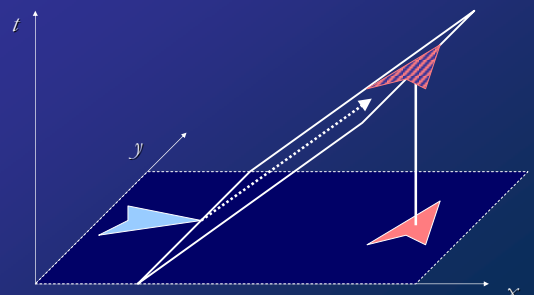
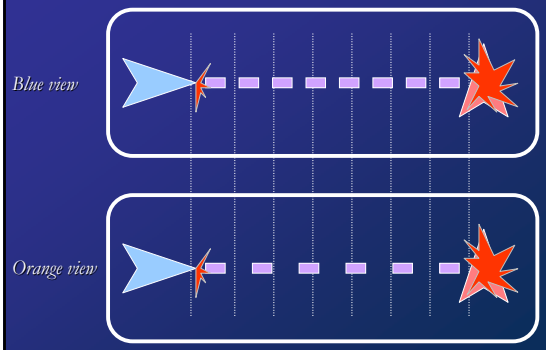
Temporal Contour (from the Blue Player's Perspective)

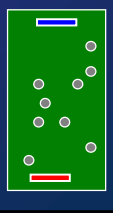


Temporal Distortion



Properties of the Co-ordinate System

- The co-ordinate system is defined independently for each player
- Depends on the player's current position and the delay of arriving information
- Changes dynamically as the player moves or as the network properties change
- Defines how a passive object should be rendered
- Two interacting objects are rendered at the same time reference point
- Each user perceives all collisions correctly
- Objects that approach the local user are rendered in the user's time
- Smooth movement

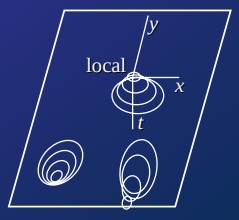


Generalizing the Local Temporal Contour

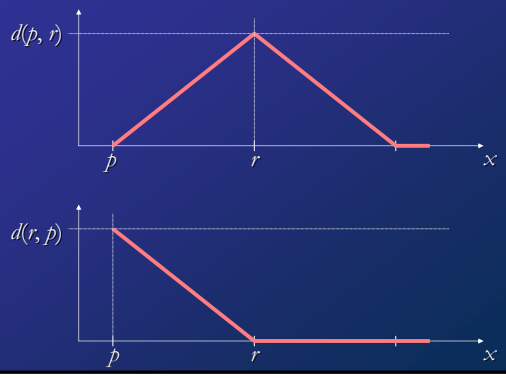
- Limitations:
  - players are capable of moving along a single axis only
  - supports two active objects only
- Generalization to a 4D co-ordinate system requires preserving for the local user:
  - interacting naturally with passive objects in vicinity
  - seeing remote interactions (passive-to-passive, passive-to-active) naturally
  - perceiving smooth motion of remote objects

Local Temporal Contour

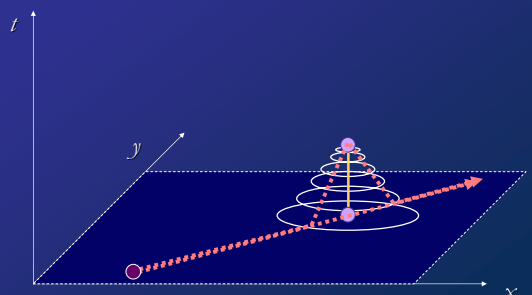
- The local user at (0, 0, 0)
- Each active object is assigned a *t* value corresponding to its latency
- Interpolate the contour over all active objects including local
- Contour defines a suitable *t* value for each spatial point



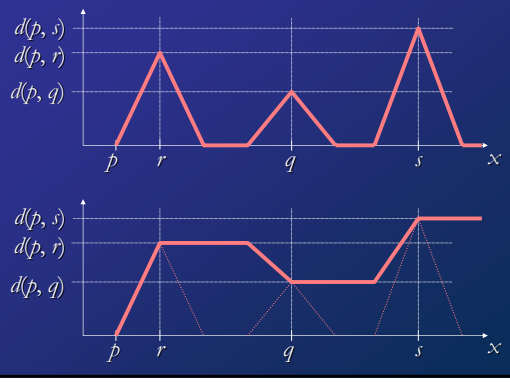
Linear Temporal Contours



2½-Dimensional Temporal Contour



Multiple Players: Aggregating the Temporal Contours



Worth Noting

- simple linear functions instead of continuous temporal contours
- LPFs are the 'opposite' of dead reckoning
  - no prediction for remote players
- the closer the players get, the more noticeable the temporal distortion becomes
  - in critical proximity interaction becomes impossible
  - no mêlée

Problems

- possibly visual disruptions on impact ⇒ shadows (see the lecture notes for details)
- sudden changes in the player's position or delay can cause unwanted effects
  - if a player leaves the game, what happens to the temporal contour?
  - third party instrusion: someone with a high delay 'blocks' the incoming entities
  - jitter: entities start to bounce back and forth in time

Bullet Time

- movies: visual effect combining slow motion with dynamic camera movement
- computer games: player can slow down the surroundings to have *more time* to make decisions
- easy in single player games: slow down the game!
- how about multiplayer games?

Bullet Time in Multiplayer Games

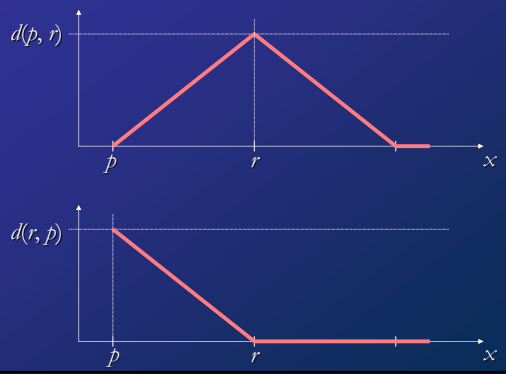
- two approaches:
  - speed up the player
  - slow down the other players
- if a player can slow down/speed up the time, how it will affect the other players?
  - localize the temporal distortion to the immediate surroundings of the player
- but how to do that?

⇒ local perception filters!

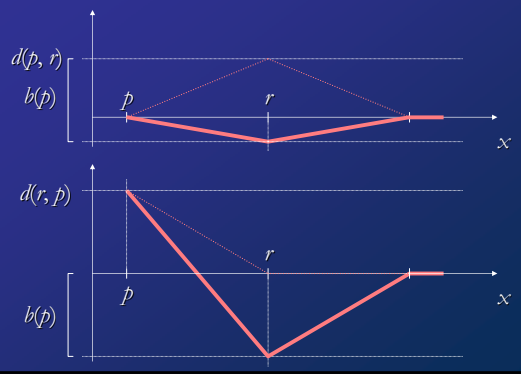
Adding Bullet Time to LPFs

- player using the bullet time has more time to react
  ⇒ the delay between bullet-timed player and the other players increases
- add artificial delay to the temporal contour

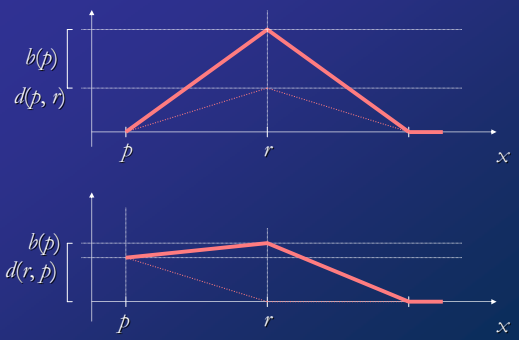
p Shoots *r* Without Bullet Time



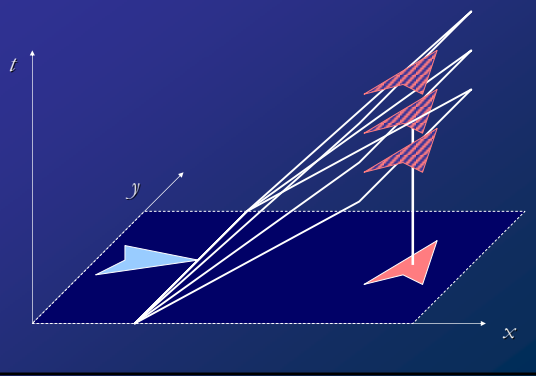
p Shoots *r* While *p* Is Using Bullet Time



p Shoots *r* While *r* Is Using Bullet Time



2½-Dimensional Temporal Contour and Bullet Time



Open Questions

- non-linear temporal contours
  - how to compute quickly?
  - noticeable benefits (if any)?
- numerical evaluation
  - measuring the distortion and its effects
- practical evaluation
  - how well does it work?
  - does it allow new kinds of games?