Why motion planning?
- Direct control is difficult for inexperienced users
- Often results in rather ugly camera motions that easily lead to motion sickness

Parameters of Camera
- Position
- Orientation
- Zoom-factor
- Type of projection
- Shear
In this paper:
- 3 parameters for position
- 3 parameters for viewing direction (1 redundant)
- 1 parameter for roll

Camera Movement
- Translation
- Pan (side view)
- Tilt (top view)
- Roll (front view)

Guidelines on the best motions
- The camera should preferably not pass too close to obstacles.
- The horizon of the camera should be kept as straight as possible to prevent a “drunk” feeling by the viewer.
- When the camera makes a sharp turn, its speed should be lowered.
- The speed of the camera should always be as high as possible
- The viewer should get cues about where the camera is going.

Goal
- Given an environment consisting of a set of obstacles $O$, a start configuration $s$, and a goal configuration $g$, compute a short, natural looking camera motion from $s$ to $g$ avoiding collisions with all $o \in O$. 

2003-04 Semester 1, David Hsu
Preprocess (create the roadmap)
- Probabilistic Roadmap Method (PRM)
- Consider the camera as a sphere $S_\delta(c)$ of radius $\delta$ around camera position $c$.
- $\text{CLEAR}(c)$: is $S_\delta(c)$ collision free?
- $\text{LINK}(c, c')$: is the "cylinder" $C_\delta(c, c')$ collision free?

Algorithm

```
Algorithm 1: CONSTRUCT-ROADMAP
Let $V_0 = \emptyset$, $E_0 = \emptyset$
1: loop
2: $v = $ a (random) position
3: if $\text{CLEAR}(v)$ is collision-free then
4: $V = V_0 \cup \{v\}$
5: $N_v = $ a set of neighbor nodes chosen from $V$
6: for all $v' \in N_v$ do
7: if $\text{LINK}(v, v')$ is collision-free then
8: add the edge $v'v$ to $E$
```

An example of a roadmap for a house

Path plan - position
- Add start and goal
- Find a fastest path
- Improve the path
- Set speed
- Shorten the path

Add start and goal
- Connect $s$ and $g$ to their neighbor nodes.
- If failed, expand the graph by adding a number of additional random positions in the vicinity of $s$ and $g$. 
What is a fastest path?

Penalty function

- Use an edge-based version of Dijkstra algorithm
- The penalty value for an edge $e$ when arriving from edge $e'$ is $p(e,e')$
- The length of $e$ is $l(e)$
- The total distance measure $d(e)$ for $e$ arriving from $e'$ is $p(e,e')+l(e)$

Path plan - position

- Add start and goal
- Find a fastest path
- Improve the path
- Set speed
- Shorten the path

$C^0$ continuous $\rightarrow$ $C^1$ continuous

- $F(x) \& G(x)$
- If $F(a)=G(a)$, then $F(x)$ and $G(x)$ is $C^0$ continuous at the position $a$
- If $F'(a)=G'(a)$, then $F(x)$ and $G(x)$ is $C^1$ continuous at the position $a$

Add circular blends

Avoid the obstacle

- Use binary search on the radius to find the largest possible radius
- Use a number of short segments instead of the arc to do collision check
A 3-D example of a path with circular blends

Path plan - position
- Add start and goal
- Find a fastest path
- Improve the path
- Set speed
- Shorten the path

Initial speed diagram

Acceleration & deceleration

Final speed diagram

Path plan - position
- Add start and goal
- Find a fastest path
- Improve the path
- Set speed
- Shorten the path
Shorten the path

Path plan - orientation
- Smoothing the viewing direction
- Up vector
- Start and goal

Viewing direction
- Should viewing direction be equal to the direction of motion?
  - NO!
- Viewing direction should give the viewer cues about where the motion is going.
- Viewing direction should be $C^1$ continuous, but motion direction is only $C^0$ continuous.

Viewing direction

Looking at a position $t_d$ ahead

Proof of $C^1$ continuity of $D(t)$ - 1
- $W(t)$: the position of the camera in the world at time $t$.
- $D(t)$: the viewing direction. Along the line from $W(t)$ to $W(t+t_d)$.
  Proof that $D(t)$ is $C^1$ continuous
Proof of $C^1$ continuity of $D(t)$ - 2
- Path: $P(i): [0..l] \rightarrow \mathbb{R}^3$, $l$ is the length of the path.
- Speed: $V(i): [0..l] \rightarrow \mathbb{R}^3$
- $U(t)$: given a time $t$, returns a position in path $i$.
- $W(t)=P(U(t))$
- $T(i) = \int_{x}^{1} \frac{1}{V(x)} \, dx$ : the time to reach the position $l$
- Then $U(t)=T^{-1}(t)$

Proof of $C^1$ continuity of $D(t)$ - 3
- $P(i)$ is $C^1$ continuous and $V(i)$ is $C^0$ continuous.
- So, $1/V(i)$ is also $C^0$ continuous.
- $T'(i) = 1/V(i)$
- $\therefore T'(i)$ is $C^1$ continuous.
- Taking the inverse of a one-to-one function does not change the continuity. So $U(t)$ is also $C^1$ continuous.
- So, $W(i)=P(U(t))$ is $C^1$ continuous.

Proof of $C^1$ continuity of $D(t)$ - 4

Path plan - orientation

Up vector
- When the up vector changes too much, the user feels the effect of being in a roller coaster.
- Always let the up vector point in the direction of the y-axis.
- Problem:
  - When the camera makes a looping-like motion, it rotates the camera 180° in a very short period of time

Solution
- Avoid situations where a twist can occur.
- Give such situations penalty function.
- Experiment showed that an exponential penalty function works best.
Path plan - orientation
- Smoothing the viewing direction
- Up vector
- Start and goal

Start and goal
- Problem:
  The viewing directions of the start configuration and the goal configuration are usually not same to the direction of the first edge and the last edge.
- The most natural solution:
  Start with a rotation and end with a rotation!

Result

2D example

3D example

Thank you!