Supplement to Lecture 12

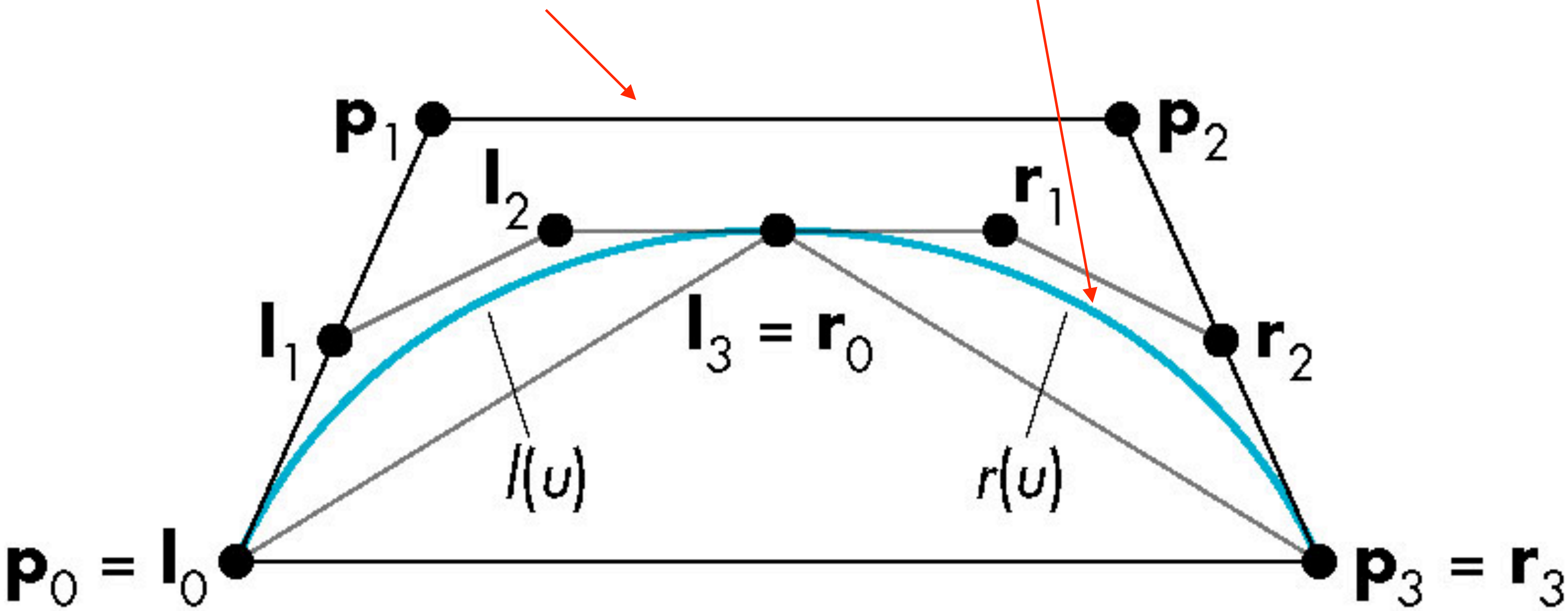
Bezier-B-spline-Curve/ Surface Subdivision

deCasteljau Subdivision

- We can use the convex hull property of Bezier curves to obtain an efficient recursive subdivision method that does not require any function evaluations
  - Uses only the values at the control points
- Based on the idea that "any polynomial and any part of a polynomial is a Bezier polynomial for properly chosen control data"

Cubic Bezier Subdivision

p_0, p_1 , p_2 , p_3 determine a cubic Bezier polynomial and its convex hull

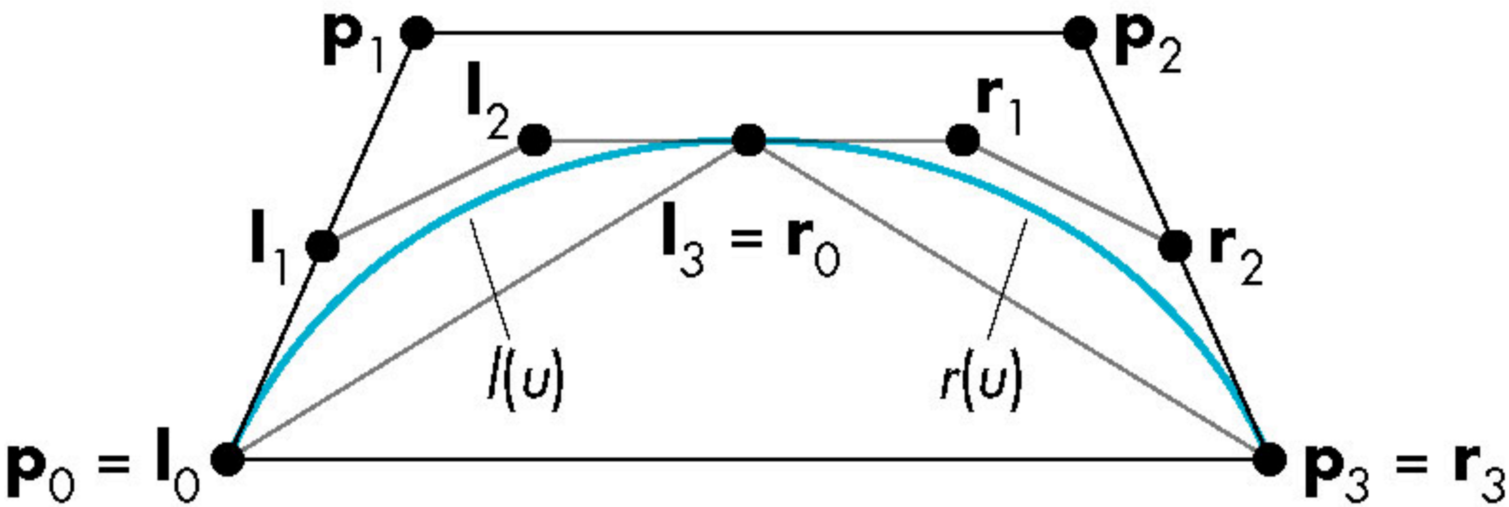


Control points of left half l(u) and right half r(u)

Variation Diminishing Property

$\{l_0, l_1, l_2, l_3\}$ and $\{r_0, r_1, r_2, r_3\}$each have a convex hull that that is closer to p(u) than the convex hull of $\{p_0, p_1, p_2, p_3\}$ This is known as the *variation diminishing property*.

The polyline from l_0 to l_3 (= r_0) to r_3 is an approximation to p(u). Repeating recursively we get better approximations.



Subdivision Equations

Start with Bezier equations $p(u)=\mathbf{u}^T M_B \mathbf{p}$

$l(u)$ must interpolate $p(0)$ and $p(1/2)$

$l(0) = l_0 = p_0$
$l(1) = l_3 = p(1/2) = 1/8(p_0 + 3 p_1 + 3 p_2 + p_3)$

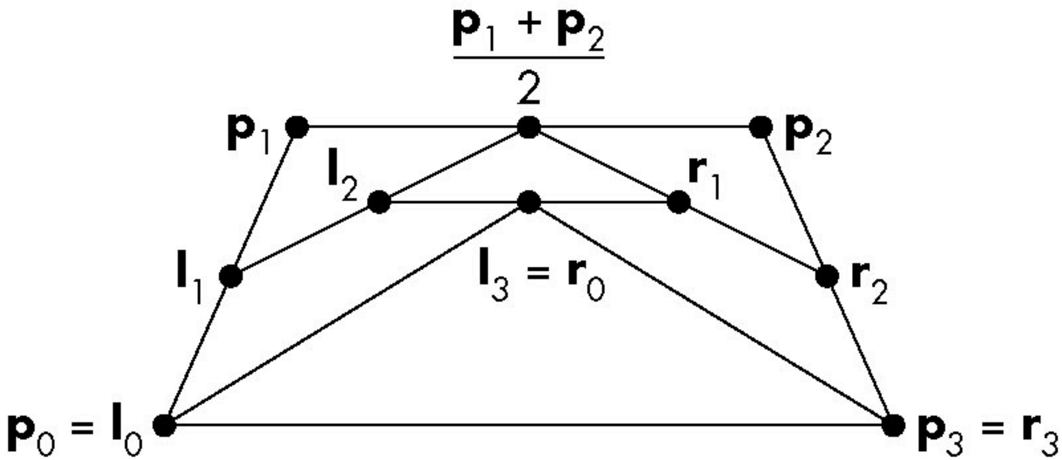
Matching slopes, taking into account that $l(u)$ and $r(u)$ only go over half the distance as $p(u)$

$l'(0) = 3(l_1 - l_0) = p'(0) = 3/2(p_1 - p_0)$
$l'(1) = 3(l_3 – l_2) = p'(1/2) = 3/8(- p_0 - p_1 + p_2 + p_3)$

Symmetric equations hold for $r(u)$

Efficient Version

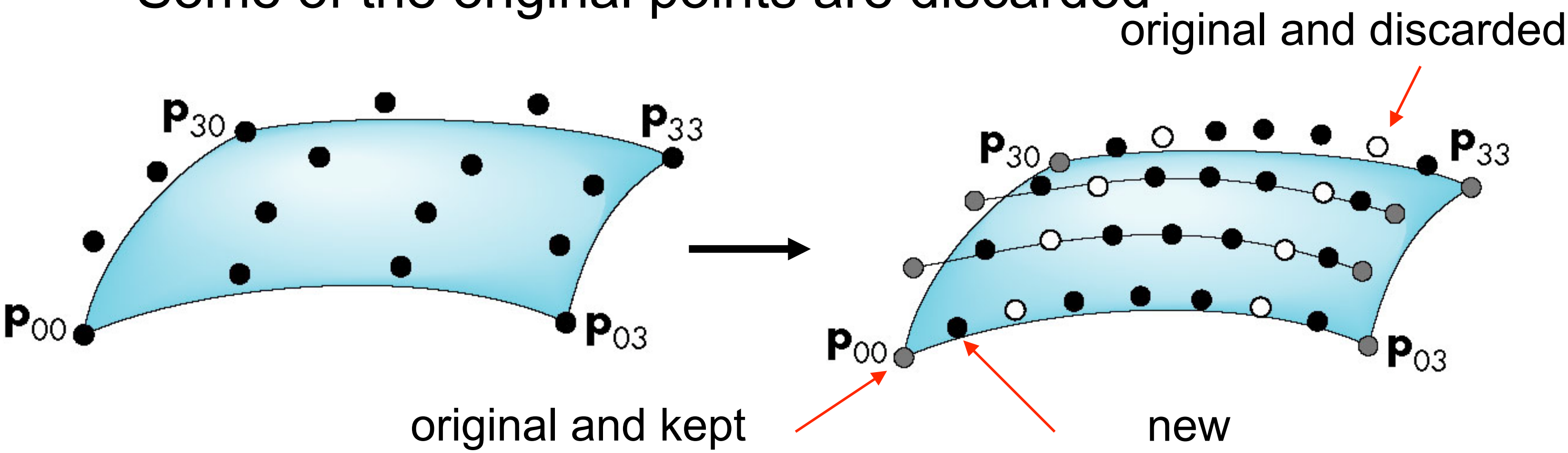
$l_0 = p_0$
$r_3 = p_3$
$l_1 = ½(p_0 + p_1)$
$r_1 = ½(p_2 + p_3)$
$l_2 = ½(l_1 + ½(p_1 + p_2))$
$r_1 = ½(r_2 + ½(p_1 + p_2))$
$l_3 = r_0 = ½(l_2 + r_1)$



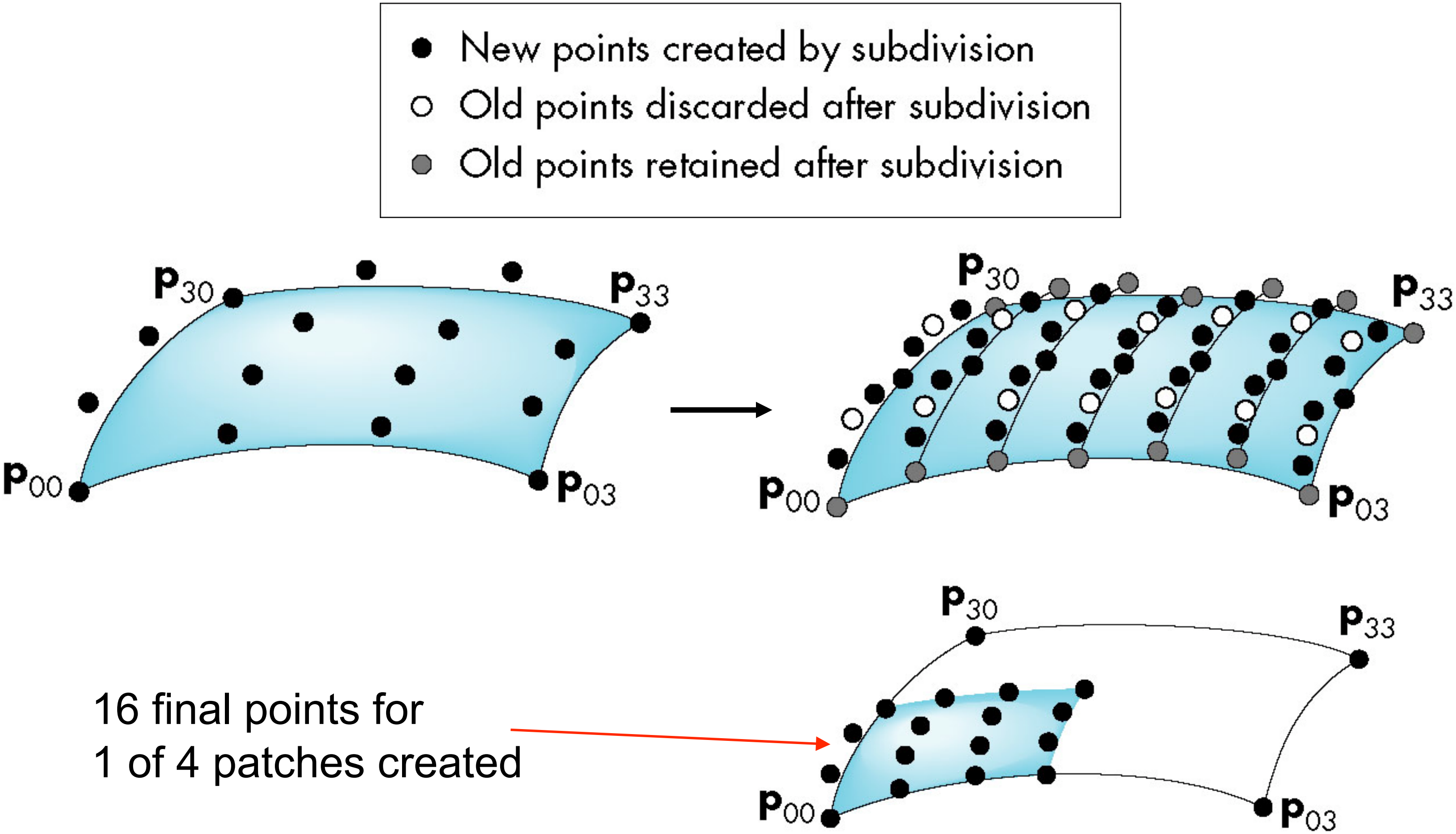
Requires only shifts and adds!

Bezier Surface Subdivision

- We can apply the recursive method to surfaces since the Bezier surface patch curves of constant u (or v) are Bezier curves in u (or v)
- First subdivide in u
  - Process creates new points
  - Some of the original points are discarded



Recursive Subdivision



16 final points for
1 of 4 patches created

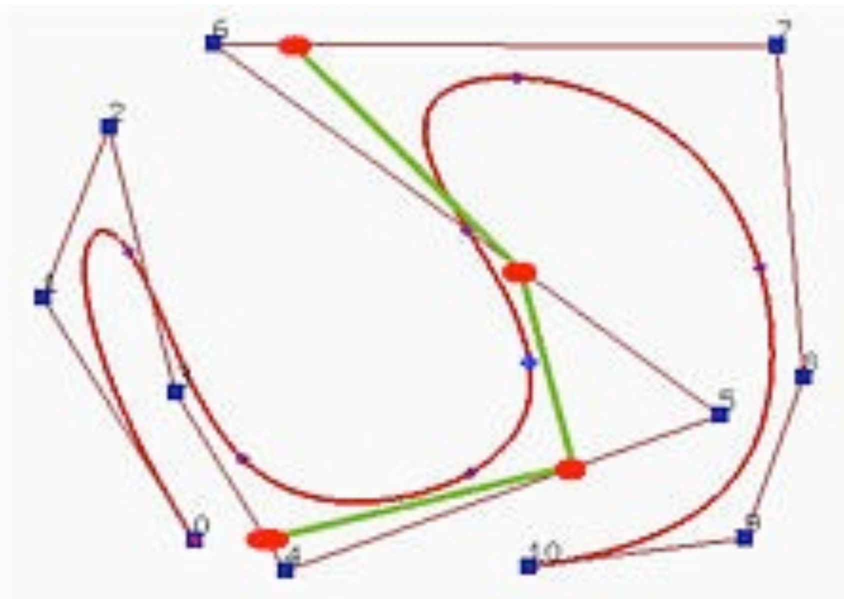
Monday, February 18, 13

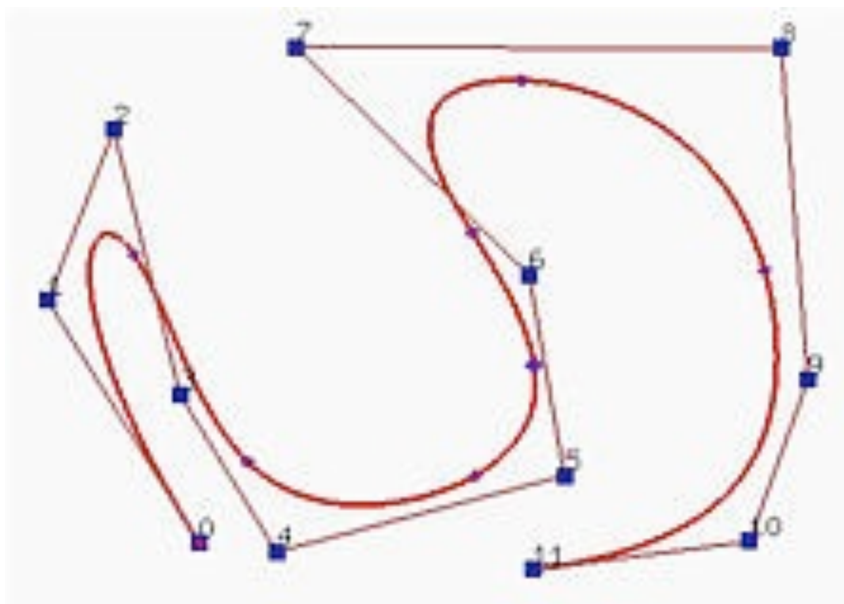
Knot Insertion : B-Spline

- *knot insertion* is adding a new knot into the existing knot vector without changing the shape of the curve.
- new knot may be equal to an existing knot → the multiplicity of that knot is increased by one
- Since, number of knots = k + n + 1
- If the number of knots is increased by 1→ either degree or number of control points must also be increased by 1.
- Maintain the curve shape →maintain degree →change the number of control points.

Knot Insertion : B-Spline

- So, inserting a new knot causes a new control point to be added. In fact, some existing control points are removed and replaced with new ones by corner cutting





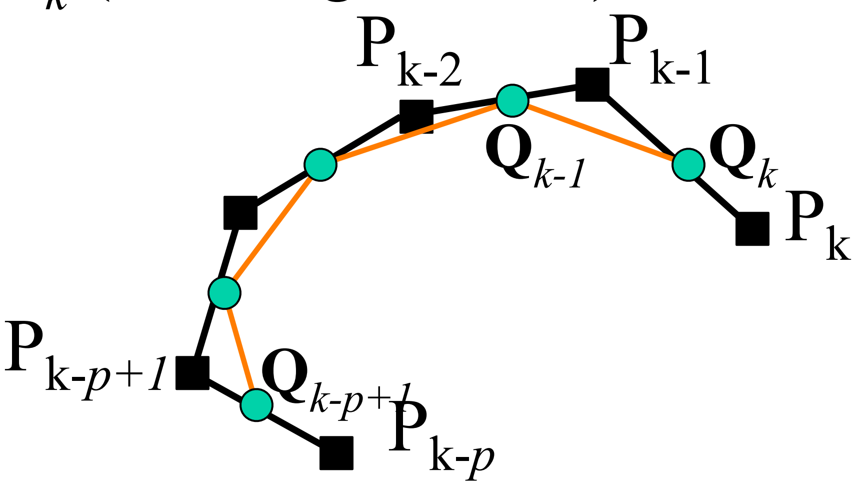
Insert knot u = 0.5

Single knot insertion : B-Spline

- Given n+1 control points – P_0, P_1, .. P_n
- Knot vector, $U= (u_0, u_1,\ldots u_m)$
- Degree = p, order, k = p+1
- Insert a new knot t into knot vector without changing the shape.
- → find the knot span that contains the new knot. Let say $[u_k, u_{k+1})$

Single knot insertion : B-Spline

- This insertion will affected to k (degree + 1) control points (refer to B-Spline properties) → P_k, P_{k-1}, P_{k-1},…P_{k-p}
- Find p new control points $\mathbf{Q}_k$ on leg $\mathbf{P}_{k-1}\mathbf{P}_k$, $\mathbf{Q}_{k-1}$ on leg $\mathbf{P}_{k-2}\mathbf{P}_{k-1}$, ..., and $\mathbf{Q}_{k-p+1}$ on leg $\mathbf{P}_{k-p}\mathbf{P}_{k-p+1}$ such that the old polyline between $\mathbf{P}_{k-p}$ and $\mathbf{P}_k$ (in black below) is replaced by $\mathbf{P}_{k-p}\mathbf{Q}_{k-p+1}\ldots\mathbf{Q}_k\mathbf{P}_k$ (in orange below)



Single knot insertion : B-Spline

- All other control points are not change
- The formula for computing the new control point $\mathbf{Q}_i$ on leg $\mathbf{P}_{i-1}\mathbf{P}_i$ is the following
  - $\mathbf{Q}_i = (1-a_i)\mathbf{P}_{i-1} + a_i\mathbf{P}_i$
  - $a_i = \dfrac{t - u_i}{u_{i+p} - u_i}$ $\qquad k-p+1 <= i <= k$

Single knot insertion : B-Spline

- Example
- Suppose we have a B-spline curve of degree 3 with a knot vector as follows:

u_0 to u_3	u_4	u_5	u_6	u_7	u_8 to u_{11}
0	0.2	0.4	0.6	0.8	1

Insert a new knot t = 0.5 , find new control points and new knot vector?

Single knot insertion : B-Spline

Solution:

- $t = 0.5$ lies in knot span $[u_5, u_6)$
- the affected control points are $\mathbf{P}_5$, $\mathbf{P}_4$, $\mathbf{P}_3$ and $\mathbf{P}_2$
- find the 3 new control points Q_5, Q_4, Q_3
- we need to compute a_5, a_4 and a_3 as follows
  - $a_5 = \dfrac{t - u_5}{u_8 - u_5} = \dfrac{0.5 - 0.4}{1 - 0.4} = 1/6$
  - $a_4 = \dfrac{t - u_4}{u_7 - u_4} = \dfrac{0.5 - 0.2}{0.8 - 0.2} = 1/2$
  - $a_3 = \dfrac{t - u_3}{} = \dfrac{0.5 - \quad 0}{} = 5/6$

Single knot insertion : B-Spline

- Solution (cont)
- The three new control points are
- $\mathbf{Q}_5 = (1\text{-}a_5)\mathbf{P}_4 + a_5\mathbf{P}_5 = (1\text{-}1/6)\mathbf{P}_4 + 1/6\mathbf{P}_5$
- $\mathbf{Q}_4 = (1\text{-}a_4)\mathbf{P}_3 + a_4\mathbf{P}_4 = (1\text{-}1/6)\mathbf{P}_3 + 1/6\mathbf{P}_4$
- $\mathbf{Q}_3 = (1\text{-}a_3)\mathbf{P}_2 + a_3\mathbf{P}_3 = (1\text{-}5/6)\mathbf{P}_2 + 5/6\mathbf{P}_3$

Single knot insertion : B-Spline

- Solution (cont)
- The new control points are $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{Q}_3$, $\mathbf{Q}_4$, $\mathbf{Q}_5$, $\mathbf{P}_5$, $\mathbf{P}_6$, $\mathbf{P}_7$
- the new knot vector is

u_0 to u_3	u_4	u_5	u_6	u_7	u_8	u_9 to u_{12}
0	0.2	0.4	0.5	0.6	0.8	1