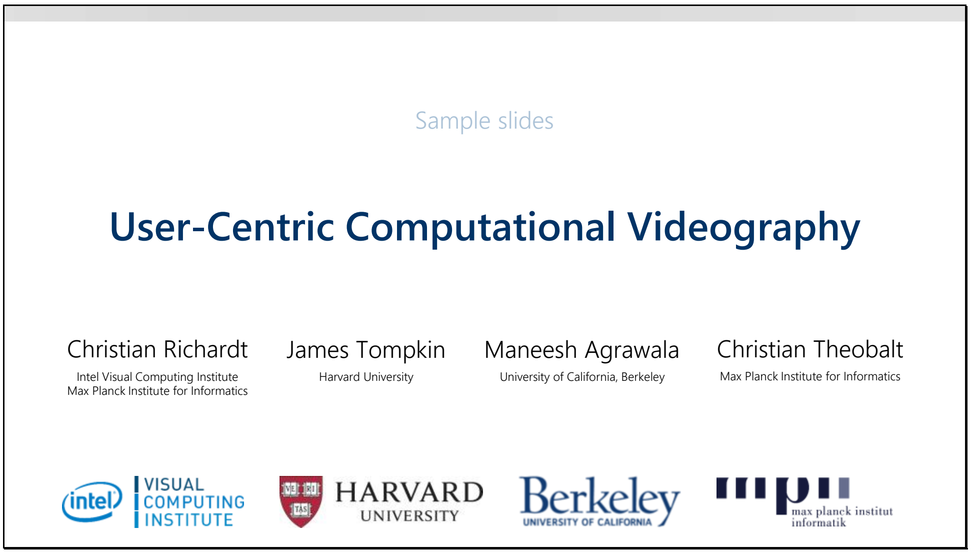
Slide 1



Sample slides

User-Centric Computational Videography

Christian Richardt
Intel Visual Computing Institute
Max Planck Institute for Informatics

James Tompkin
Harvard University

Maneesh Agrawala
University of California, Berkeley

Christian Theobalt
Max Planck Institute for Informatics

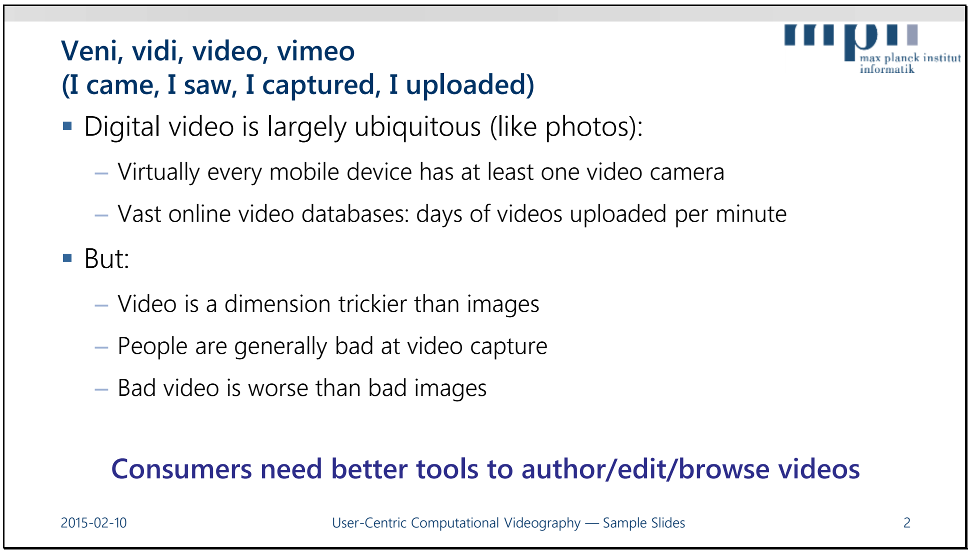
These are sample slides from our course submission #142 to SIGGRAPH 2015 on User-Centric Computational Videography.

The slides cover three parts of our proposed course:
(1) the introduction of our course which motivates the need for user-centric video tools,
(2) some of the background materials on existing video editing tools, and
(3) part of a technical section on model-free video editing.

Together, these slides represent about 10% of our proposed 3.25 hour course.

You can find these slides in different formats (including videos) on our course website:
http://gvv.mpi-inf.mpg.de/teaching/uccv_course_2015/

Slide 2



Veni, vidi, video, vimeo
(I came, I saw, I captured, I uploaded)

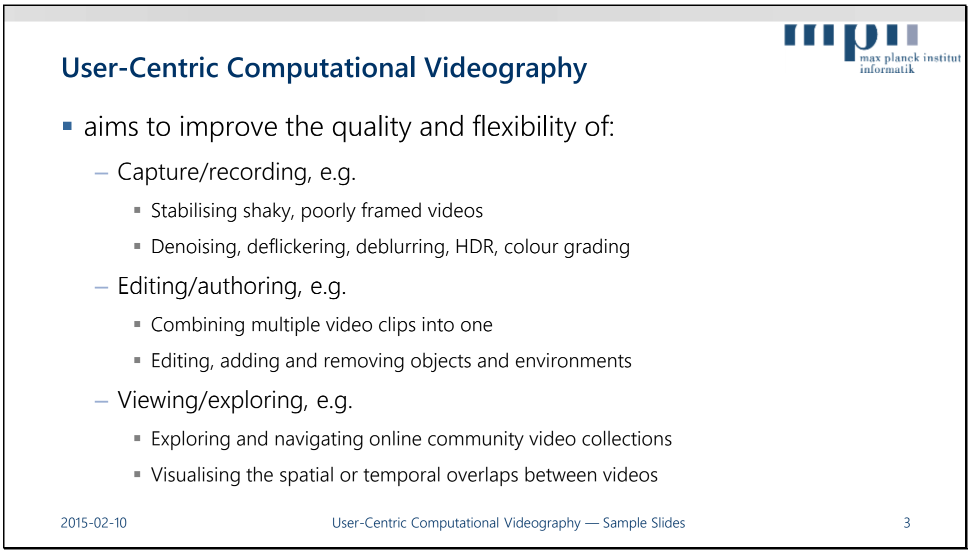
- Digital video is largely ubiquitous (like photos):
  - Virtually every mobile device has at least one video camera
  - Vast online video databases: days of videos uploaded per minute
- But:
  - Video is a dimension trickier than images
  - People are generally bad at video capture
  - Bad video is worse than bad images

Consumers need better tools to author/edit/browse videos

2015-02-10 User-Centric Computational Videography — Sample Slides 2

[Presenter: Christian Richardt]
Let me start by adapting one of Julius Caesar's most famous quotes to the current day: veni, vidi, video, vimeo – I came, I saw, I captured and I uploaded.
What this is hinting at is that digital video is becoming more and more ubiquitous, just like digital photos have been over the last few years.
Most mobile devices now come with at least one high-resolution video camera, and fast mobile internet enables the upload of 100s of hours of video to online video communities such as YouTube, Google+ and Vine, every minute.
However, video is harder than images, mostly because of the added time dimension. Video is harder to capture as handheld mobile devices often produce shaky and wobbly footage. Video is also harder to get right, as bad video is generally worse than bad images. And video is harder to edit despite a huge range of consumer and professional video editing tools.
To us, it is clear that consumers clearly need better tools for authoring, editing and browsing videos.
In this course, we look at the progress made so far on this topic, and discuss current trends in the software industry as well as in research, and end by proposing directions for future research.

Slide 3



User-Centric Computational Videography

- aims to improve the quality and flexibility of:
  - Capture/recording, e.g.
    - Stabilising shaky, poorly framed videos
    - Denoising, deflickering, deblurring, HDR, colour grading
  - Editing/authoring, e.g.
    - Combining multiple video clips into one
    - Editing, adding and removing objects and environments
  - Viewing/exploring, e.g.
    - Exploring and navigating online community video collections
    - Visualising the spatial or temporal overlaps between videos

2015-02-10 User-Centric Computational Videography — Sample Slides 3

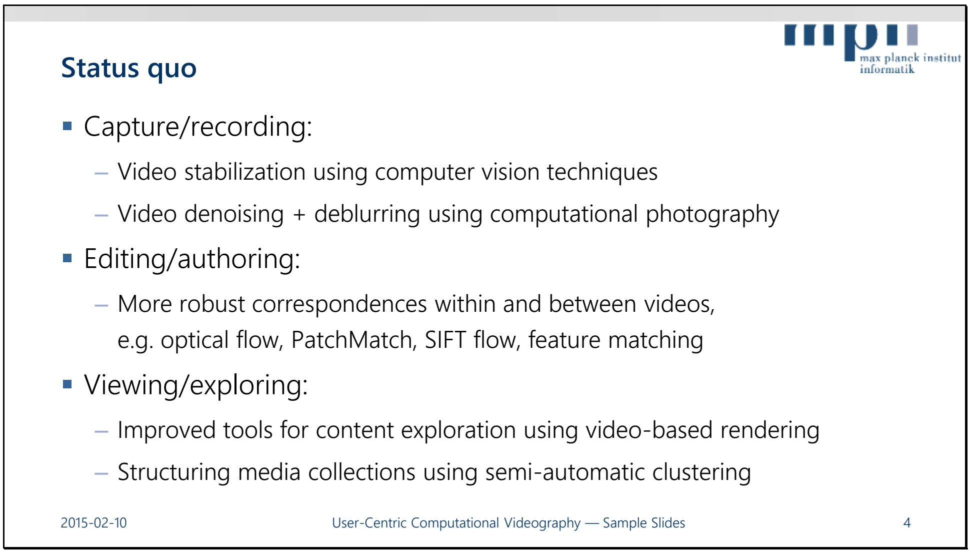
This is what we understand by the term "User-Centric Computational Videography". It covers all aspects of handling consumer videos, from their capture with handheld digital video cameras, over video editing and authoring, all the way to how to make large collections of videos easily explorable. Let's look at what users need in terms of each of these activities.

In terms of video capture, the most important task for software tools is to get the most out of the recorded footage, to stabilise shaky and wobbly videos, improve the framing of videos if possible, denoise, deflicker and deblur them, and perhaps reconstruct high-dynamic-range video, or fix the captured colours. In our course, we consider these things as necessary preprocessing tasks, which we will briefly cover to provide some background, but we will mostly focus on the other two areas.

Video editing and authoring covers all aspects of creating a new video, usually by combing multiple video clips, photos and music. The long-term goal of users is full flexibility in their edits: they want to edit objects directly by cutting and pasting them in a different place and perhaps time, and they want to change the look of videos by modifying the environments and lighting in existing videos.

Increasingly important becomes the task of how videos can be viewed and explored. Online video communities comprise millions of videos, but the only way to search them is using textual keywords, instead of visualising the spatial or temporal overlaps between videos.

Slide 4



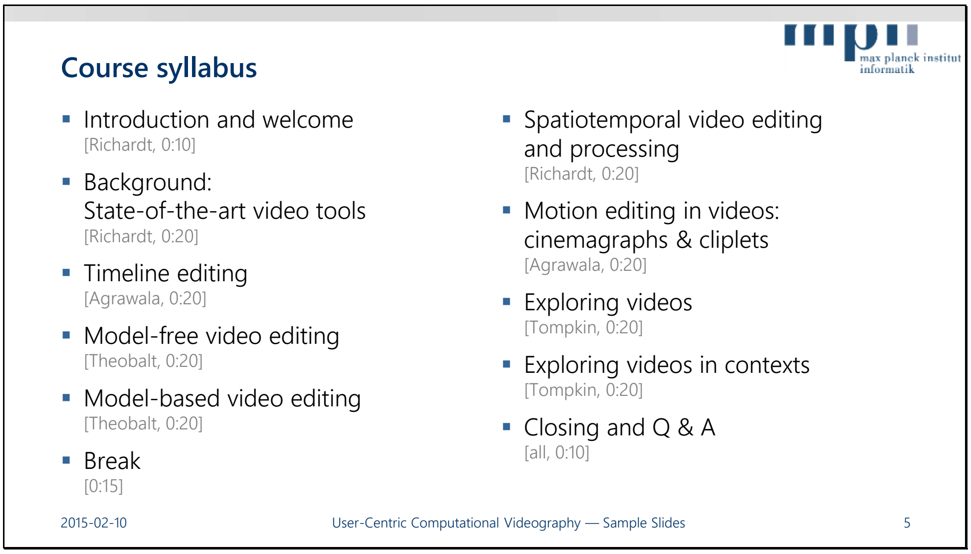
Status quo

- Capture/recording:
  - Video stabilization using computer vision techniques
  - Video denoising + deblurring using computational photography
- Editing/authoring:
  - More robust correspondences within and between videos, e.g. optical flow, PatchMatch, SIFT flow, feature matching
- Viewing/exploring:
  - Improved tools for content exploration using video-based rendering
  - Structuring media collections using semi-automatic clustering

2015-02-10 User-Centric Computational Videography — Sample Slides 4

So where are we as a community relative to these goals?
Video capture has become more flexible as more robust video stabilisation techniques have found their way into consumer devices, and computational photography techniques for denoising and deblurring videos will probably soon follow them.
Video editing has benefited from improved and more robust correspondences techniques that are applied within and between videos, such as […], which allow the exploitation of content relationships for new empowering video experiences.
And video exploring stands to be revolutionized by novel tools for interactive content exploration that are based on automatically structured media collections.

Slide 5



Course syllabus

- Introduction and welcome
  [Richardt, 0:10]
- Background:
  State-of-the-art video tools
  [Richardt, 0:20]
- Timeline editing
  [Agrawala, 0:20]
- Model-free video editing
  [Theobalt, 0:20]
- Model-based video editing
  [Theobalt, 0:20]
- Break
  [0:15]
- Spatiotemporal video editing and processing
  [Richardt, 0:20]
- Motion editing in videos: cinemagraphs & cliplets
  [Agrawala, 0:20]
- Exploring videos
  [Tompkin, 0:20]
- Exploring videos in contexts
  [Tompkin, 0:20]
- Closing and Q & A
  [all, 0:10]

2015-02-10 User-Centric Computational Videography — Sample Slides 5

In this course, we are covering the following topics over the next 3 hours.
We start with a look at existing state-of-the-art video editing tools, what functionality they provide, what is missing and how work on community photo collections could be extended to community video collections.
Maneesh then dives into timeline editing, which is concerned with the temporal arrangement of videos, their synchronisation, alignment and summarisation.
After that, Christian Theobalt discusses model-free and model-based video editing techniques, that range from video cut-and-paste to inverse rendering for manipulation of 3D objects and environments.
We'll then have a short break of 15 minutes, to get up and stretch, before continuing with spatiotemporal video editing and processing. This part covers advanced video transitions, virtual cinematography, hyperlapses and more.
Next, Maneesh will talk about motion editing in videos in the form of cinemagraphs and cliplets.
After that, James will give an overview of existing video exploration tools that allow for browsing videos or collections of video by themselves, or within the context of other videos.
We will then close with an outlook of what is to come, and a final Q&A session.

Slide 6

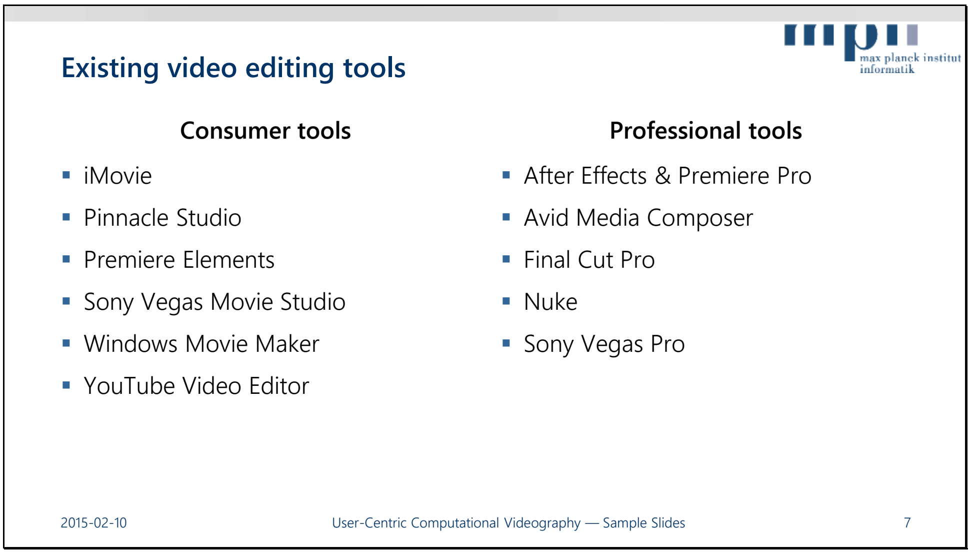
Christian Richardt

Background: State-of-the-Art Video Tools

2015-02-10 User-Centric Computational Videography — Sample Slides 6

Let’s start by looking into the existing tools for video editing and processing.

Slide 7



Existing video editing tools

Consumer tools

- iMovie
- Pinnacle Studio
- Premiere Elements
- Sony Vegas Movie Studio
- Windows Movie Maker
- YouTube Video Editor

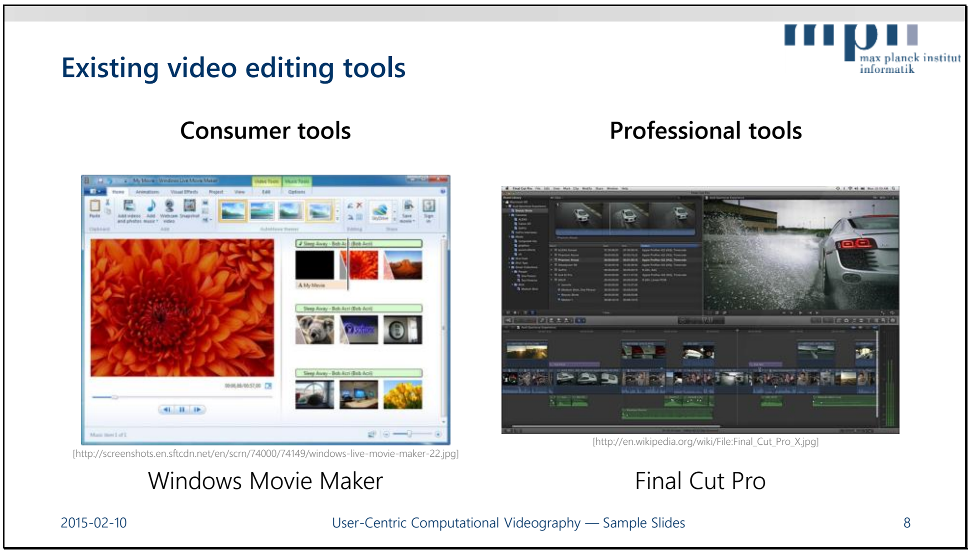
Professional tools

- After Effects & Premiere Pro
- Avid Media Composer
- Final Cut Pro
- Nuke
- Sony Vegas Pro

2015-02-10 User-Centric Computational Videography — Sample Slides 7

There are quite a few software tools for video editing available, which can be broadly categorised into consumer and professional or prosumer tools. I am sure you will have heard of many of these tools, and most likely also used one or the other of them for your personal videos or at work. And while these tools cover a huge spectrum of functionality, and also price, they do have some common basic functionality.

Slide 8



Existing video editing tools

Consumer tools

Professional tools

[http://screenshots.en.sftcdn.net/en/scrn/74000/74149/windows-live-movie-maker-22.jpg]

Windows Movie Maker

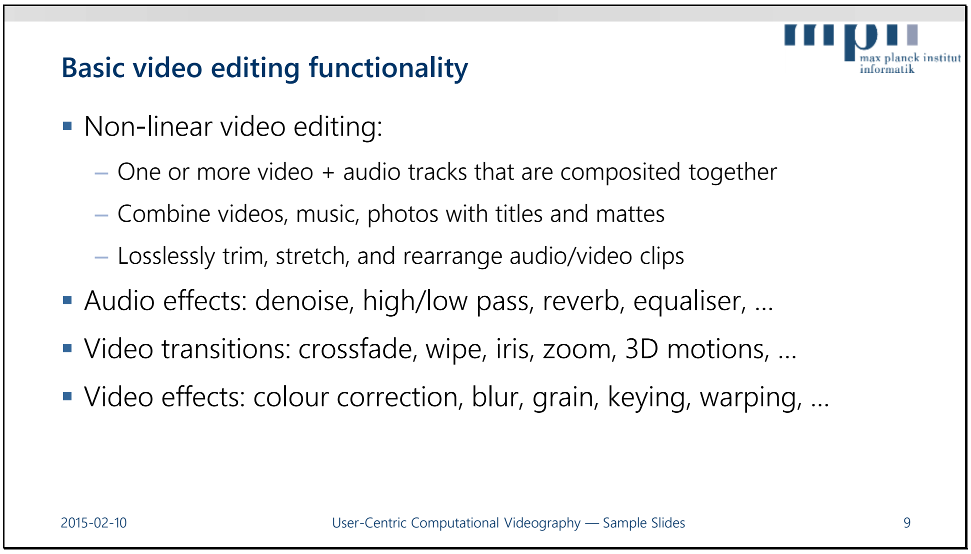
[http://en.wikipedia.org/wiki/File:Final_Cut_Pro_X.jpg]

Final Cut Pro

At their core, both categories focus on timeline-based non-linear video editing, in which multiple videos, photos, and music are combined with titles and transitions into an edited video sequence. Although the interfaces greatly vary in complexity, they both allows lossless triming, stretching and rearranging video and audio clips.

Sources:
http://screenshots.en.sftcdn.net/en/scrn/74000/74149/windows-live-movie-maker-22.jpg
http://en.wikipedia.org/wiki/File:Final_Cut_Pro_X.jpg

Slide 9



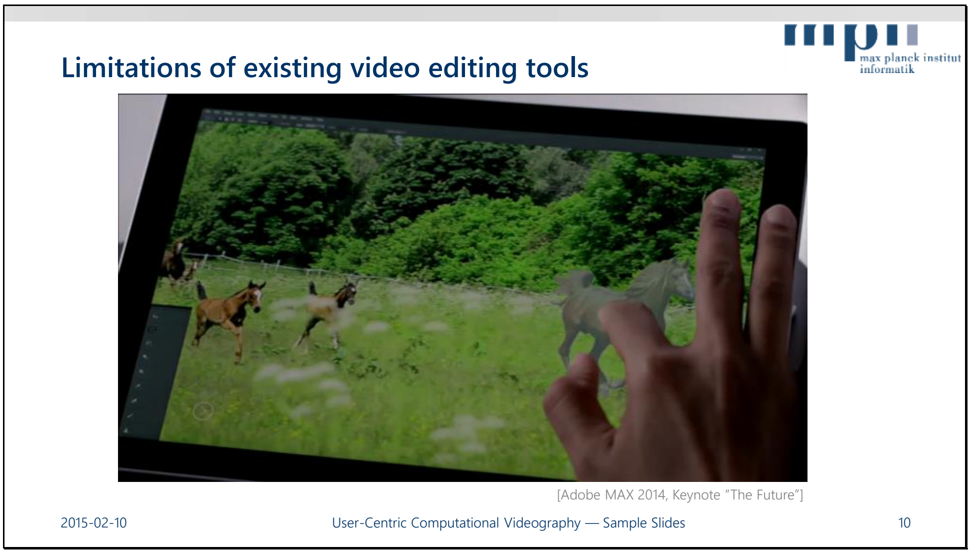
Basic video editing functionality

- Non-linear video editing:
  - One or more video + audio tracks that are composited together
  - Combine videos, music, photos with titles and mattes
  - Losslessly trim, stretch, and rearrange audio/video clips
- Audio effects: denoise, high/low pass, reverb, equaliser, …
- Video transitions: crossfade, wipe, iris, zoom, 3D motions, …
- Video effects: colour correction, blur, grain, keying, warping, …

2015-02-10 User-Centric Computational Videography — Sample Slides 9

Beyond the basic non-linear editing functionality, video editing tools also provide a wide range of audio and video effects and transitions. These can be used to alter the look and sound of a video clip in line with an artistic vision, and also to glue together smaller clips in a visually smooth or interesting fashion.

Slide 10



Limitations of existing video editing tools

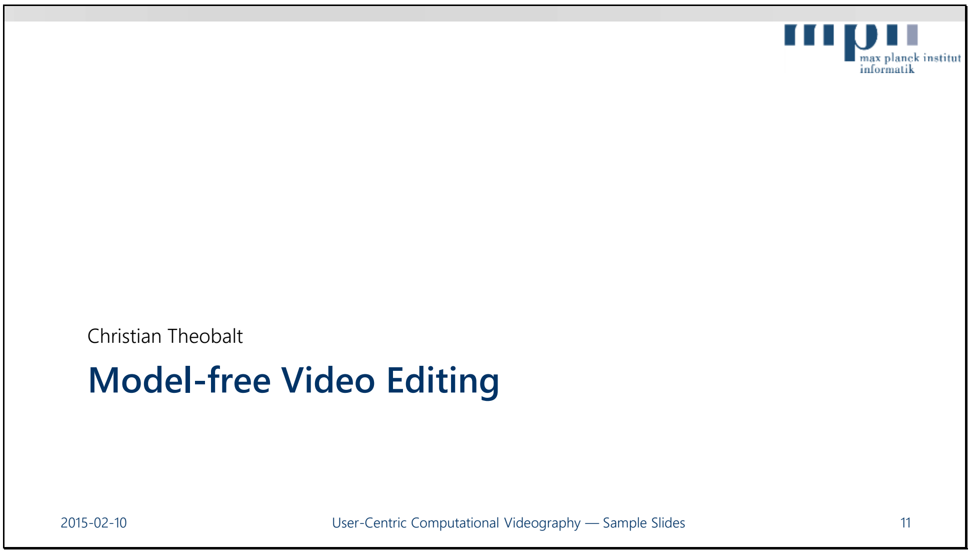
[Adobe MAX 2014, Keynote "The Future"]

2015-02-10 User-Centric Computational Videography — Sample Slides 10

However, there are some tasks that none of these existing tools can perform out of the box, such as this vision of content-aware video editing shown at Adobe MAX 2014.
© Here, a user rearranges a video by simply dragging the horses around.
The building blocks of this vision are video segmentation and video inpainting, both of which exist in the research literature, but we are still far from the robust and fast implementations required to pull off this sort of video editing.
This vision is a great example of the kind of editing tasks that User-Centric Computational Videography will enable in the future.

Source:
http://max.adobe.com/sessions/max-online/ (session "The Future", 00:16:08–00:16:15)

Slide 11



max planck institut informatik

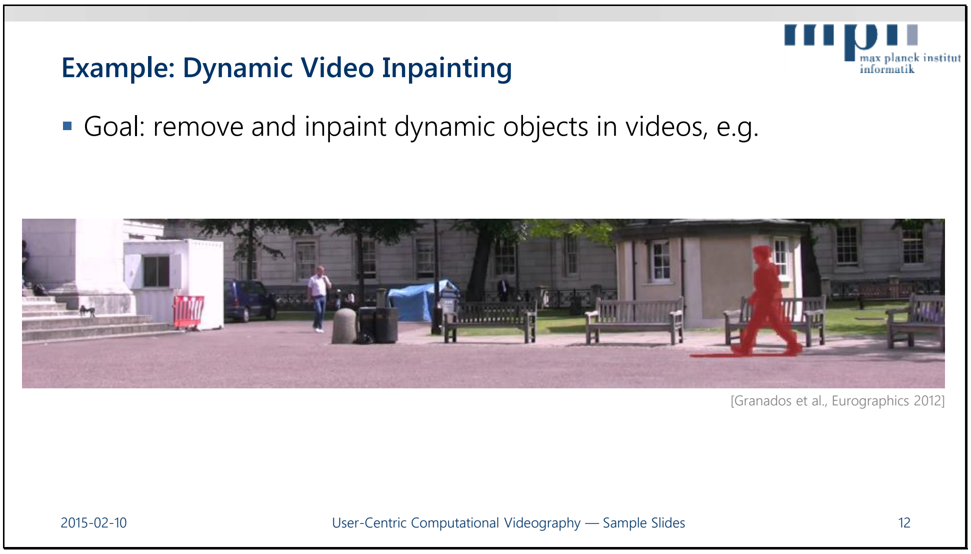
Christian Theobalt

Model-free Video Editing

2015-02-10 User-Centric Computational Videography — Sample Slides 11

Many video editing tasks can be performed without requiring any strong models of the observed scene in a video, of the lighting as well as the people or objects in a video, and this is what we call “model-free video editing”.

Slide 12



Example: Dynamic Video Inpainting

- Goal: remove and inpaint dynamic objects in videos, e.g.

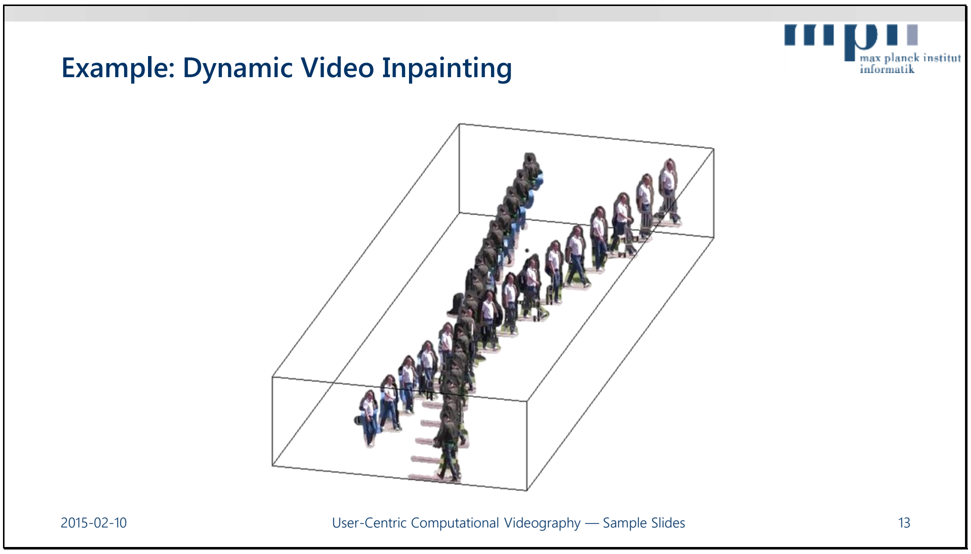
[Granados et al., Eurographics 2012]

2015-02-10 User-Centric Computational Videography — Sample Slides 12

As an example, let us consider the task of dynamic video inpainting. Say that we want to remove the walking person highlighted in red from this video, and to paint in plausible video content in the hole that is left. Here, the person on the right (as well as his shadow) is already segmented out and highlighted in red.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 13



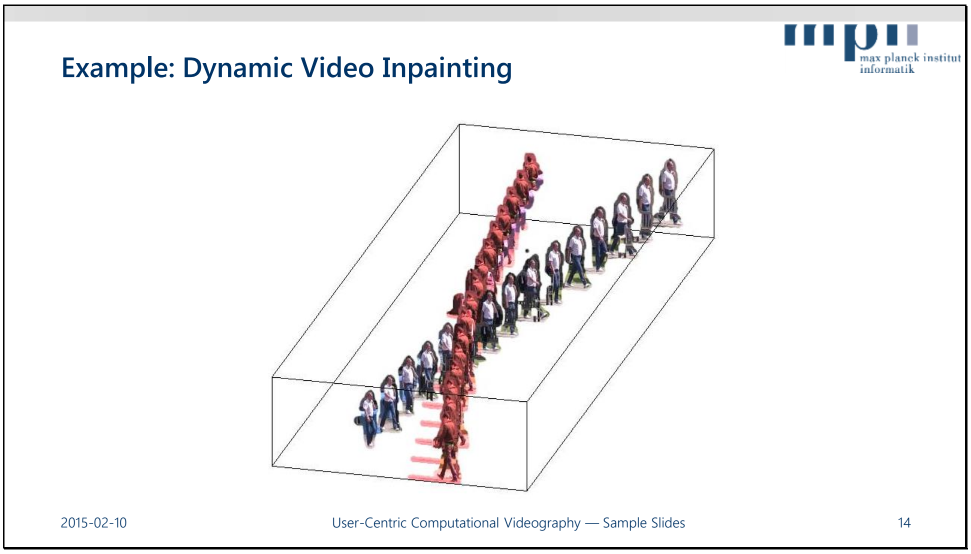
Example: Dynamic Video Inpainting

2015-02-10 User-Centric Computational Videography — Sample Slides 13

This is a volume visualization of the same video, in which the largely static background is hidden and only the walking people (and their shadows) remain for better visualization.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 14



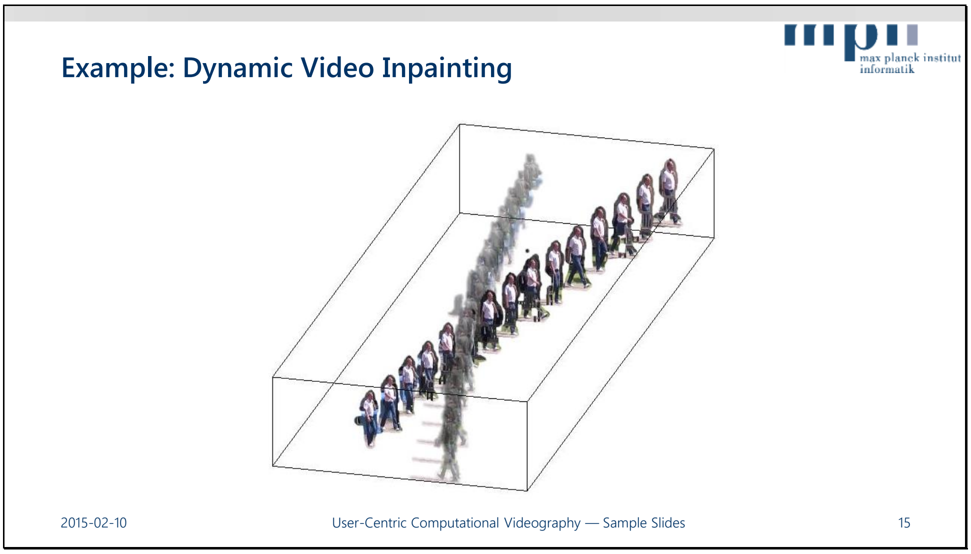
Example: Dynamic Video Inpainting

2015-02-10 User-Centric Computational Videography — Sample Slides 14

The region in red is what we want to remove from the video.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 15



Example: Dynamic Video Inpainting

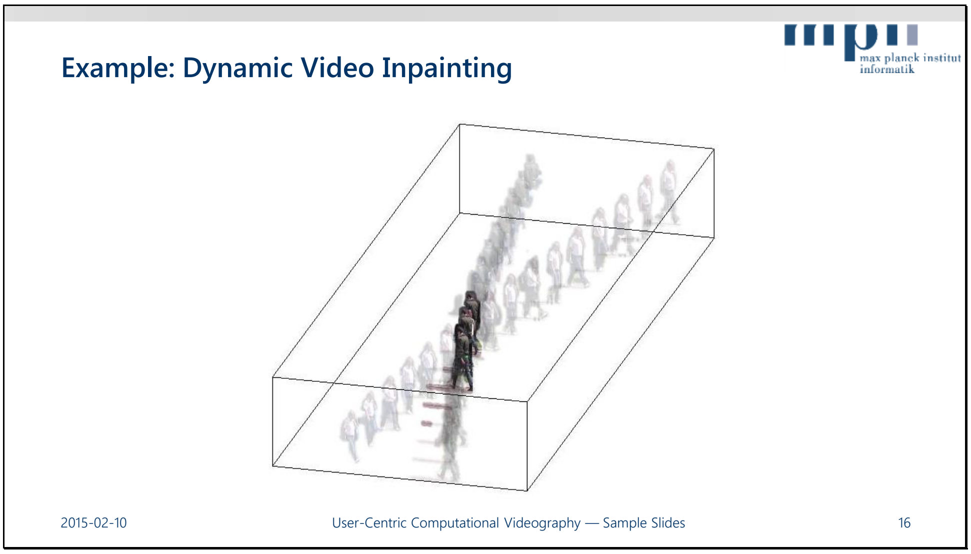
2015-02-10 User-Centric Computational Videography — Sample Slides 15

Simply removing the selected regions leaves a hole in every single frame of the video, which needs to be filled with whatever is behind the removed person.

In this video, the background is largely static, so we can use image content from nearby video frames to fill in most holes.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 16



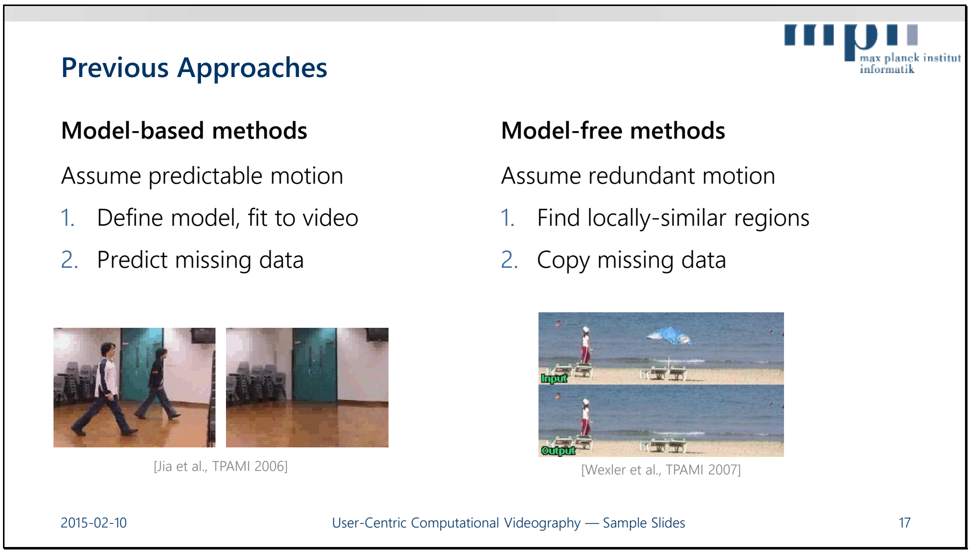
Example: Dynamic Video Inpainting

2015-02-10 User-Centric Computational Videography — Sample Slides 16

However, things get tricky where the other person was covered, as restoring this dynamic background is much more difficult. Manual inpainting of these regions is not an option as it is a very tedious process that can take many hours, even for professional artists.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 17



Previous Approaches

Model-based methods

Assume predictable motion

1. Define model, fit to video
2. Predict missing data

[Jia et al., TPAMI 2006]

Model-free methods

Assume redundant motion

1. Find locally-similar regions
2. Copy missing data

Input
Output

[Wexler et al., TPAMI 2007]

2015-02-10 User-Centric Computational Videography — Sample Slides 17

In the literature, there are two main approaches for doing this.

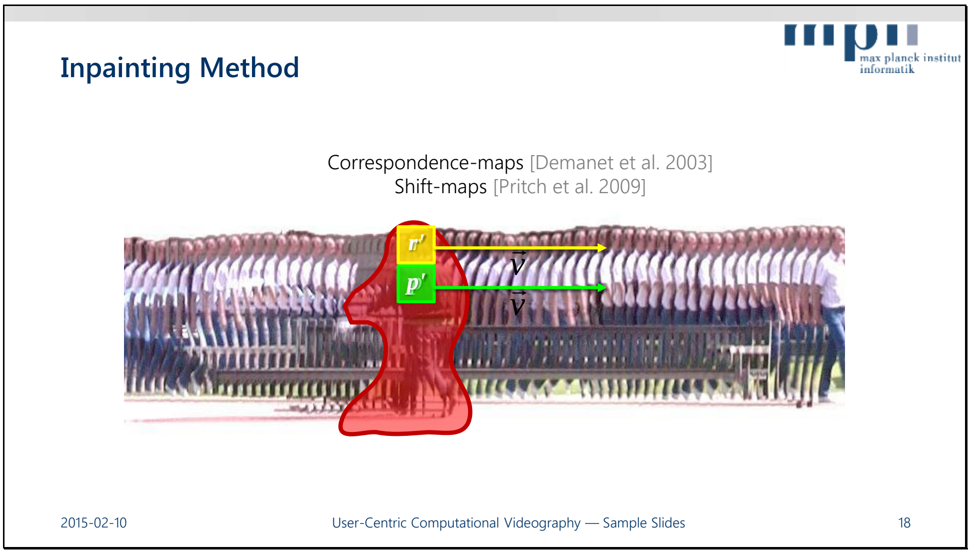
The first is © a model-based approach:
© they assume that the behavior of the occluded objects can be predicted
© So you choose a model for the objects, and you train it with your input video,
© and then using this model you predict the occluded part.
This type of method can be more accurate but it mostly relies on cyclic motions.

On the other hand, © model-free methods don't assume anything about the scene,
© only that there is enough redundancy in the video to fill the missing data.
© So they find other regions in the video that are similar to the boundary of the missing data,
© and copy them in a way that the result is consistent with the rest of the video.

Right now, I will focus on the model-free methods for video editing, and will discuss the model-based video editing methods later.

Sources:
Jia et al., Video repairing under variable illumination using cyclic motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5): 832–839, 2006.
Wexler et al., Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3): 463–476, 2007.

Slide 18



Inpainting Method

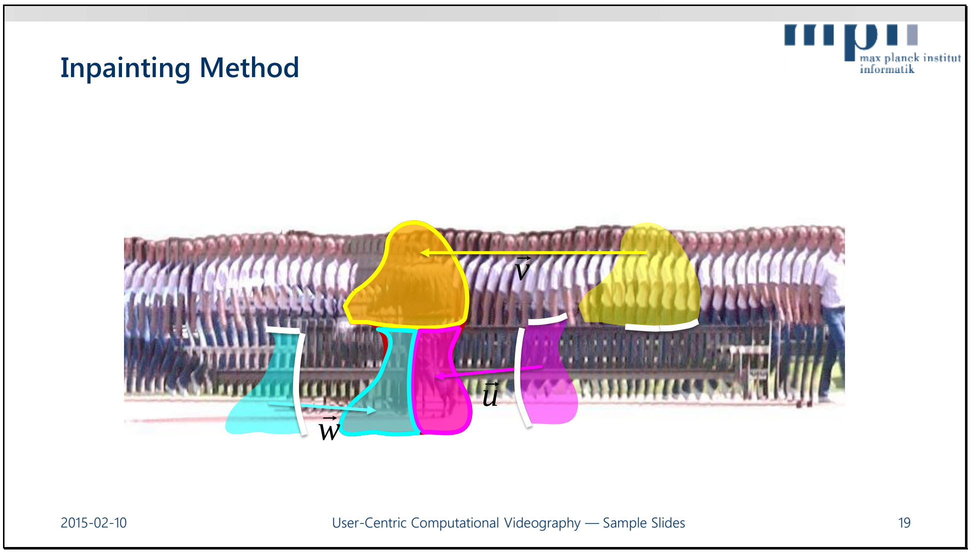
Correspondence-maps [Demanet et al. 2003]
Shift-maps [Pritch et al. 2009]

2015-02-10 User-Centric Computational Videography — Sample Slides 18

Let’s take a front view of the video, and the person we want to remove is © here.
We assume that there is a high degree of redundancy in videos, which means that every © missing pixel can be filled using some © other pixel somewhere else in the video.
Pixels in the missing region that are close by © are also likely to be replaced with © pixels outside the hole that are also close.
Instead of computing absolute pixel locations, we compute © offsets, so that adjacent pixels can have the same offset.
© This offset field is known as a “correspondence map” or a “shift map”, and has previously been used for image inpainting.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 19



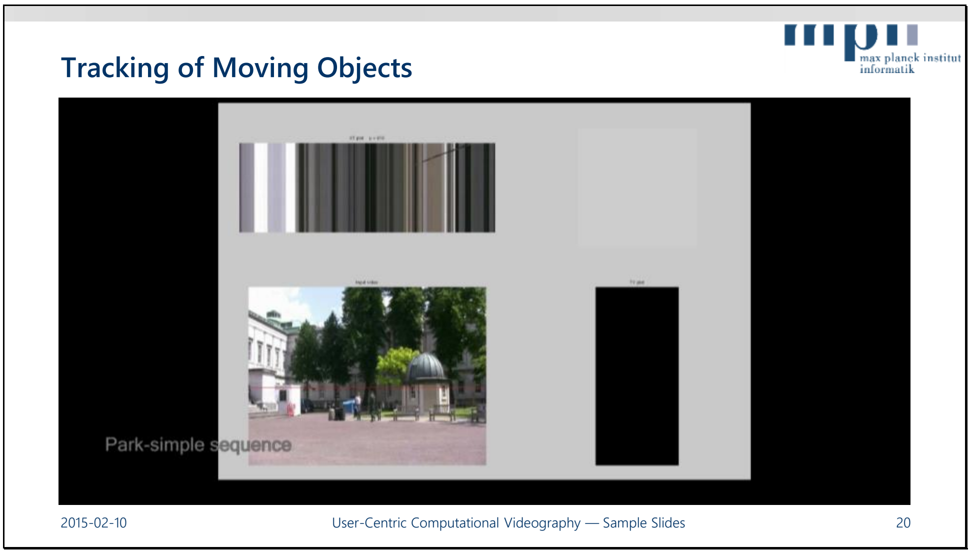
Inpainting Method

2015-02-10 User-Centric Computational Videography — Sample Slides 19

However, finding a single large region to fill each hole restricts the quality of the final inpainting, as the filled video content needs to be plausible in each frame but also coherent over time.
A better approach is to fill the hole with smaller pieces of the video, like in a 3D jigsaw puzzle, where we cut © © © pieces from everywhere in the video, no matter what size or shape they have.
The only restriction is that the boundary of each piece has to match those of © © © all the adjacent pieces.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 20



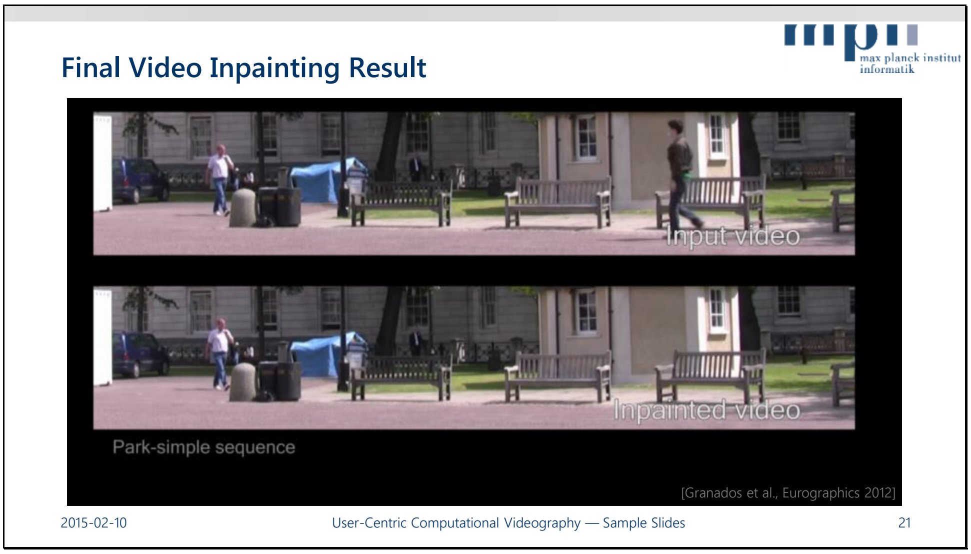
Tracking of Moving Objects

2015-02-10 User-Centric Computational Videography — Sample Slides 20

Searching for matching regions across the entire video makes for an extremely large search space, but one can reduce the search space for dynamic, moving objects interactively. The user simply marks the trajectory of the person, and the video inpainting then only needs to search within this bounding box.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 21



Final Video Inpainting Result

[Granados et al., Eurographics 2012]

2015-02-10 User-Centric Computational Videography — Sample Slides 21

And this is the final inpainting results at the bottom compared to the input video at the top. If you only saw the inpainted video, you probably wouldn't even notice that a complete person has been removed largely automatically. Only if one looks very carefully, one can see a few minor artifacts in the dynamic inpainting areas [point it out].

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 22

One More Video Inpainting Result

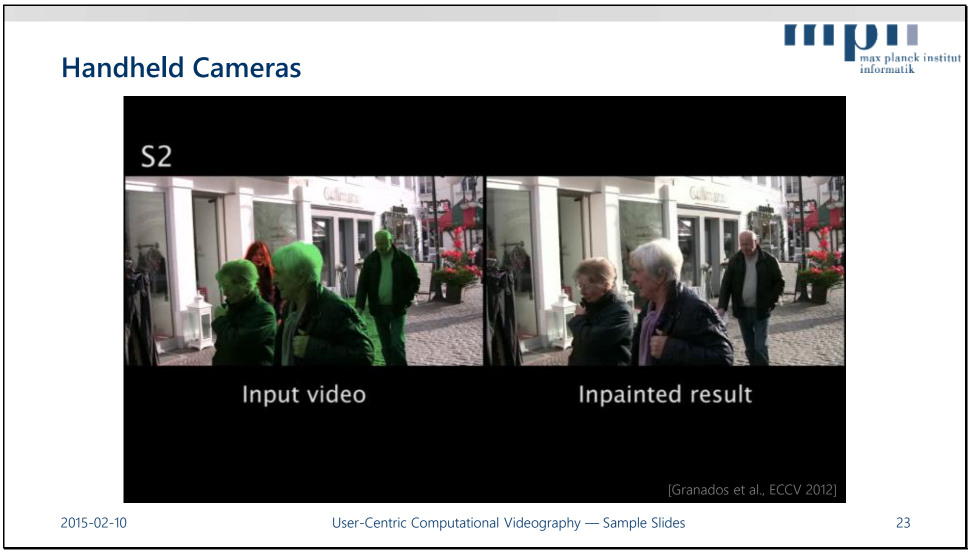
[Granados et al., Eurographics 2012]

2015-02-10 User-Centric Computational Videography — Sample Slides 22

(starts paused) This is one of my favorite results because it shows that automatic inpainting can be very high quality, even on high resolution videos © Here we'll remove the two pedestrians walking in front.
(wait) This is the result. (wait) Here we can see both the input and the output …
(wait) Most people realize that the reflections in the glass are still there but the result is very plausible.

Source:
Granados et al., How Not to Be Seen – Object Removal from Videos of Crowded Scenes. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2):219–228, 2012.

Slide 23



Handheld Cameras

[Granados et al., ECCV 2012]

2015-02-10 User-Centric Computational Videography — Sample Slides 23

Video inpainting can also be extended to the much more challenging case of handheld cameras. Here, the woman marked in red is removed entirely from the video. Crucial for good results is the stabilization of the background of the video, which is achieved in this work using multiple planar homographies that track and align the piecewise planar scene geometry over time.

Source:
Granados et al., Background Inpainting for Videos with Dynamic Objects and a Free-moving Camera. *ECCV* 2012.