Looking Back, Moving Forward

Computational Photography

Derek Hoiem, University of Illinois

Today

- Requested topics
  - 3D reconstruction
  - Light transport
  - Event cameras

- Beyond this class…

This course has provided fundamentals

- How photographs are captured from and relate to the 3D scene

- How to think of an image as: a signal to be processed, a graph to be searched, an equation to be solved

- How to manipulate photographs: cutting, growing, compositing, morphing, stitching

- Basic principles of computer vision: filtering, correspondence, alignment

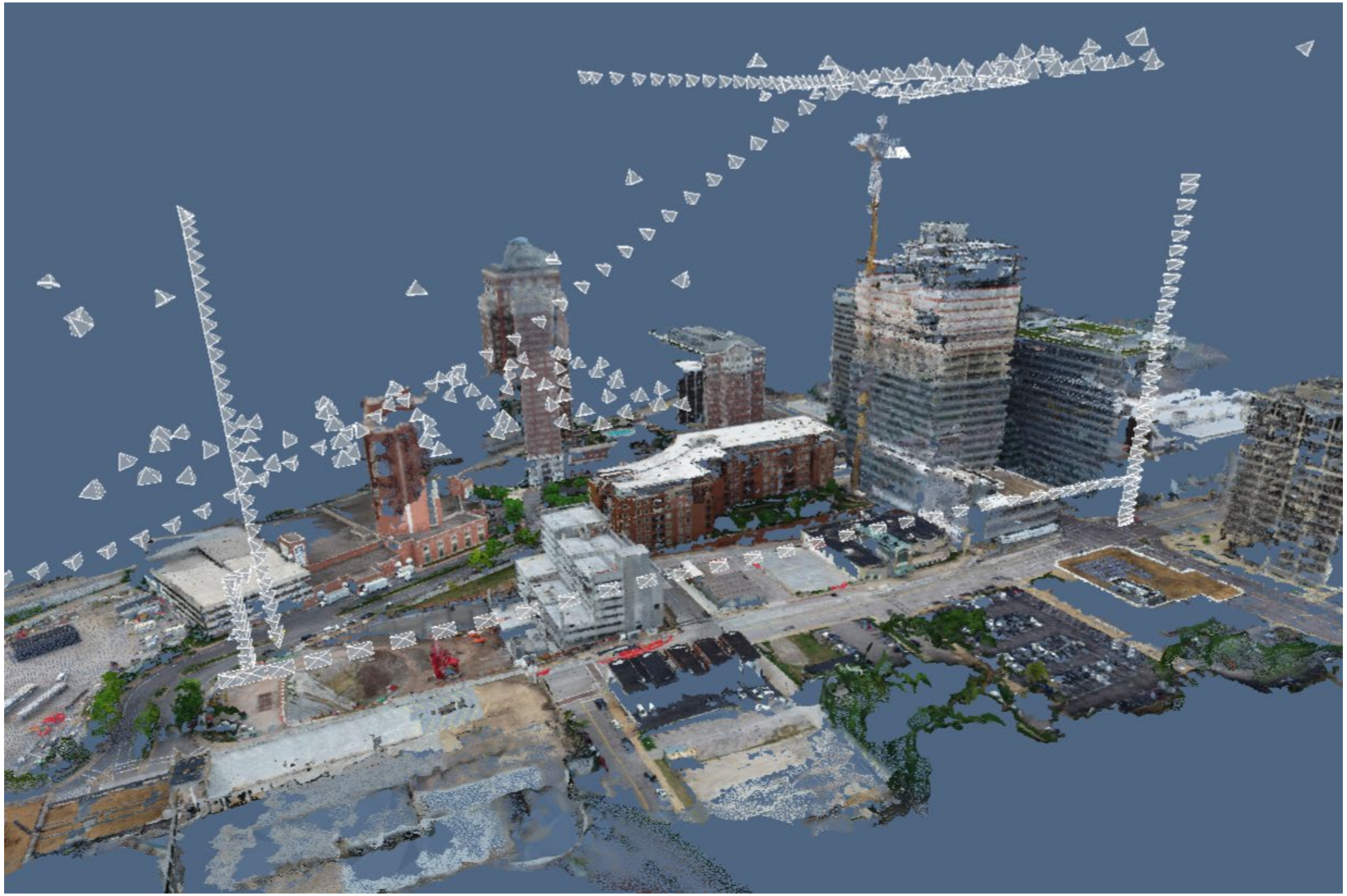
What else is out there?

Lots!

- Machine learning
- Videos and motion
- 3D reconstruction
- Scene understanding
- Better/cheaper devices
- …

How to create 3D model from multiple images

1. Solve for camera poses
2. Propose and verify 3D points by matching
3. Fit a surface to the points



Incremental Structure from Motion (SfM)

Goal: Solve for camera poses and 3D points in scene

Incremental SfM

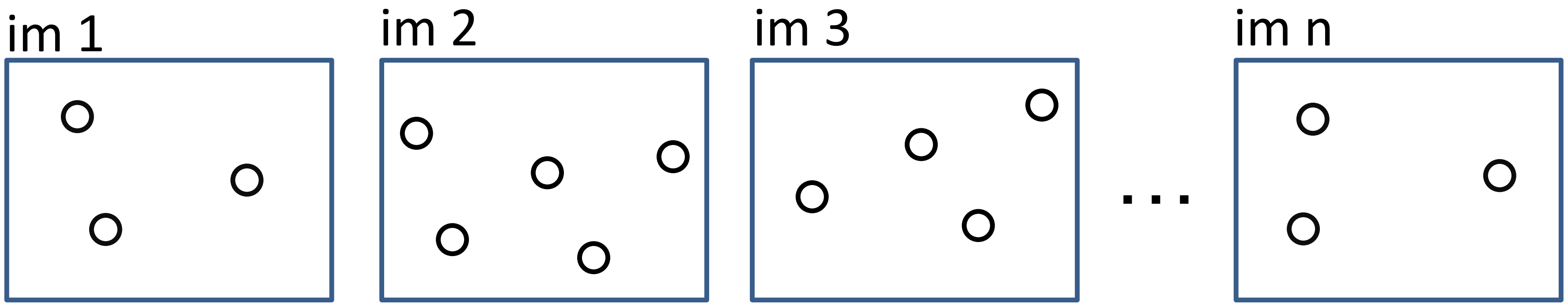
1. Compute features

2. Match images

3. Reconstruct
    a) Solve for pose and 3D points in two cameras
    b) Solve for pose of additional camera(s) that observe reconstructed 3D points
    c) Solve for new 3D points that are viewed in at least two cameras
    d) Bundle adjust to minimize reprojection error

Incremental SFM: **detect features**

- Feature types: SIFT, ORB, Hessian-Laplacian, …



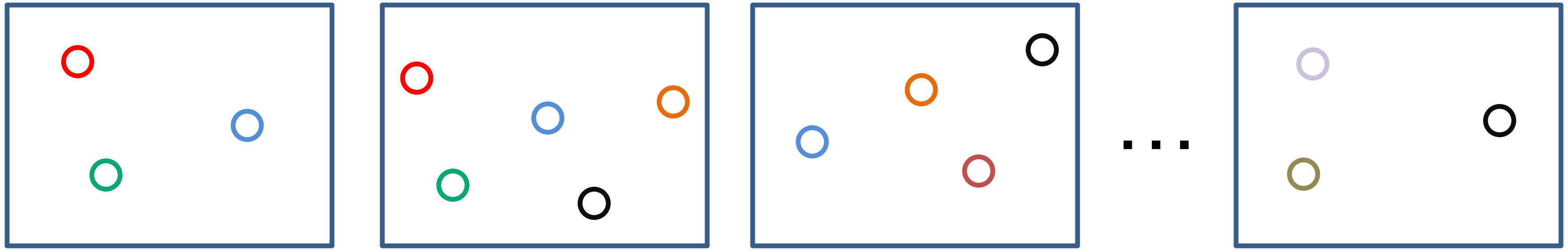
Each circle represents a set of detected features

Incremental SFM: **match features and images**

For each pair of images:

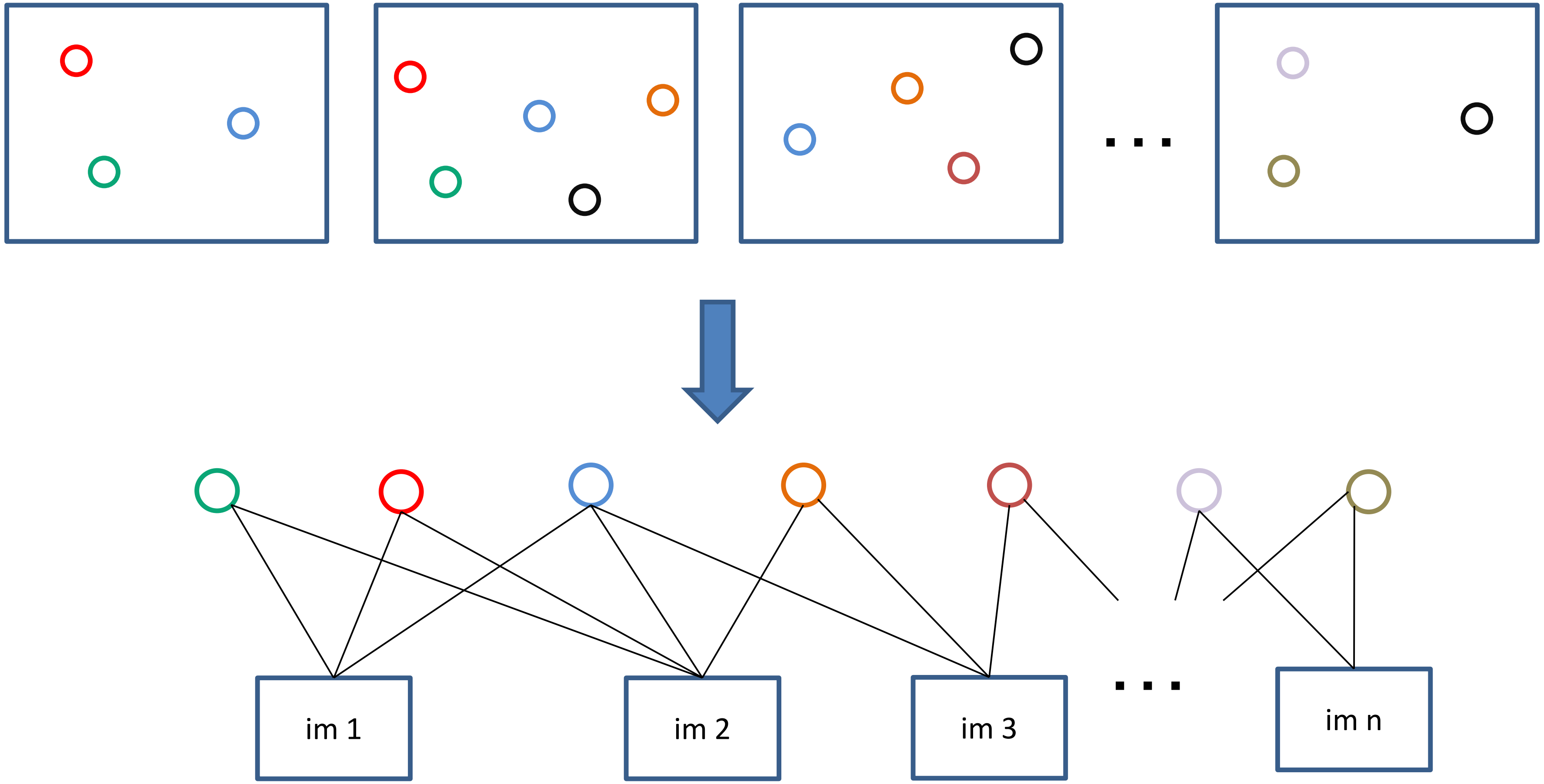
1. Match feature descriptors via approximate nearest neighbor and apply Lowe's ratio test
2. Solve for F and find inlier feature correspondences

- Speed tricks
  - Use vocabulary tree to get image match candidates
  - Use GPS coordinates to get match candidates, if available



Points of same color have been matched to each other

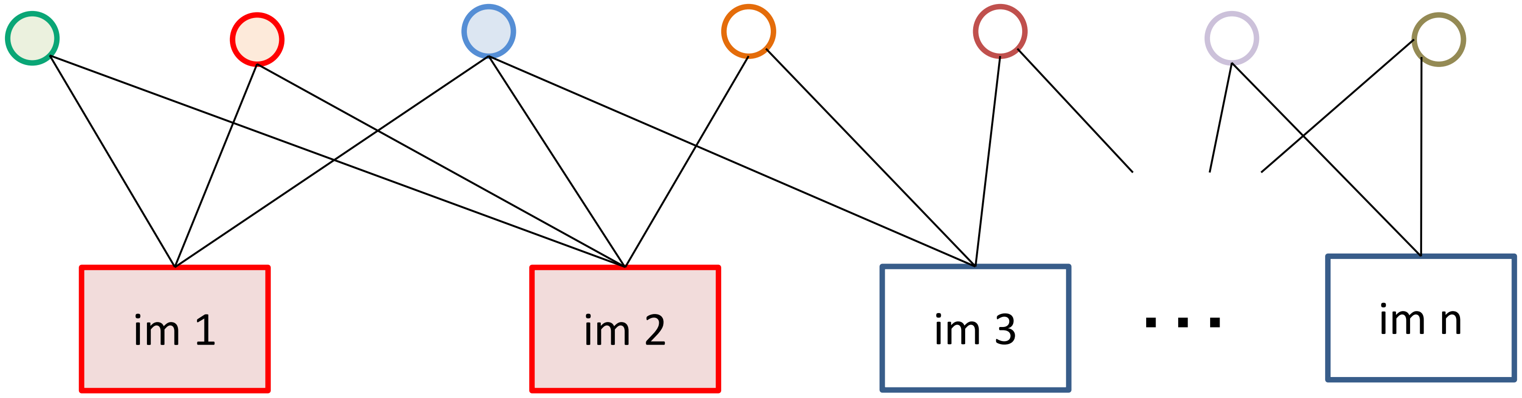
Incremental SFM: **create tracks graph**



tracks graph: bipartite graph between observed 3D points and images

Incremental SFM: **initialize reconstruction**

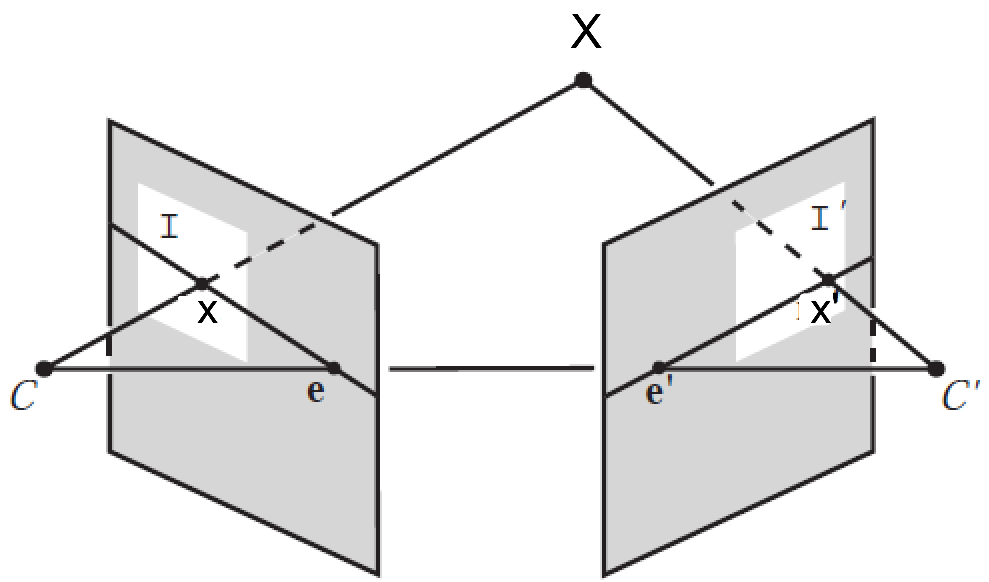
1. Choose two images that are likely to provide a stable estimate of relative pose
   - E.g., $\frac{\#\text{ inliers for } H}{\#\text{ inliers for } F} < 0.7$ and many inliers for F
2. Get focal lengths from EXIF, estimate essential matrix using 5-point algorithm, extract pose R_2, t_2 with $R_1 = \boldsymbol{I}, t_1 = \boldsymbol{0}$
3. Solve for 3D points given poses
4. Perform bundle adjustment to refine points and poses



filled circles = "triangulated" points
filled rectangles = "resectioned" images (solved pose)

Triangulation: Linear Solution

- Generally, rays C→x and C'→x' will not exactly intersect
- Can solve via SVD, finding a least squares solution to a system of equations



$$\mathbf{x} = \mathbf{P}\mathbf{X} \qquad \mathbf{x}' = \mathbf{P}'\mathbf{X}$$

$$\mathbf{A}\mathbf{X} = \mathbf{0} \quad \mathbf{A} = \begin{bmatrix} u\mathbf{p}_3^T - \mathbf{p}_1^T \\ v\mathbf{p}_3^T - \mathbf{p}_2^T \\ u'\mathbf{p}_3'^T - \mathbf{p}_1'^T \\ v'\mathbf{p}_3'^T - \mathbf{p}_2'^T \end{bmatrix}$$

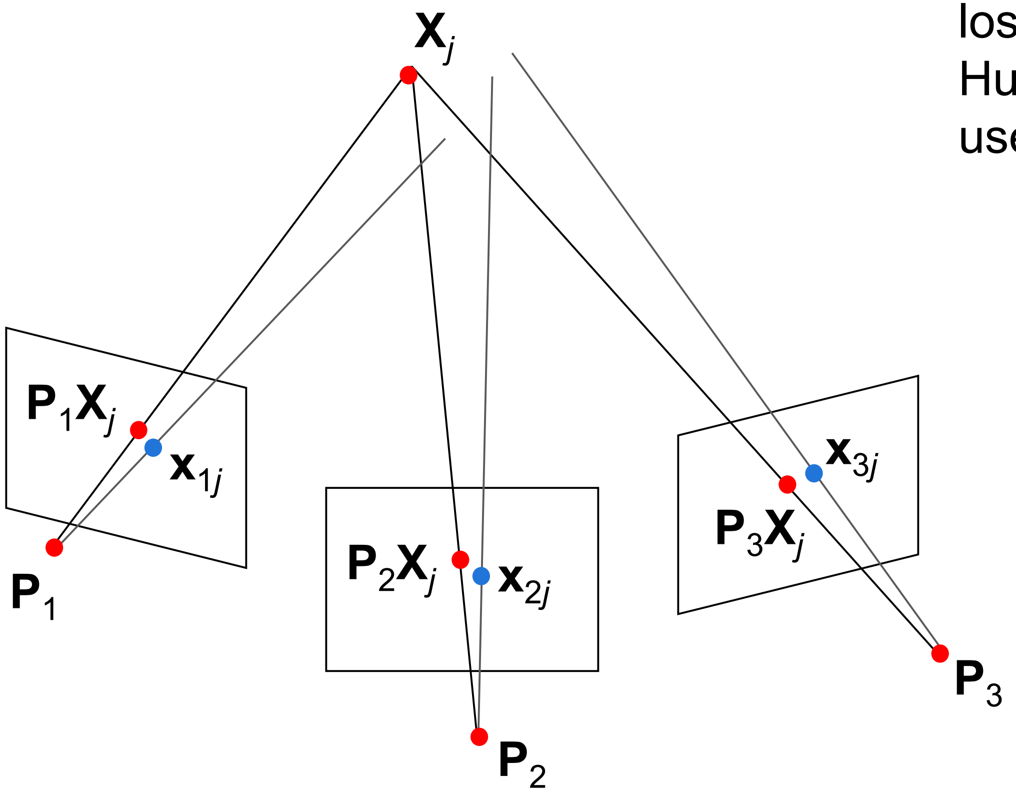
Further reading: Hartley-Zisserman p. 312-313

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing reprojection error

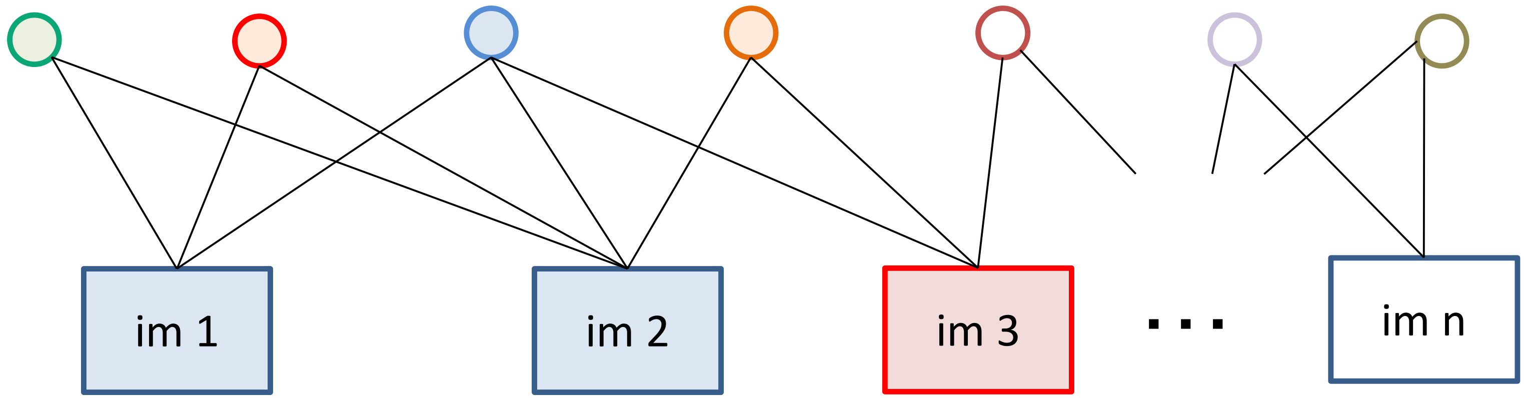
$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

Often a robust loss, such as Huber loss is used



Incremental SFM: **grow reconstruction**

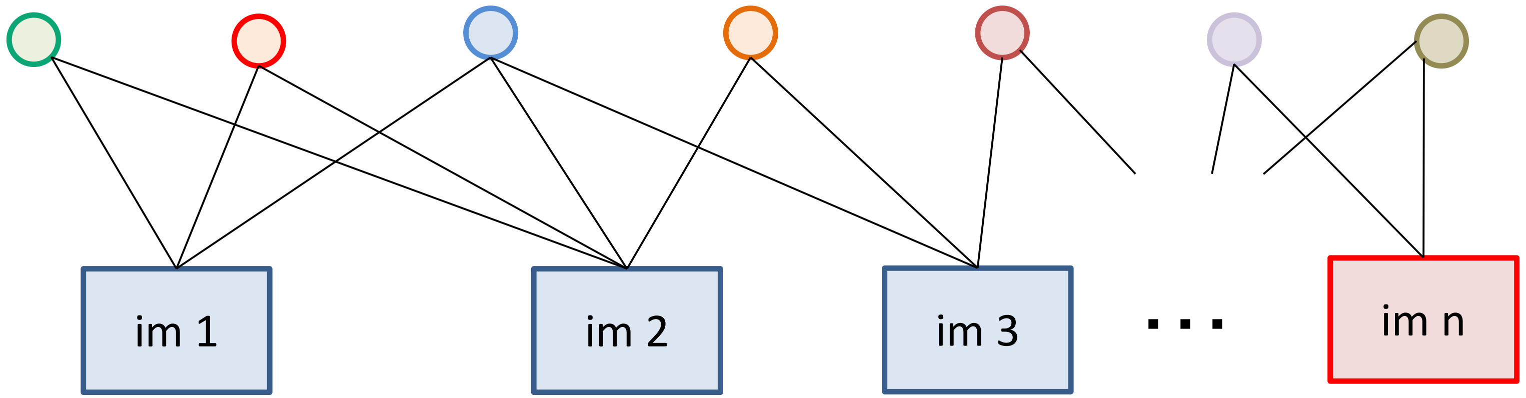
1. Resection: solve pose for image(s) that have the most triangulated points
2. Triangulate: solve for any new points that have at least two cameras
3. Remove 3D points that are outliers
4. Bundle adjust
   - For speed, only do full bundle adjust after some percent of new images are resectioned
5. Optionally, align with GPS from EXIF or ground control points (GCP)



filled circles = "triangulated" points
filled rectangles = "resectioned" images (solved pose)

Incremental SFM: **grow reconstruction**

1. Resection: solve pose for image(s) that have the most triangulated points
2. Triangulate: solve for any new points that have at least two cameras
3. Remove 3D points that are outliers
4. Bundle adjust
   - For speed, only do full bundle adjust after some percent of new images are resectioned
5. Optionally, align with GPS from EXIF or ground control points (GCP)



filled circles = “triangulated” points
filled rectangles = “resectioned” images (solved pose)

Important recent papers and methods for SfM

- Snavely thesis (2008): intro to SfM in Chapter 3

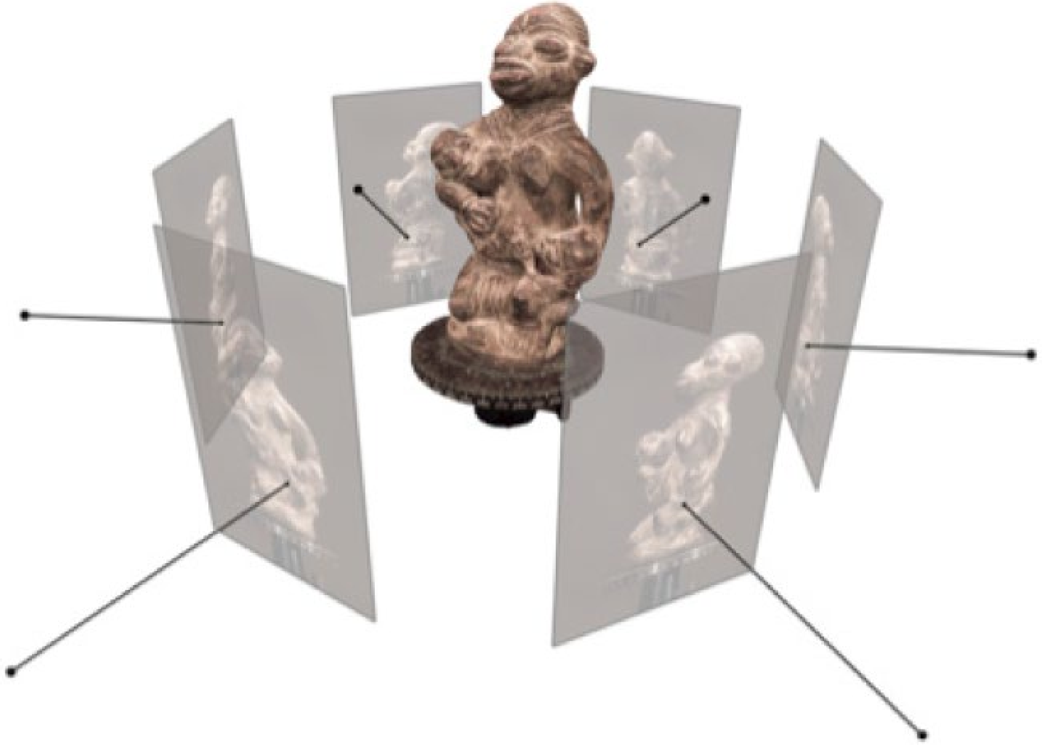
- Visual SfM: Visual SfM (Wu 2013)
  - Used to be the best incremental SfM software (but not anymore and closed source); paper still very good

- COLMAP
  - Good open source system based on "Structure-from-motion revisited" (Schonberger Frahm 2016)

- OpenSfM:
  - Python open-source system, easy to read and modify

Reconstruction of Cornell (Crandall et al. ECCV 2011)

Multiview Stereo: propose and verify 3D points by matching pixel patches across images



Select depth at each pixel that minimizes NCC of patches with other images

Key Assumptions

- Enough texture to match
- Surface looks the same from each view (non-reflective)

Figure from Furukawa & Hernandez (2015)

Multiview Stereo: recommended reading

"Multiview Stereo: a tutorial" by Yasu Furukawa

http://www.cse.wustl.edu/~furukawa/papers/fnt_mvs.pdf

COLMAP:

- Code based on "Pixelwise View Selection for Unstructured Multi-View Stereo" by Schonberger et al. 2016

Surface Reconstruction

Floating scale surface reconstruction:

http://www.gcc.tu-darmstadt.de/home/proj/fssr/

Constrained Delaunay triangulation

- Create 3D triangulation of dense points and remove faces that conflict with observed points

CVPR 2018

Deep Image Prior

Dmitry Ulyanov
Skolkovo Institute of Science and Technology, Yandex
dmitry.ulyanov@skoltech.ru

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

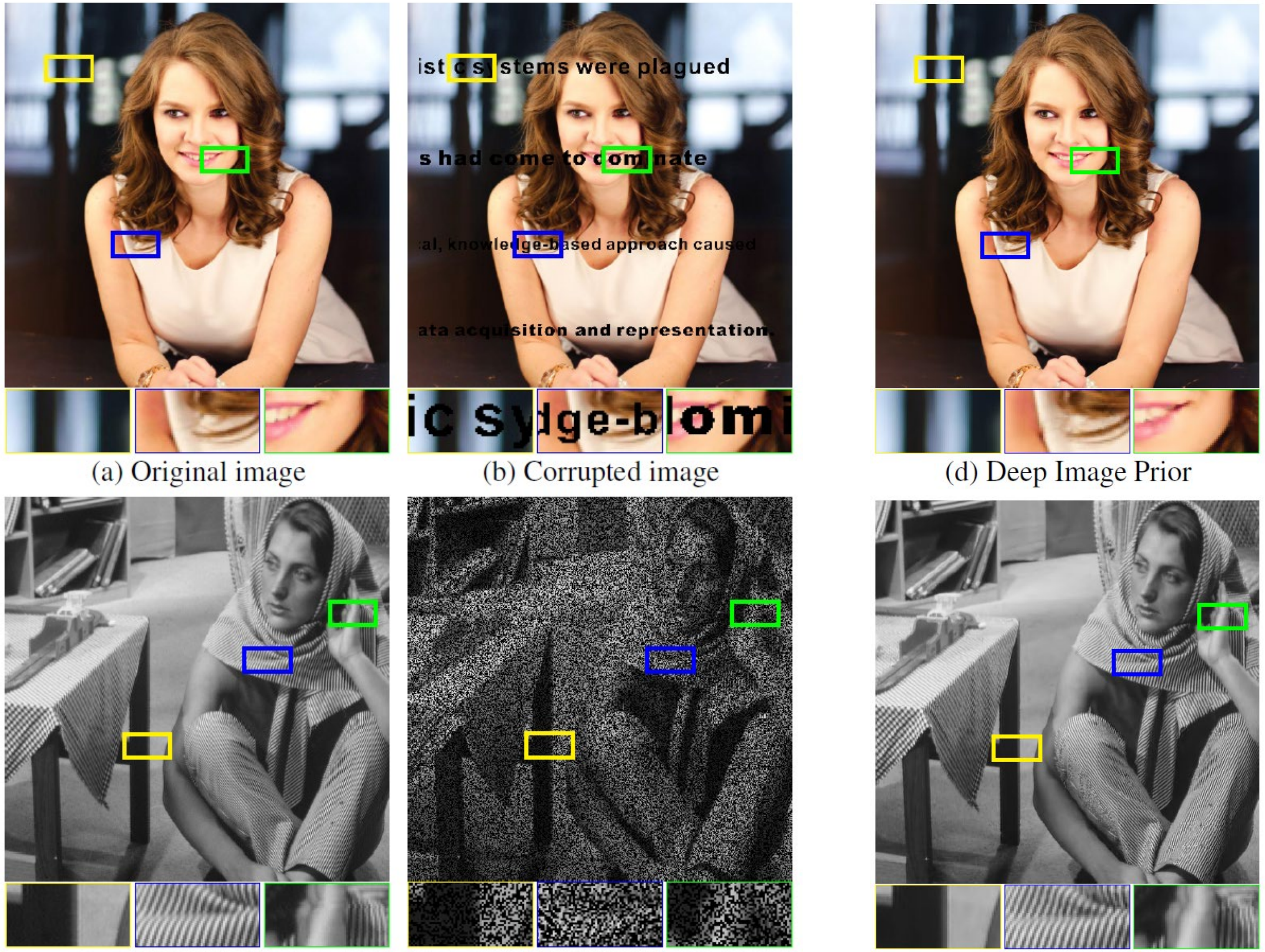
Victor Lempitsky
Skolkovo Institute of Science and Technology (Skoltech)
lempitsky@skoltech.ru

Surprising result: A randomly initialized decoder network, when trained to reproduce a corrupted image, fixes the noise, holes, etc.

The network structure acts as a prior!

(a) Corrupted image (b) Global-Local GAN [15] (c) Ours, LR = 0.01

Magic or math? Gradient descent on encoder network to reproduce Original produces a cleaner image. Even better than recent methods designed to solve this problem.



(a) Original image (b) Corrupted image (d) Deep Image Prior

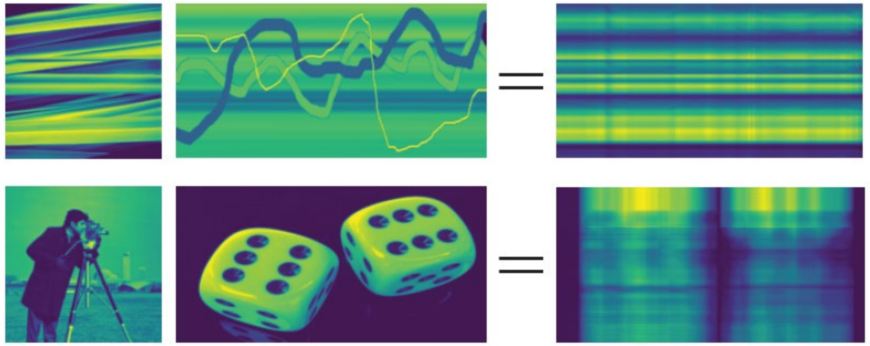
(e) Original image (f) Corrupted image (h) Deep Img. Prior, PSNR = 32.22

Computational Mirrors: Blind Inverse Light Transport by Deep Matrix Factorization

NIPS 2019

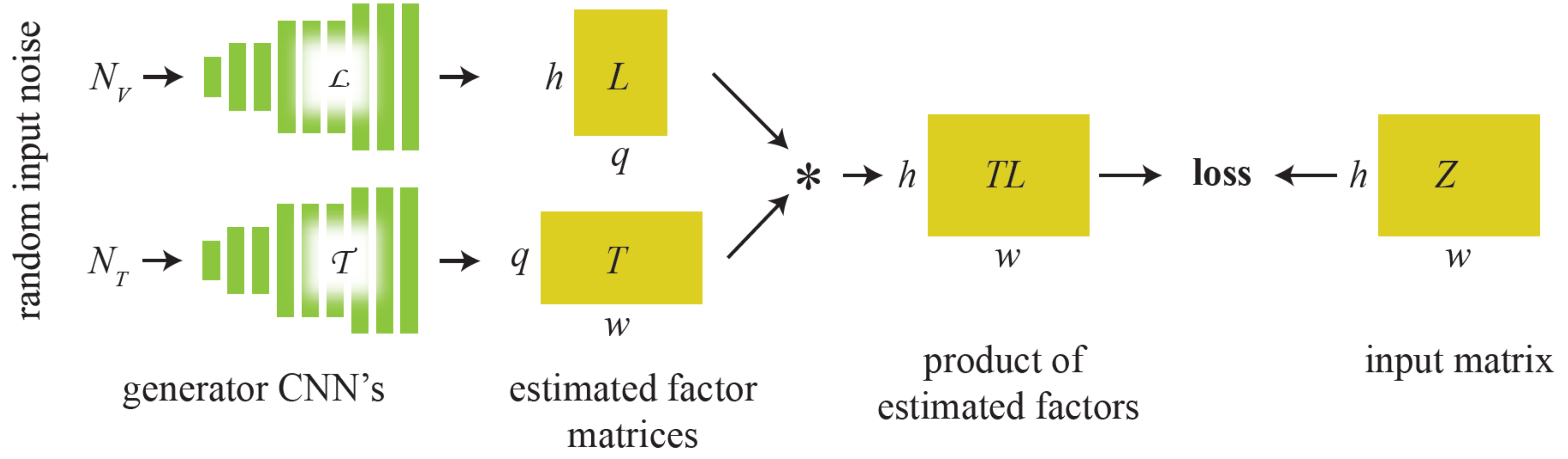
Miika Aittala MIT miika@csail.mit.edu
Prafull Sharma MIT prafull@mit.edu
Lukas Murmann MIT lmurmann@mit.edu
Adam B. Yedidia MIT adamy@mit.edu
Gregory W. Wornell MIT gww@mit.edu
William T. Freeman MIT, Google Research billf@mit.edu
Frédo Durand MIT fredo@mit.edu

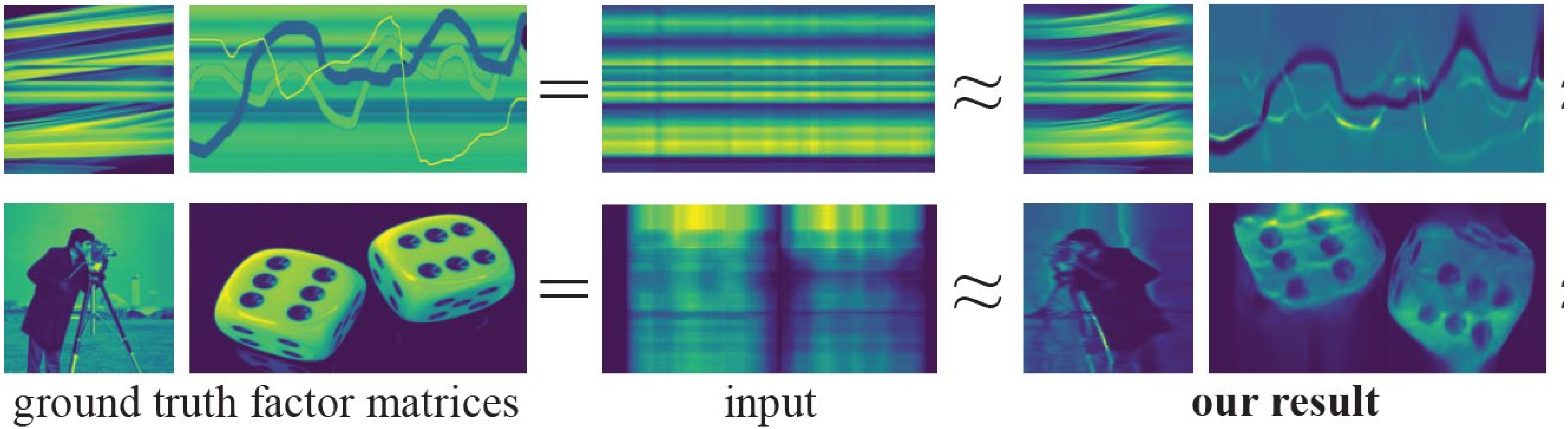
Now take it a step further. If you have the product of two images, you can recover the factors.



ground truth factor matrices input

Note: there are practically infinitely many useless solutions to this problem.


random input noise
N_V
N_T
$\mathcal{L}$
$\mathcal{T}$
generator CNN's
h L q
q T w
estimated factor matrices
$*$
h TL w
product of estimated factors
loss
h Z w
input matrix


=
≈
=
≈
ground truth factor matrices
input
our result

- Each “pixel” of light on the projector lights the scene, producing an image
- The total image is the sum of images from each pixel.
- Observed image can be factorized into surface colors and projected image (assuming no ambient light)

https://www.youtube.com/watch?v=bzsfREU2dDM

Event cameras

- First commercially produced in 2008
- Respond only when individual pixels change intensity
  - Corresponds to camera or scene motion
- 1 micro-second latency
- High dynamic range
- 100x less power than standard camera

Overview: https://www.youtube.com/watch?v=LauQ6LWTkxM
3D Reconstruction: https://www.youtube.com/watch?v=fA4MiSzYHWA

Two more

- Handwriting beautification (Zitnick SG’13)
  - Example of user assistance
- Semantic image synthesis (Park et al. CVPR 2019)

Trends and Future of Computational Photography

- Camera phones continue to serve as a platform for latest advances in hardware and software
  - Depth may be commonly available
- VR / AR blend graphics with tracking and understanding of environment
  - Killer app outside of games and teleconferencing?
- Design smart programs that work together with people
  - This is #1 from Harry Shum, Exec VP of AI and Research at Microsoft

How can you learn more?

- Relevant courses
  - Production graphics (CS 419)
  - Machine learning (CS 446 and others)
  - Deep learning
  - Computer vision (CS 543)
  - Optimization methods (CS 544)
  - Parallel processing / GPU
  - HCI, data mining, NLP, robotics

How can you learn more?

- Conference proceedings
  - Vision: CVPR, ICCV, ECCV, NIPS
  - Computational photography: ICCP
  - Graphics: SIGGRAPH, SIGGRAPH Asia

Computer Vision (with Prof Gupta Spring 2020)

Similar stuff to CP

- Camera models, filtering, single-view geometry, light and capture

New stuff

- Mid-level vision
  - Edge detection, clustering, segmentation
- Machine learning
- Recognition
  - Image features and classifiers
  - Object category recognition
  - Action/activity recognition
- Videos
  - Tracking, optical flow
  - Structure from motion
- Multi-view geometry

How do you learn more?

Explore and fiddle!

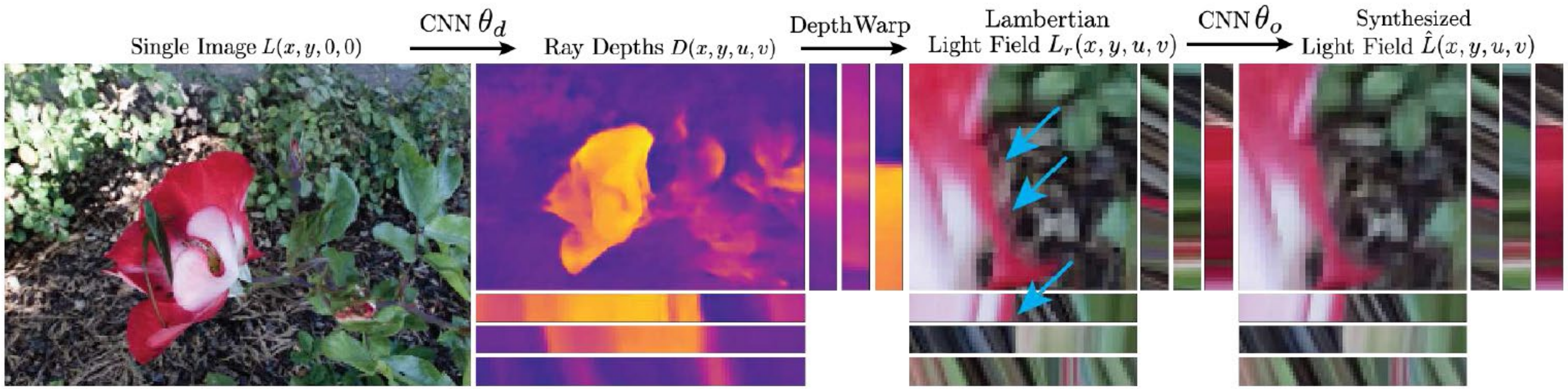
Thank you!

ICES forms

Image → Light Field

Learning to Synthesize a 4D RGBD Light Field from a Single Image

Pratul P. Srinivasan[1], Tongzhou Wang[1], Ashwin Sreelal[1], Ravi Ramamoorthi[2], Ren Ng[1]
[1]University of California, Berkeley [2]University of California, San Diego



https://www.youtube.com/watch?v=yLCvWoQLnms

Superresolution

EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis

Mehdi S. M. Sajjadi Bernhard Schölkopf Michael Hirsch

Bicubic ENet-E ENet-PAT Ground Truth

E: Optimize least squares objective with upsampling network
PAT: Optimize “perceptual” (VGG features) loss, adversarial loss, texture corr loss

(a) Input (b) SR [18] (c) SR [18]+Deblur [33] (d) Deblur [33] (e) Deblur [33]+SR [18] (f) Ours (g) GT

Learning to Super-Resolve Blurry Face and Text Images

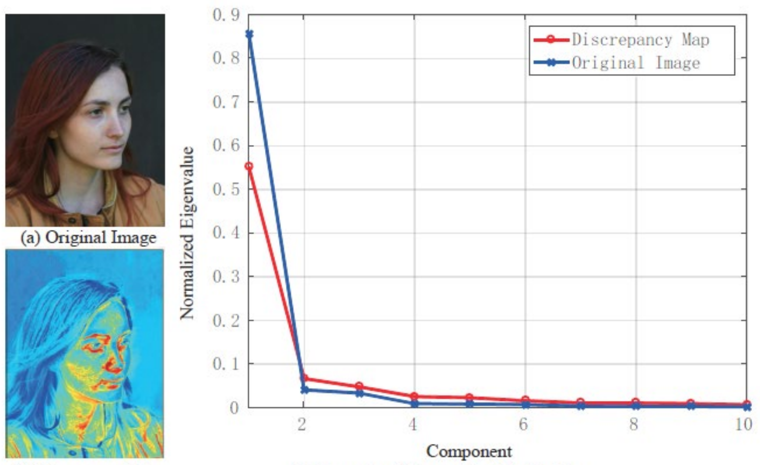
Xiangyu Xu[1,2,3] Deqing Sun[3,4] Jinshan Pan[5] Yujin Zhang[1]
Hanspeter Pfister[3] Ming-Hsuan Yang[2]
[1]Tsinghua University [2]University of California, Merced [3]Harvard University
[4]Nvidia [5]Nanjing University of Science & Technology

Pretty similar to above, more limited domain

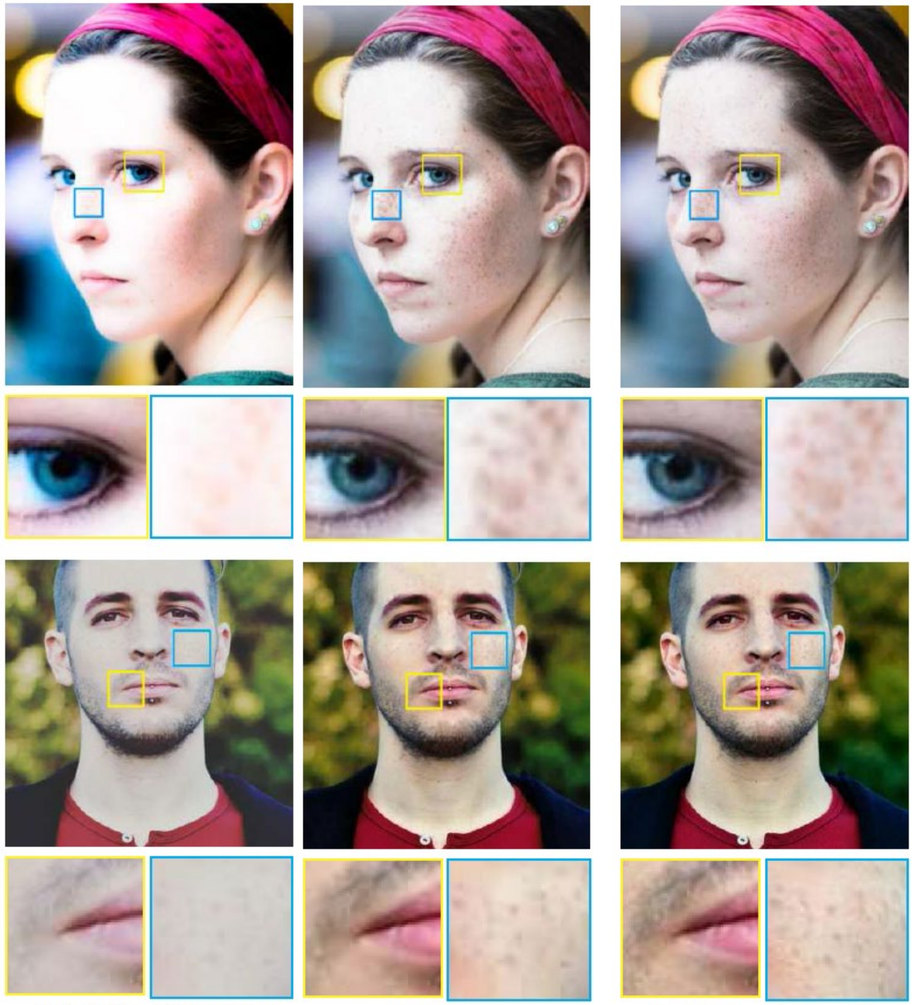
De-beautification

Makeup-Go: Blind Reversion of Portrait Edit*

Ying-Cong Chen[1] Xiaoyong Shen[2] Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong [2]Tencent Youtu Lab
ycchen@cse.cuhk.edu.hk dylanshen@tencent.com leojia9@gmail.com



(a) Original Image (b) Discrepancy Map (c) Normalized Eigenvalue Distribution



(a) Edited Image (d) Ours (e) Ground truth

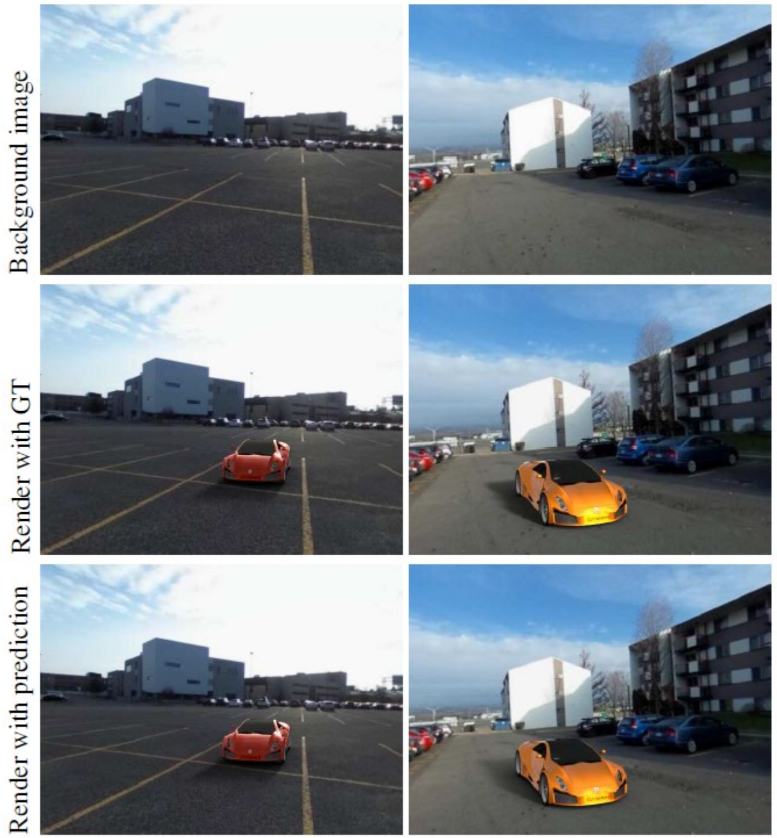
Network regresses principal components of discrepancy map

LDR --> HDR

Learning High Dynamic Range from Outdoor Panoramas

Jinsong Zhang Jean-François Lalonde
Université Laval, Québec, Canada
jinsong.zhang.1@ulaval.ca, jflalonde@gel.ulaval.ca
http://www.jflalonde.ca/projects/learningHDR

- Regress HDR from one LDR image
- Train on synthetic data
- Limited to outdoor scenes, rotated so that sun is on top



Smarter user assistance

- Handwriting beautification (Zitnick SG’13)
- 3D object modeling (Chen et al. SGA’13)
- 3D object modeling (Kholgade et al. SG’14)

Video and motion

- Video = sequence of images
  - Track points → optical flow, tracked objects, 3D reconstruction
  - Find coherent space-time regions → segmentation
  - Recognizing actions and events
- Examples:
  - Point tracking for structure-from-motion
    - Boujou 1
  - Facial transfer: Xu et al. SG2014

Scene understanding

Interpret image in terms of scene categories, objects, surfaces, interactions, goals, etc.

- Remove the guy lying down (Alyosha)
- Make the woman dance or the guy get up
- Fill in the window with bricks
- Find me images with only Alyosha and Piotro

Scene understanding

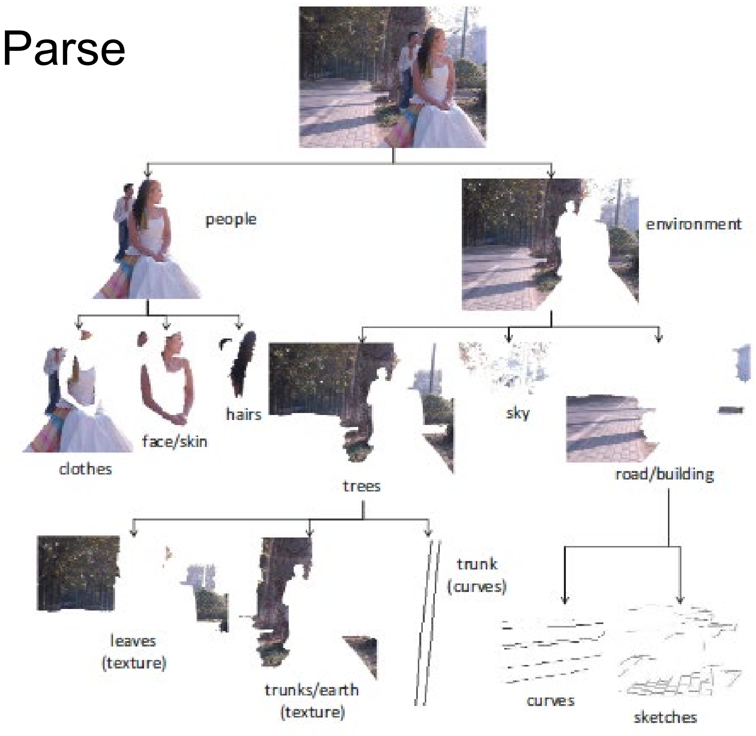
- Mostly unsolved, but we’re getting there (especially for graphics purposes)
- Examples
  - “From Image Parsing to Painterly Rendering” (Zeng et al. 2010)
  - “Sketch2Photo: Internet Image Montage” (Chen et al. 2009)
  - Editing via scene attributes (Laffont et al. 2014)

Image Parsing to Painterly Rendering

Zeng et al. SIGGRAPH 2010

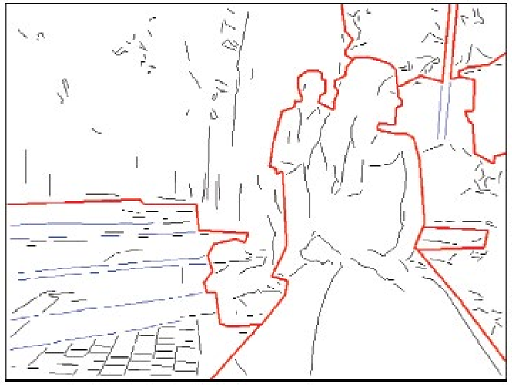
Image Parsing to Painterly Rendering

Parse



(a)

Sketch



Brush Orientations

Brush Strokes

Zeng et al. SIGGRAPH 2010

Image Parsing to Painterly Rendering

Zeng et al. SIGGRAPH 2010

Image Parsing to Painterly Rendering

More examples

- Sketch2photo: http://www.youtube.com/watch?v=dW1Epl2LdFM

- Animating still photographs

Chen et al. 2009

Modeling humans

- Estimating pose and shape
  - http://clothingparsing.com/
  - Parselets (Dong et al., ICCV 2013)

- Motion capture
- 3D face from image (Kemelmacher ICCV'13)

Better and simpler 3D reconstruction

MobileFusion (2015): https://youtu.be/8M_-lSYqACo