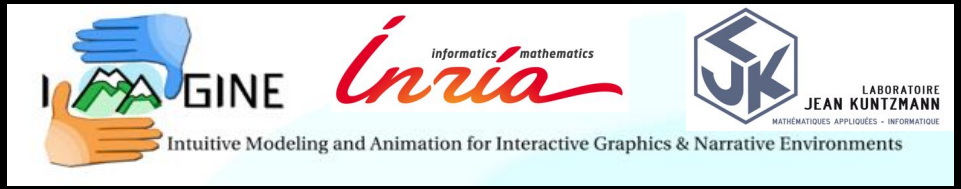
The Prose Storyboard Language
A Tool for Annotating and Directing Movies

Rémi Ronfard

Séminaire Image Numérique, ENS, 11 février 2016



Parcours

- **Thèse en 1991**
- Dassault Systèmes
- Institut National de l'Audiovisuel
- IBM Watson Research Center
- **INRIA en 2002**
- Xtranormal Technologies
- **HDR en 2009**

Vision par ordinateur

Informatique graphique

Analyse automatique de films, des images aux actions (1999-2009)

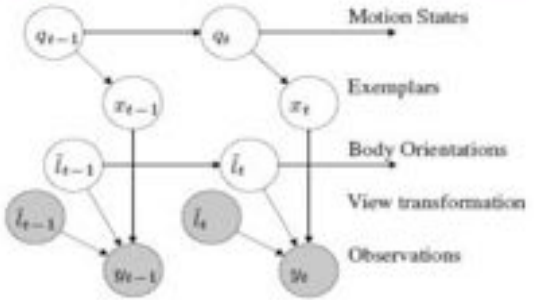
- Modèle structurel des acteurs: parties du corps et relations spatiales

  [ECCV 2002, CVPR 2013]

- Modèle structurel des actions: poses et relations temporelles

  [CVPR 2006, ICCV 2007]

  [CVIU 2006, CVIU 2011]



5

Mettre en scène les mondes virtuels, des actions aux images (2012-2016)

- Mettre la création de films narratifs à la portée de tous — en animation 3D.

Du scénario au storyboard

Du storyboard à la scène

De la scène à l'écran

6

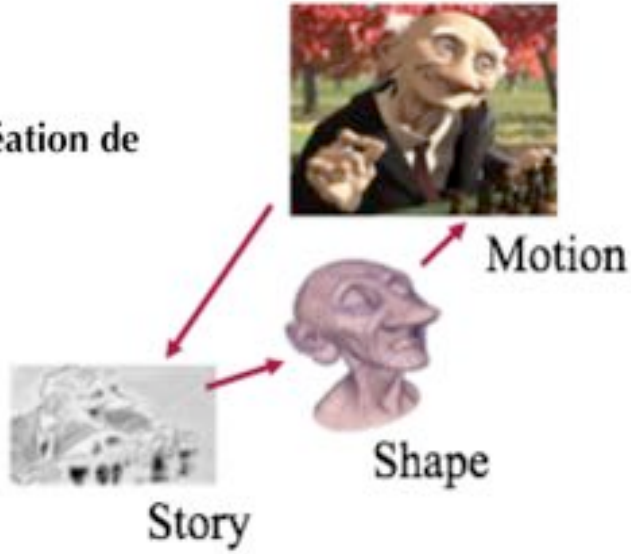
EPI IMAGINE: Création de formes, mouvements et histoires

- **Interaction fructueuse entre création de contenus 3D et mise en scène**

  Acteurs virtuels

  Storyboards interactifs

  Figurines augmentées



11

Part 1

Overview

- INTRODUCTION
- SIMPLE, COMPLEX AND COMPOSITE SHOTS
- FRAME COMPOSITION
- PARSING MOVIES INTO PSL
- GENERATING MOVIES FROM PSL
- CONCLUSION

Introduction

- Motivation
  - Learning cinematography from examples
  - Scoring cinematography, like music
- Cinematography is « writing with motion »
  - Prose storyboard is a short-hand notation
  - Shots are sentences, not words
  - Frame compositions and actions are words

State of the art

- Much work in virtual cinematography deals with one-shots and two-shots
- Little work on three-shots and more
- Little agreement between researchers on how to describe complex shots
  - In time and in space

How to describe this ?

How to describe this ?

How to describe this ?

How to describe this ?

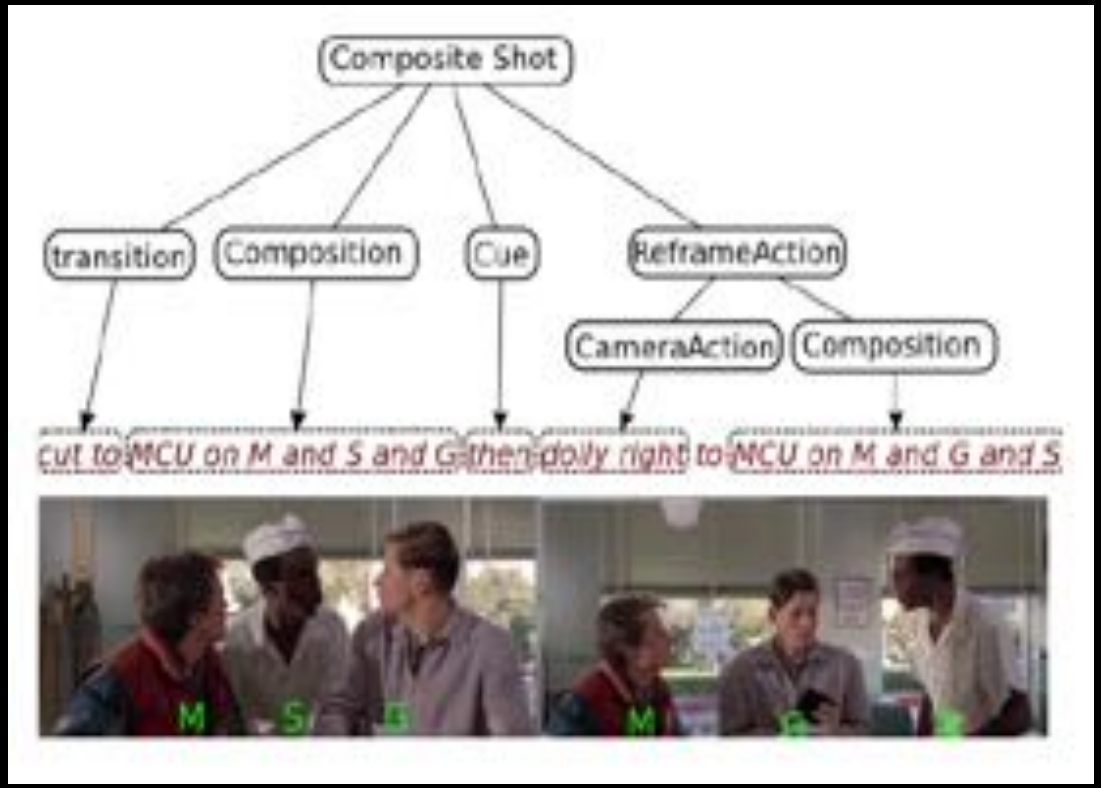
How to describe this?

How to describe this ?

Requirements

- PSL should be intuitive
  - Human readable, pseudo natural language
- PSL should be compact and expressive
  - Simple cases are simply described
  - Can be used to notate all movies
- PSL should be a notation
  - machine-readable, can be compiled and executed

Proposed solution : prose storyboard



References

- Van Rijsselbergen et al. Movie script markup language. ACM symposium on Document engineering, 2009.
- Proferes, Film directing fundamentals, See your film before shooting it, Focal Press 2008.
- Thomson, Grammar of the Shot, Focal Press, 1988.

Simple shots

- Camera does not move or rotate or zoom
- Frame composition changes only when actors move in frame
- Prose storyboard notation : frame composition + actor movement

Complex shots

- Camera does not move
- Frame composition changes
  - Due to camera pan, tilt and zoom
  - Due to actor movement
  - Or both
- Prose storyboard notation : frame composition + actor movement + camera rotation

Composite shots

- Camera and actors move freely
- Frame composition changes
  - Due to actors in the frame
  - Due to camera motion
  - Or both
- Prose storyboard notation : frame composition + actor movement + camera movement

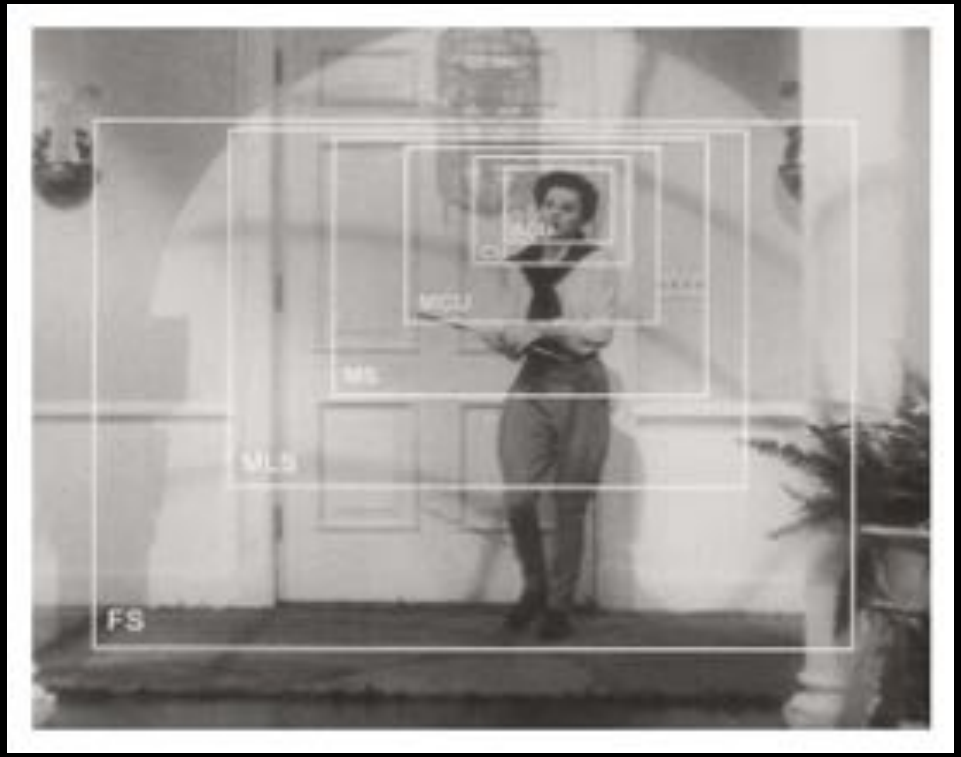
Frame composition in PSL

- From left to right, back to front

```
<Composition>      ::= [<angle>] {<FlatComposition>}+

<FlatComposition>::= <size>  on <Subject>[ <profile>][ <screen>]
        { and <Subject>[ <profile>][ <screen>][in (back|fore)ground]}*
```

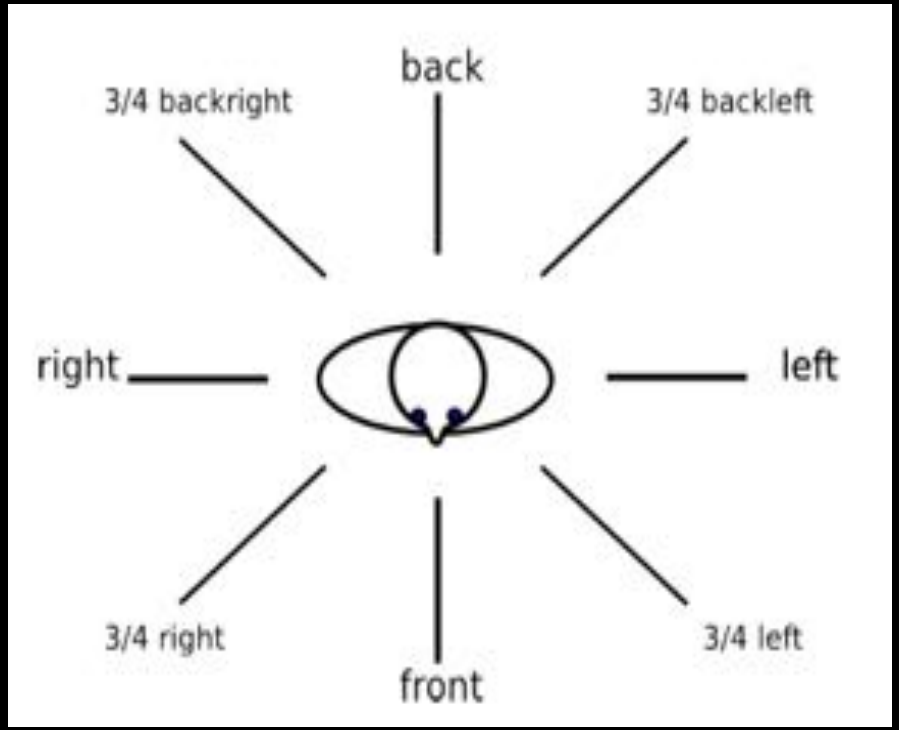
```
<angle>        ::= (high|low)angle

<size>         ::= ECU|BCU|CU|MCU|MS|MLS|FS|LS|ELS

<profile>      ::= 34leftback|left|34left|front|34right|right|34leftback|back

<screen>       ::= screen(center|left|right)
```

Frame composition : Size

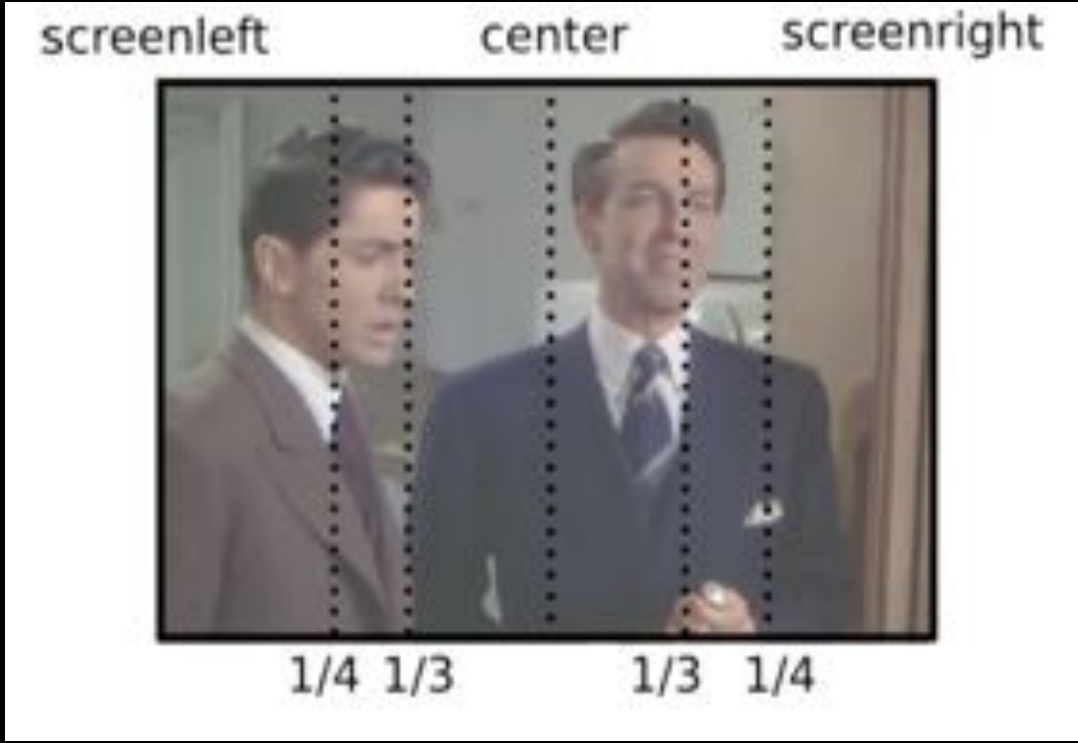


B. Salt. Moving Into Pictures. Starword, 2006.

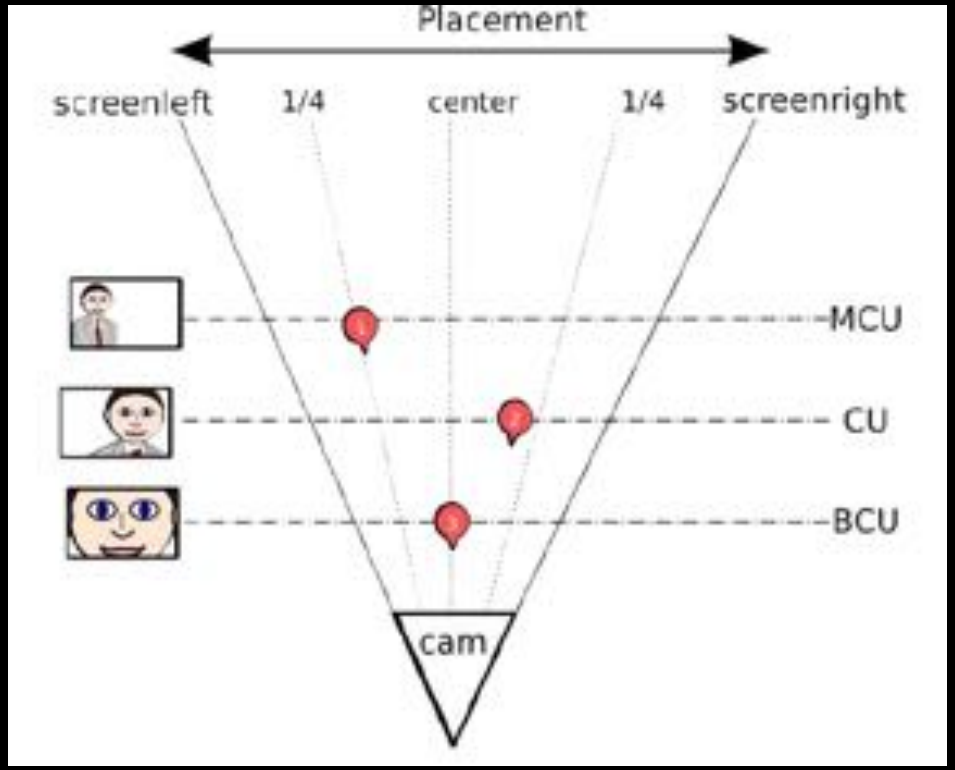
Elements of composition : Profile angle



Elements of composition : Screen position



Cinematographic staging



BCU Girl 34left (screen center)

MCU Marianne (front screen center) and BCU Cissors in foreground

MS Girl 34back right screen left and Ferdinand (front screen right)

FS Cyd right (screen left) left and Fred left (screen right)

MLS Father screen left and ELS Kane in background and MS Man and Mother in foreground screen right

Shot in PSL

- Frame compositions, cues and actions

```
<Shot>  ::= [<transition>] to [<Camera>] <Composition> {<Fragment>}*
```

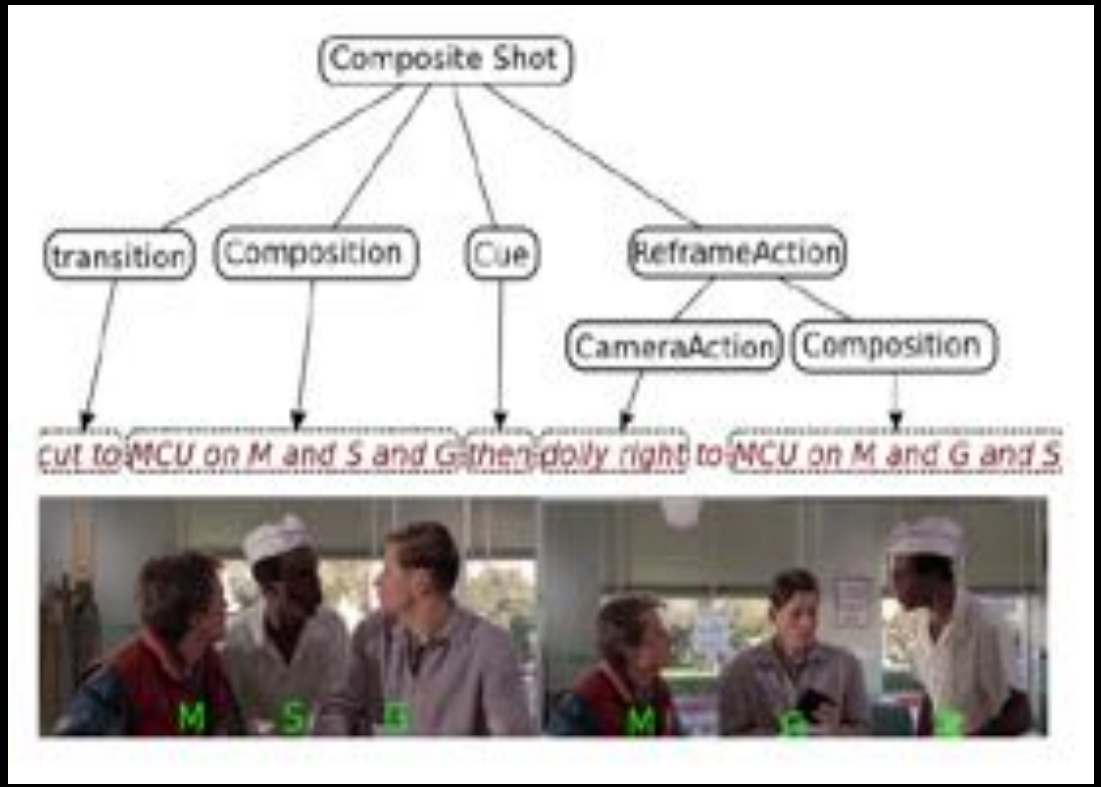
```
<Fragment>      ::= <Cue> (<RenameAction> | <ReframeAction>)
```

```
<Cue>   ::= At <timeref> | As <Actor> <Action> | then
```

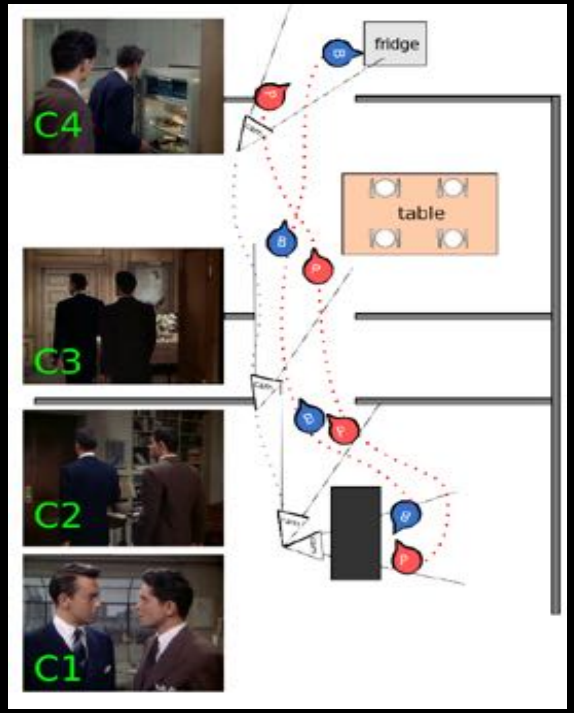
```
<RenameAction>  ::= (lock|continue) to <Composition>
<ReframeAction> ::= <CameraAction> (to|with) <Composition>
```

```
<CameraAction>::= [Speed]  pan [left|right|up|down]
                         | dolly [in|out|left|right]
                         | crane [up|down]
```

Example : back to the future



Examples from Hitchcock's Rope (1948)



First sequence


MCU on Brandon and Philip


As they start moving
slow pan left to ...


MS on Brandon,Philip 3/4leftback


then dolly with MS on Brandon,Philip
3/4leftback


As Philip crosses left over Brandon
continue to ...

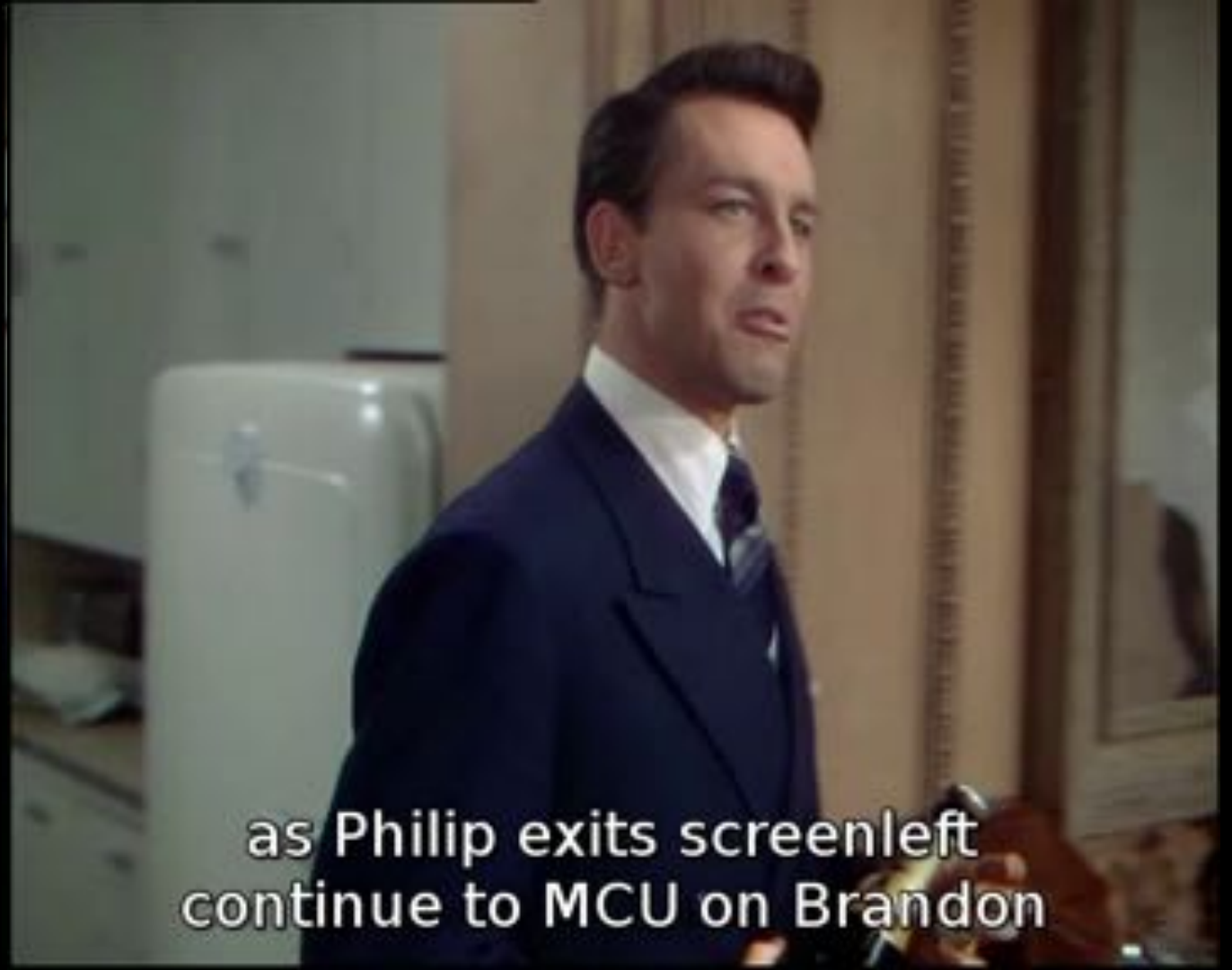

MS on Philip,Brandon 3/4leftback


then dolly in to ...


MCU on Philip and Brandon
screencenter in BG


as Brandon starts moving screenright
dolly right to ...


MCU on Brandon and Philip


as Philip exits screenleft
continue to MCU on Brandon


then dolly out to MS on Brandon

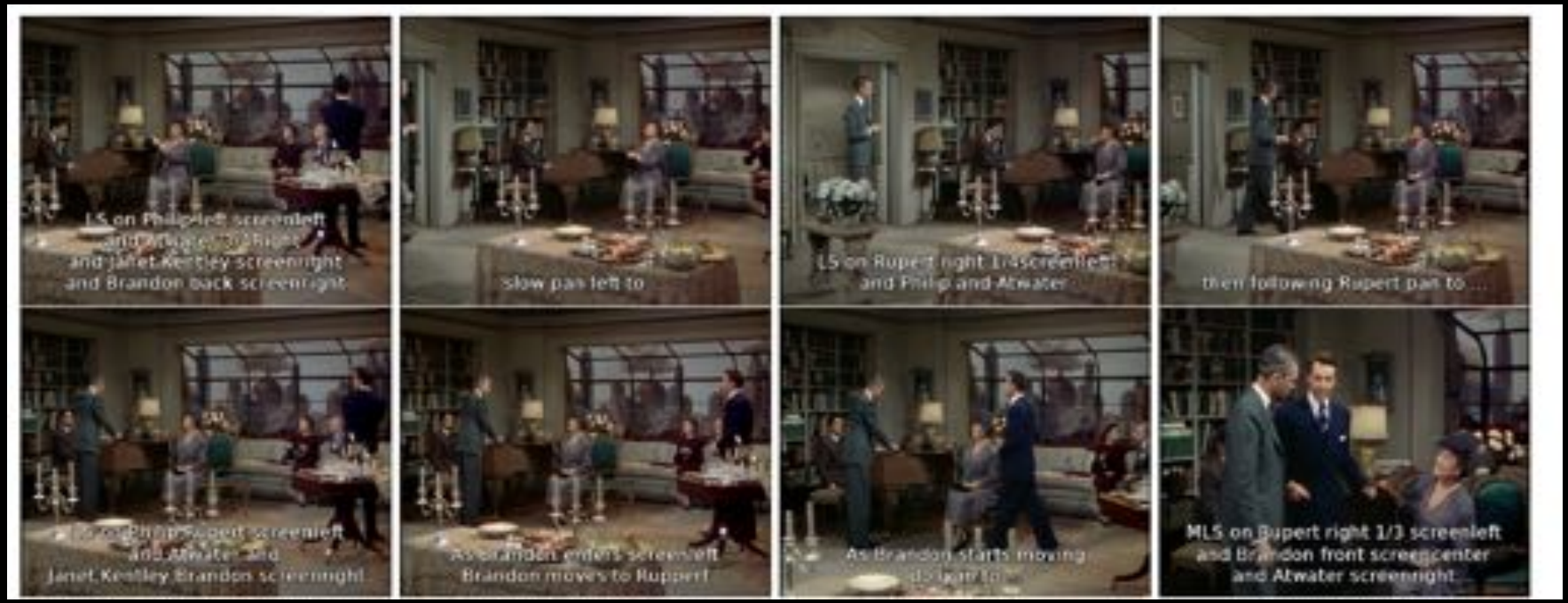

as Philip enters screenleft
continue to ...


MS on Philip and Brandon


as Philip touche glass
slow dolly in to ...


MCU on Brandon and Philip

Second sequence




LS on Philip left screenleft
and Atwater 3/4Right
and Janet,Kentley screenright
and Brandon back screenright


slow pan left to ...


LS on Rupert right 1/4screenleft
and Philip and Atwater


then following Rupert pan to ...


LS on Philip,Rupert screenleft
and Atwater and
Janet,Kentley,Brandon screenright


As Brandon enters screenleft
Brandon moves to Ruppert


As Brandon starts moving
dolly in to ...


MLS on Rupert right 1/3 screenleft
and Brandon front screencenter
and Atwater screenright


as Rupert starts moving screenright
dolly right to ...


MS on Brandon,Rupert right screenleft
and Janet and Kentley left 1/3
screenleft


as Rupert starts moving to Kenneth
dolly left to ...


MCU on Rupert,Kenneth facing

Parsing Movies into PSL

- Manual annotation
  - Parse annotation to PSL
- Automatic analysis
  - Actor detection and identification
    - Detecting and Naming Actors in Movies using Generative Appearance Models, CVPR 2013
  - Event recognition – ENTER, EXIT, CROSS, SIT, STAND, etc.
  - Synchronisation with Stochastic Timed Petri Nets

Generating movies from PSL

- Frame composition
  - Actor sizes, screen positions and profile angles from stylesheet produce a sketch storyboard
  - Relative camera and actor positions can be computed using « through the lens » camera control
- Camera and actor movements
  - Trajectories interpolating camera and actor positions
  - Timed Petri Nets for synchronizing camera and actors

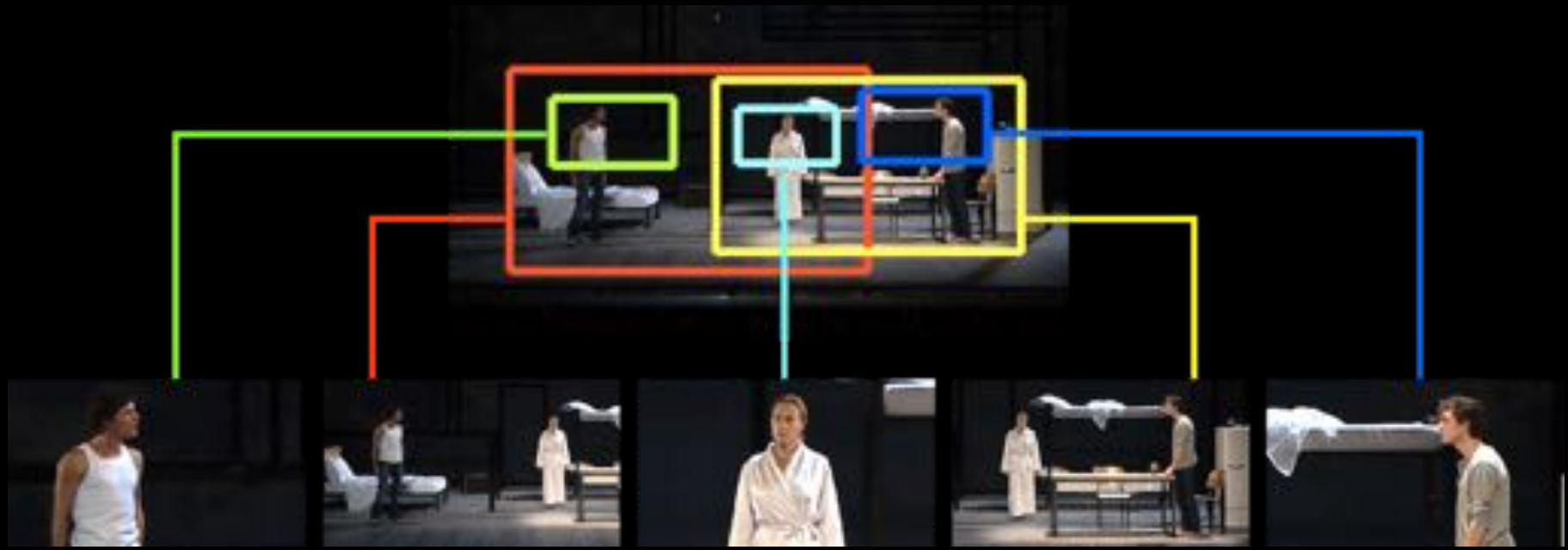
Horizontal and vertical editing

Up until now, motion picture editors have thought almost exclusively in the horizontal direction. The question to be answered was simply, Whats next?

In the future, that number is going to become even more cosmic because film editors will have to start thinking vertically as well, which is to say: What can I edit within the frame?

Walter Murch, In the blink of an eye, 2001.

Application to theatre performance



Content-Aware Keyframe Interpolation for Vertical Video Editing

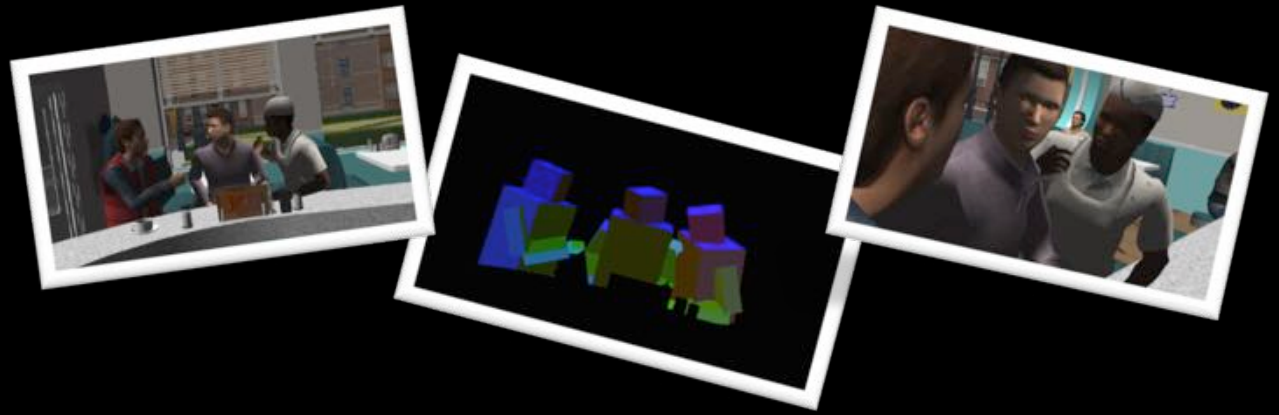
Submission id:

Conclusion

- PSL is a « pseudo-natural » language for cinematography and editing
  - Transcribe movies
  - Control virtual cameras and edit movies
  - Auto-annotate virtual movies
  - Share and compare movies
- Future work will add sets, lighting and audio

Part 2

Continuity Editing for 3D Animation



Twenty-Ninth AAAI Conference
2015

Quentin Galvane
Rémi Ronfard
Christophe Lino
Marc Christie

Motivation

- Grand Challenge: Automated film-making

  (Mark Riedl AAAI – WICED 2014)

- Specialized agents with cinematographic knowledge
  - Director, actor, cameraman, film-editor
- With applications to games

Objectives

- Read actions and dialogues from script
- Generate 3D animation
- Place cameras and lights, generate rushes
- Edit the rushes into a movie

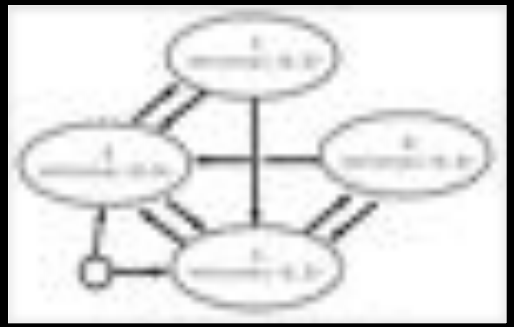
Continuity errors

left-to-right ordering Jump Cut screen position gaze direction

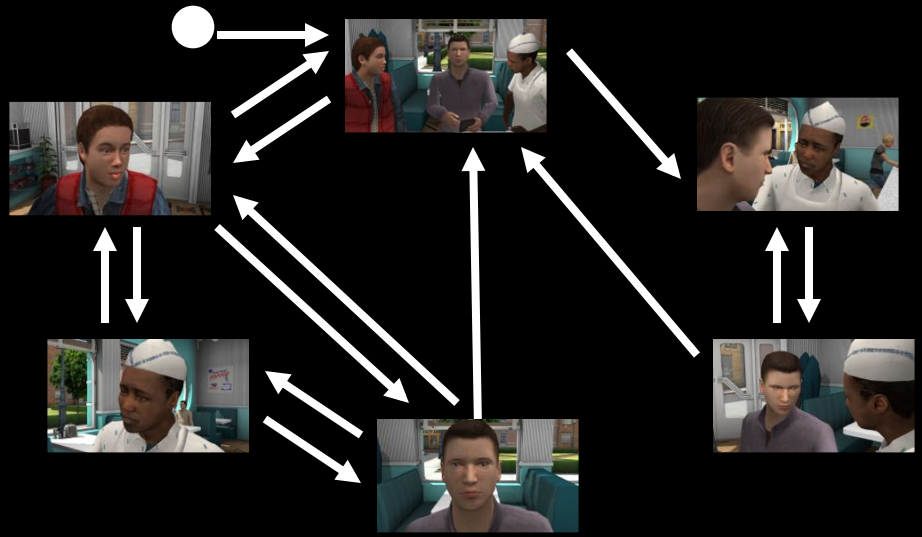
Non-motivated shots and cuts

Our solution

Idiom based solutions

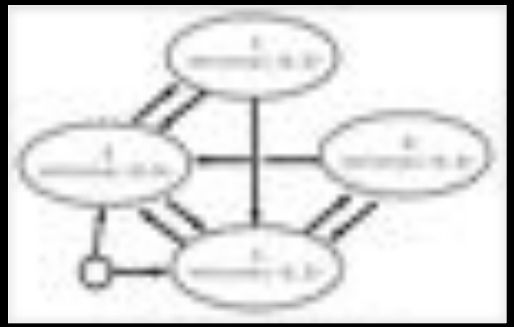


Virtual cinematographer
[Christianson et al. 1996]

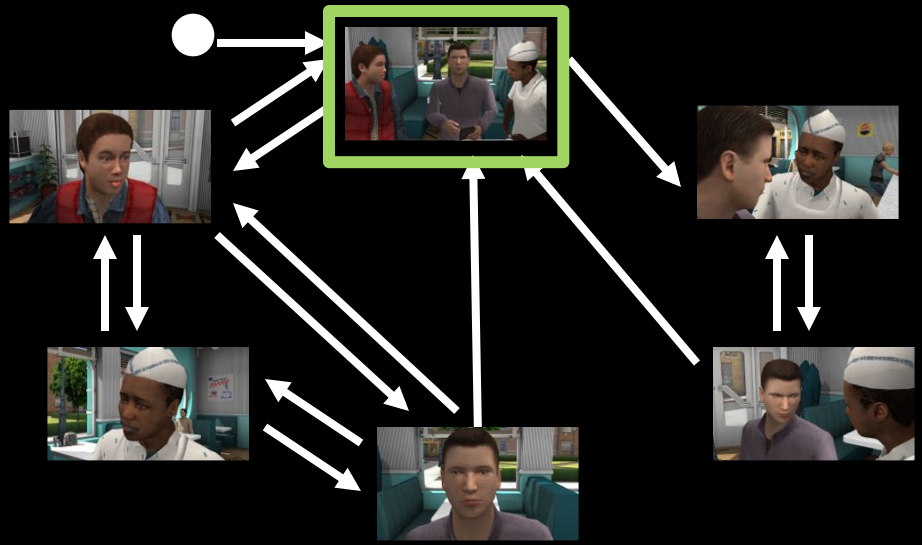


Scenario

Idiom based solutions

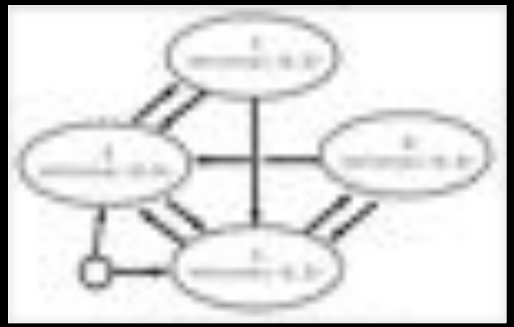


Virtual cinematographer
[Christianson et al. 1996]

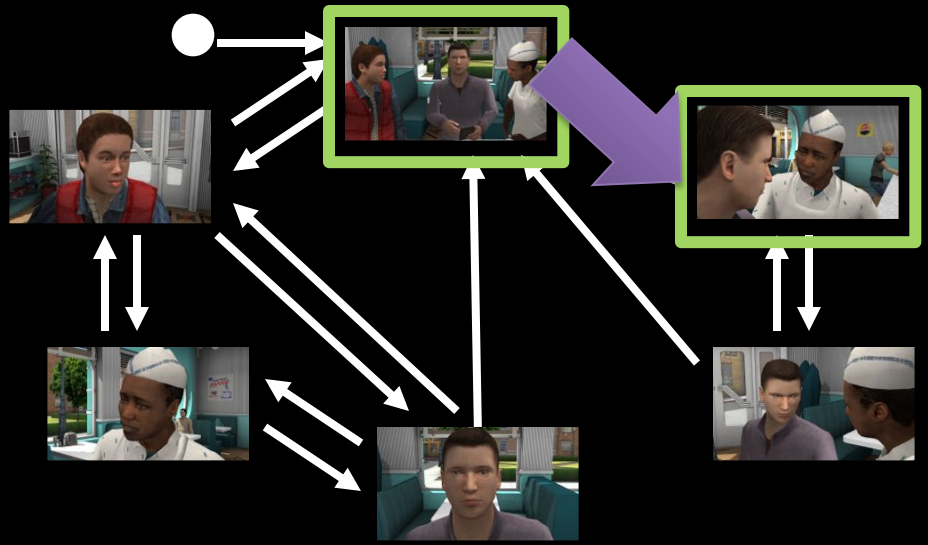


Scenario

Idiom based solutions

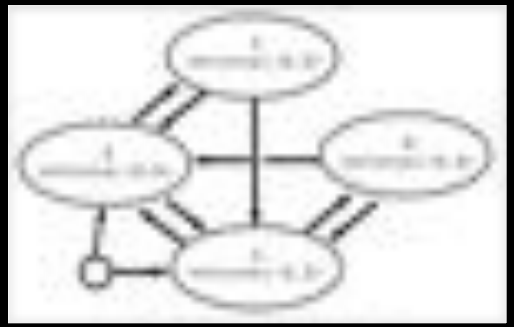


Virtual cinematographer
[Christianson et al. 1996]

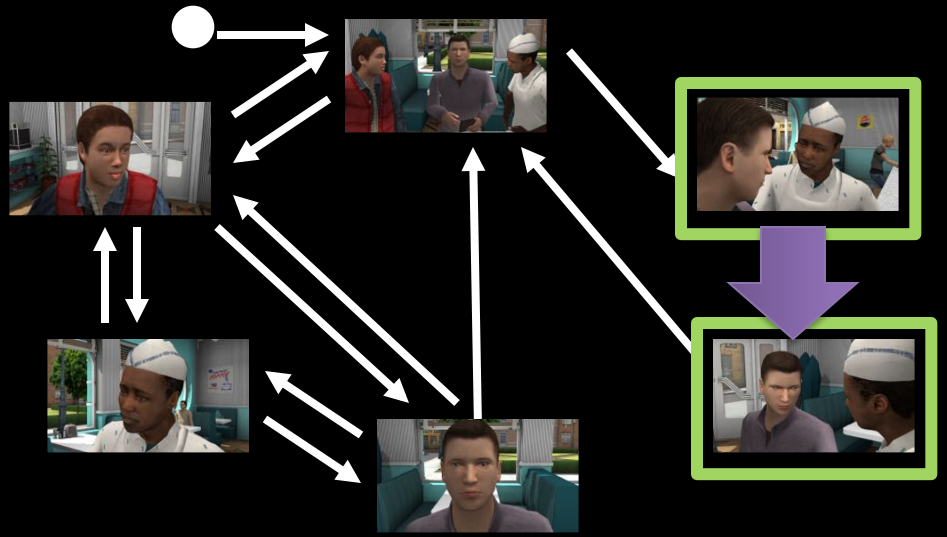


Scenario

Idiom based solutions

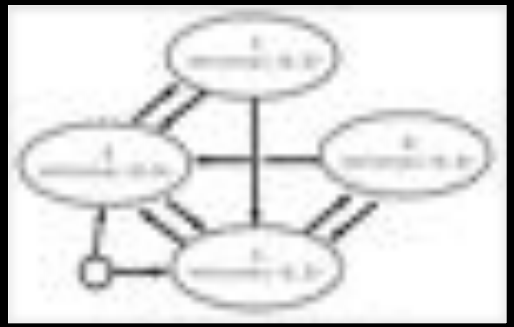


Virtual cinematographer
[Christianson et al. 1996]

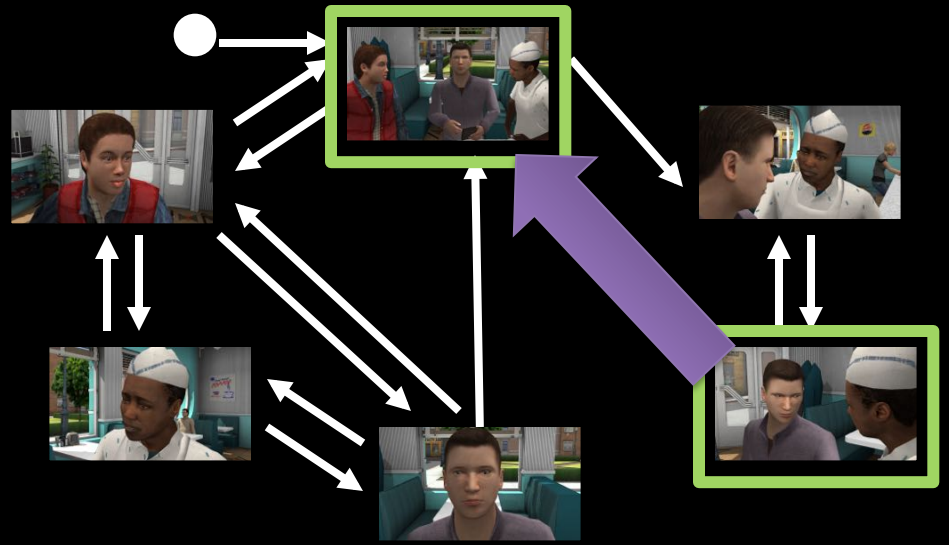


Scenario

Idiom based solutions

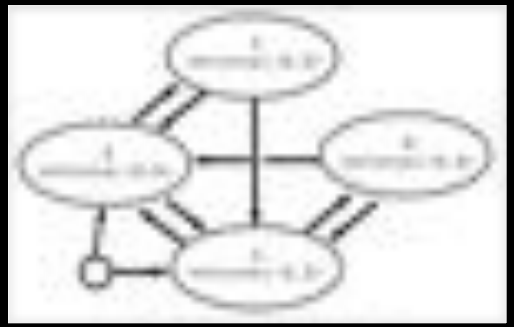


Virtual cinematographer
[Christianson et al. 1996]

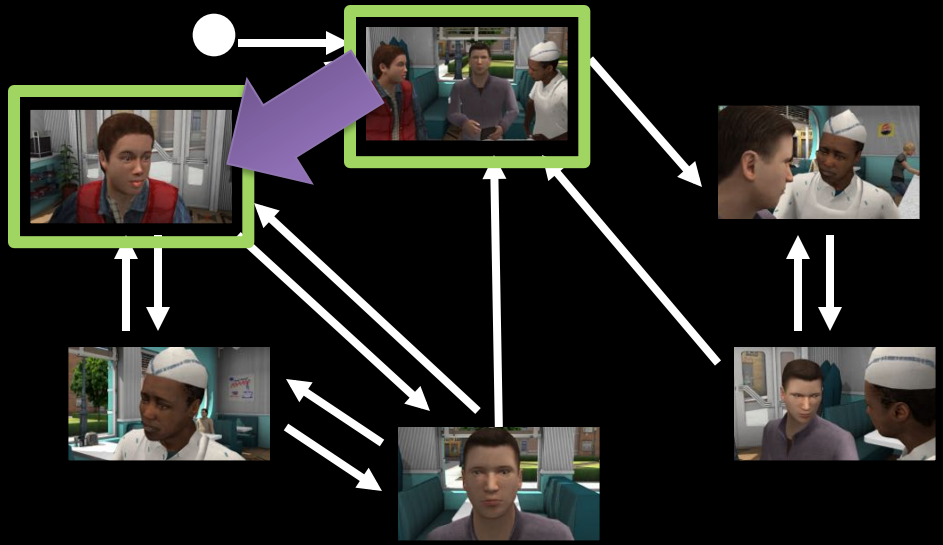


Scenario

Idiom based solutions



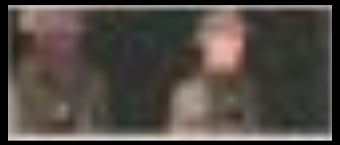
Virtual cinematographer
[Christianson et al. 1996]



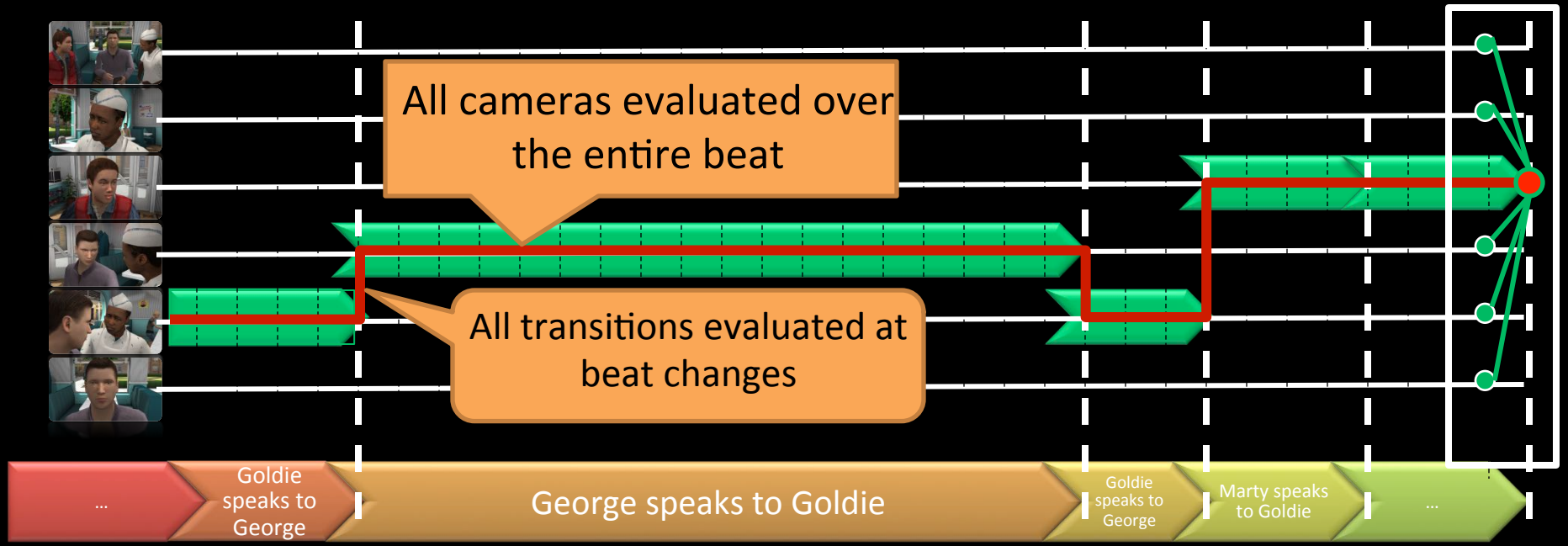
Scenario

Optimization based approach

Dynamic programming

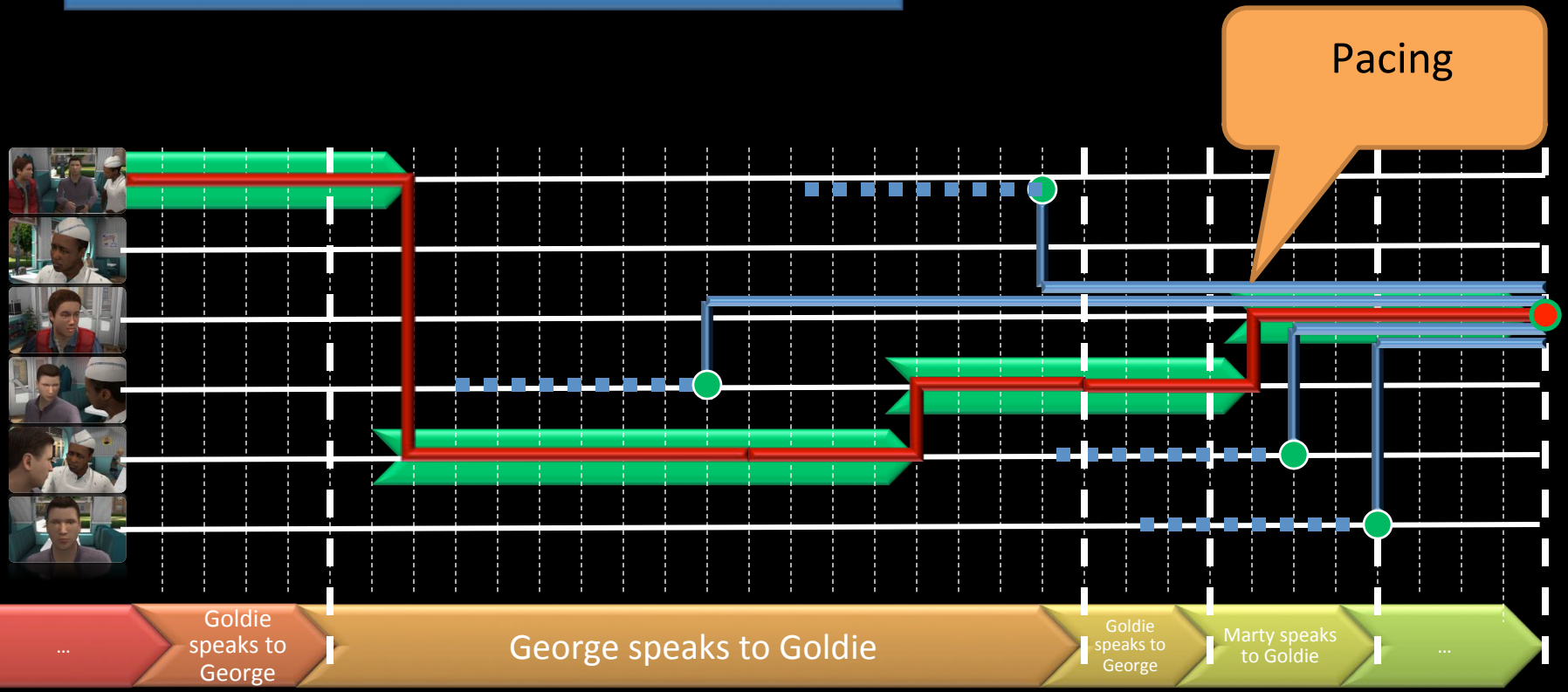


[Riedl, M. et al., 2008]



Scenario

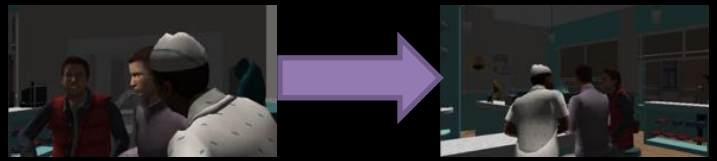
Evaluate all possible transitions



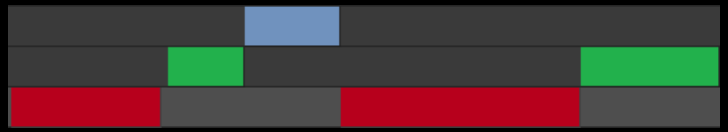
Scenario

- Film editing as an optimization problem
  - Semi-Markov chains
- Create an editing graph that evaluates 3 aspects:
  - Shot quality

  - Cut quality



  - Pacing



Film editing as optimization

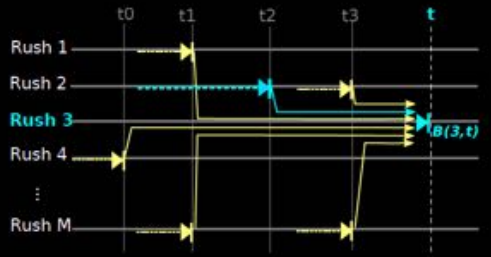
- Search over semi-Markov chains s = (r_j, d_j) given actions a(t)
- Minimize cost function:

$$C(s,a) = \sum_j \sum_{t_j \le t \le t_j + d_j} C^A(r_j, t) + \sum_{1 \le j} C^T(r_{j-1}, r_j, t_j) + \sum_j C^R(d_j)$$

Action cost (Shot quality)

Transition cost (Cut quality)

Rhythm cost (Pacing Quality)



The final editing is given by the shortest path in the editing graph

➢ Shot quality:

- Hitchcock principle

The size of a character on the screen should be proportional to its narrative importance in the story.

- Narrative importance from script
- Visible area V = S – O for each rush

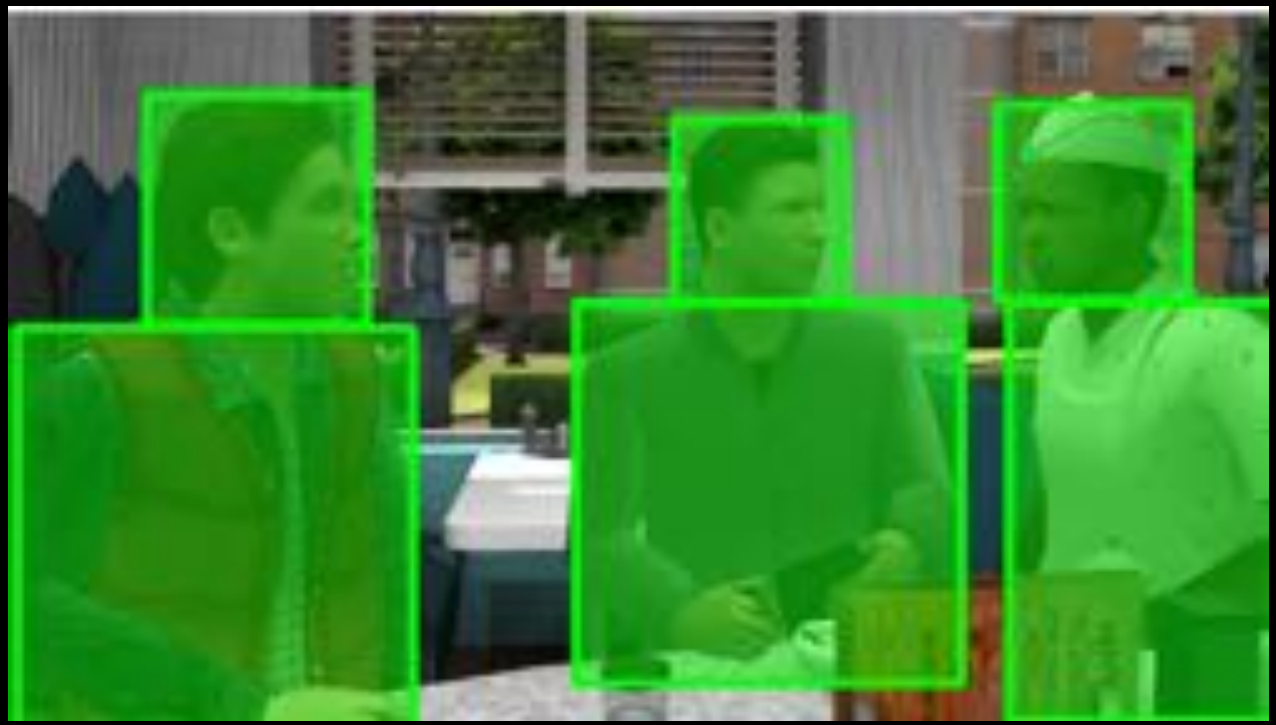
$$C_H^A(r,t) = \sum_{c \in C} \left| \frac{I(c,t)}{\sum_{c'} I(c',t)} - \frac{V(c,r,t)}{\sum_{c'} V(c',r,t)} \right|$$

Actors and actions

High action visibility

Low action visibility

High action proximity



Low action proximity

Look room

Head room

Screen continuity

Screen continuity

Screen discontinuity

Screen discontinuity

Motion continuity

Motion continuity

Motion discontinuity

Motion discontinuity

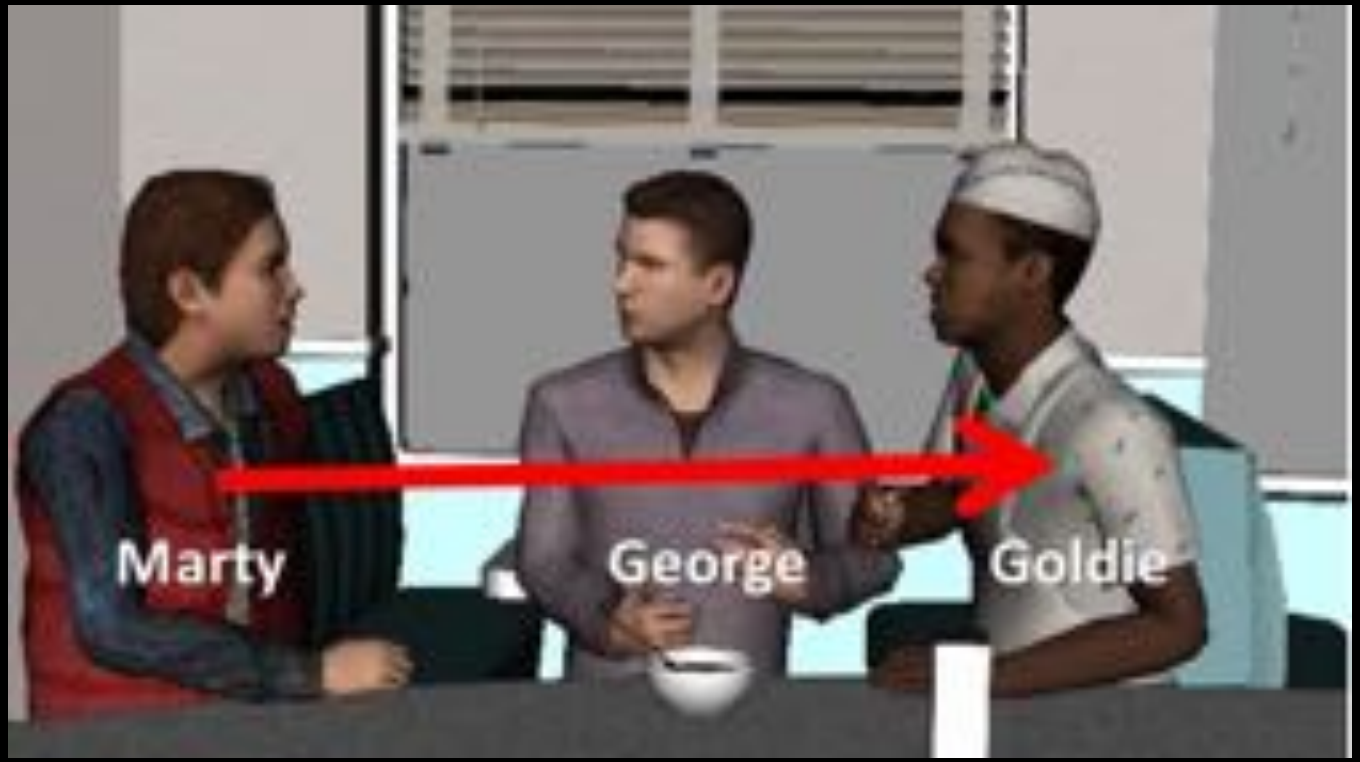
Gaze continuity

Gaze continuity

Gaze discontinuity

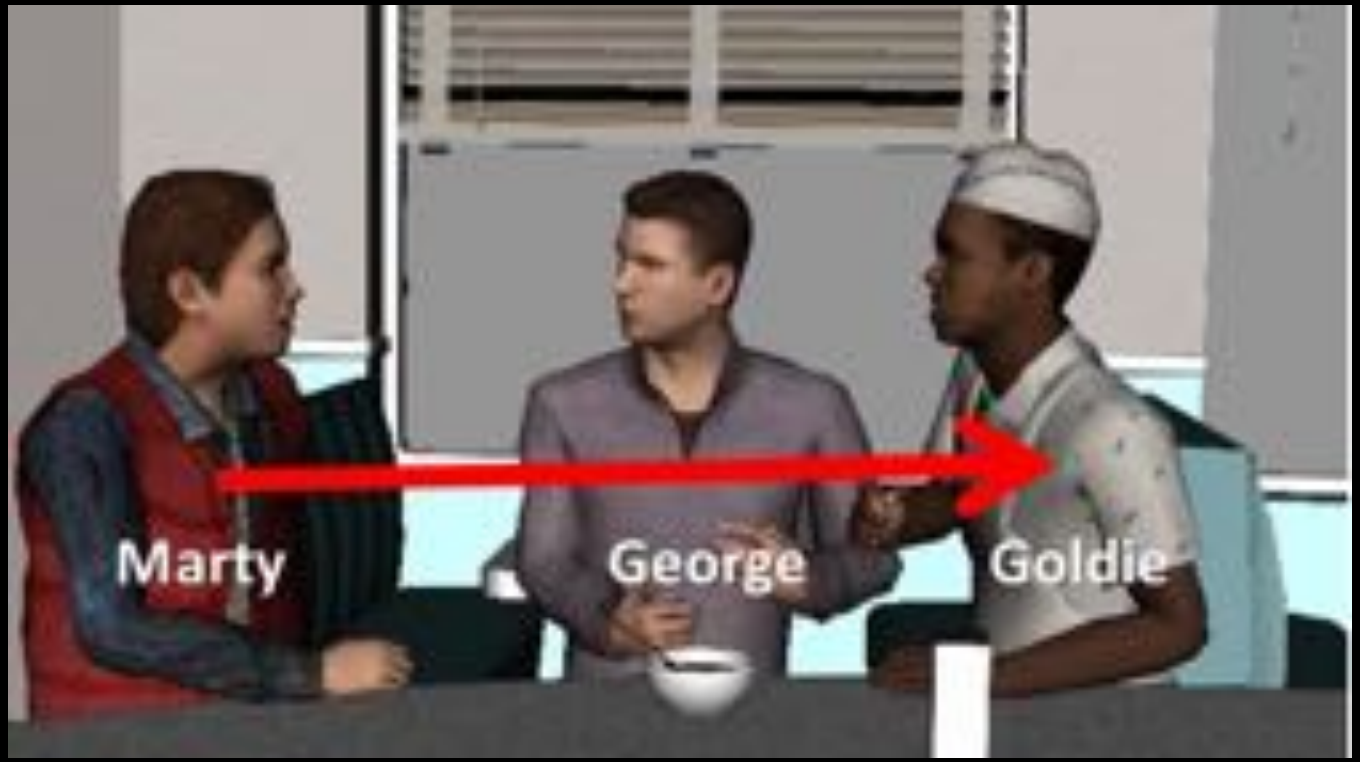
Gaze discontinuity

Left-to-right continuity



Left-to-right continuity

Left-to-right discontinuity



Left-to-right discontinuity

Jump cuts

Jump cut

Not a jump cut

Not a jump cut

Results

Input

Computed Editing

Camera : 6

Last shot cost: 13.50s
Value: 0.43

Last cut from 16 to 6 at 7.54s
Value: 0.03

Pacing from 7.54s to 16.50s (8.96s)
Value: 0.06

Results

- 3 different versions edited:
  - Original pacing
  - Slower pacing
  - Faster pacing

- User study: confirms that all 3 terms are important
  - Shot quality, cut quality and pacing

Limitations

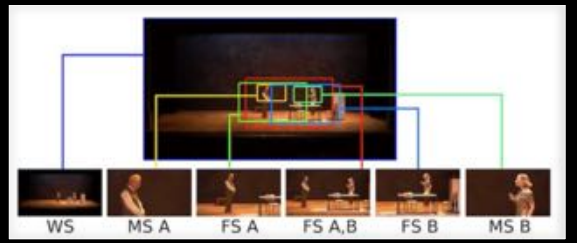
- Cameras must be pre-computed
- Cannot handle book-ending
  - Context free grammar
- Cannot handle ellipsis or flashbacks

Future work

- Optimize over camera positions and movements

[Galvane et al., 2014]

- Extend to real live video



[Gandhi et al., 2014]

- Learn other editing styles from real movies

Conclusion

- Computational model for film editing
- Incorporates knowledge about shot composition and film grammar
- Efficient offline algorithm
- Suitable for automatic film making

Dataset and results:

- *https://team.inria.fr/imagine/continuity-editing/*

Continuity editing

Back to the Future

R.Zemeckis (1985)

Thank you !