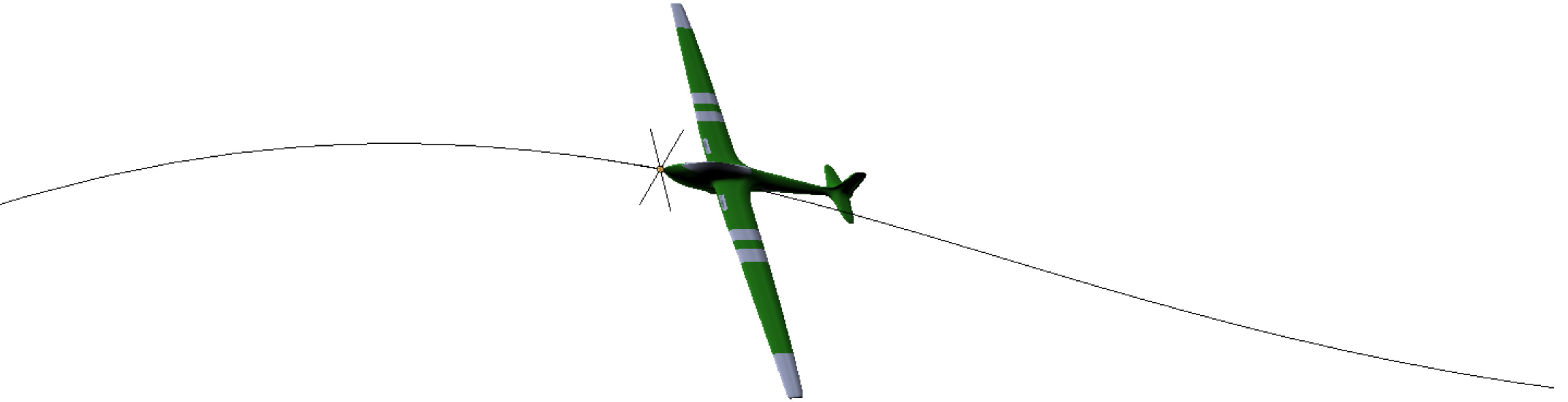
Animation:
Moving along Curves



Follow a path with an object or camera

- A simple animation : everything stays static except the camera (or an object) (walk throughs or flybys) along a path.
- The camera: like any other object with regard to orientation and positioning.
- The user must build a path in the space that will be followed by the observer / camera and then define **position, velocity** and **orientation**.
- **Path = Place key frame + interpolation of inbetween frames.**
- If the path is obtained by digitization then it needs a smooth. The path can also be defined on the surface of another object.

Orientation along a path

- **Follow a path:**
  - The keys are only given for the position
  - How to change your orientation in a "natural" way.

- **Example:** a glance while you walk
  - Look in the direction you walk

Orientation → several options:

- Frenet Frame
- Center of Interest
- Follow Object

Control the motion along the curve

Path shape:

The parametric curve P (u) (eg Spline) in parameter u

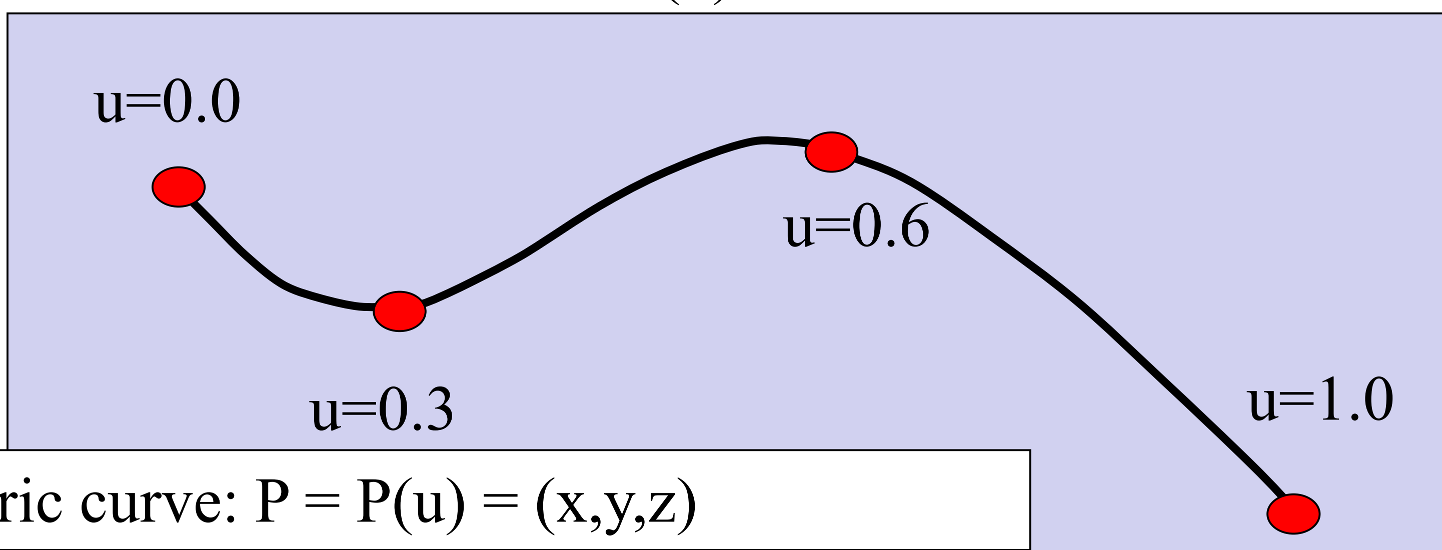
Motion control along the path: The speed at which the curve is crossed depends on the parameterization

Constant speed advance (t incremented by Δt) does not match uniform parameterisation (equal points along the curve)

The constant speed is only ensured by

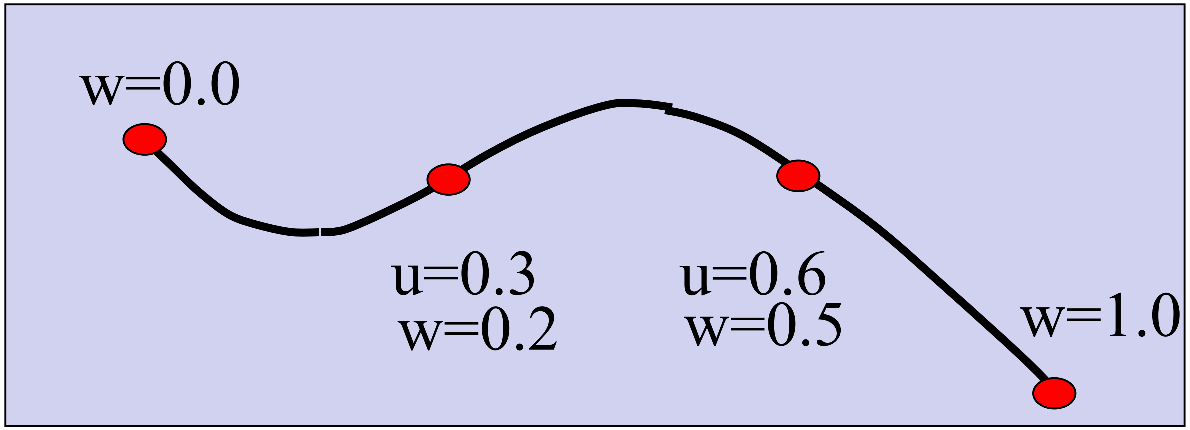
parameterization to arc length (arc-length)

Points on the curve P = P(u) 0.0 <=u<=1.0



Parametric curve: P = P(u) = (x,y,z)

Arc-length reparameterization



Arc-length (lunghezza ad arco):

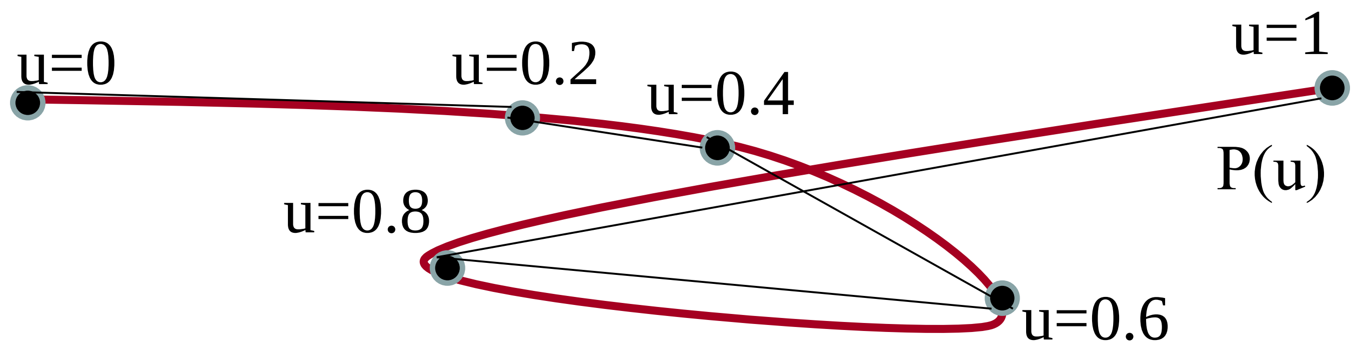
$$S = \int_0^L \left\| \frac{dP}{du} \right\| du$$

Chord-length: Approximate the curve with a polygonal line
Approximate the arc-length for Δu ->0

$$u_i = i\Delta u, \quad P_i = P(u_i)$$

$$S = \sum_i \left\| P_{i+1} - P_i \right\|$$

Chord-length reparameterization



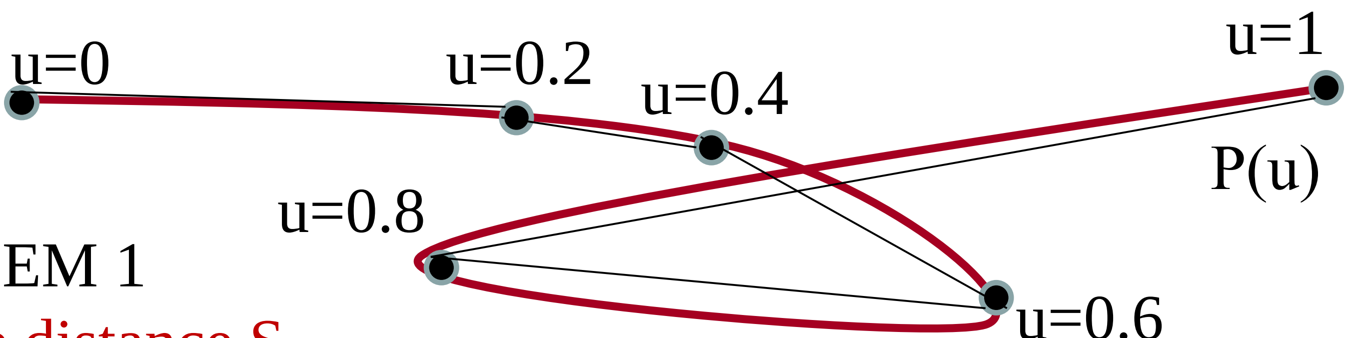
Let $s(u_i)$ the approximate chord length of the curve from u=0.0 a $u=u_i$

$$s(u_i) \approx \sum_{j=1}^{i} \left\| P_j - P_{j-1} \right\| = s(u_{i-1}) + \left\| P_i - P_{i-1} \right\|$$

u	s(u)
0.0	0.0
0.2	2.5
0.4	4.0
0.6	5.0
0.8	7.0
1.0	8.5

- We build chord length table for constant increments of parameter u
- Interpolate between u values for constant s(u) values

Chord-length reparameterization



PROBLEM 1

Find the distance S
along the curve at time (parameter) u*.

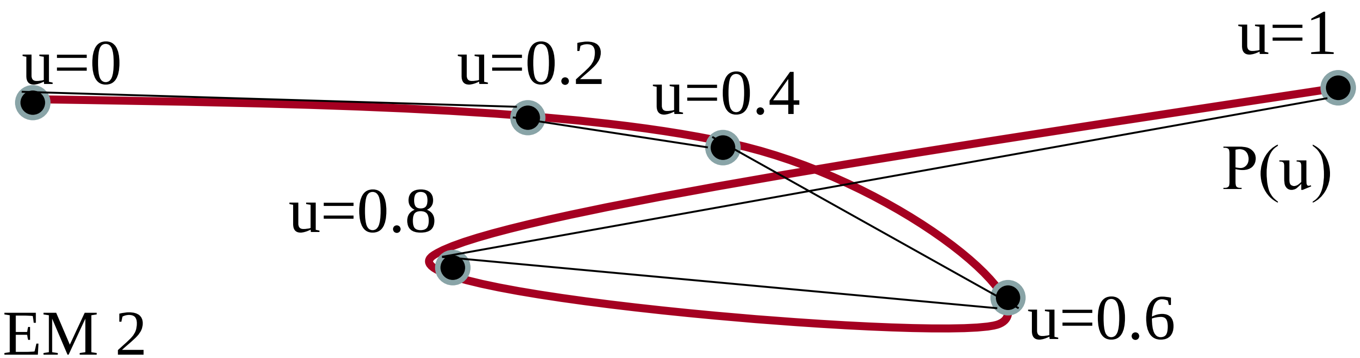
Let i be the largest parametric entry u(i) < u*.
Interpolate linearly between the two values
such that u* between [u1,u2]

$$S = s(i) + \frac{(u^* - u(i))}{(u(i+1) - u(i))}(s(i+1) - s(i))$$

Es: If u* = 6.5 then u is between [0.6, 0.8]

i	u	s(u)
1	0.0	0.0
2	0.2	2.5
3	0.4	4.0
4	0.6	5.0
5	0.8	7.0
6	1.0	8.5

Chord-length reparameterization



PROBLEM 2

Finding the value of u
given the arc-length s*

Let i be the largest parametric entry s(i) < s*.
Interpolate between two values such that
s* in [s1,s2]

$$u = u(i) + \frac{(s^* - s(i))}{(s(i+1) - s(i))}(u(i+1) - u(i))$$

i	u	s(u)
1	0.0	0.0
2	0.2	2.5
3	0.4	4.0
4	0.6	5.0
5	0.8	7.0
6	1.0	8.5

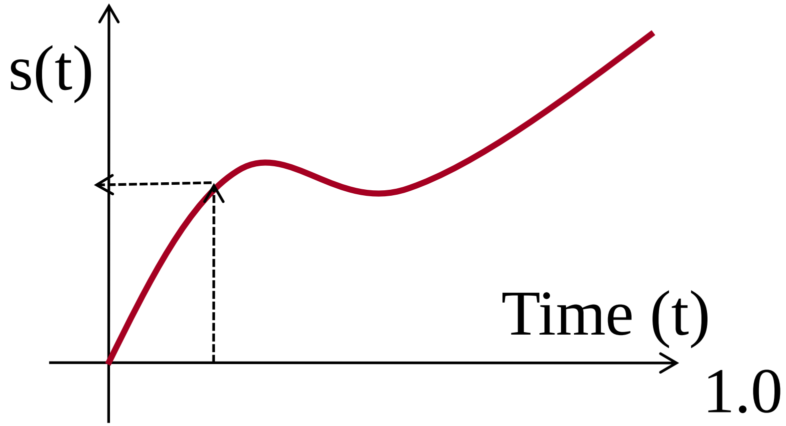
Speed Control

USER: Defines a function (speed control function) s(t) that determines where the object (camera) must be found at each time t

The curve of the path determines where to go, the curve s(t) establishes when; they are independent of each other

Eg: $s(t)=vt$ motion with uniform speed

$s(t)=v_0t+at^2/2$ motion with uniform acceleration



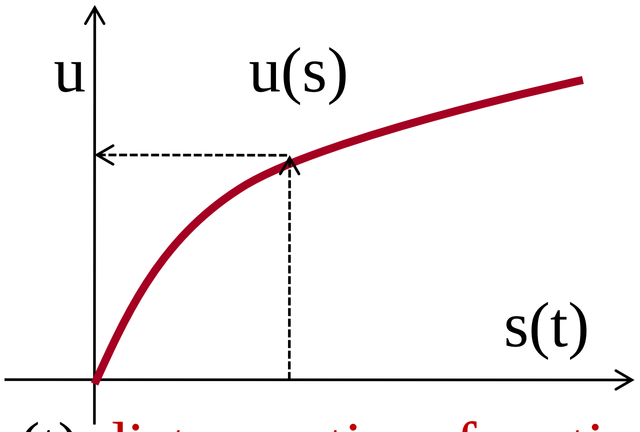
s(t) distance-time function (arc-length)

Monotonic in t otherwise you will go back, continue to have no speed jumps

Reparameterization to control the motion

For each time t, s(t) is the desired distance along the curve, to get the corresponding position on the P(u) curve, we must know the parameter u

SYSTEM: The system calculates the parameter u which corresponds to a certain s(t) by consulting the chord-length table (u, s(u))



P(u(s(t)))

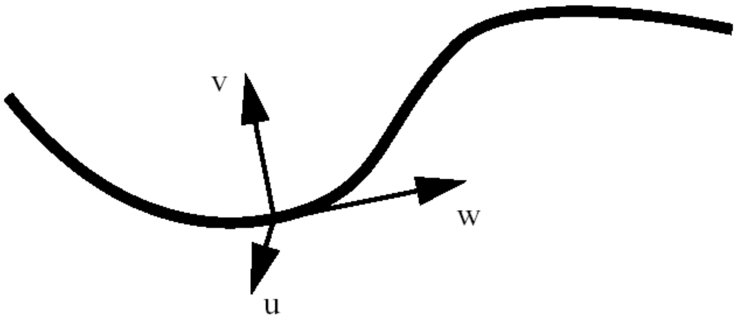
s(t) distance-time function

Orientation control

- FRENET Frame along a path: the orientation is defined by the properties of the curve path
- Local right coordinate system (U, V, W), with system origin (POS) determined by a point along a path P(s). The system changes direction along the curve.

For each POS (point on the path):

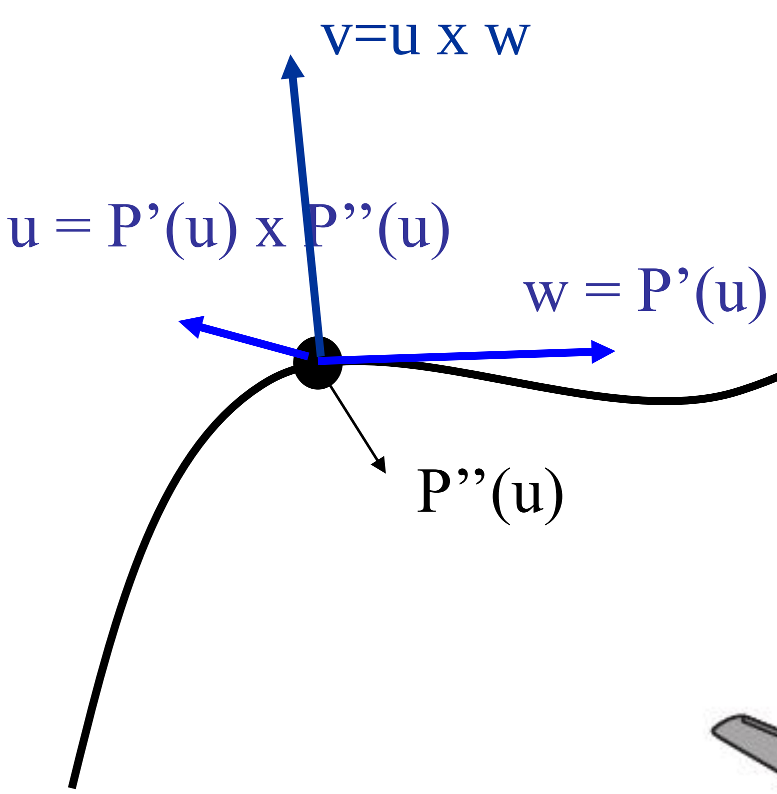
- W: direction view defines orientation
- V: up-vector orthogonal to w (the observer's head is aligned with the up-vector)
- U: Vector orthogonal to the two object)



- Normalize the vectors

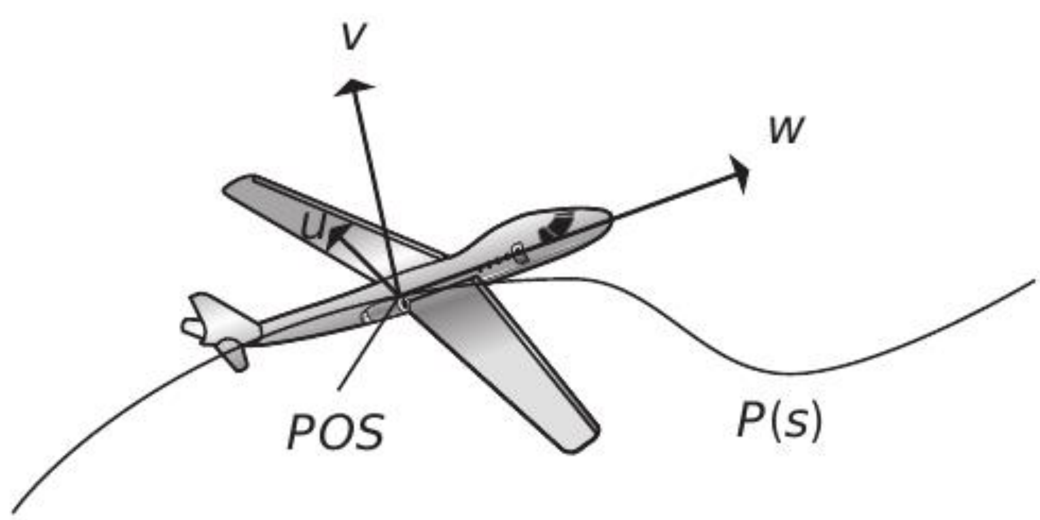
Frenet Frame

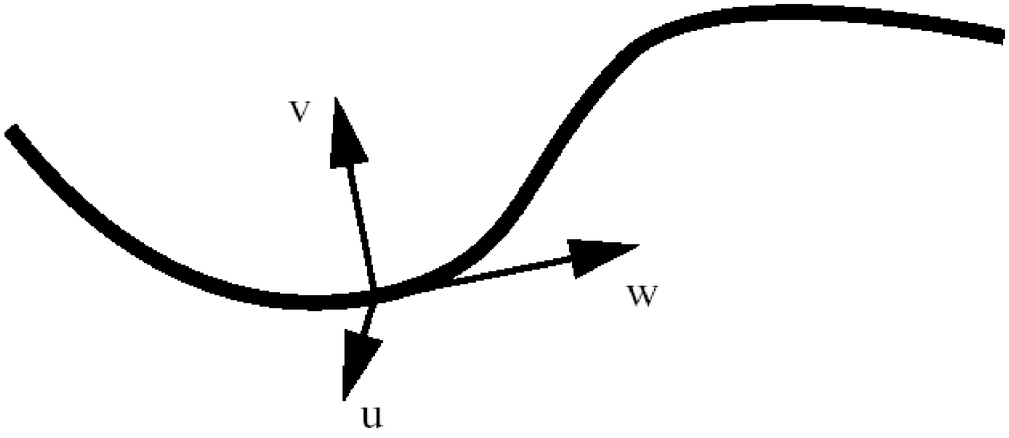
W direction of view is
the direction tangent to the curve



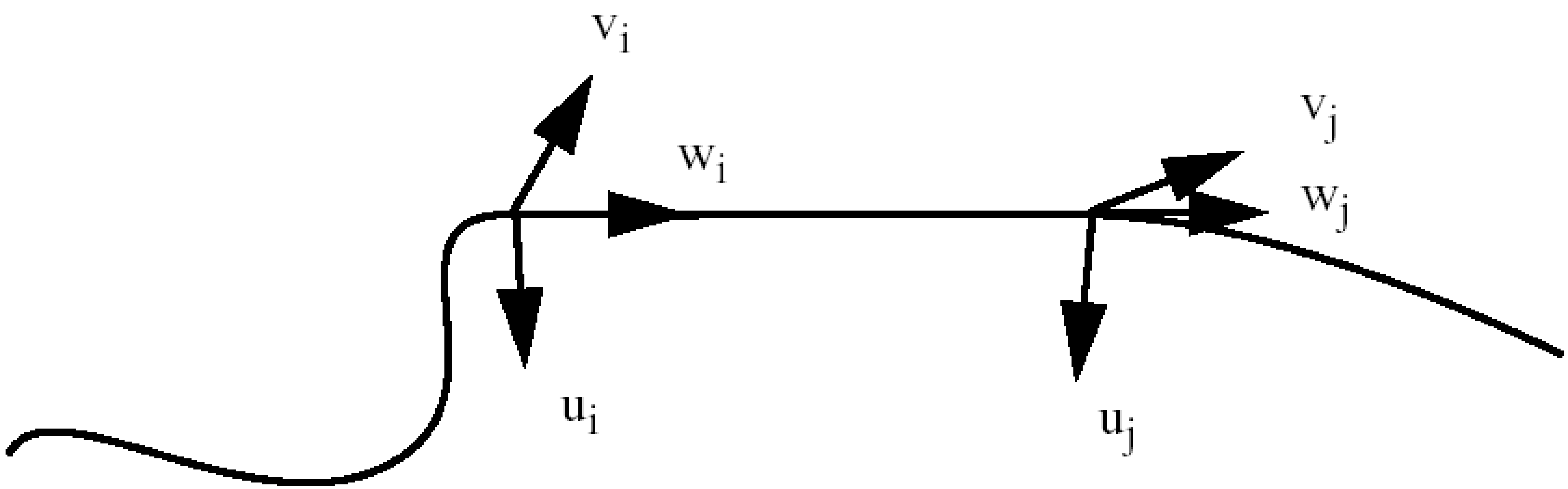
U-vector orthogonal to w, ortho to the plan defined by tangent and curvature

V up-vector orthogonal to w, ortho to the plan defined by tangent and u





Problem: P’’(s)=0. Solution: interpolate the frenet frames at the extrema of the segment with zero curvature.



Frenet frame: problems

- **Curvature zeros along a segment of path**

  Example: straight line

- **Solution:** Interpolate Extreme Frames Different only by rotation around the axis w (straight line)
- **Zero curvature at one point:**

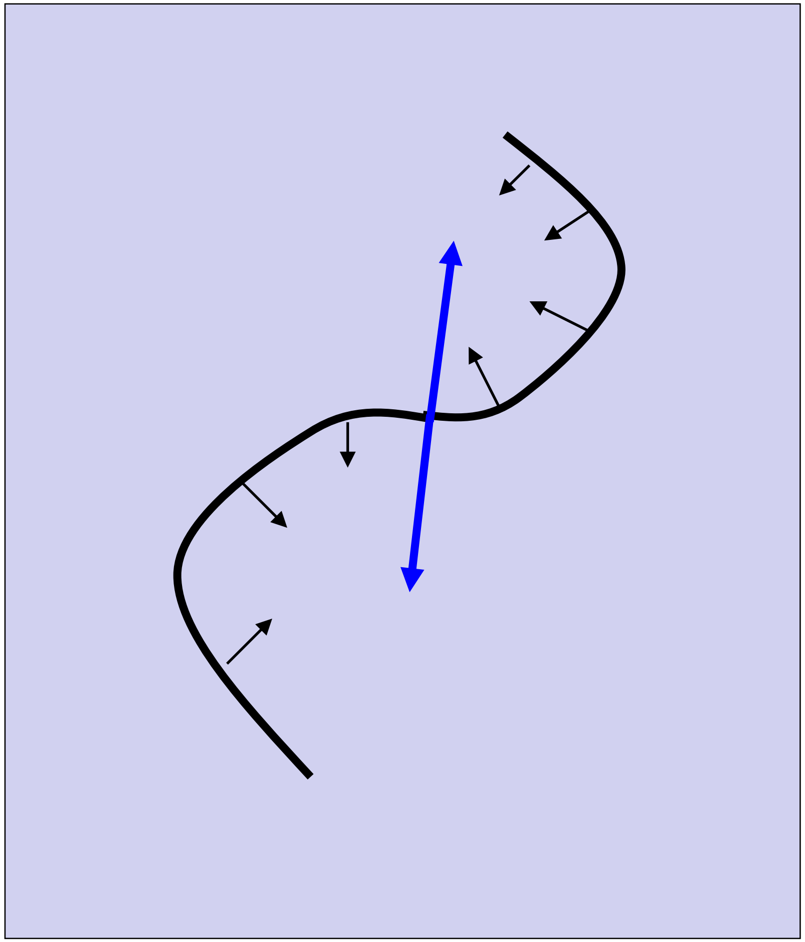
  Possible flip (eg camera tilt)

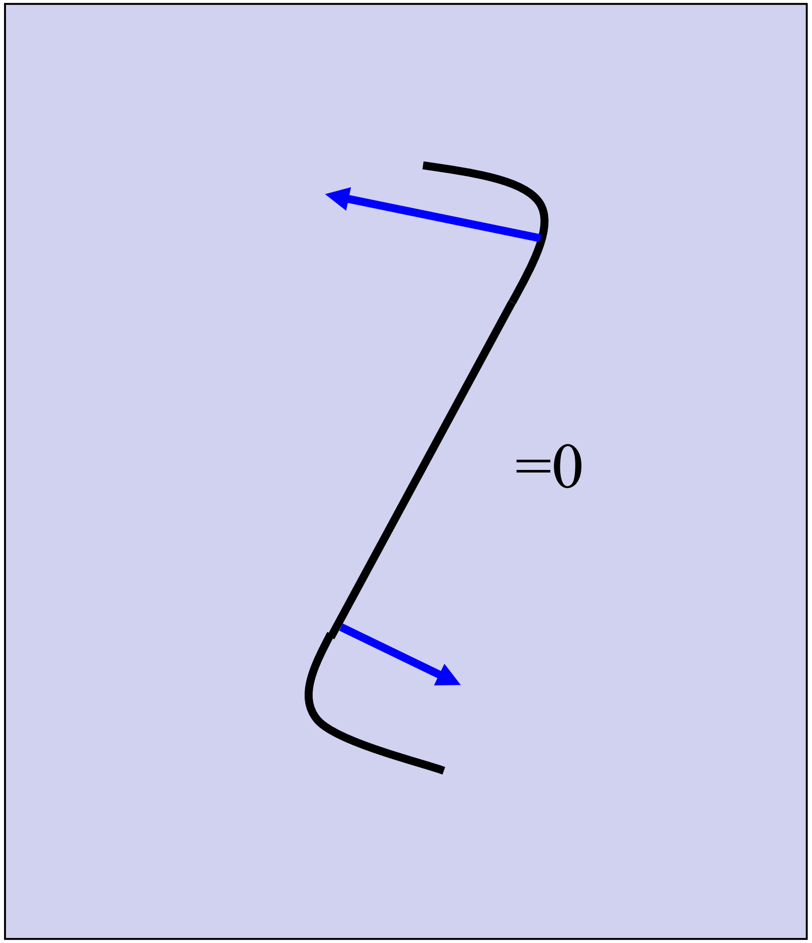
- **Discontinuity in bending:**

  Sudden Object Orientation Changes

- **These effects are sometimes intolerable**

Continuity of the curvature



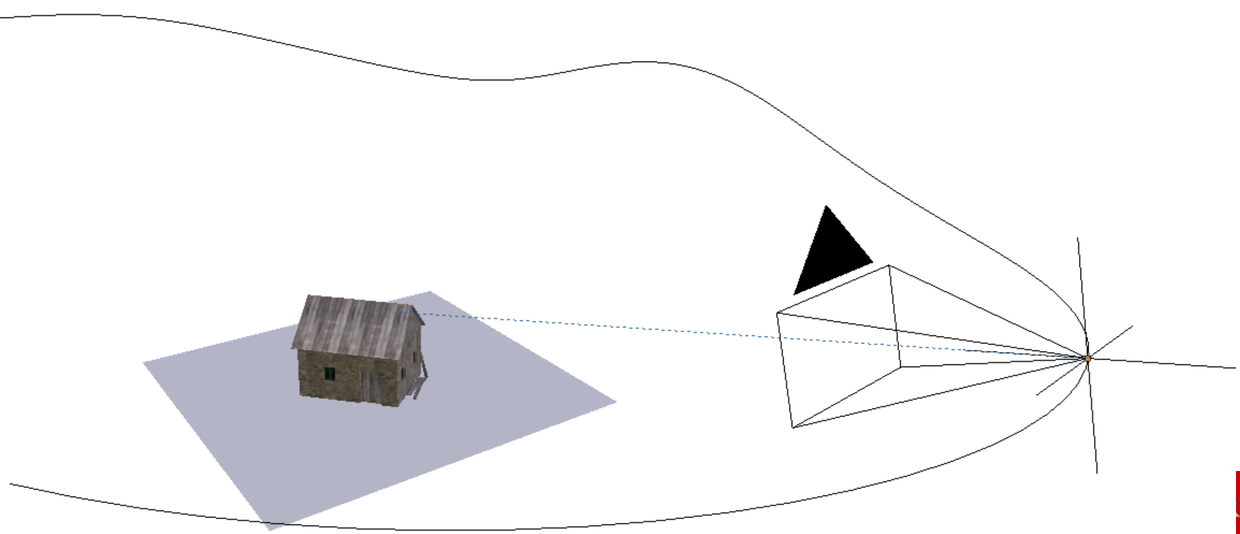


Alternatively…

- **The tangent vector as orientation vector is ok for objects**
  - Not ideal for the camera

- **For camera orientation**

  "Center of Interest" (COI)



(1)Camera Path Following COI

- The center of interest can be kept fixed while the observer's position is interpolated along the curve.

- **W View direction -> obtained by the observer position (POS) and by the center of interest (COI)**

$$w = COI - POS$$

$$u = w \times y - axis$$

$$v = u \times w$$

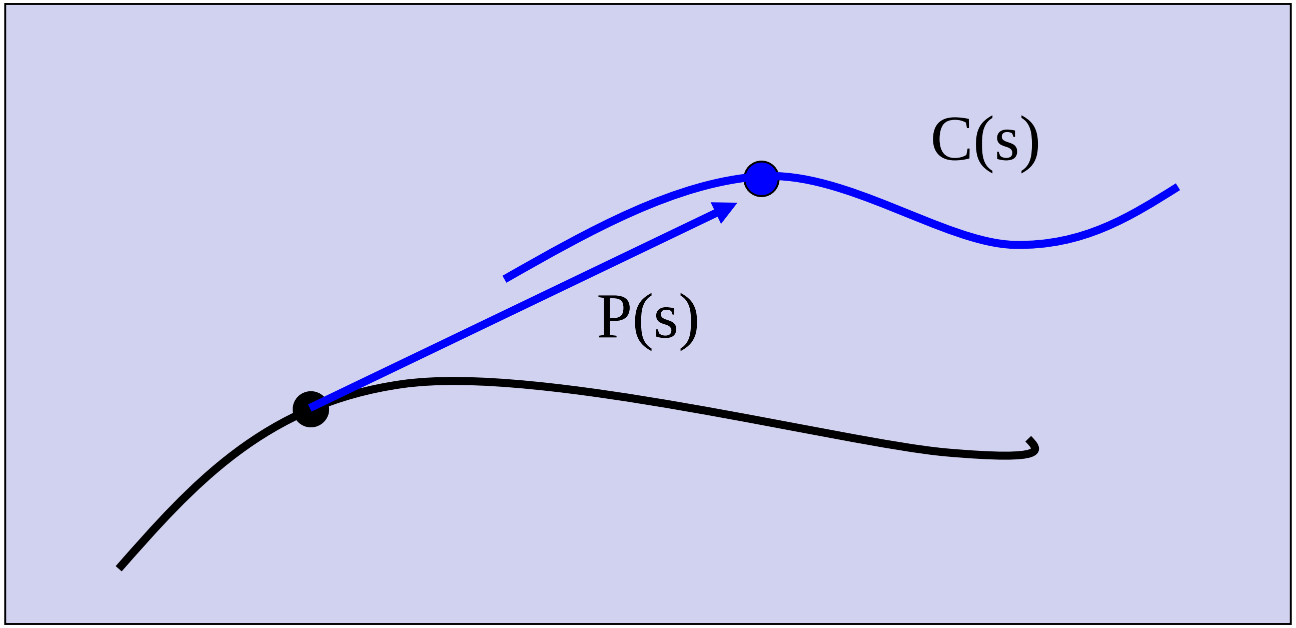
- Useful when the observer is flying over the environment focusing attention on a specific location.

(2)Camera Path Following: COI(s)

- Construction of a path of centers of interest,

  eg. a series of building in an environment.
- Often you want to focus your attention on a building for some frames before proceeding with the next building.
- The COI can be attached to objects in animation.
- A separate path for the COI
  - You can animate this point separately by using additional key positions.

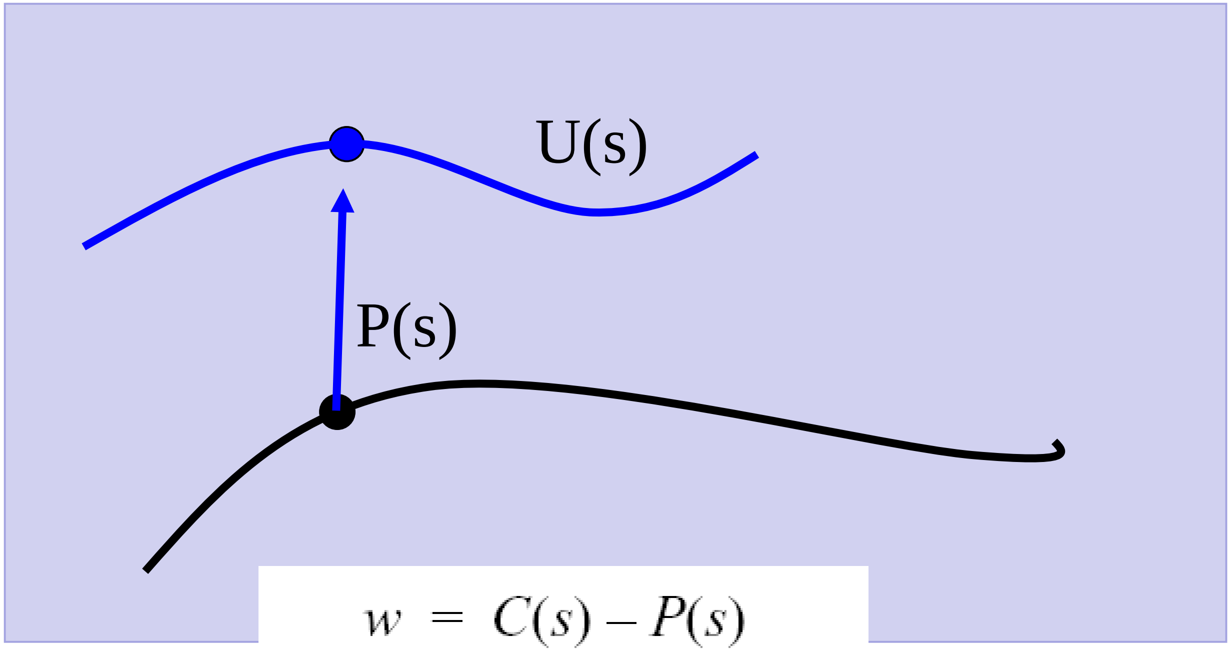
Define the direction vector



COI is on a path C(s) which differs from the camera P(s)

Define the "up" vector:

In the plane generated by the view dir. and the global UP vector



$$w = C(s) - P(s)$$

$$u = (U(s) - P(s)) \times w$$

$$v = w \times u$$

The up-vector v is on a U(s) path different from the path P(s)

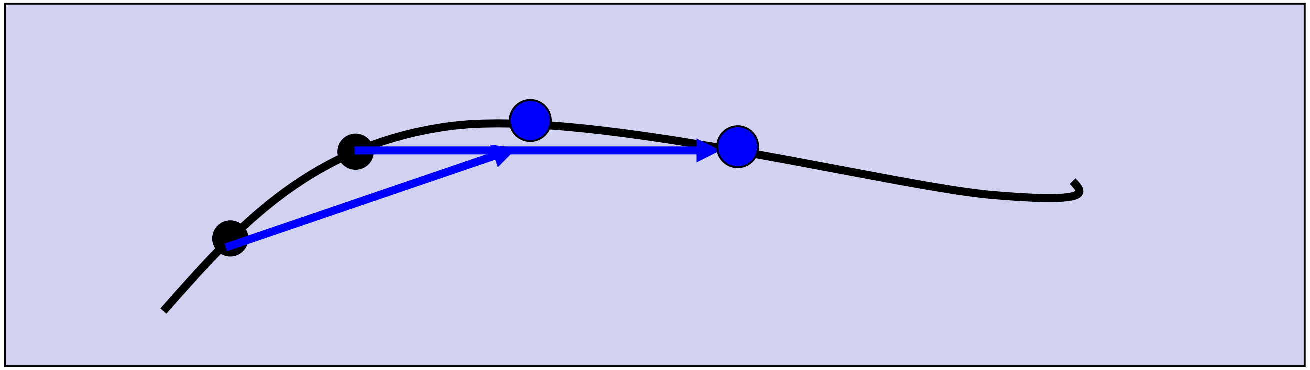
(2)Camera Path Following: look ahead

- **COI is on the camera path**

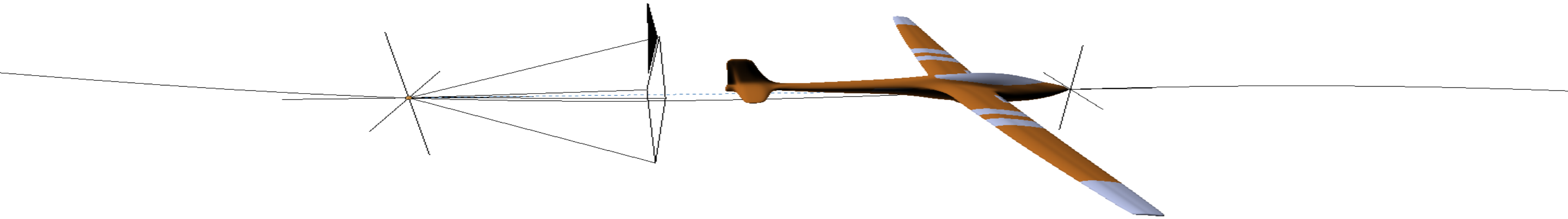
COI(s)=P(s+ds)

(after arc-length reparameterization)

At the curve extreme (s =1), COI becomes the final tangent vector

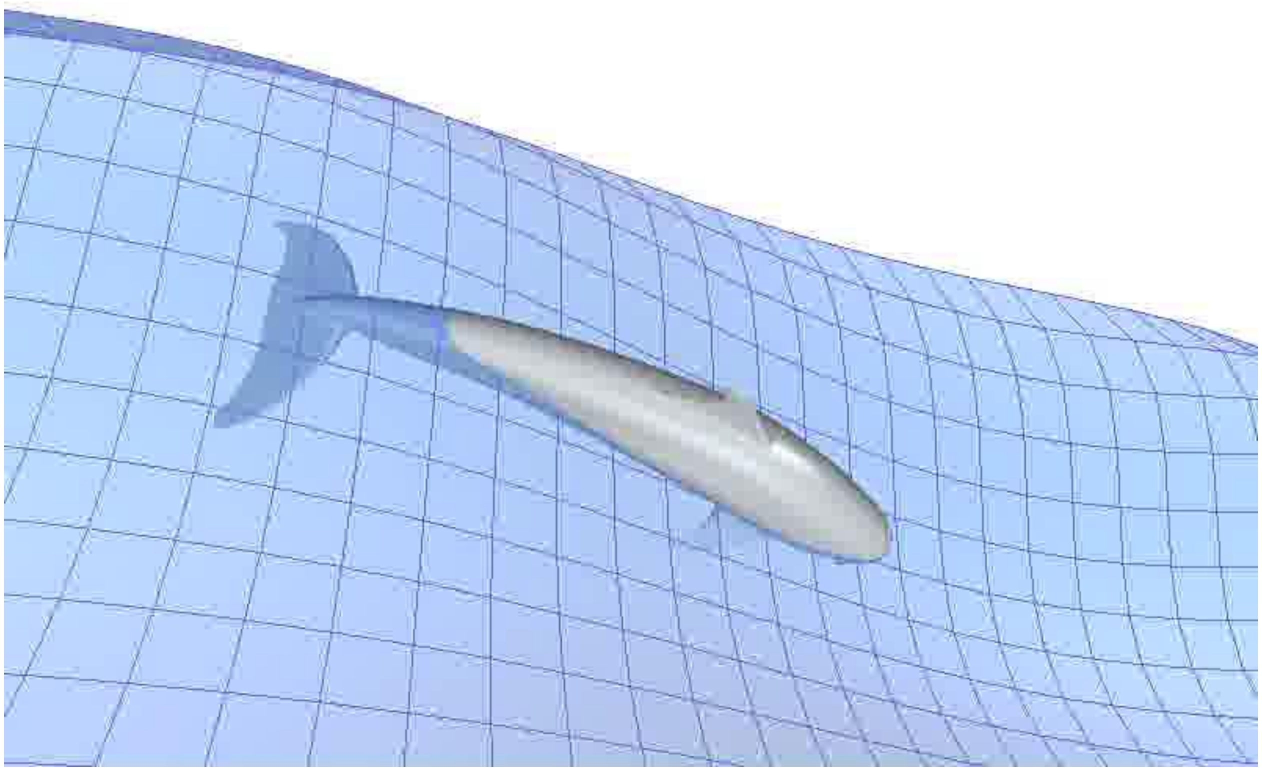


Solution **"Look ahead" (Follow object)**



Determining a path along a surface

An object moves along the surface of another object.



Determining a path along a surface

- Assume starting point P1 and final point P2 of the path

- Path along a surface mesh
  - Determine a plane the share the two points P1 and P2 and is almost perpendicular to the surface (quasi perpendicolare= average of the two normal vectors at the extreme of the path). Interpolate plane/mesh
- Path along a parametric surface (i.e. spline)
  - Draw a path on the parametric domain (either a straight line joining the two points) and determine the corresponding 3D curve on the surface itself.