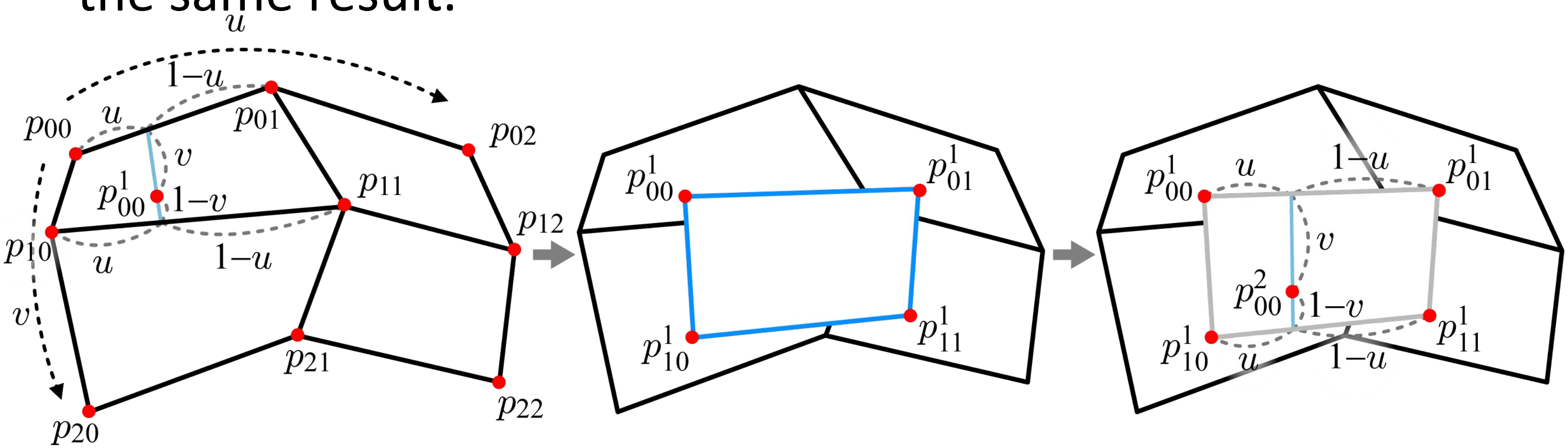
B□□□□□□□□□B□□□□□□□□□□□(□□□□□)□

- Another example



B[illegible] B[illegible] ([illegible])

- So far, we have seen *two-stage explicit evaluation*.
- Now, let's see *repeated bilinear interpolations*, which produce the same result.



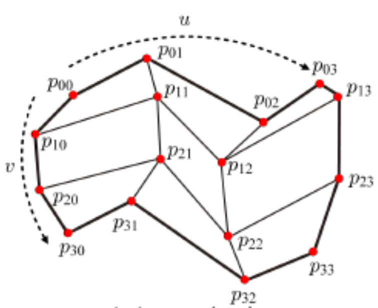
$$p^1_{00} = (1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11}$$
$$p^1_{01} = (1-u)(1-v)p_{01} + u(1-v)p_{02} + (1-u)vp_{11} + uvp_{12}$$
$$p^1_{10} = (1-u)(1-v)p_{10} + u(1-v)p_{11} + (1-u)vp_{20} + uvp_{21}$$
$$p^1_{11} = (1-u)(1-v)p_{11} + u(1-v)p_{12} + (1-u)vp_{21} + uvp_{22}$$

$$p^2_{00} = (1-u)(1-v)p^1_{00} + u(1-v)p^1_{01} + (1-u)vp^1_{10} + uvp^1_{11}$$

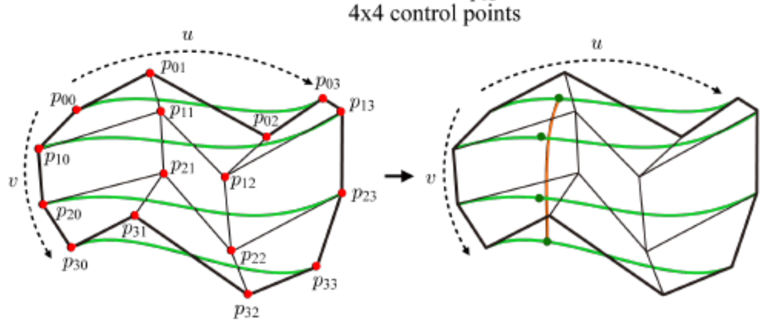
B[illegible] B[illegible]

- A simple extension of biquadratic Bézier patch

$$p(u,v) = \left((1-v)^3 \; 3v(1-v)^2 \; 3v^2(1-v) \; v^3\right) \begin{pmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{pmatrix} \begin{pmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 3u^2(1-u) \\ u^3 \end{pmatrix}$$



4x4 control points

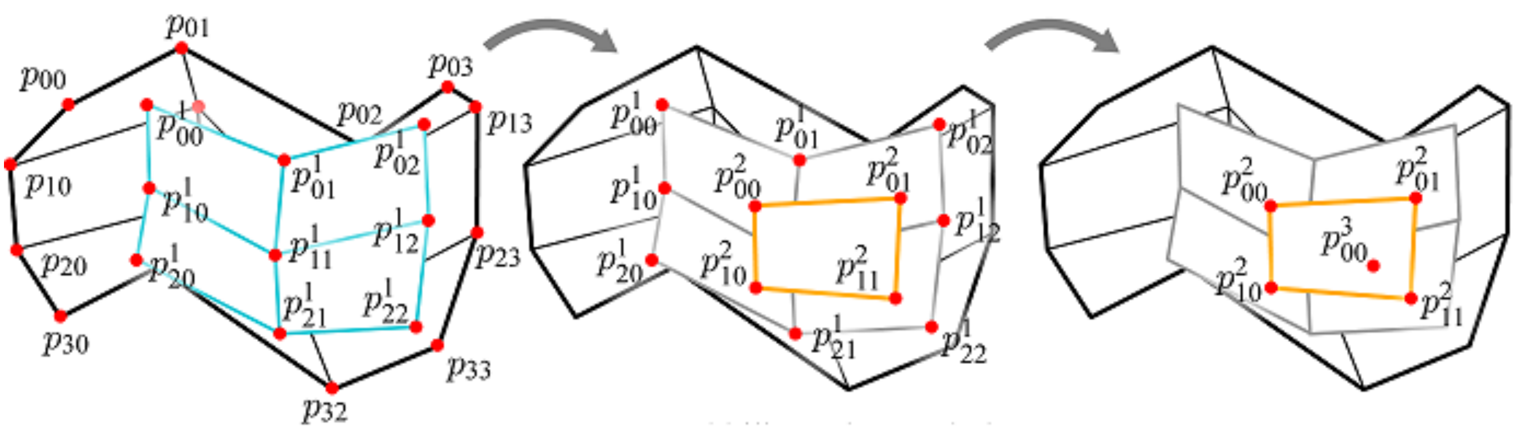


two-stage explicit evaluation

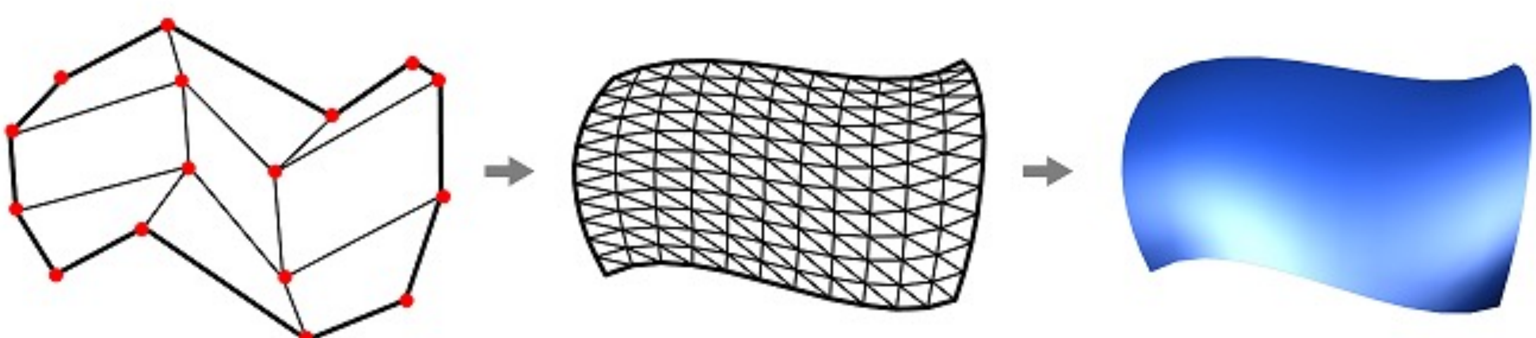
- We can of course reverse the order of *u* and *v*.

B[illegible] B[illegible] ([illegible])

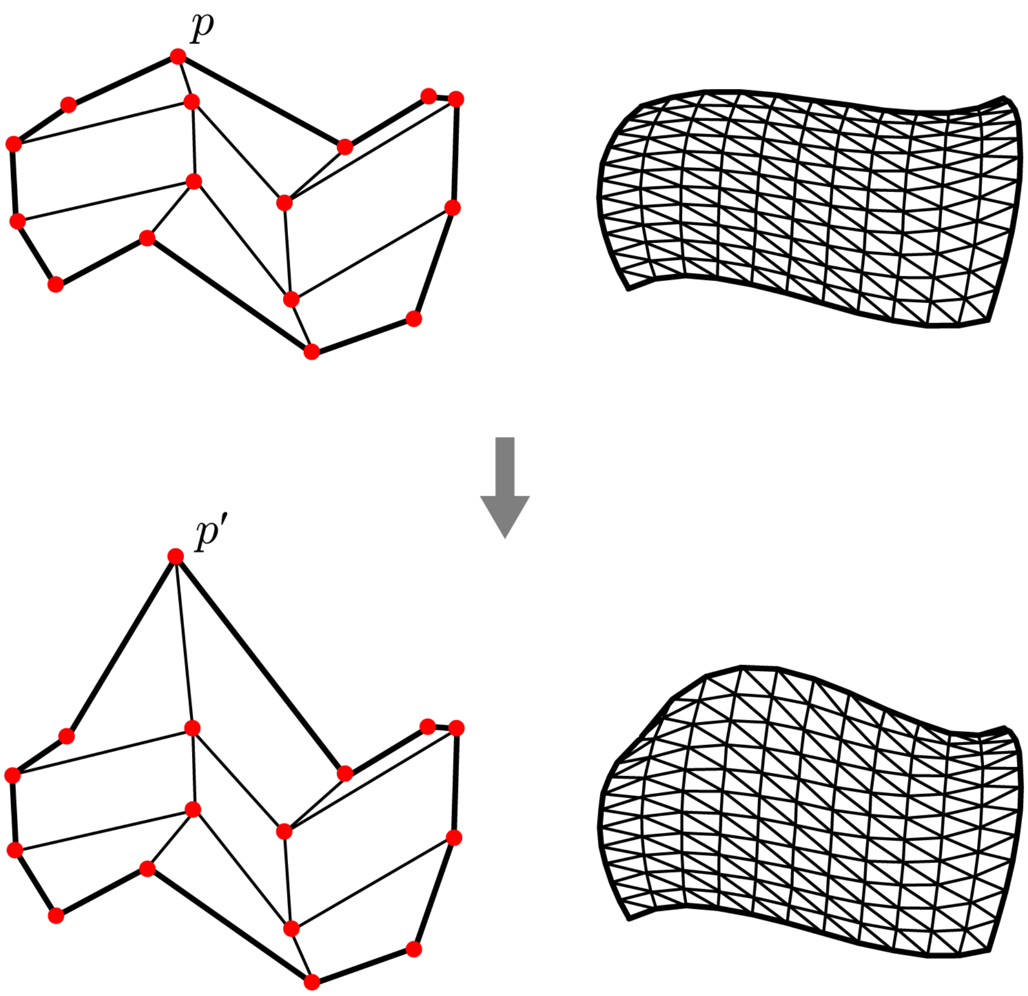
- Let's apply *repeated bilinear interpolations*.



- Tessellation and rendering result



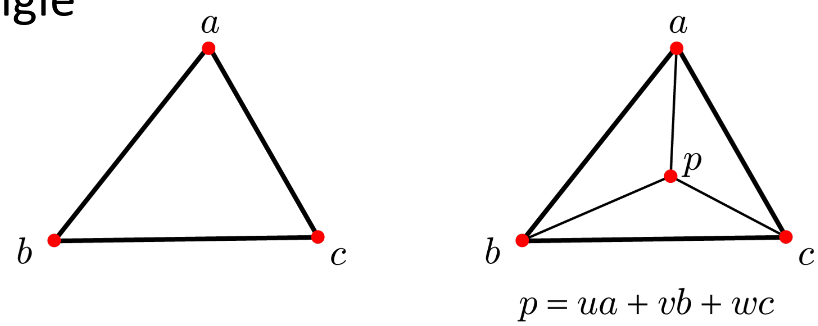
[illegible]



p
p'

B

- Degree-1 Bézier triangle



(a) (b)

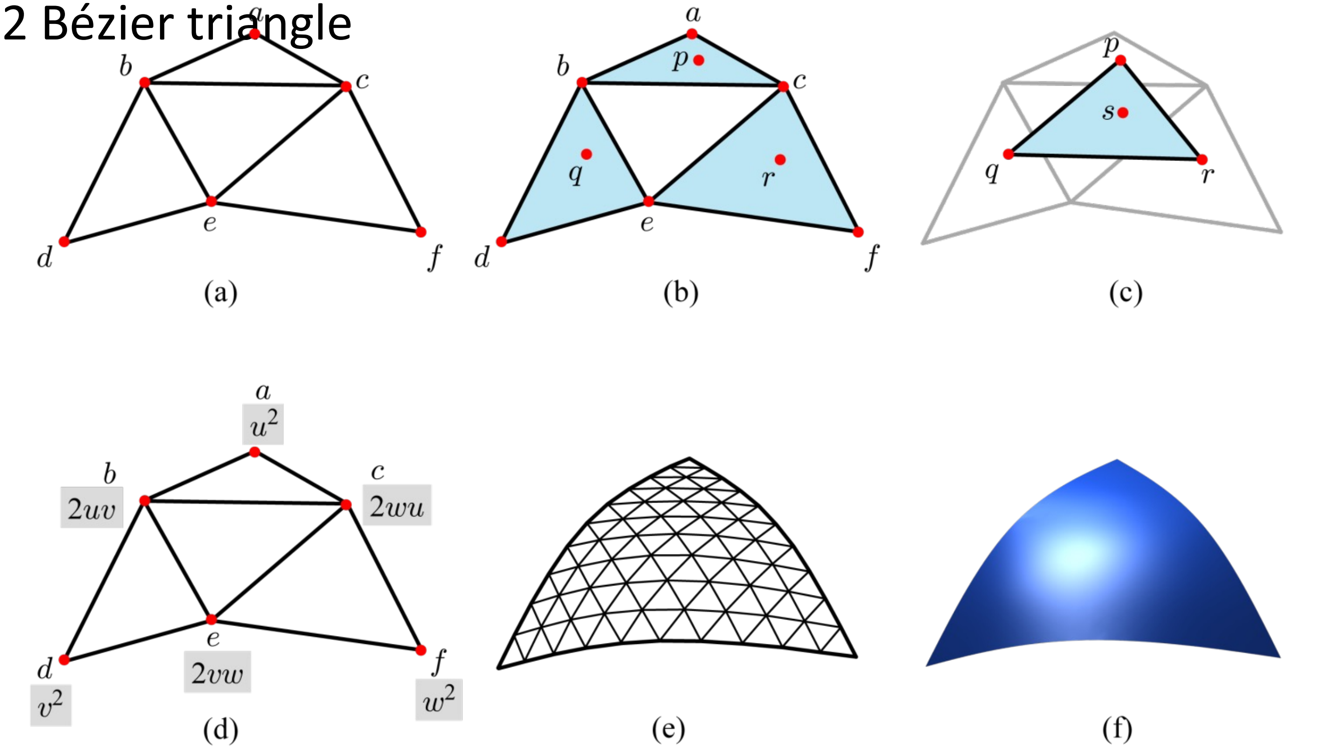
$$p(u, v, w) = ua + vb + wc$$

$$u = \frac{area(p, b, c)}{area(a, b, c)}, v = \frac{area(p, c, a)}{area(a, b, c)}, w = \frac{area(p, a, b)}{area(a, b, c)}$$

- The weights (u,v,w) are called the *barycentric coordinates* of p.
- The triangle is divided into three sub-triangles, and the weight given for a control point is proportional to the area of the sub-triangle "on the opposite side."
- Obviously, $u+v+w=1$, and therefore w can be replaced by $(1-u-v)$.

B

- Degree-2 Bézier triangle



$$p = ua + vb + wc$$
$$q = ub + vd + we$$
$$r = uc + ve + wf$$

$$\begin{aligned} s &= up + vq + wr \\ &= u(ua + vb + wc) + v(ub + vd + we) + w(uc + ve + wf) \\ &= u^2 a + 2uvb + 2wuc + v^2 d + 2vwe + w^2 f \end{aligned}$$

- See what *s* represents when *u*=0 and *u*=1.

B[illegible]([illegible])

- Degree-3 Bézier triangle