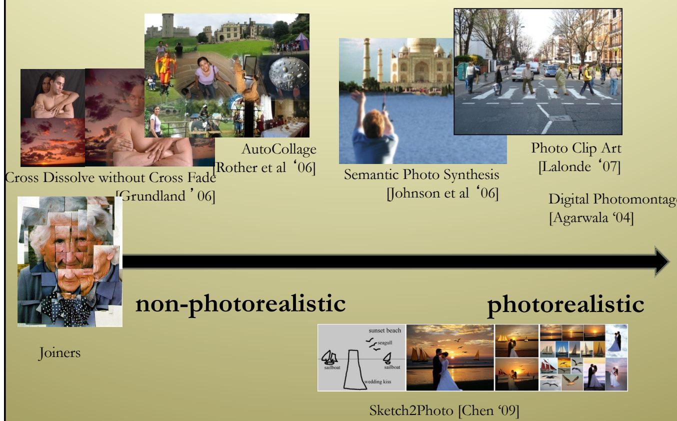
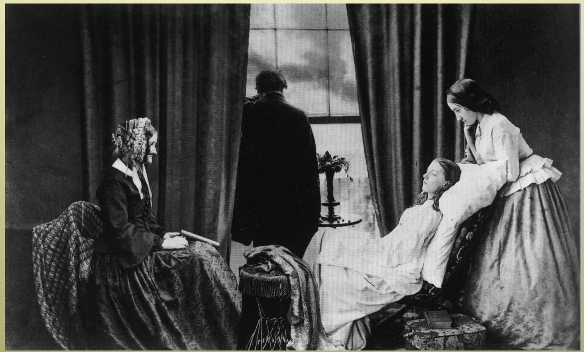
Making Collages

Goal: Combine parts of a set of photos together into a single composite image

Combining Multiple Images



Application: Artistic

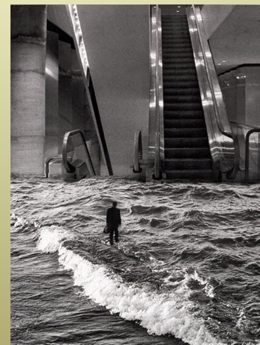


Henry Peach Robinson, Fading Away, 1858
(made from 5 negatives)

Application: Artistic

Henry Peach Robinson, When the Day's Work is Done, 1877

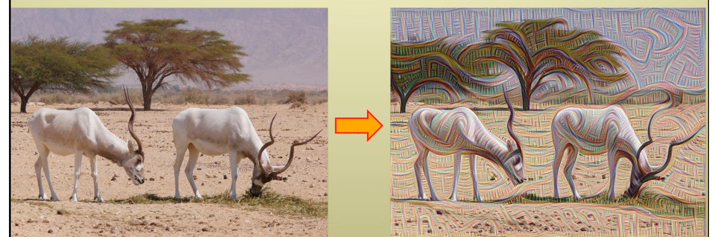
Application: Artistic



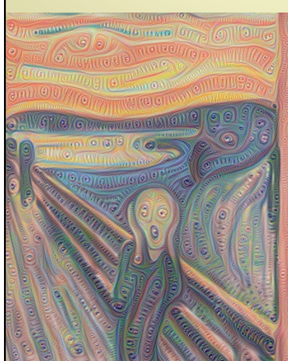
Scott Mutter

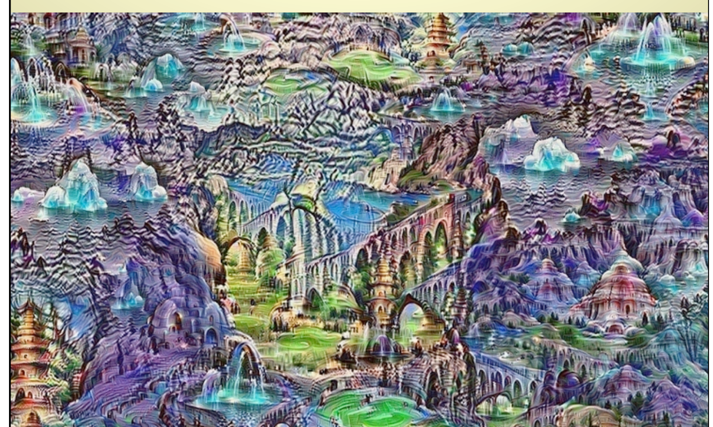
Deep Dreams / Inceptionism

Google project by A. Mordvintsev, C. Olah, and M. Tyka



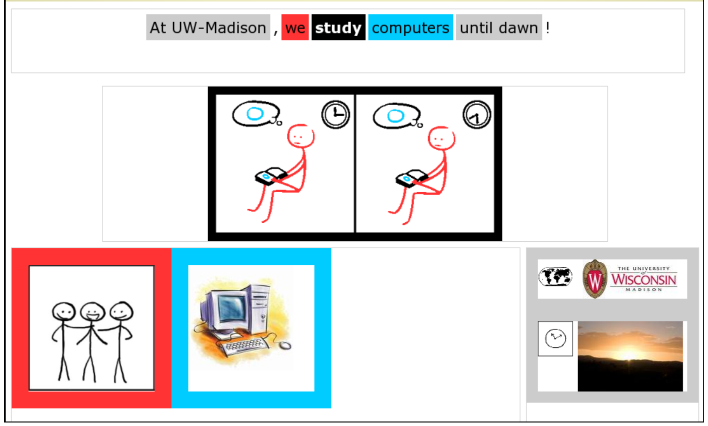
Thanks Aaron Wurtinger-Knaack



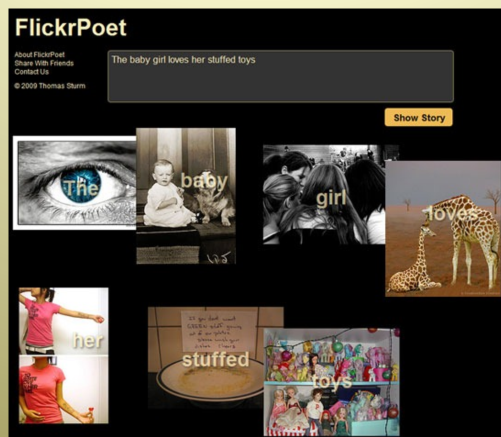


Application: Group Portraits

Visual Storytelling: Text-to-Picture



Visual Storytelling: FlickrPoet



Challenges

- Image selection
- Region selection
  - Segmentation / ROI
- Placement / layout
- Blending

Interactive Digital Photomontage

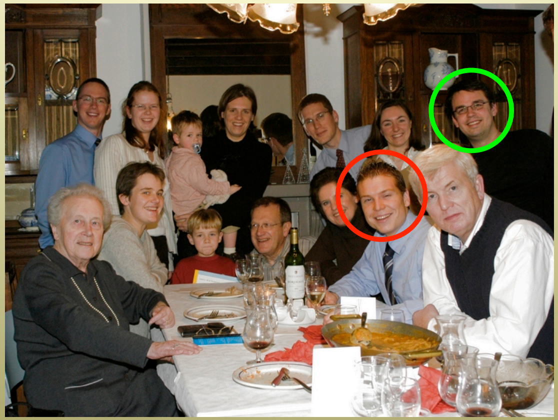
Aseem Agarwala, Mira Dontcheva,
Maneesh Agrawala, Steven Drucker, Alex Colburn,
Brian Curless, David Salesin, Michael Cohen

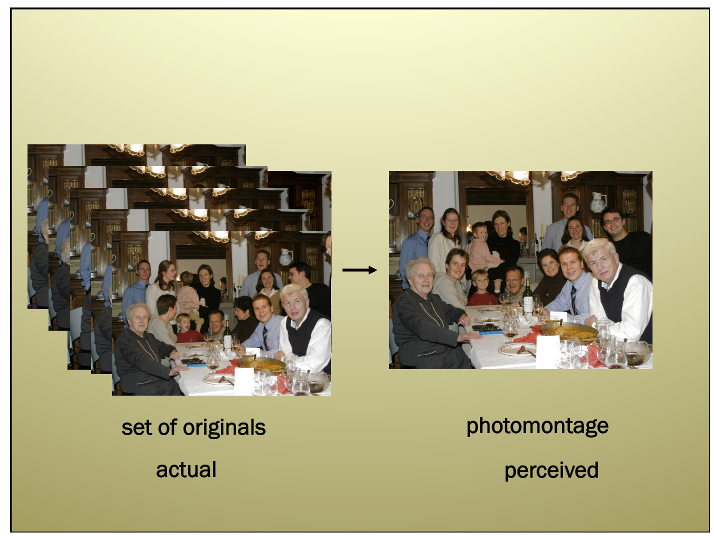
University of Washington & Microsoft Research

Proc. SIGGRAPH 2004

Example: The "Moment Camera"

Camera is *always* recording images using a finite round-robin buffer of perhaps 500 frames, or 5 seconds, resulting in a space-time slab




set of originals
actual
photomontage
perceived

Overview of Approach

Given a stack of images that are approximately spatially in register and closely-related photos of a scene, interactively create a composite image:

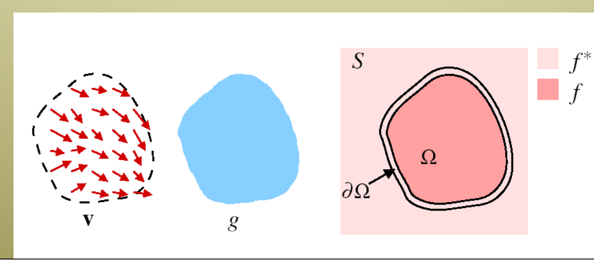
- Iteratively refine the composite image "labeling," i.e., assign at each pixel one of the source images where the corresponding pixel's color is copied from
- Use a graph-cut optimization method to determine what regions from each image are used
- Use Poisson image editing (gradient domain blending) to blend image regions seamlessly together

Gradient Domain Blending

- Perez, Gangnet and Blake, Poisson Image Editing, *SIGGRAPH* 2003
- Rather than copying pixels, copy the ***gradients*** instead; then compute the pixel values by solving a Poisson equation that matches the gradients while also satisfying fixed boundary conditions at seam

Poisson Cloning: "Guiding" the Completion

- Use gradients from a source image to *guide* the completion
- Find new image f defining the pixel values in the hole whose gradients are closest to the gradients (vector field $\boldsymbol{v}$) of a source image, g, while holding $f = f^*$ over the boundary $\delta\Omega$



Formulate as Optimization Problem

Cost function to minimize:

$$C(L) = \sum_{p} C_d(p, L(p)) + \sum_{p,q} C_i(p, q, L(p), L(q))$$

Data penalty (**Image Objective**) — Interaction penalty (**Seam Objective**)

L(*p*) = "labeling" for pixel *p*, i.e., which source image is used for pixel *p*

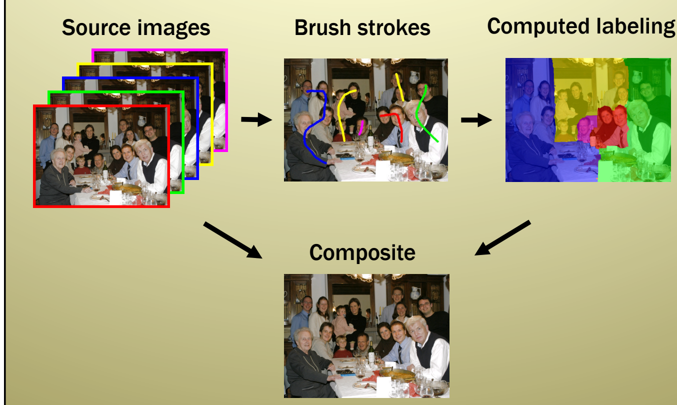
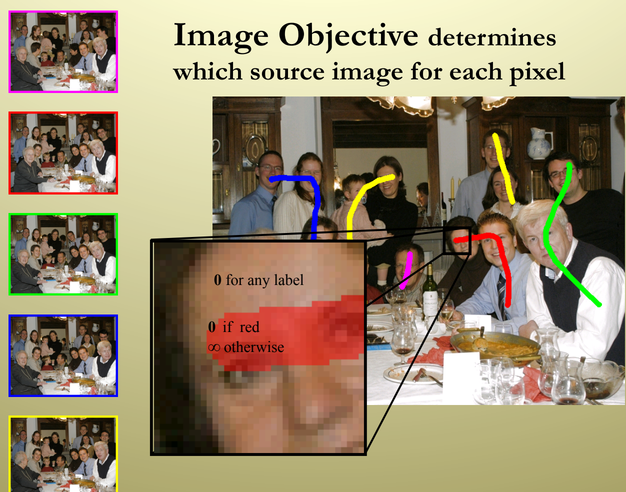


Image Objective determines which source image for each pixel



Some Possible Image Objectives

User-selected pixel property that is used to decide for each pixel in composite, which corresponding pixel from the stack of images should be used:

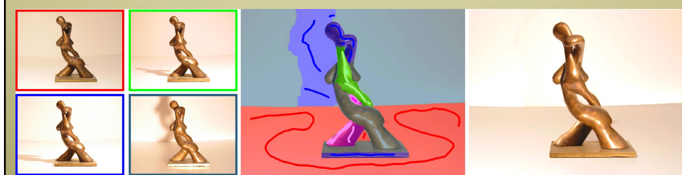
- Designated image
- Designated color
- Min/Max Luminance
- Min/Max Contrast
- Min/Max Likelihood
- Eraser
- Min/Max Difference

Designated Image

- For each stroke, the user specifies which source image the pixels come from

Designated Color

- Specify a target color and find source images that have similar or different colors
- Cost function given by Euclidean distance in RGB space



Min/Max Luminance

- Min (max) of luminance channel. Good for adding shadows/highlights
- The darkest / lightest pixel in the span

Seam Objective measures suitability of a seam between two regions, i.e., adjacent pixels assigned different labels

Seam Objectives

- **Colors**: match colors across seams
- **Colors & Gradients**: match colors and color gradients across seams
- **Colors & Edges**: match colors across seams but prefer seams that lie *along* edges (i.e., edge-sensitive)

Luminance Objective

courtesy of P. Debevec

Video Sequence defines Source Image Stack

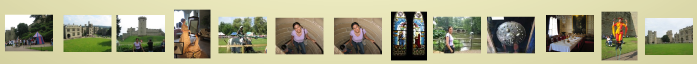
Iteration 1: Maximum Likelihood Objective Defines Static Background

Iteration 2: Maximum Difference Objective

AutoCollage

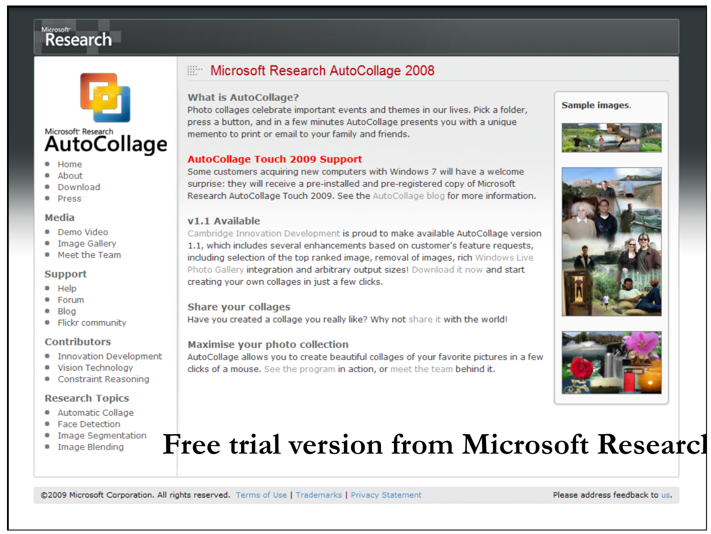
C. Rother *et al.*, *SIGGRAPH 2006*

Microsoft Research AutoCollage



Goals:

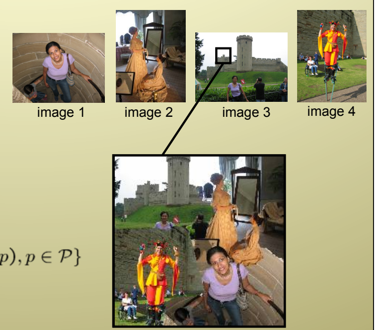
- Representativ
- One coherent from each ima
- Pack many in (sky at top)
- Smooth imag



Free trial version from Microsoft Researcl

Formulate as Optimization Problem

- Pixel in AutoCollage $p \in \mathcal{P}$
- Label of a pixel $L(p) = (n, s)$
  where n = input image
  s = offset from p, i.e., $I(p) = I_n(p - \mathbf{s})$
  (In Interactive Digital Photomontage [Agarwala et al. '04] s is known)



- **Goal:** find best labelling $L = \{L(p), p \in \mathcal{P}\}$
- **Minimize energy:**

$$E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$$

object sensitivity | representative images | important image parts | smooth image transitions

Object Sensitivity

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

Goal:

- Include faces (important objects) as a whole

Face detection [Viola, Jones '01]

- Good object placement

AutoCollage

Representative Images

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

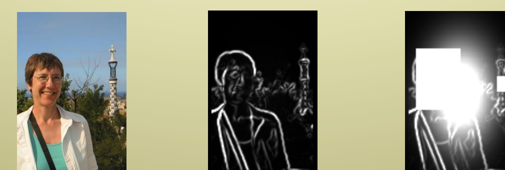
Goal: Select high-quality images that are interesting and distinct

- Energy: $E_{rep}(L) = -\sum_n \text{Qual}(n) + \min_m \text{Diff}(n, m)$ (for images n, m present)
- Image quality: $\text{Qual}(n) = [n \text{ has a face}] + \text{Entropy}(n)$ (color texture variation)
- Image difference: $\text{Diff}(n, m)$ distance between histograms of images *n* and *m*
- Additional Information:
  - Detecting same scene (e.g. [Noah '06])
  - Study aesthetics of photographs (e.g. [Datta '06])

Important Image Regions

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

Goal: Estimate an important region in an image



Local entropy around a pixel (contrast) | Add center weighting and faces

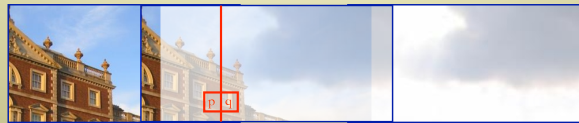
$$E_{imp}(L) = -\sum_p localEntropy(p) \times imageCentrality(p)$$

- Alternatively: use visual saliency model [Itti et al. '98]

Image Transition

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

Goal: Invisible transitions between adjacent images



Pairwise term: $E_{tran}(L) = \sum_{p,q \in Neigh} \text{Diff}(p) + \text{Diff}(q)$

Standard texture synthesis
[Kwatra et al. '03]

$\min\ [\text{Diff}(p)/\text{Edge}(p,q), \text{Diff}(q)/\text{Edge}(p,q)]$
similar to [Kwatra et al. '03; Agarwala et al. '04]

Energy Minimization

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

Approximated sequentially by:

Image Ranking
↓
Region of Interest
↓
Constraint Satisfaction
↓
Graph-Cut Optimization

Extra constraints:
- **Coverage**
  All pixels p must have a label $L(p)$
- **Information bound**
  disallow small image fragments $E_{imp}(n) \geq 90\%$

Comparison to Tapestry [Rother et al '05]:
- One optimization step (graph cut)
- About 16 times slower
- Information bound not feasible

Example Image Set

Step 1: Rank Images

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

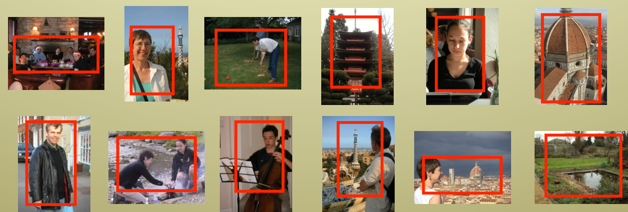
Method: Order representative images in a greedy way using E_{rep}

12 top-ranking images:

Step 2: Detect Region of Interest

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

Method: Exhaustive search using integral images [Crow '84]
so that all faces included and $E_{imp}(n) \geq 90\%$



Constraint Satisfaction

Energy: $E(L) = E_{obj}(L) + w_{rep}E_{rep}(L) + w_{imp}E_{imp}(L) + w_{trans}E_{trans}(L)$

Goal: Find global layout that satisfies all constraints

(NP-hard problem)

ROI packing –
Information bound constraint

(no 2 ROIs overlap)

Image packing –
Coverage constraint

(every pixel covered)

After Constraint Satisfaction

After Graph-Cut Optimization

Final Result after Blending

Results

Results

Results


Results


Results

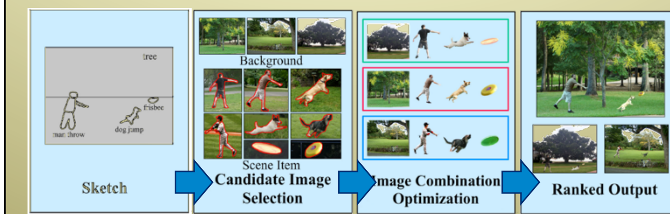
Sketch2Photo

Tao Chen Ming-Ming Cheng Ping Tan
Ariel Shamir Shi-Min Hu

Tsinghua University, National University of Singapore

Proc. SIGGRAPH Asia, 2009

Approach



Challenges

- Image search
- Image segmentation
- Image composition

  Difficult problems in general!
- Key idea:

  Use simple images by filtering!

Results

Results

Photo Clip Art

Jean-François Lalonde, Derek Hoiem, Alexei A. Efros
Carnegie Mellon University

Carsten Rother, John Winn and Antonio Criminisi
Microsoft Research Cambridge

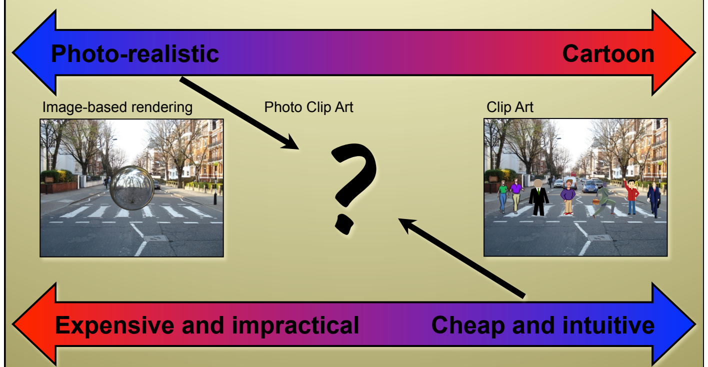
Proc. SIGGRAPH, 2007

Inserting Objects into Images

Inserting Objects in Images: Clip Art

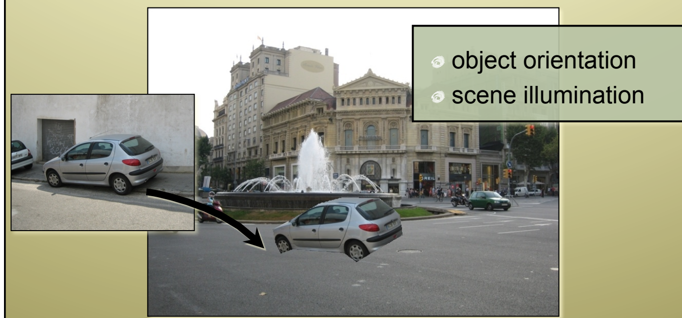
- Easy, intuitive, cheap
- Not realistic

Goal



Challenges

- Insert **THIS** object: impossible!



The Use of Data

- Insert **SOME** object: much easier!

Photo Clip Art

Jean-François Lalonde, Derek Hoiem, Alexei A. Efros,
Carsten Rother, John Winn and Antonio Criminisi

Carnegie Mellon GRAPHICS

Microsoft Research

Results: Street Accident

Results: Bridge

Results: Painting

Results: Alley