Computer Vision

Lecture 01 - Introduction

“a professor teaching a class in computer vision” – StableDiffusion Jan’24

Computer Vision

Lecture 01 - Introduction

“a professor teaching a class in computer vision” – FluxAI “Schnell” Jan’25

Computer Vision

Lecture 01 - Introduction

“a professor teaching a class in computer vision” – FluxAI “Schnell” Jan’25

Course Structure

- 20 lectures
  - pptx on moodle the day before the lecture.
- 4 classes
- 3+1 practicals
- Sit-down (closed book) exam

Classes and Practicals

- 4 classes in 4 groups – please distribute!
  - Mostly maths and conceptual questions
  - Will be similar to the exam questions

- 4 practicals with 3 sheets
  - Mostly code

Practicals

1. Image filtering and transformations
2. Image Classification & Interpretability
3. Object Detection & Diffusion
4. Coordinate Networks and Representation Learning (optional)

Signing off will normally happen in the last half hour of each session or at the beginning of the following one.

As usual, when checking your work, the demonstrator will want to see a working version of the program in action, as well as appropriate commenting of your code and sketches indicating the design steps. Try to make your report as concise as possible, perhaps in the form of appropriate comments to your code.

Books

- Multiple View Geometry in Computer Vision
  Richard Hartley and Andrew Zisserman, Cambridge University Press, link

- Deep Learning
  Ian Goodfellow, Yoshua Bengio and Aaron Courville. MIT Press 2016, link

- Pattern Recognition and Machine Learning
  Christopher M Bishop, Springer 2007, link

Python

- python 3 (typical packages: opencv-python, numpy, sci-py, matplotlib, pytorch, ...)
- Everything runs inside Google Colab notebooks https://colab.research.google.com
- Alternative: you can run things locally on your laptop
  - You will need to manage your environment yourself (use conda)
  - Demonstrators will not have time to help you with setup problems
- Most lectures come with a Colab notebook containing code for the examples in the slides.

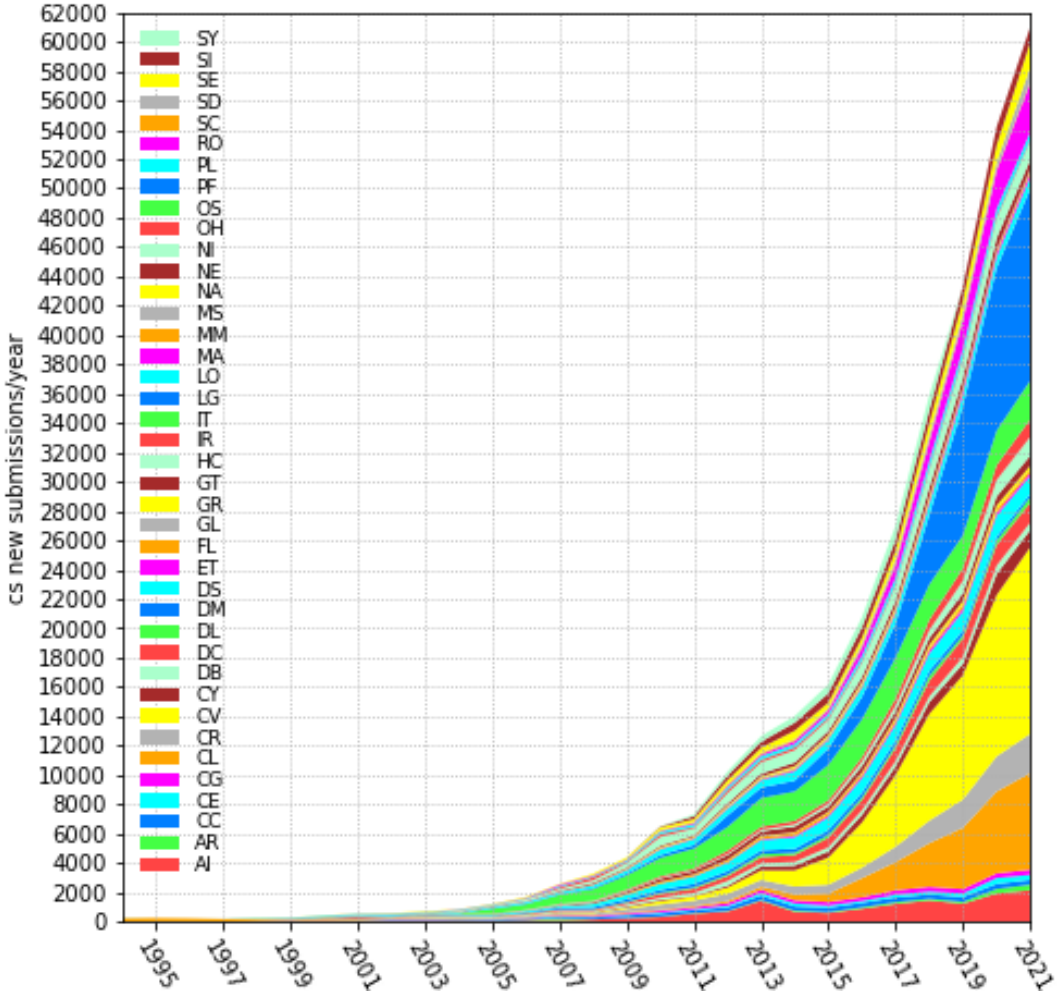
This Lecture Today

- Should give you a general understanding of what kind of problems we think about in CV.
- Will introduce some historical context and how things evolved to where we are.
- Might motivate why we will look at fundamentals before we dive into the flashy topics later in the course.

Computer Vision Now

Rapid progress

- arXiv/cs 200 submissions/day (2021)
- Keeping up is difficult
- Benchmarks evaluate progress for individual tasks
- Ideas translate between sub-fields

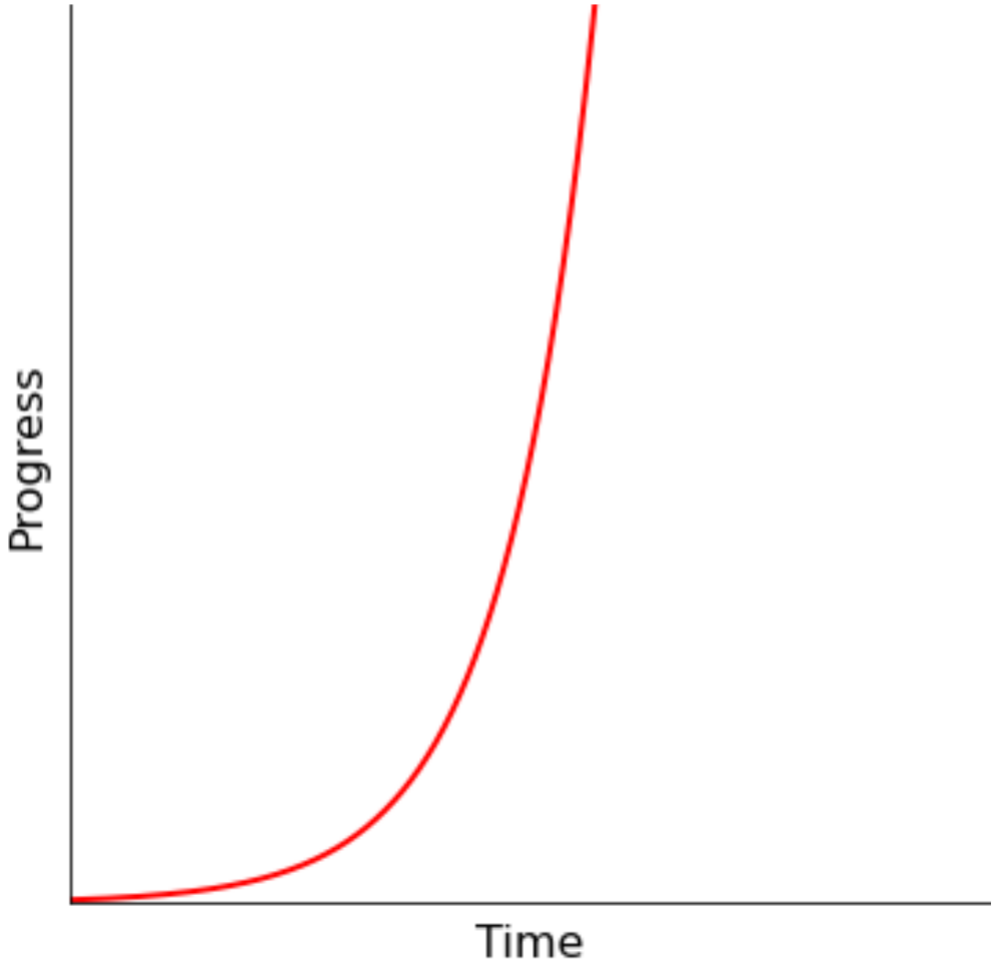


Progress in Computer Vision

- Currently: 4 main drivers of improvement:
  - Data
  - ML improvements (optimisers, architectures, etc.)
  - Engineering/tuning (often task specific)
  - Good ideas (rare compared to other factors)

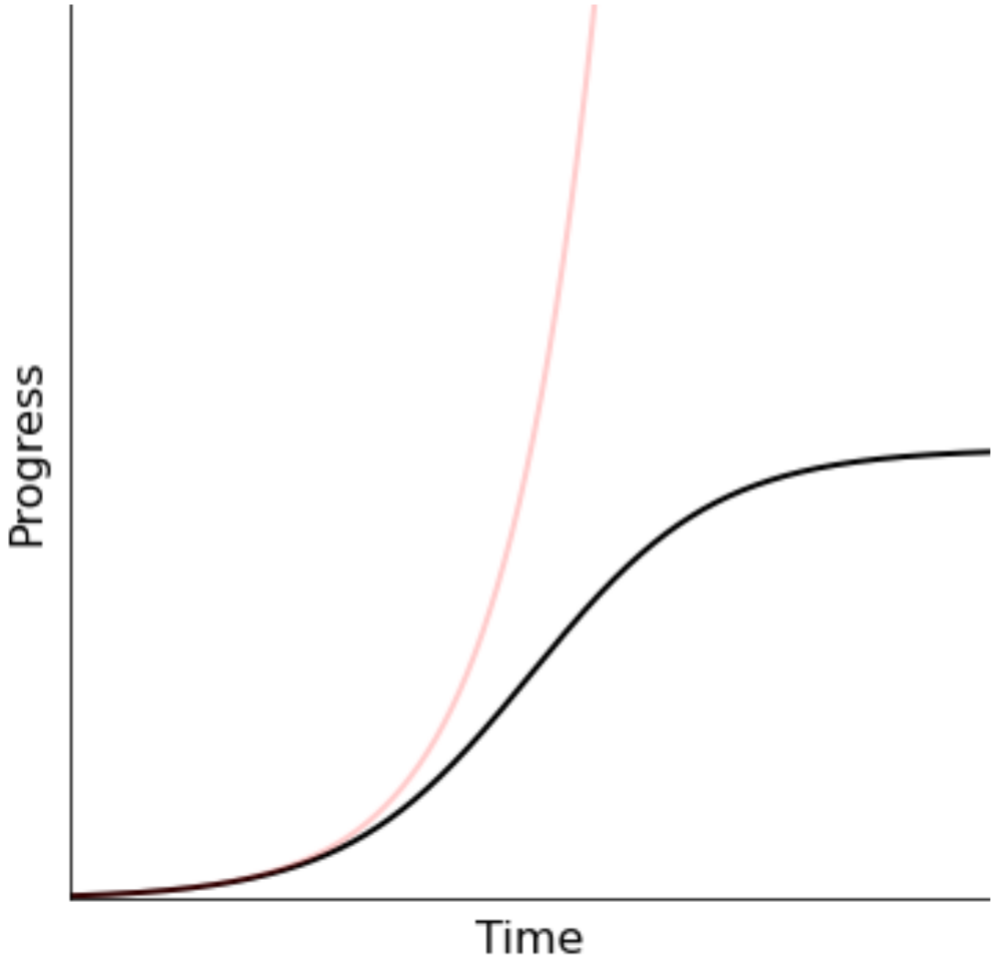
- Papers often contain a mixture
- This course: (mostly) good ideas
  -> easier to judge for older approaches

Where is Computer Vision now?



Slide credits: John Barron – Scholars and Big Models Workshop, CVPR 2023

Where is Computer Vision now?

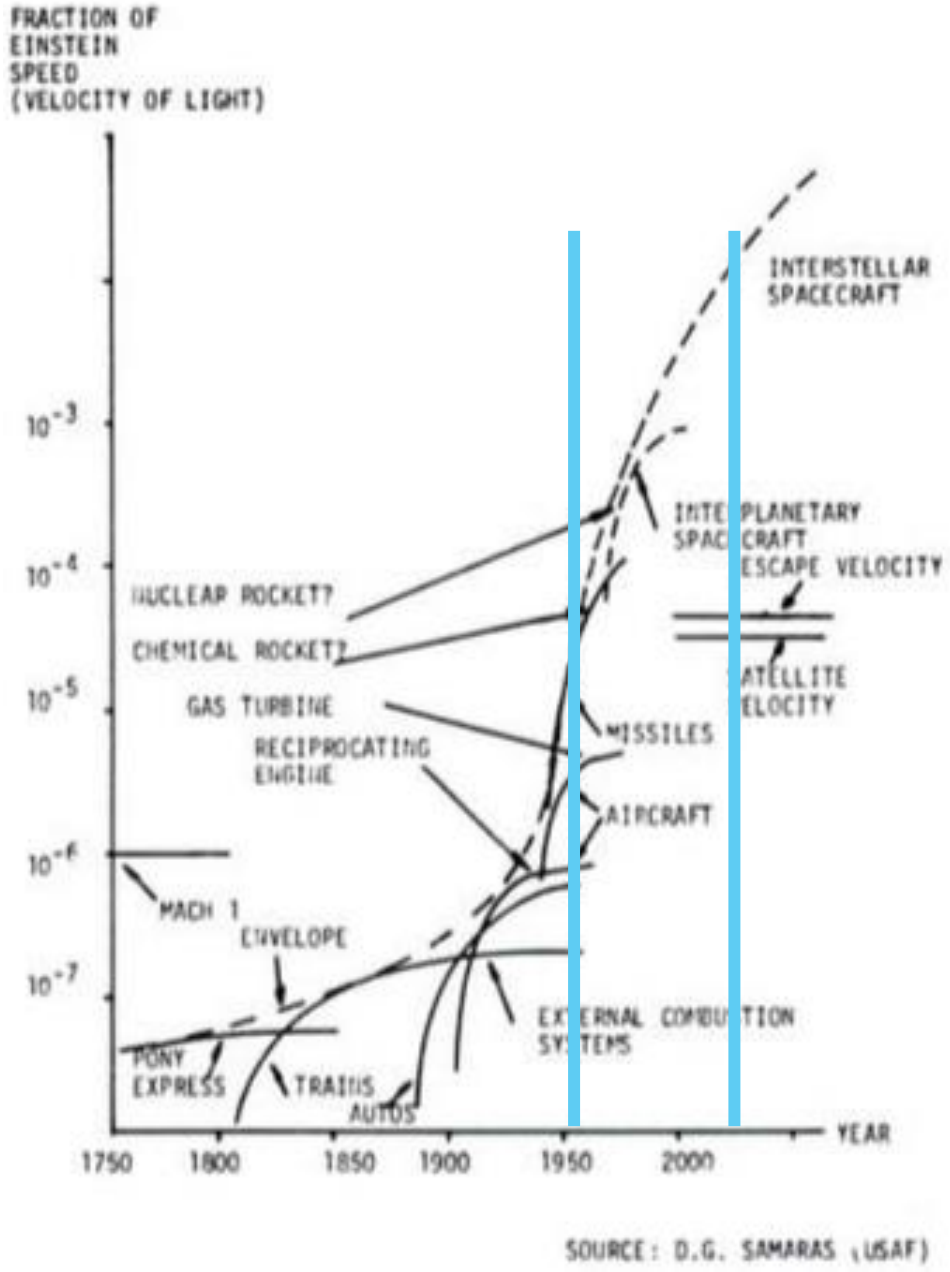


Slide credits: John Barron – Scholars and Big Models Workshop, CVPR 2023

Rocket speed - 1953

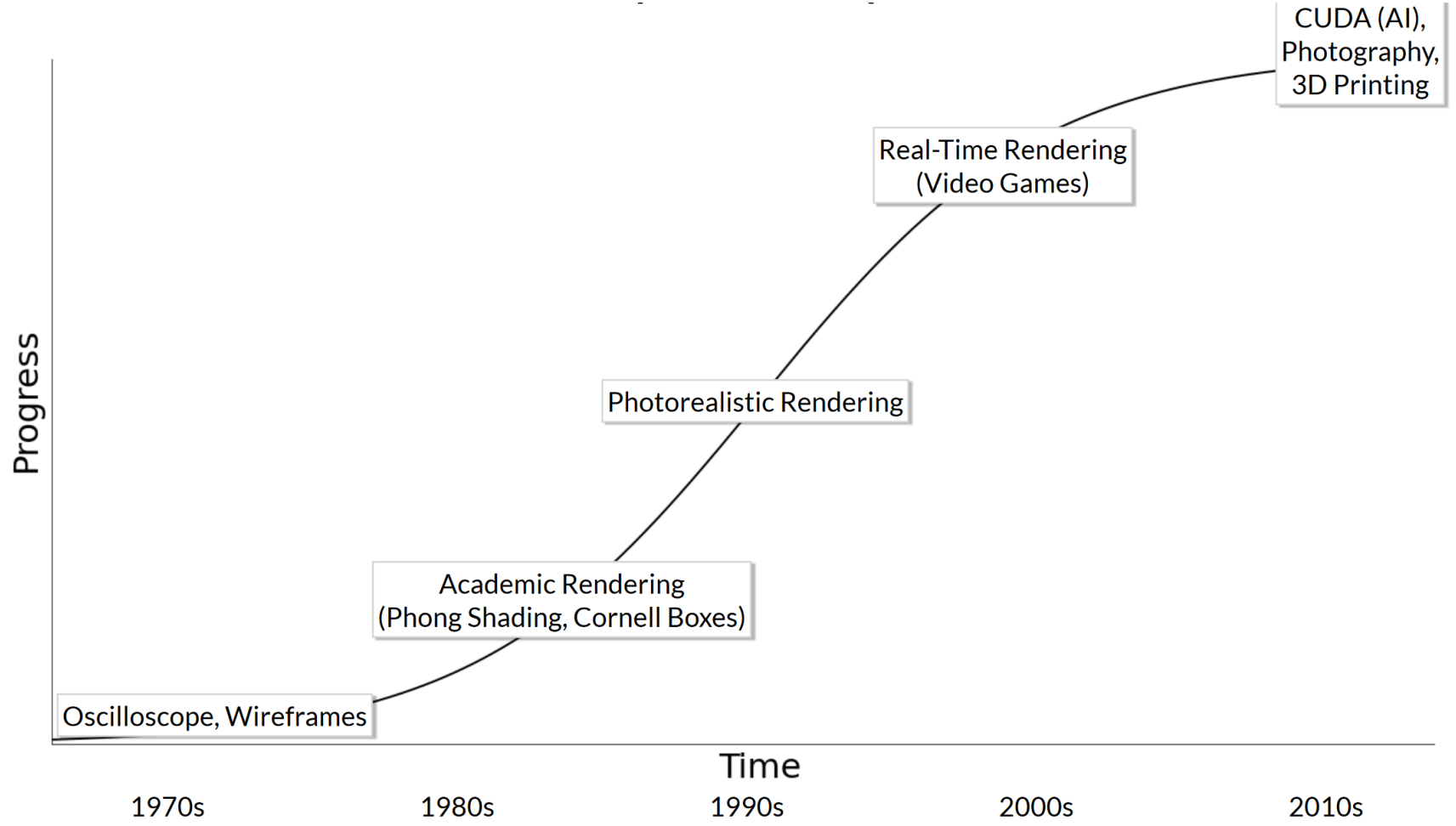
- USAF prediction for rocket speed
- Time vs. fraction of light speed
- Vastly overestimates progress

- Nuclear rockets
- Interstellar travel



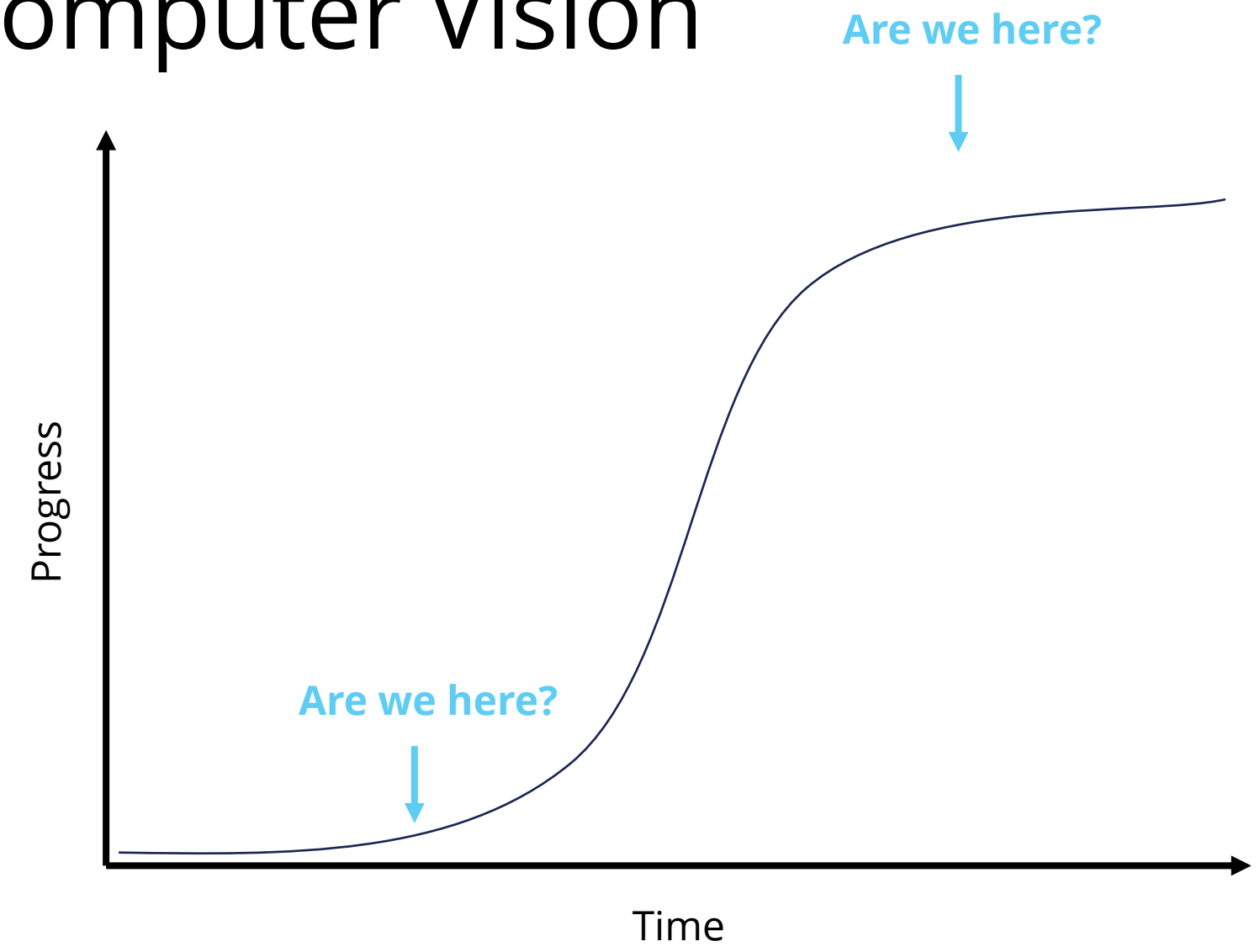
Slide credits: John Barron – Scholars and Big Models Workshop, CVPR 2023

Computer Graphics



Slide credits: John Barron – Scholars and Big Models Workshop, CVPR 2023

Computer Vision



Computer Vision

- It is very difficult to predict where we are
- Towards the end of this curve progress transitions from academia to industry
- So far, we thought we are at the end many times and were wrong (AI, self-driving, CNNs, etc.)
- It might not be a sigmoid after all!

(Recent) CV History Overview

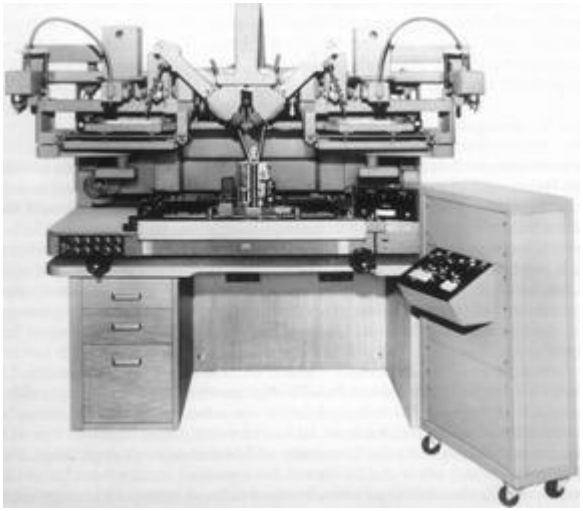
- 1960-1970: Blocks, Edges and Model Fitting
- 1970-1981: Low-level vision: stereo, flow, shape-from-x
- 1985-1988: Neural Networks, backprop., self-driving
- 1990-2000: Dense stereo and multi-view stereo, MRFs
- 2000-2010: Features, descriptors, structure-from-motion
- 2010-20??: Learning, deep learning, large datasets, rapid growth

History slides credits: Andreas Geiger - History of Computer Vision 2021 & Svetlana Lazebnik – Computer Vision: Looking Back to Look Forward

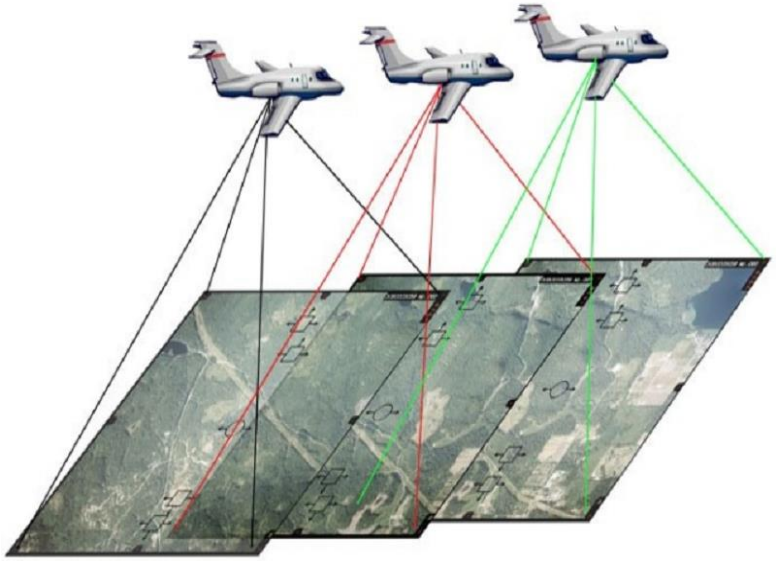
Historical Context

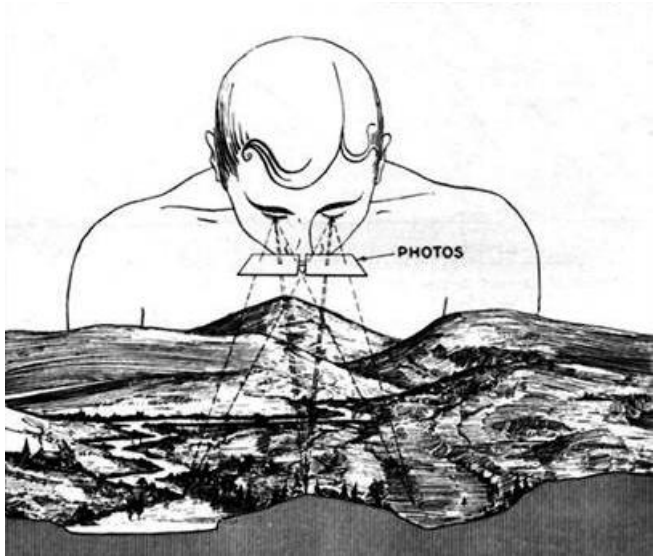
1957: Stereo(photogrammetry)

- Gilbert Hobrough: analog implementation of stereo image correlation
- Used to create elevation maps (Photogrammetry, since 1840)



Wild B8 (721x produced 1961 - 1972)

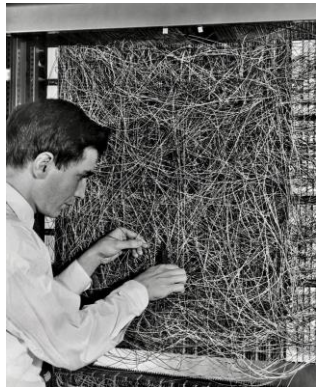




Louis, Hobrough Gilbert. "Methods and apparatus for correlating corresponding points in two images." U.S. Patent No. 2,964,642. 13 Dec. 1960

1958-1962: Rosenblatt's Perceptron

- First algorithm and implementation for training single linear "threshold neuron"
- Perceptron criterion: $L(w) = -\sum_{n \in M} w^T x_n y_n$
- Convergence proof: Novikoff



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

1958 New York Times…

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

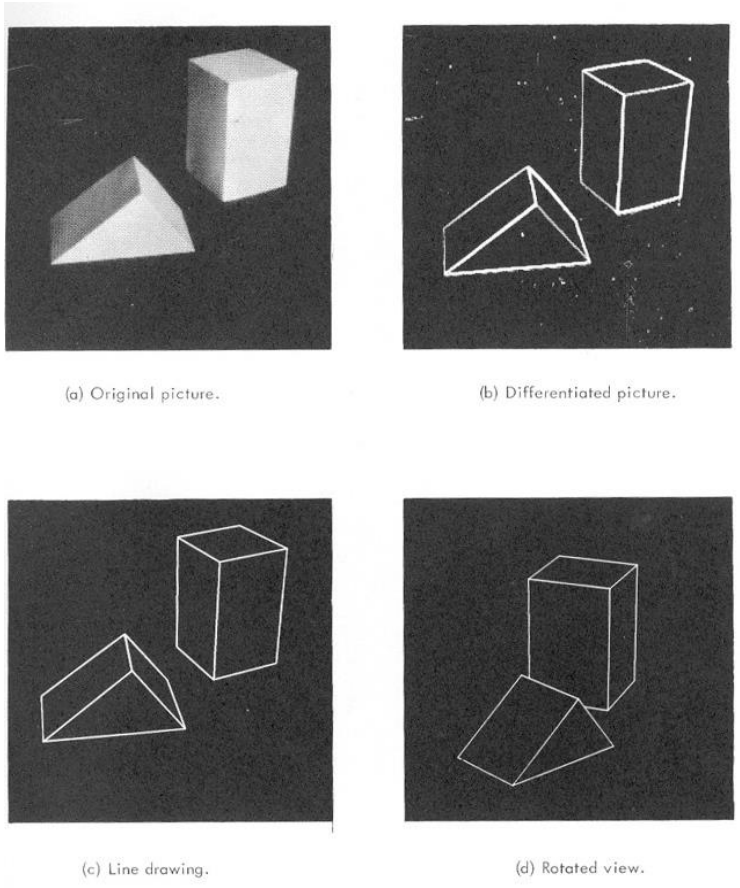
The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65.6 (1958): 386.

1963: Larry Roberts

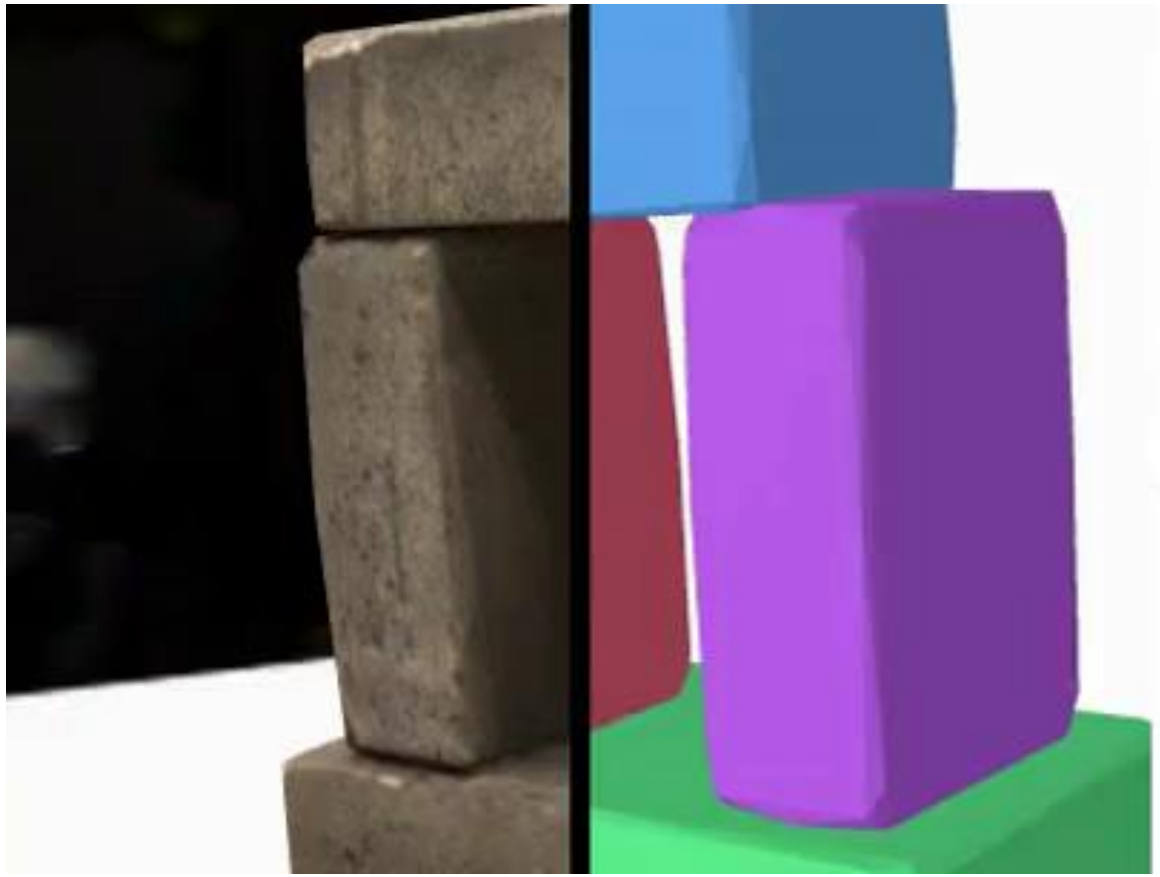
- Scene Understanding for Robotics
- Extracts edges
- Infers 3D structure from topological structure of edges

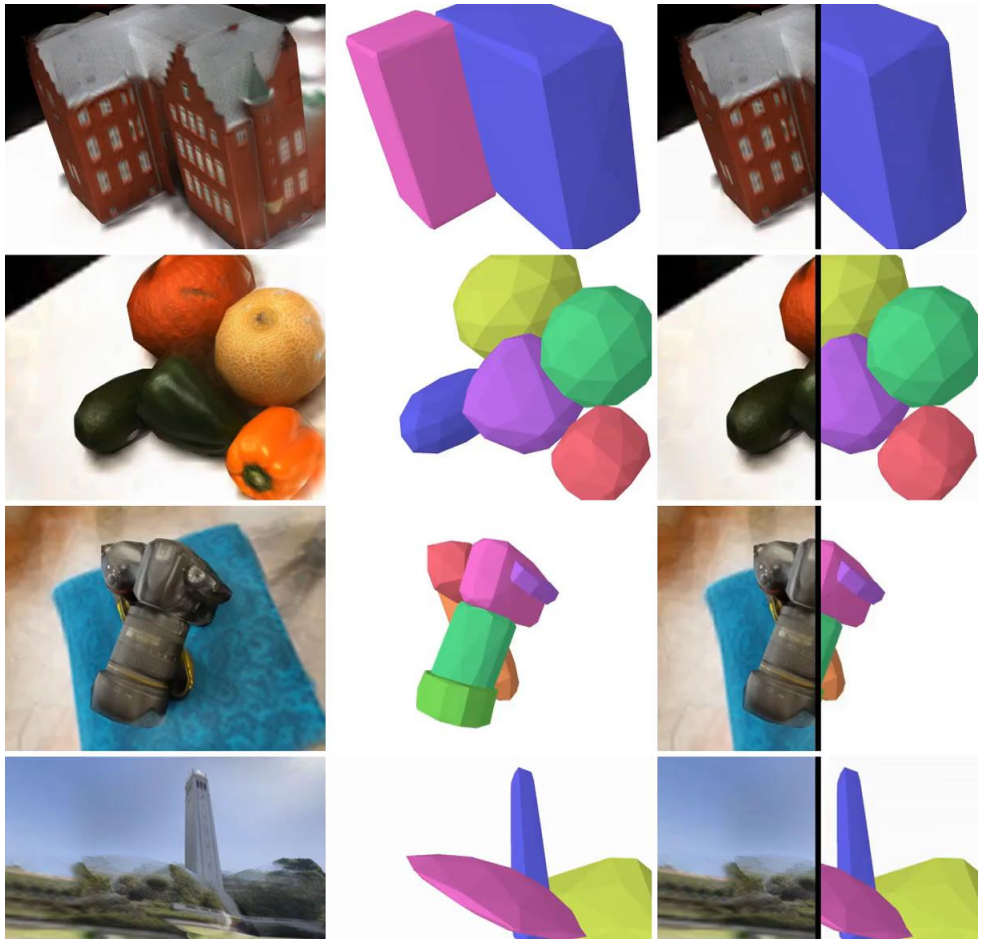
- “It is assumed that a photograph is a projection of… **known three-dimensional models**… These assumptions enable a computer to obtain a reasonable, three-dimensional description from the edge information in a photograph by means of a topological, mathematical process.”



Roberts, Lawrence G. *Machine perception of three-dimensional solids*. Diss. Massachusetts Institute of Technology, 1963.

Blocks World now

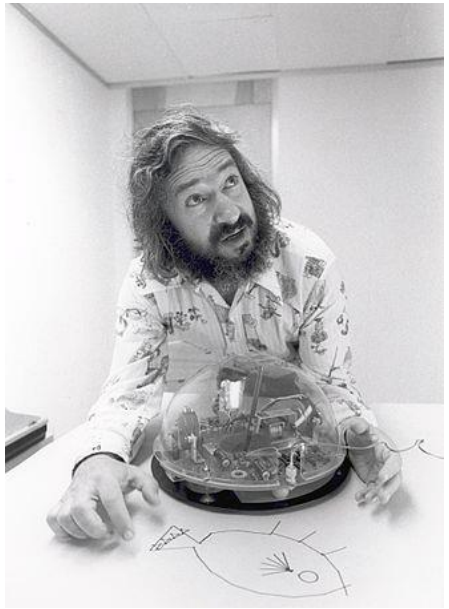




Differentiable Blocks World: Qualitative 3D Decomposition by Rendering Primitives
Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A. Efros, Mathieu Aubry, NeurIPS 2023

1966: MIT Summer Vision Project

- Solve computer vision as a summer project
- Committed to block world ideas



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

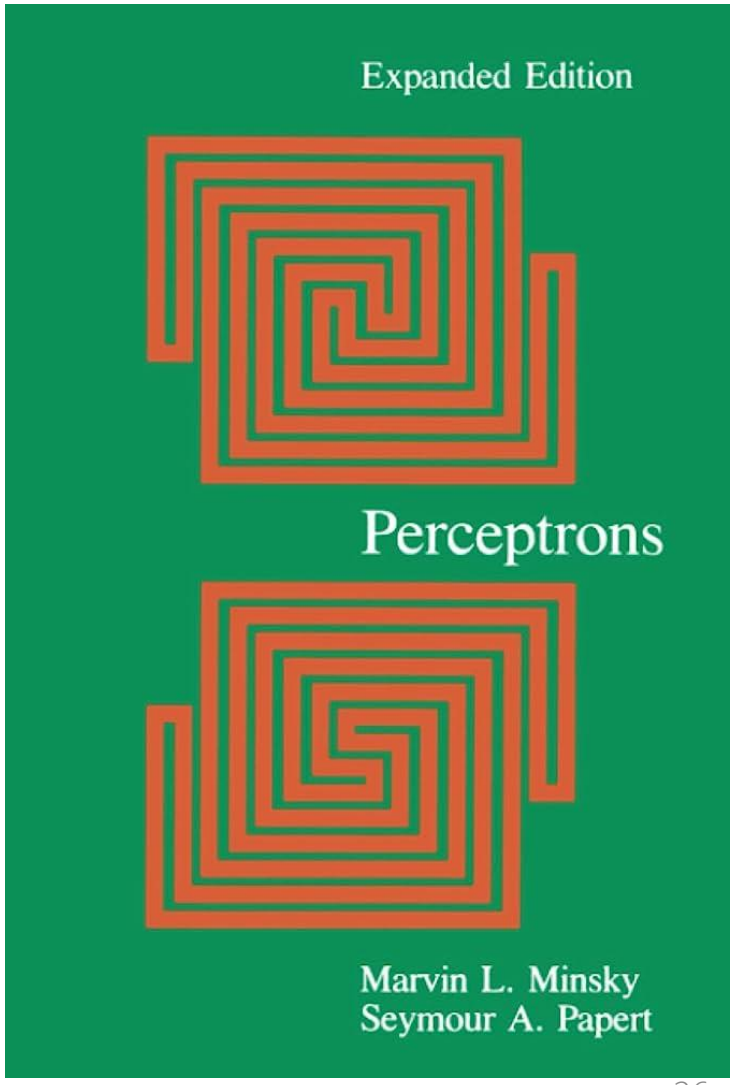
THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

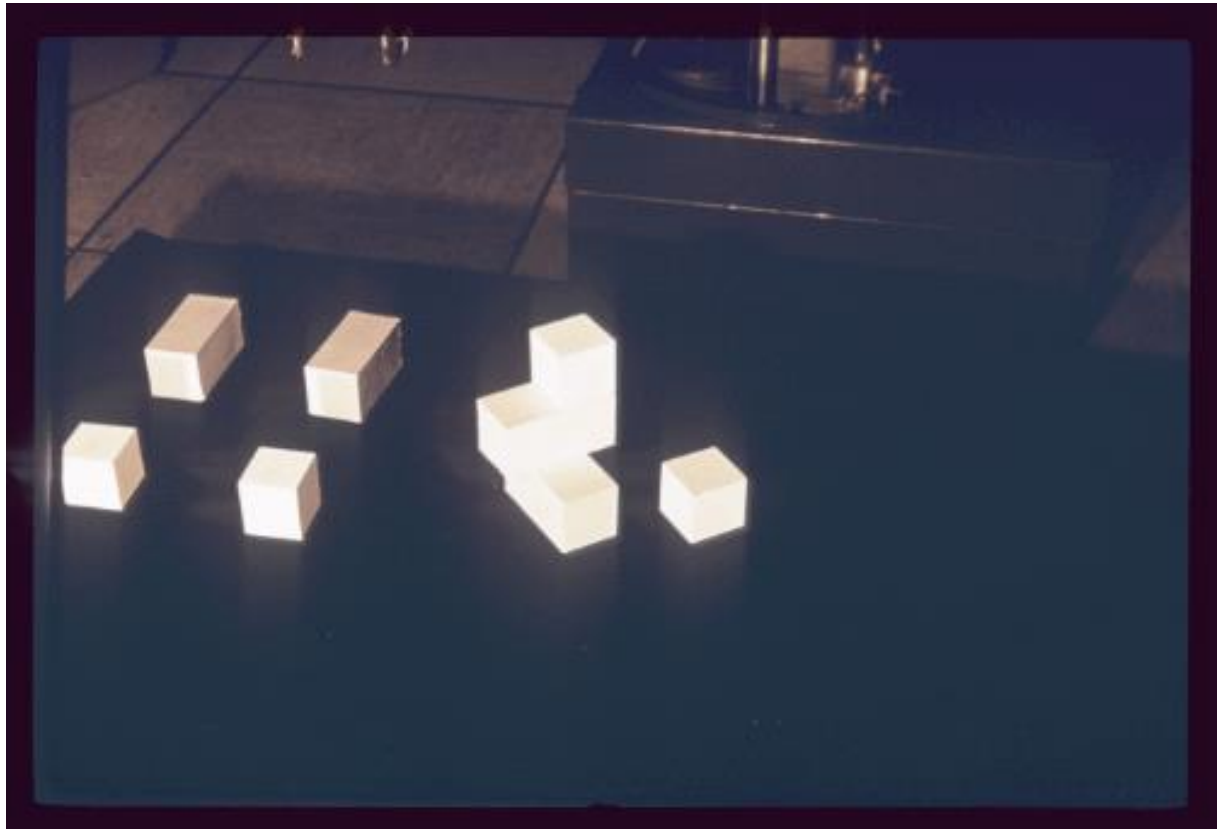
1969: Perceptrons book

- Minsky and Papert
- Several discouraging results
- *Perceptrons cannot solve the XOR problem*
- Largely contributed to the following "AI winter"
- 70s: mostly symbolic AI



1970: MIT Copy Demo

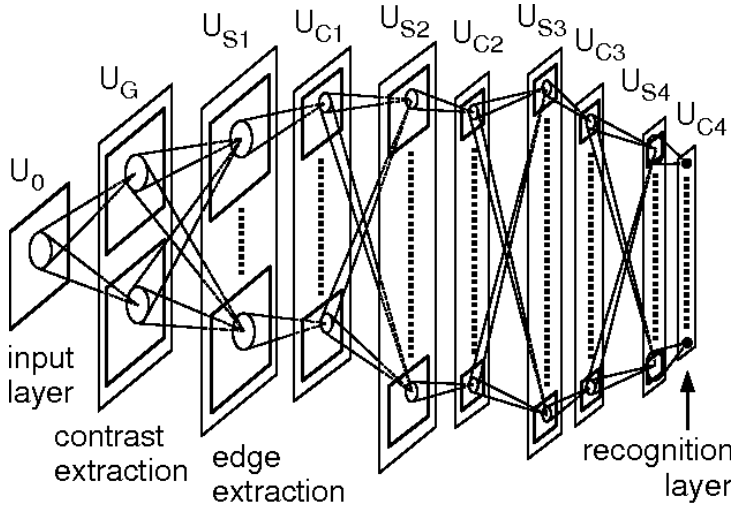
- Vision + Robotics
- Recover the structure of a scene
- Plan robot movement to copy the block arrangement
- Only works in ideal conditions
- Causes attention to robustness for low level vision tasks



Patrick Winston and the MIT AI Lab Copy Demo https://people.csail.mit.edu/bkph/phw_copy_demo.shtml

1980s: Advances in ML

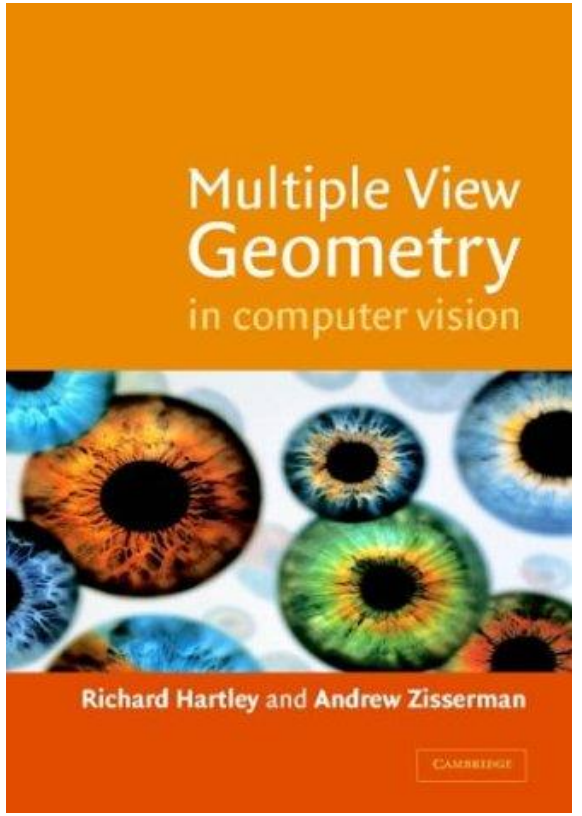
- Neocognitron: Fukushima (1980)
- Back-propagation: Rumelhart, Hinton & Williams (1986)
  - Origins in control theory and optimization: Kelley (1960), Dreyfus (1962), Bryson & Ho (1969), Linnainmaa (1970)
  - Application to neural networks: Werbos (1974)
- Parallel Distributed Processing: Rumelhart et al. (1987)
- Neural networks for digit recognition: LeCun et al. (1989)



Fukushima (1980)

1990s Theme: Geometry

- Fundamental matrix: Faugeras (1992)
- Normalized 8-point algorithm: Hartley (1997)
- RANSAC for robust fundamental matrix estimation: Torr & Murray (1997)
- Bundle adjustment: Triggs et al. (1999)
- Hartley & Zisserman book (2000)
- Projective structure from motion: Faugeras and Luong (2001)



Tracking and Optical Flow

Visual Tracking

Visual tracking involves the identification of some characteristic of the scene in successive images.

2D tracking

follow and perhaps control the image position of some entity as it moves from frame to frame over time.

3D (or pose) tracking

use image measurements (possibly involving 2D tracking) to update the 6 degrees of freedom (3 translation + 3 rotation) which define 3D pose.

2D Visual Tracking

Both, 2D and 3D tracking require a model of appearance, sufficient to identify high cross-correlation between frames.

In 2D tracking, the appearance model might relate to

- a single point
- a small patch, possibly deformable
- a contour, possibly deformable
- a line element
- ...?

Let's start simply by tracking an image point ...

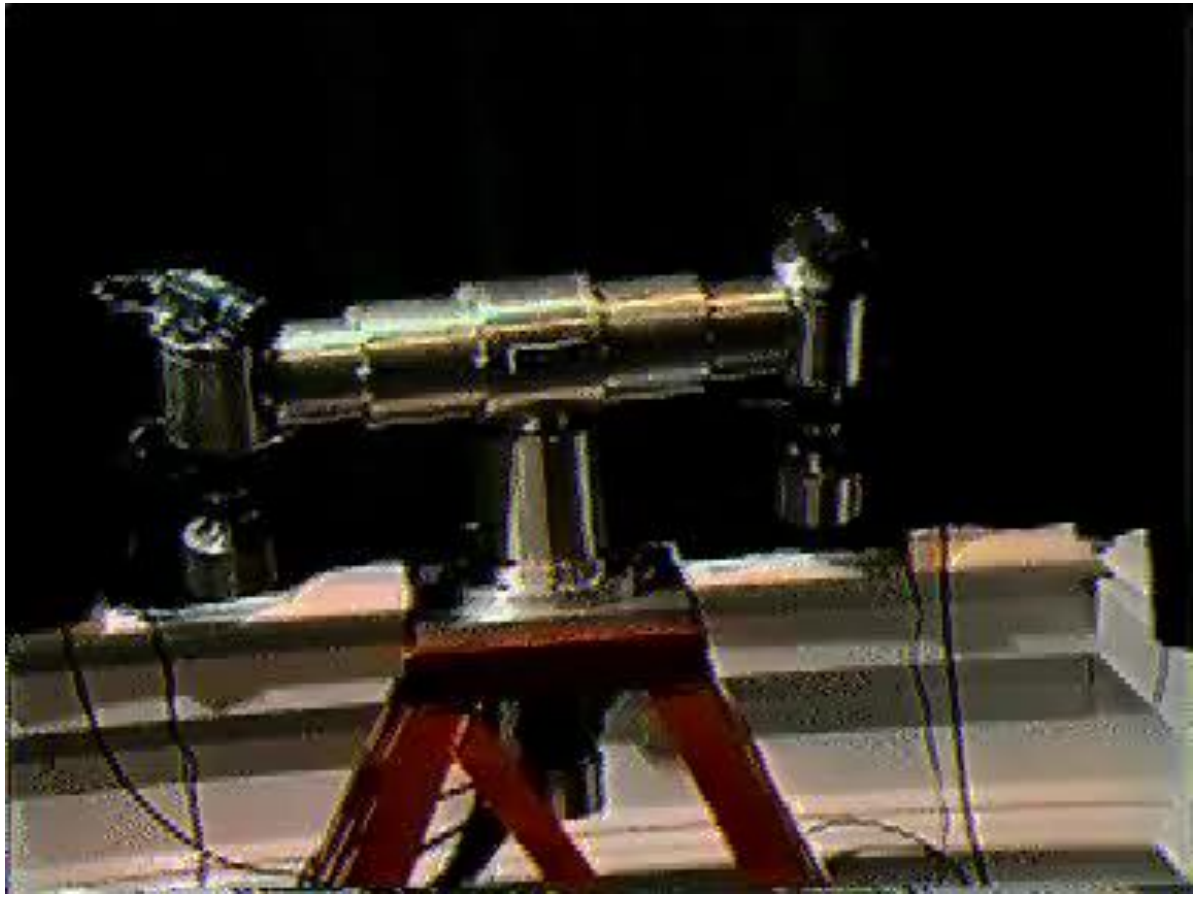
Point Tracking

- Using a bright spot detector

Not robust: we need to incorporate

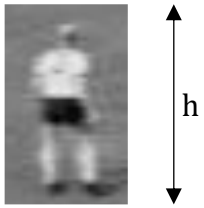
- Stronger appearance/ measurement models;
- Image and/or scene dynamics

Points -> patches



Sharkey, P., McLauchlan, P. F., Reid, I. D. and Murray, D. W.
Real-time Control of a Reactive Stereo Head/Eye Platform.
International Conference on Automation, Robotics and Computer Vision, 1992

Template Tracking



Template T

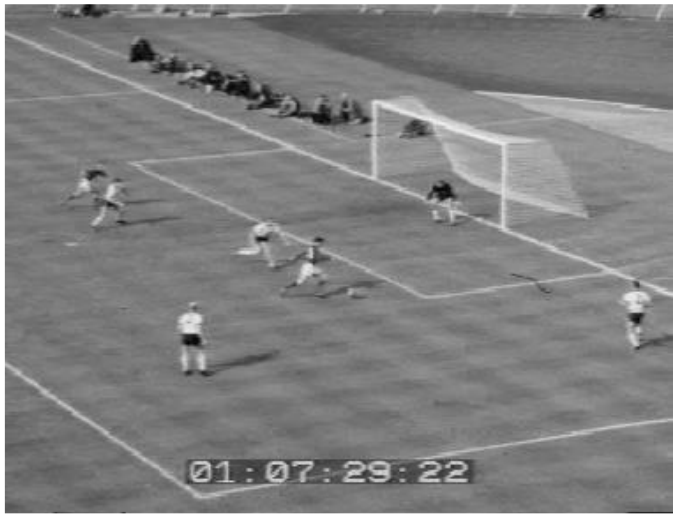
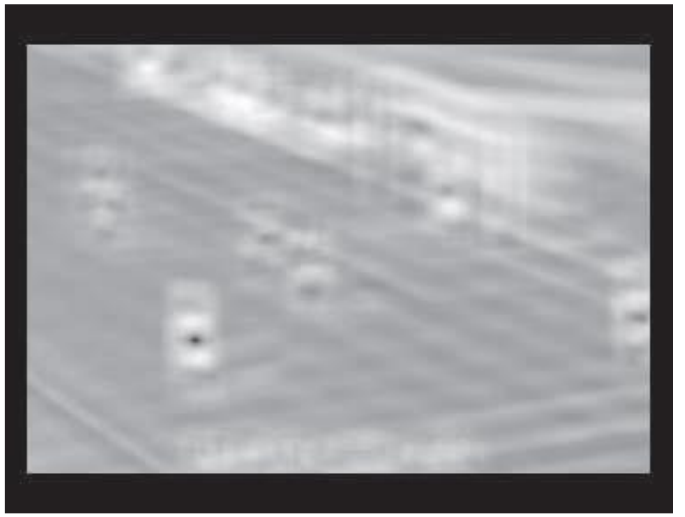


Image I

Sum of squared differences between image I and template T at every location (u, v)



Energy E

$$E(u,v) = \sum_{(x,y)\in\left[-\frac{w}{2},\frac{w}{2}\right]\times\left[-\frac{h}{2},\frac{h}{2}\right]} \left(I(u+x, v+y) - T(x,y)\right)^2$$

- Tracking by Detection (each frame processed individually)
- Very slow: (#pixels_image * #pixels_template) comparisons

1981: Lucas-Kanade Template Tracking

$$E(u,v) = \sum_{x,y} \big(I(u+x, v+y) - T(x,y)\big)^2$$

Improving the efficiency:

- Formulate search as an optimisation problem using brightness constancy as our objective function
- E is a non-convex function over the image I
- Suppose the starting point is $(t_x, t_y)^T$ e.g., detection in prev. frame
- LK searches for an update $(\delta_x, \delta_y)^T$ of the starting point

Generalised LK Tracking

- Simple update rule that iteratively refines the tracked position
- Any differentiable warp works
- Further optimization: pre-compute template gradients and warp “the other way”

- In-depth LK analysis:

  Lucas-Kanade 20 Years On: A Unifying Framework
  Simon Baker and Iain Matthews

LK Tracker Insights

- A general difficulty with trackers relying too heavily on the spatial relationships between pixels is that they are prone to break due to partial occlusion and orientation changes in the scene.

- Appearance changes can be compensated by updating the template from frame to frame
  - Can lead to “drift”:
  - The template will gradually pick up the background and eventually “stick”.
  - Can be avoided when background is simple, or
  - With a segmentation mask

2010: Tracking by Detection

- Pure tracking can fail (occlusions, drift, blurry frames, etc…)
- Idea: integrate a detector into the tracking pipeline

- Tracking-Learning-Detection (TLD) Framework (Kalal et al, 2010)
  - the target object is defined by a bounding box in a single frame.
  - a template/patch-based tracker follows the object frame to frame.
  - a random forest-based detector localizes the object and corrects the
  - tracker if needed.
  - learning is used to improve the detector.

Tracking by Detection

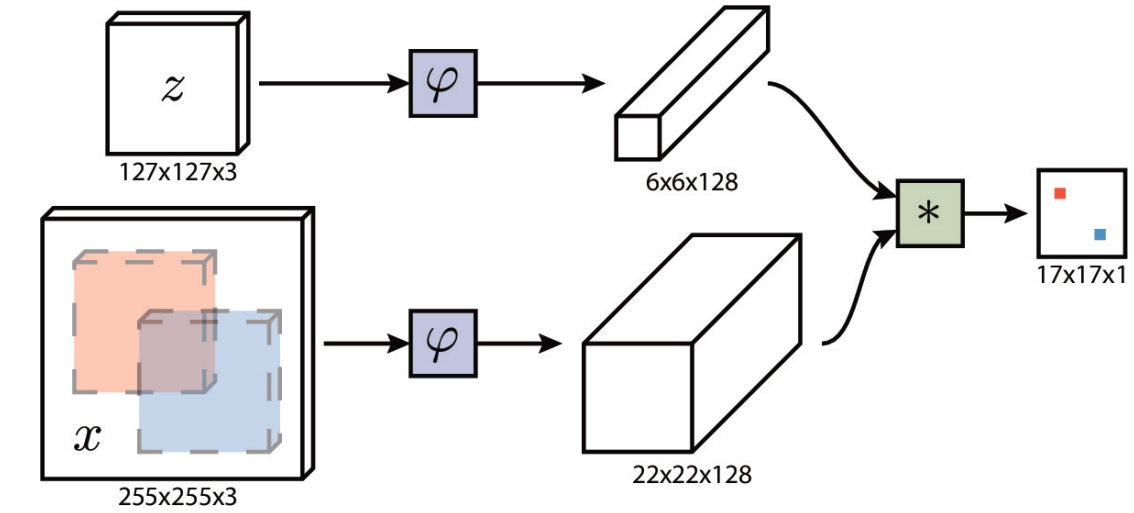
Kalal, Z., Mikolajczyk, K. and Matas, J., 2011. Tracking-learning-detection. IEEE transactions on pattern analysis and machine intelligence

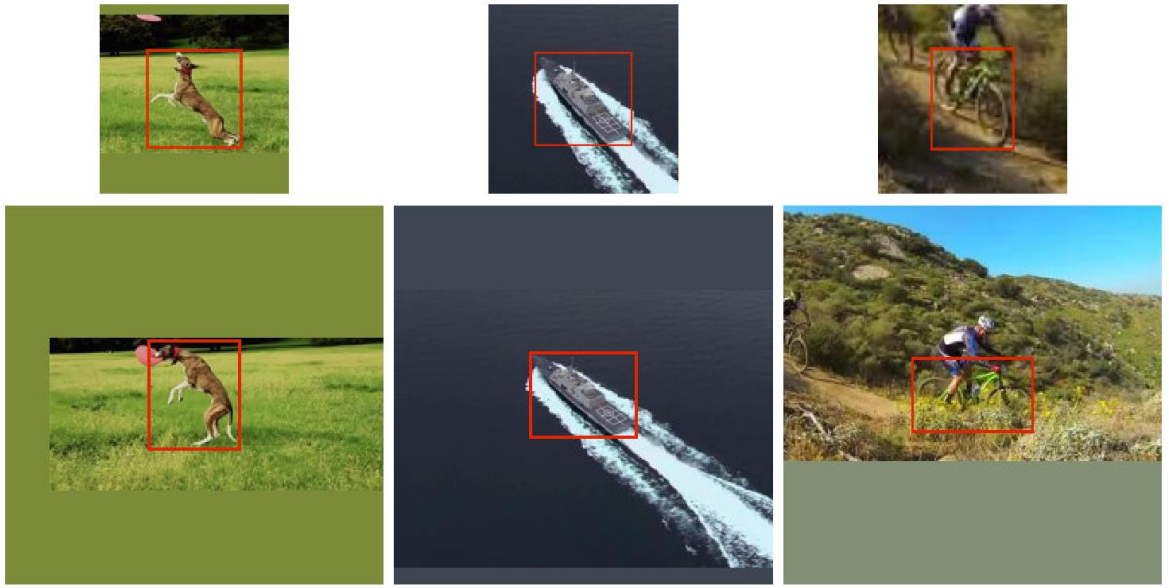
Tracking by Detection

Benfold, B. and Reid, I., 2011, June. Stable multi-target tracking in real-time surveillance video. In CVPR 2011

2016: CNN Tracking

- Instead of comparing pixel intensities: compare features
- Learn features by comparing the score map to ground-truth detections from the same video
- Very simple and fast

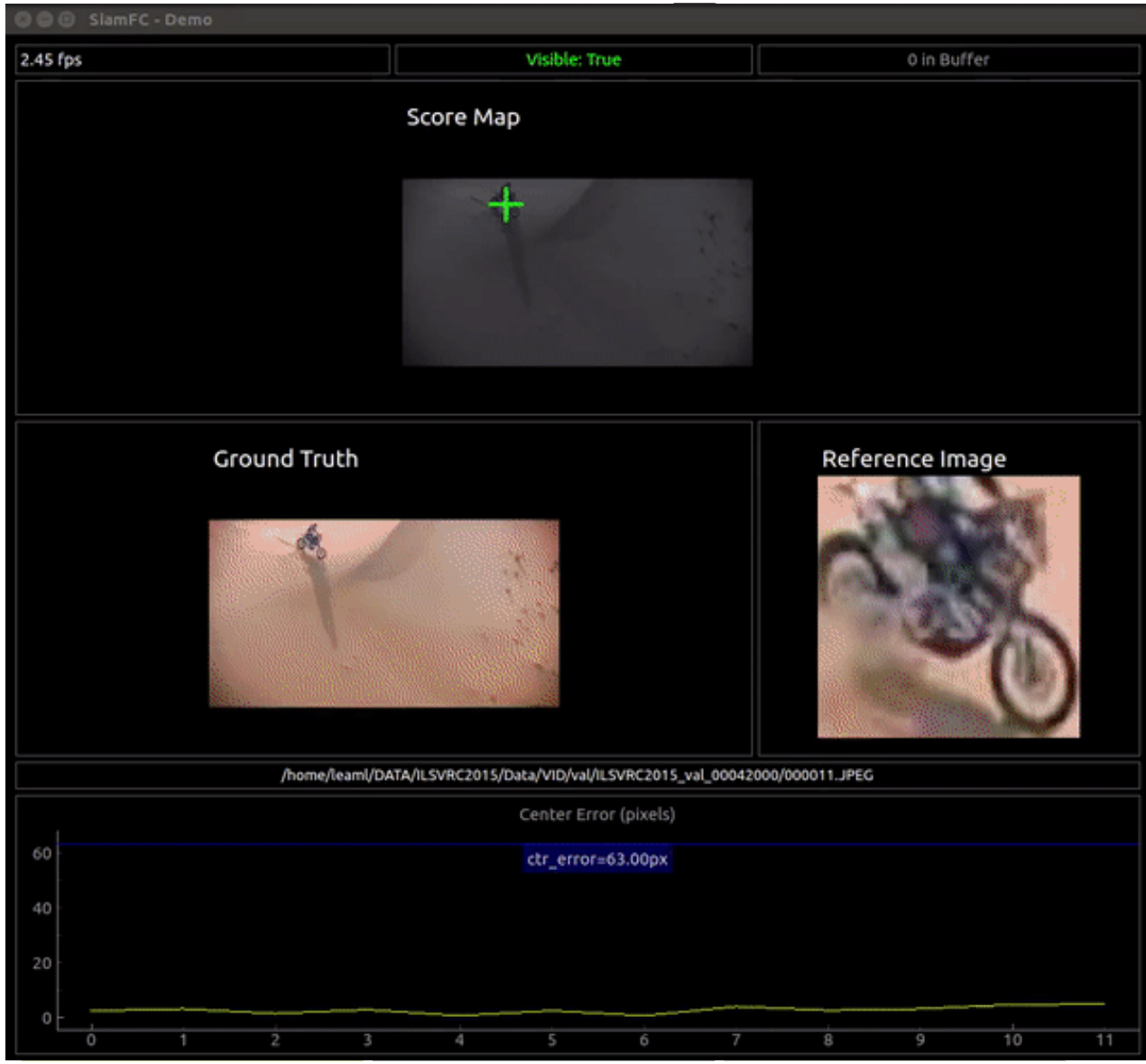




Bertinetto, Luca, Jack Valmadre, Joao F. Henriques, Andrea Vedaldi, and Philip HS Torr. "Fully-convolutional siamese networks for object tracking." ECCV 2016

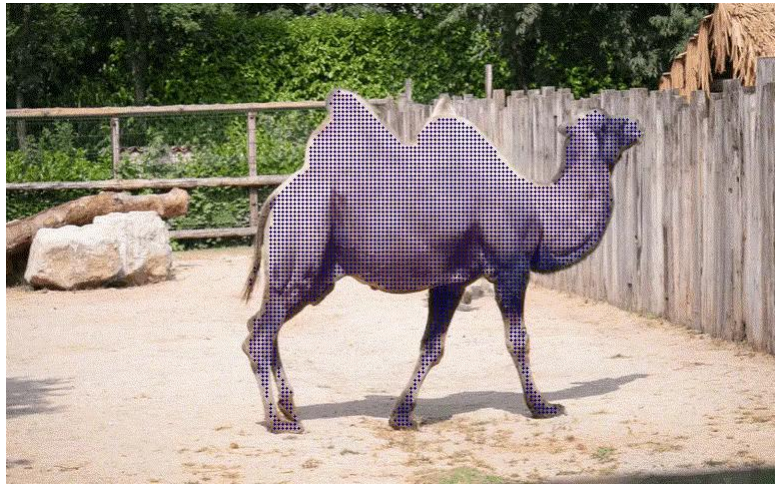
2016: CNN Tracking

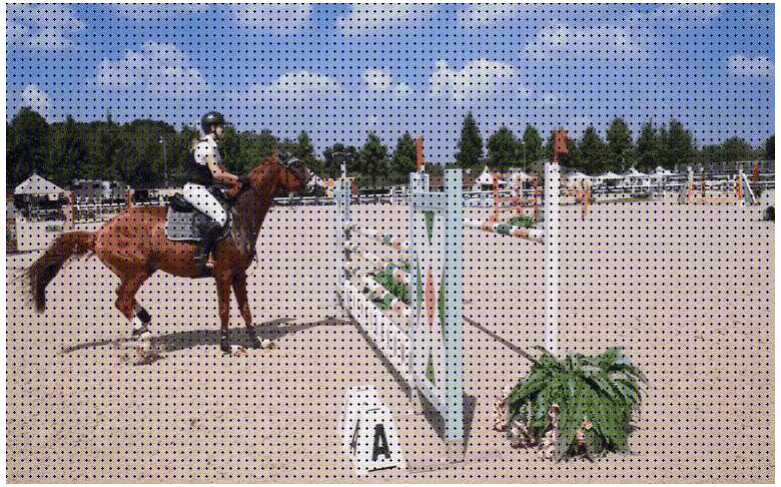
- Instead of comparing pixel intensities: compare features
- Learn features by comparing the score map to ground-truth detections from the same video
- Very simple and fast



Bertinetto, Luca, Jack Valmadre, Joao F. Henriques, Andrea Vedaldi, and Philip HS Torr. "Fully-convolutional siamese networks for object tracking." ECCV 2016

Point Tracking Revival





Harley, Adam W., Zhaoyuan Fang, and Katerina Fragkiadaki. "Particle video revisited: Tracking through occlusions using point trajectories." ECCV, 2022.

Optical Flow

- How does every pixel move from one frame to another?
- “Tracking every pixel”
- Correspondence problem
- Some pixels can go missing (occlusion, image border, etc…)
- Some pixels can appear or change colour (e.g. traffic light)

- Ground-truth difficult to obtain

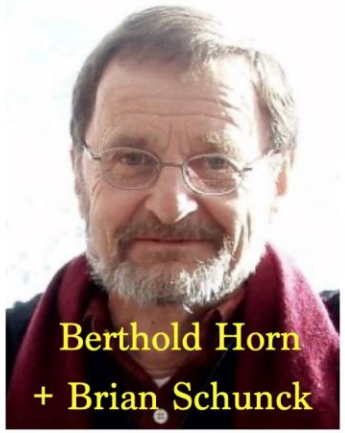
1981: Optical Flow

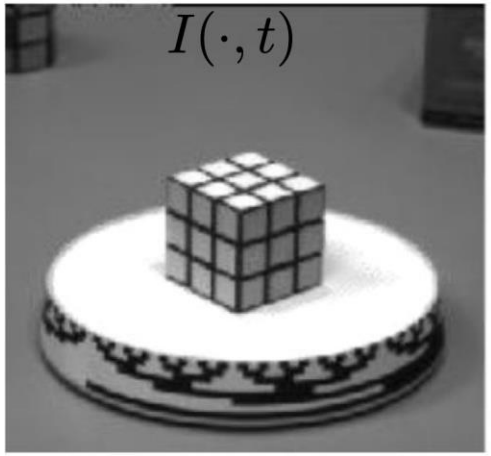
- Pattern of apparent motion: densely tracking pixels between frames
- Horn-Schunck algorithm
- 2D correspondence search: more difficult than stereo

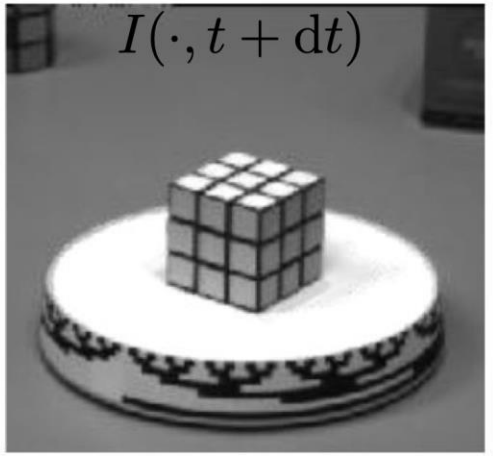
Determining Optical Flow

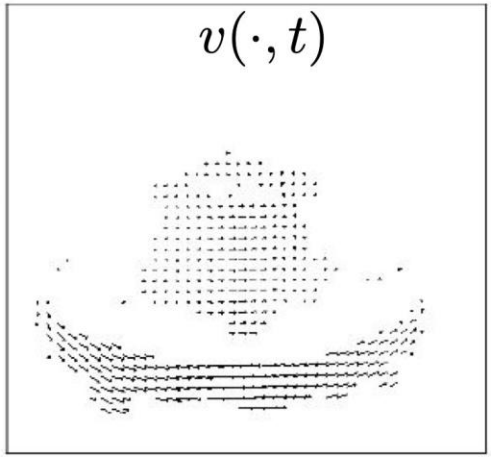
Berthold K.P. Horn and Brian G. Schunck
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

ABSTRACT
Optical flow cannot be computed locally, since only one independent measurement is available from the image sequence at a point, while the flow velocity has two components. A second constraint is needed. A method for finding the optical flow pattern is presented which assumes that the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image. An iterative implementation is shown which successfully computes the optical flow for a number of synthetic image sequences. The algorithm is robust in that it can handle image sequences that are quantized rather coarsely in space and time. It is also insensitive to quantization of brightness levels and additive noise. Examples are included where the assumption of smoothness is violated at singular points or along lines in the image.









Optical Flow – The Beginnings

Raw estimate

Smoothed estimate

Optical Flow – The Beginnings

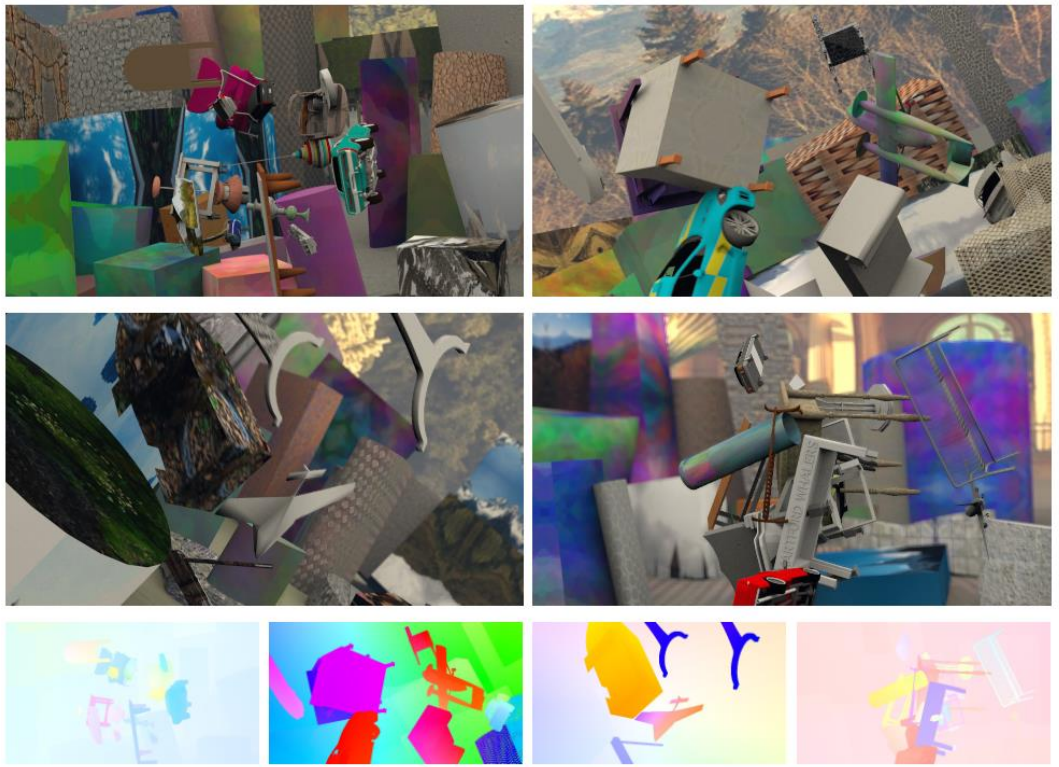
- Intensity based optical flow
- Problems with uniform-coloured regions
- Difficult to regularise
  - Can be combined with LK tracking
  - Smoothness constraints

- Difficult evaluation on very few scenes
  - Synthetic with ground truth
  - Qualitative on real scenes

Optical Flow – Flying Things

- Easier to create – automatic pipeline
- Generalises well to real data

-> optical flow is a low level vision problem and thus sim2real transfer works well

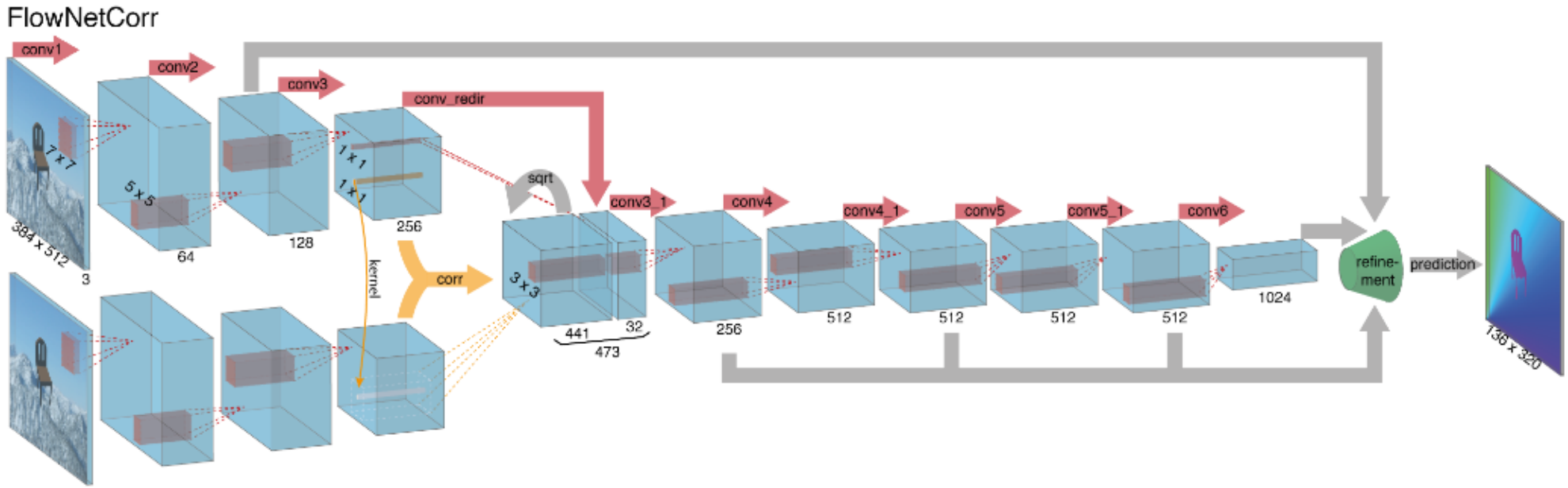


Mayer, Nikolaus, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation.“, CVPR 2016

Optical Flow - Sintel

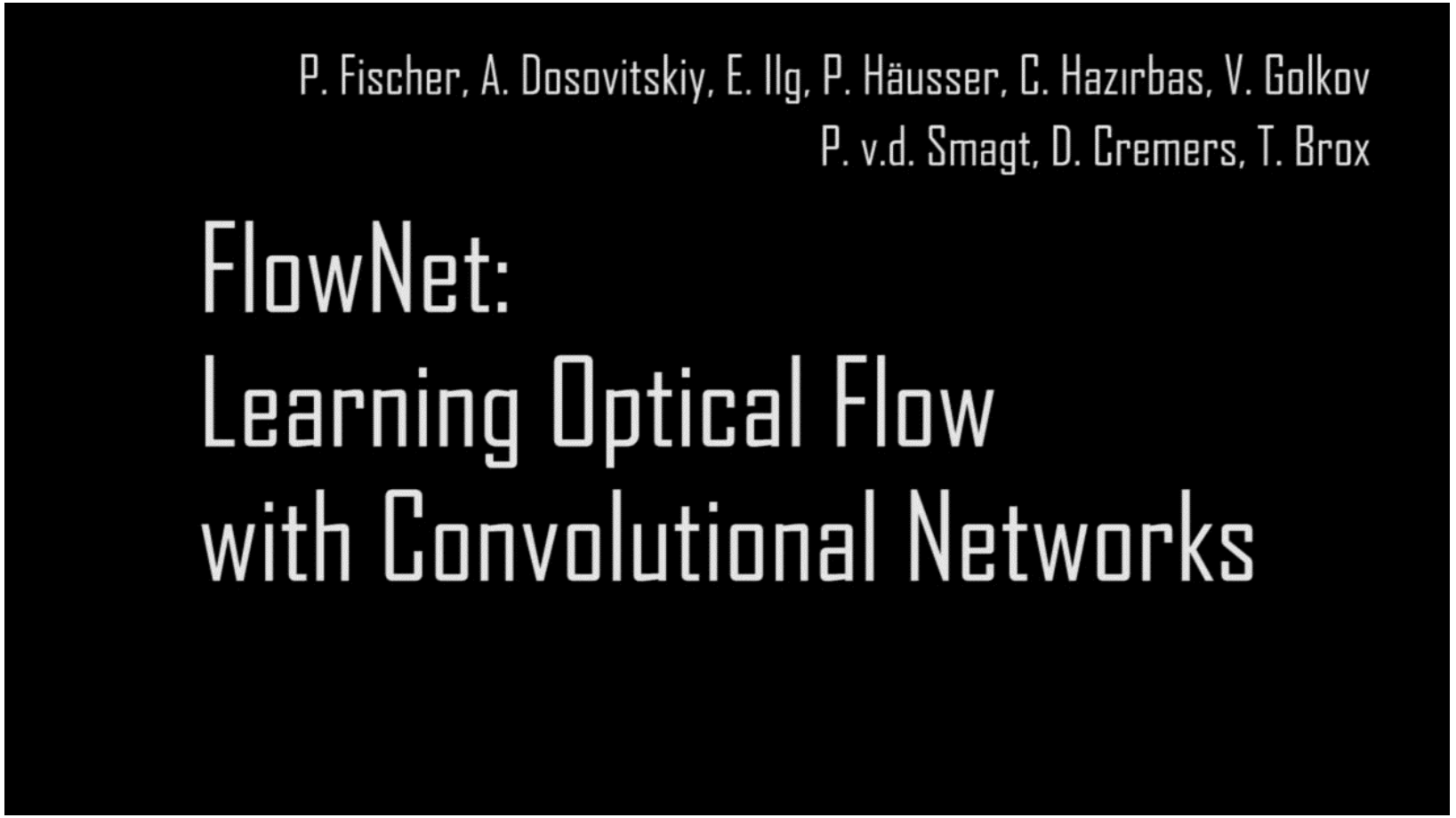
Butler, Daniel J., Jonas Wulff, Garrett B. Stanley, and Michael J. Black. "A naturalistic open source movie for optical flow evaluation." ECCV 2012

Optical Flow –Learned: FlowNet



- Again Siamese architecture
- Compute correlation volume inside networ $c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k,k] \times [-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$
- Trained on FlyingThings3D

Optical Flow –Learned: FlowNet

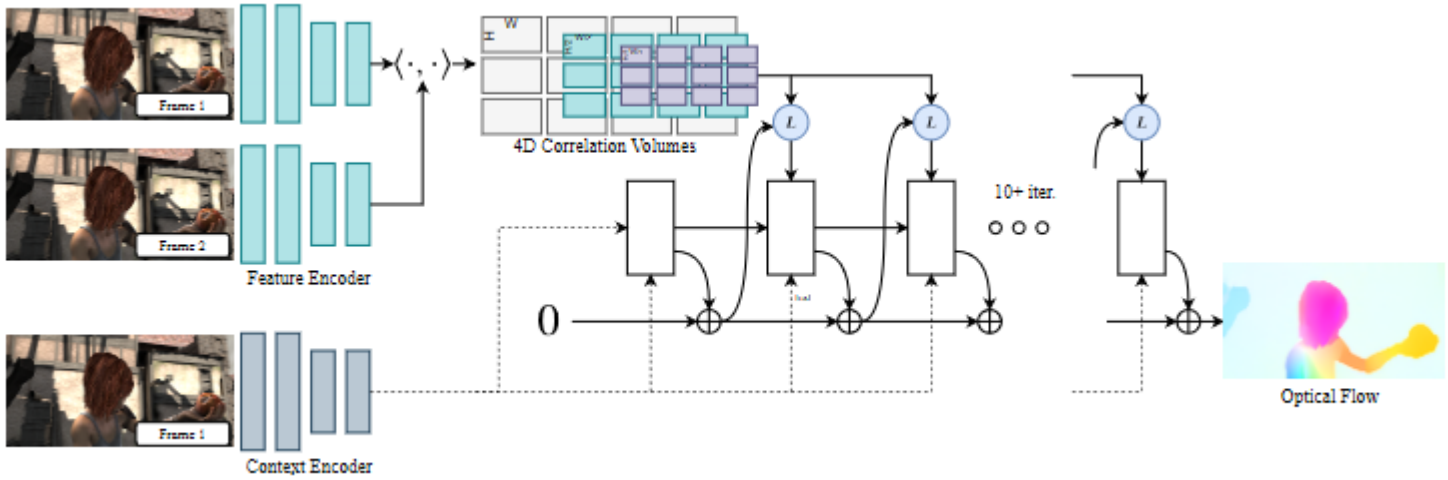


A. Dosovitskiy, P. Fischer, E. Ilg, P. Haeusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers and T. Brox "FlowNet: Learning Optical Flow ith Convolutional Networks), ICCV 2015

Optical Flow –Now(ish): RAFT

Improvements (by RAFT and other papers):

- Better backbone architectures
- Multi-scale correlation volume
- Iterative refinements: predict and update the flow estimate through several iterations



Teed, Zachary, and Jia Deng. "Raft: Recurrent all-pairs field transforms for optical flow.“, ECCV 2020

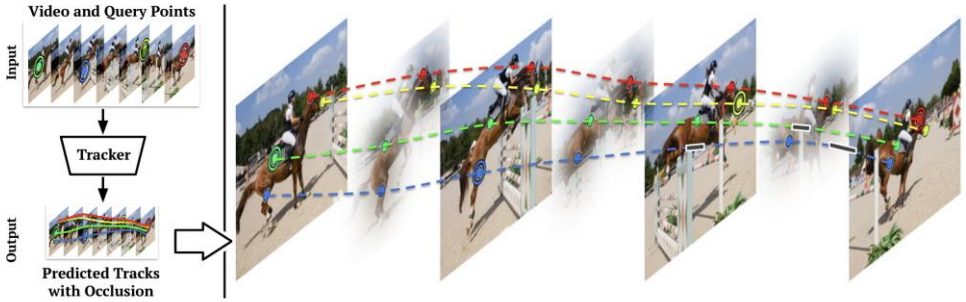
Motion Estimation

Point Tracking

Long-term tracking of individual points

PIPs

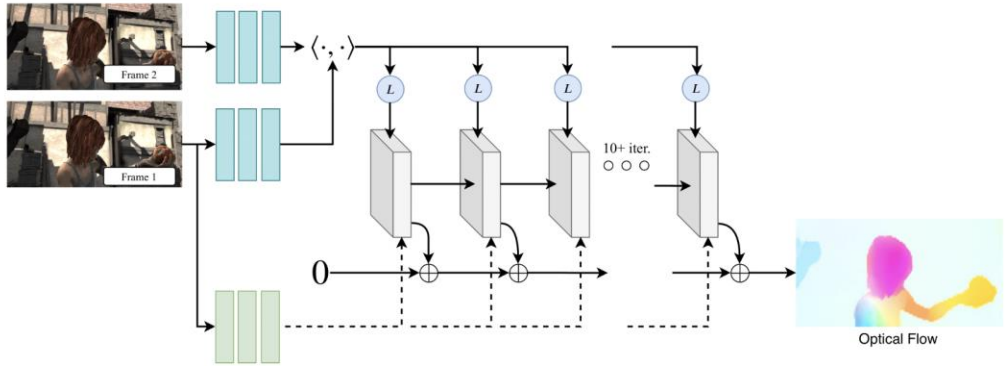
TAP-Net



Optical Flow

Dense correspondences between a pair of frames

RAFT



Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories. Harley et. al. ECCV 2022
TAP-Vid: A Benchmark for Tracking Any Point in a Video. Doersch et al., NeurIPS D&B 2022
RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. Teed et al. , ECCV 2020

Shape-from-X

1970: Shape from Shading

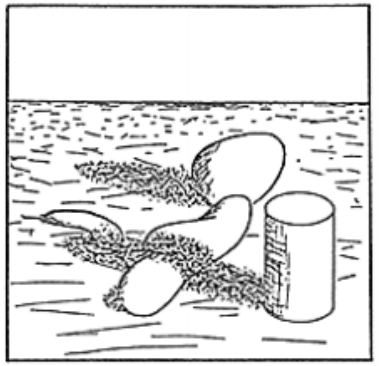
- Recover 3D from a single 2D image
- Assume simple lighting and material (Lambertian with constant albedo)
- Strong smoothness assumptions

Shape-from-X

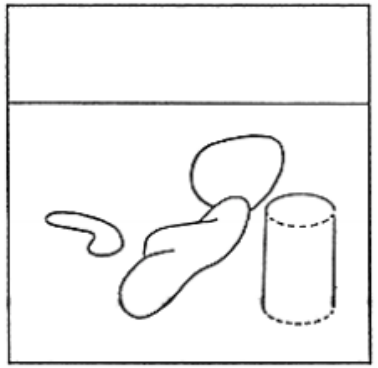
- Shading: Horn (1970)
- Contour: Guzman (1971), Waltz (1975), etc.
- Texture: Bajczy & Lieberman (1976)
- Stereo: Marr & Poggio (1976)

1978: Intrinsic Images

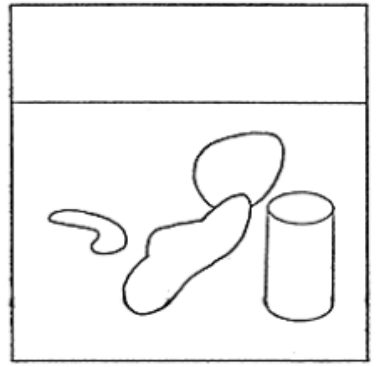
- Decompose images into its intrinsic 2D layers
  - Reflectance
  - Shading
  - Shape
  - Motion, etc.
- Useful for downstream tasks: e.g. removing lighting simplifies object detection



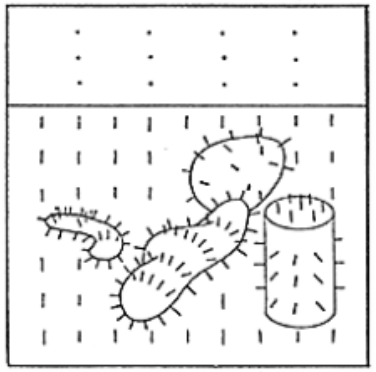
(a) ORIGINAL SCENE



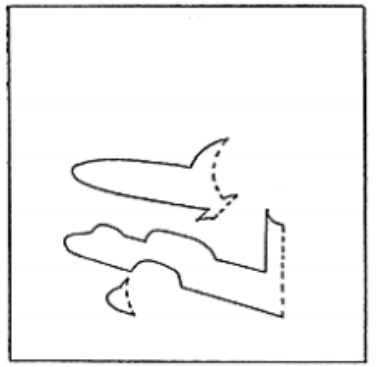
(b) DISTANCE



(c) REFLECTANCE



(d) ORIENTATION (VECTOR)



(e) ILLUMINATION

Figure 3 A set of intrinsic images derived from a single monochrome intensity image

The images are depicted as line drawings, but, in fact, would contain values at every point. The solid lines in the intrinsic images represent discontinuities in the scene characteristic; the dashed lines represent discontinuities in its derivative.

1978: Intrinsic Images

MIT Intrinsic Images dataset (2009)

Wu, Jiaye, et al. "Measured Albedo in the Wild: Filling the Gap in Intrinsics Evaluation." *arXiv preprint arXiv:2306.15662* (2023)

1980: Photometric Stereo

- Recover 3D from multiple (>2) 2D images with varying lighting
- Highly detailed and accurate
- Still Lambertian lighting assumption – relaxed later

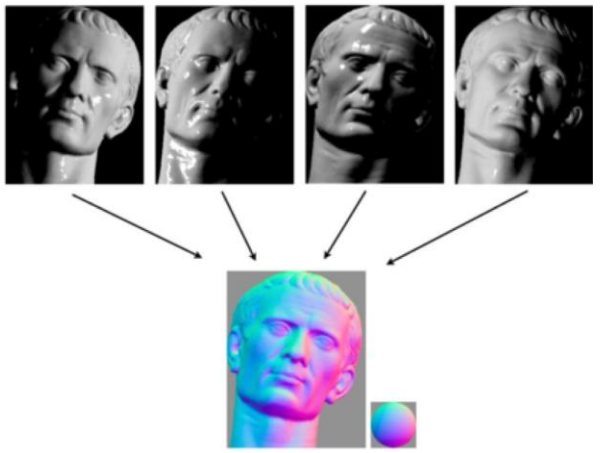
Photometric method for determining surface orientation from multiple images

Robert J. Woodham

Department of Computer Science
University of British Columbia
2075 Wesbrook Mall
Vancouver, B.C., Canada
V6T 1W5

Abstract. A novel technique called photometric stereo is introduced. The idea of photometric stereo is to vary the direction of incident illumination between successive images, while holding the viewing direction constant. It is shown that this provides sufficient information to determine surface orientation at each image point. Since the imaging geometry is not changed, the correspondence between image points is known *a priori.* The technique is photometric because it uses the radiance values recorded at a single image location, in successive views, rather than the relative positions of displaced features.

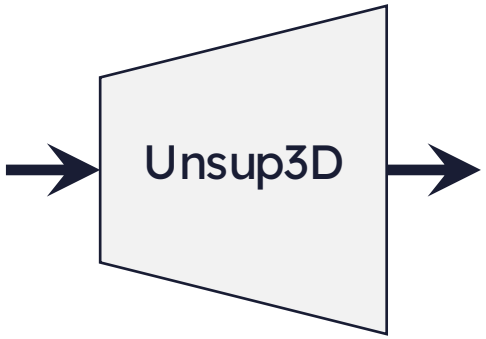
Photometric stereo is used in computer-based image understanding. It can be applied in two ways. First, it is a general technique for determining surface orientation at each image point. Second, it is a technique for determining object points that have a particular surface orientation. These applications are illustrated using synthesized examples.



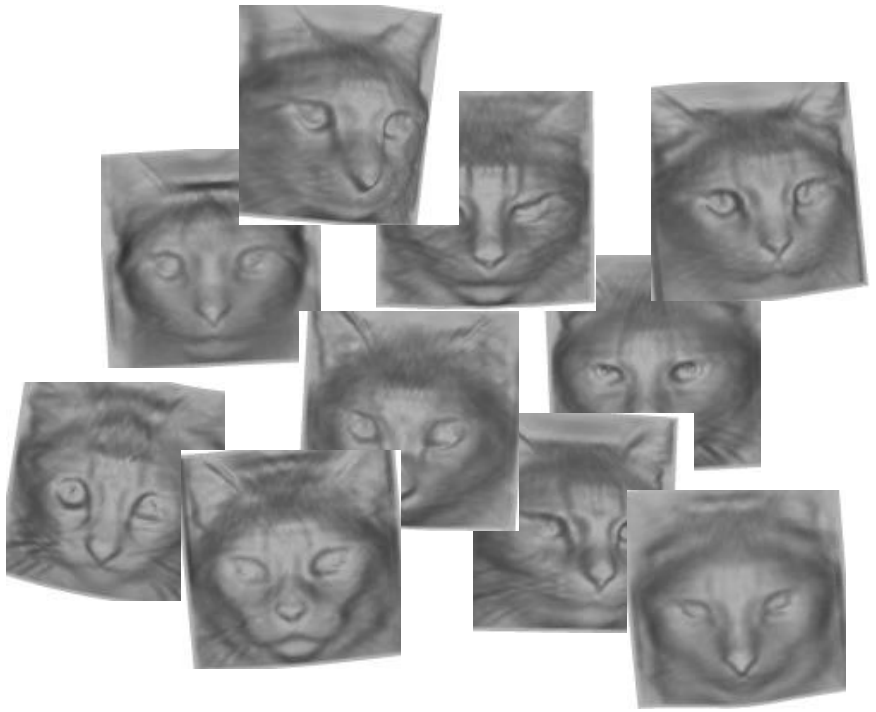
Shape-from-X now(-ish)

Training Data

single-view images of a category



Output



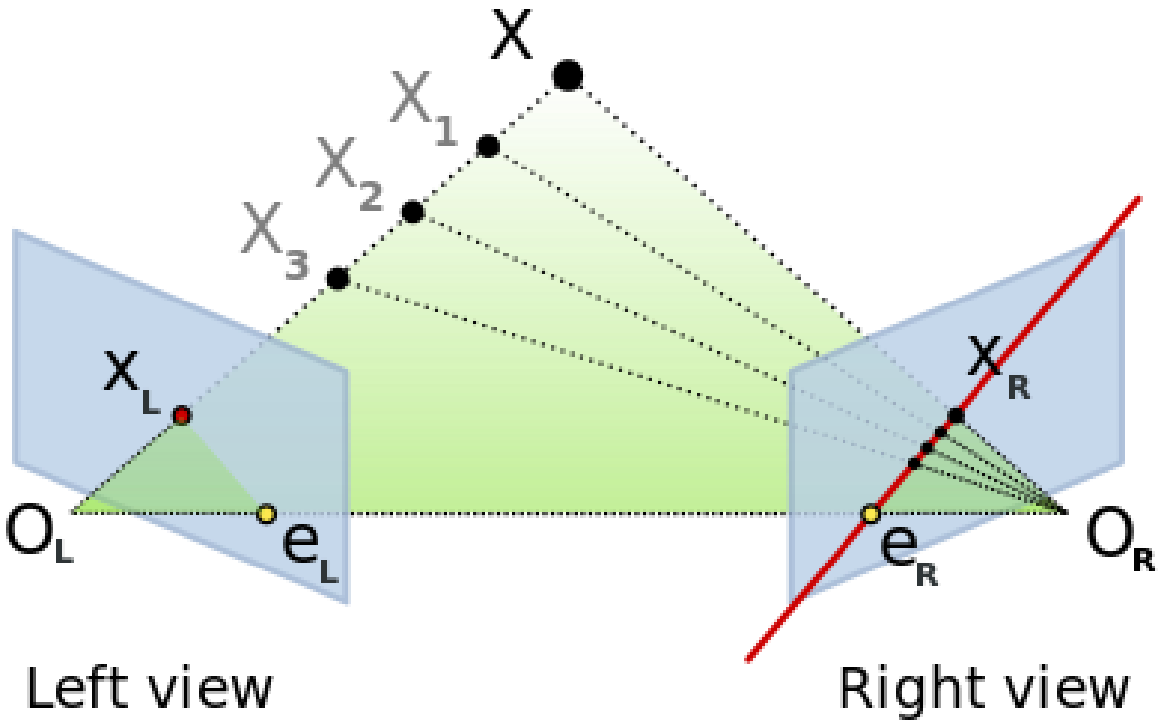
instance-specific 3D shapes

Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild
S Wu, C Rupprecht, A Vedaldi
CVPR 2020

Stereo

1981: Essential Matrix

- 2-view Geometry: a matrix that maps points to epipolar lines
- Correspondence search becomes a 1D problem
- Essential Matrix can be computed from 2D correspondences
- Rediscovery of known ideas >100 years old



H. Christopher Longuet-Higgins (September 1981). "A computer algorithm for reconstructing a scene from two projections". Nature.

1992: Structure-from-motion

- Estimating the 3D structure from image collections of static scenes
- Static scenes require only a single (moving) camera
- Closed-form SVD solution: Tomasi-Kanade factorisation for orthographic projection.
- Later: non-linear least squares for projective cameras

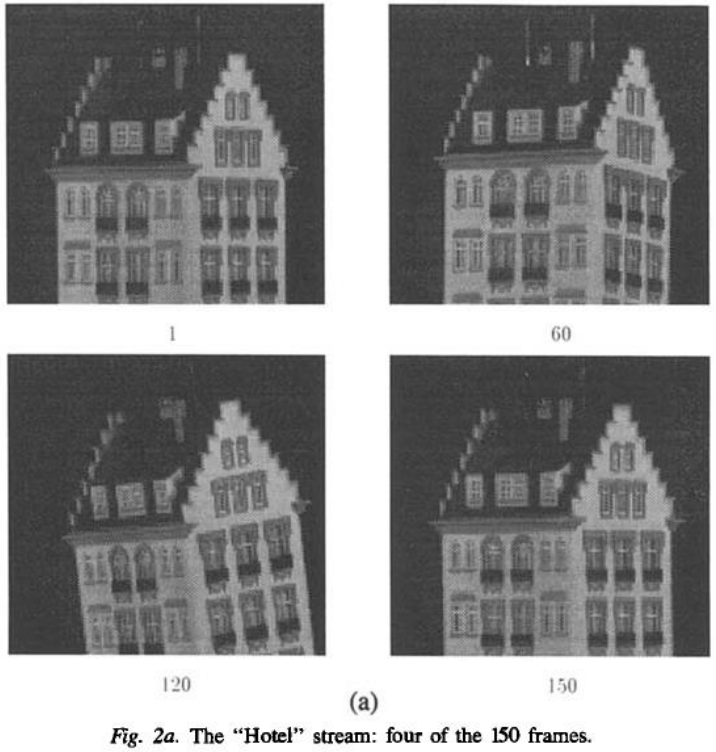
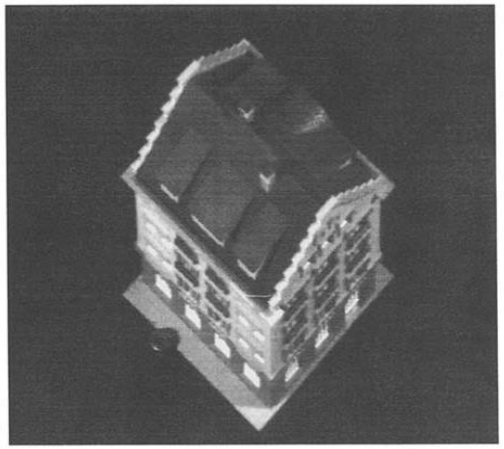


Fig. 2a. The "Hotel" stream: four of the 150 frames.



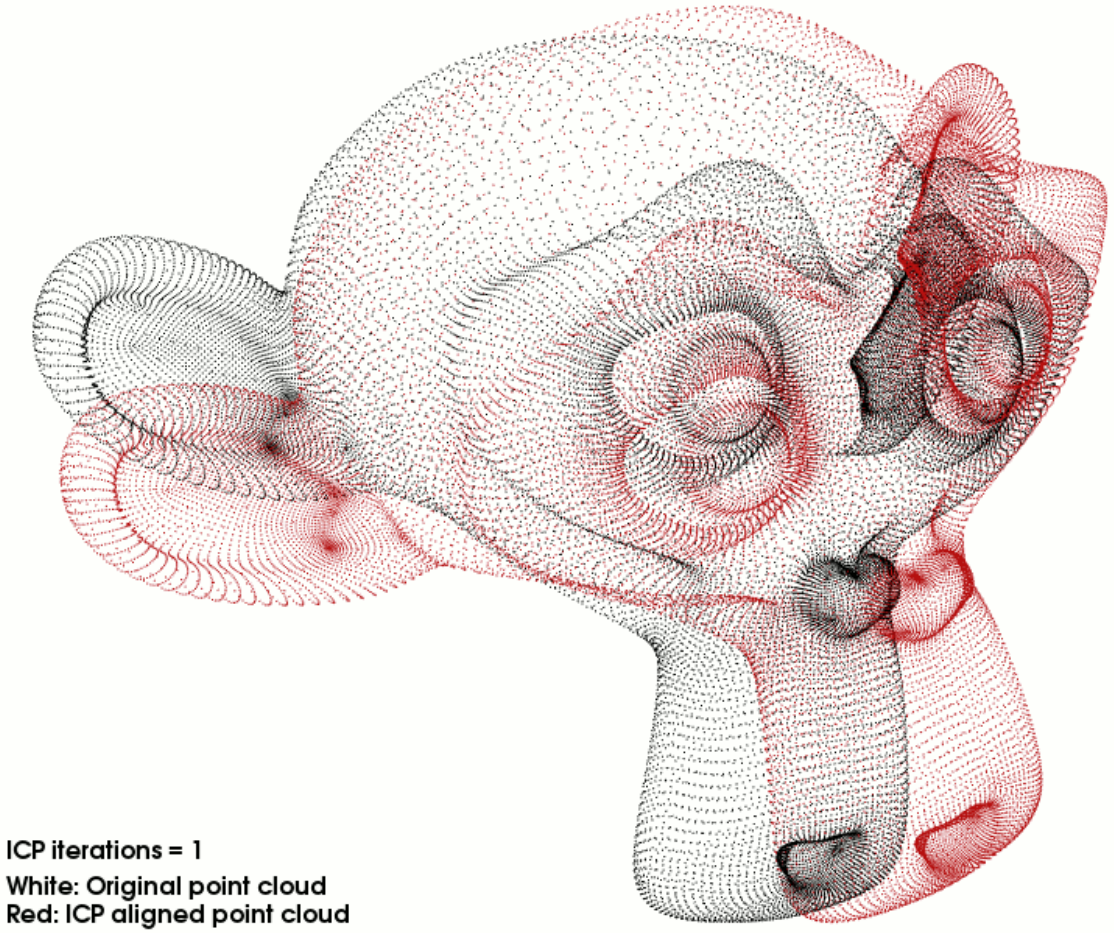
Tomasi, Carlo, and Takeo Kanade. "Shape and motion from image streams under orthography: a factorization method." *IJCV* (1992)

1992: Iterative Closest Points

- Register two point clouds by minimizing the distance between closest points

Uses:

- Align partial scans
- Estimate relative camera poses from point clouds
- Localization within 3D maps



https://github.com/yassram/iterative-closest-point

Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes." *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. Spie, 1992.

1998: Multi-view stereo

- 3D reconstruction from multiple input images – this time with level-set methods
- Surfaces instead of points

- Modelling visibility
- Convergence proofs

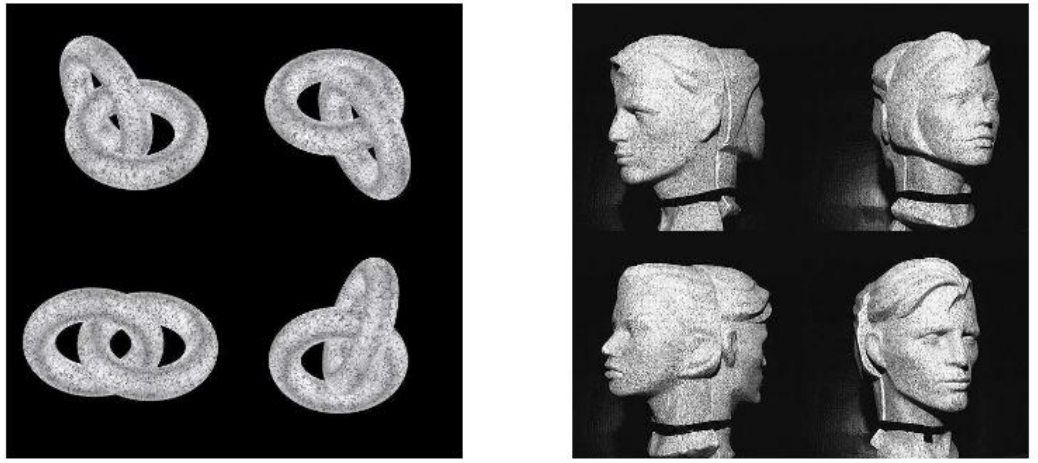


Fig. 3. Multicamera images of 3D objets. On the left hand side, two crossing synthetic tori (24 images). On the right hand side, real images: two human heads (18 images).

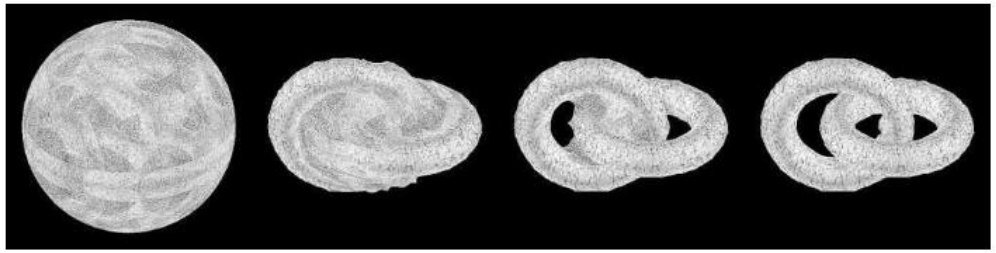


Fig. 4. Evolution of the surface for the two tori.

Faugeras, Olivier, and Renaud Keriven. "Complete dense stereovision using level set methods." *Computer Vision—ECCV'98*

2000s: Large-scale SfM

- 2006: Photo Tourism (Snavely et al,, SIGGAPH'06)
  - 3D reconstruction from internet images
  - Large scale compute
- 2009: Building Rome in a Day (Agarwal et al. ICCV'09)
  - Search "rome" on flickr
  - Reconstruction: 150k images, 21h, 500CPUs

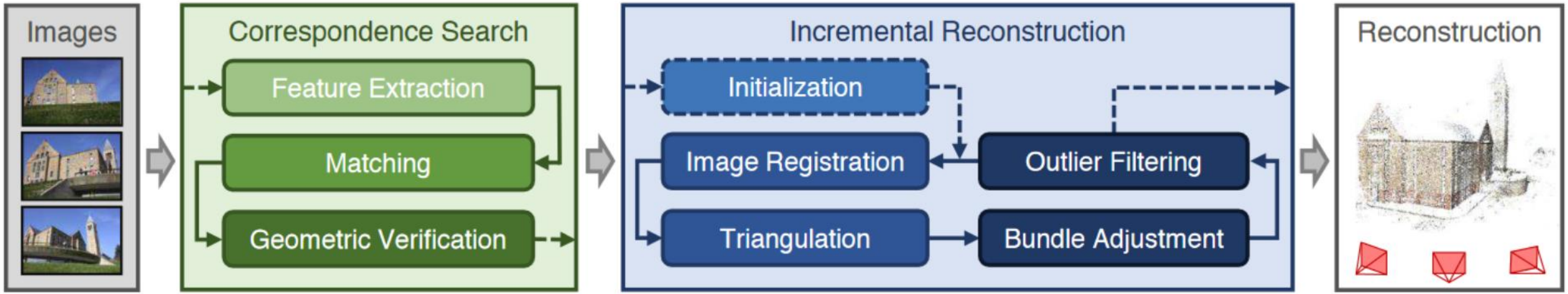
2016: COLMAP

- Open-source C++ framework
- Integrating the best features from prior work
- Defacto standard for SfM

Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline

Schonberger, Johannes L., and Jan-Michael Frahm. "Structure-from-motion revisited." CVPR 2016.

Structure from Motion (still)



- Individual components have been enhanced with DL
- COLMAP pipeline remains mostly unchanged

Neural Radiance Fields

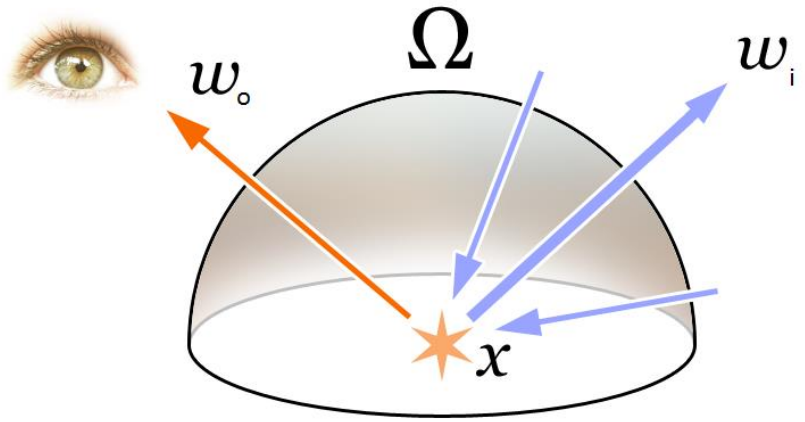
1850: Photosculpture

- 24 photographs of an object/person
- Cut contour from wood
- Assemble radial sculpture

1986: The Rendering Equation

How much light (of wavelength λ) is leaving a point x in the direction of ω_o at time t?

$$L_o(x, \omega_o, \lambda, t) = \underbrace{L_e(x, \omega_o, \lambda, t)}_{\text{emitted radiance (glowing things)}} + \underbrace{L_r(x, \omega_o, \lambda, t)}_{\text{reflected radiance}}$$



Immel, David S.; Cohen, Michael F.; Greenberg, Donald P. "*A radiosity method for non-diffuse environments*", SIGGRAPH 1986
Kajiya, James T."*The rendering equation*". Conference on Computer graphics and interactive techniques 1986

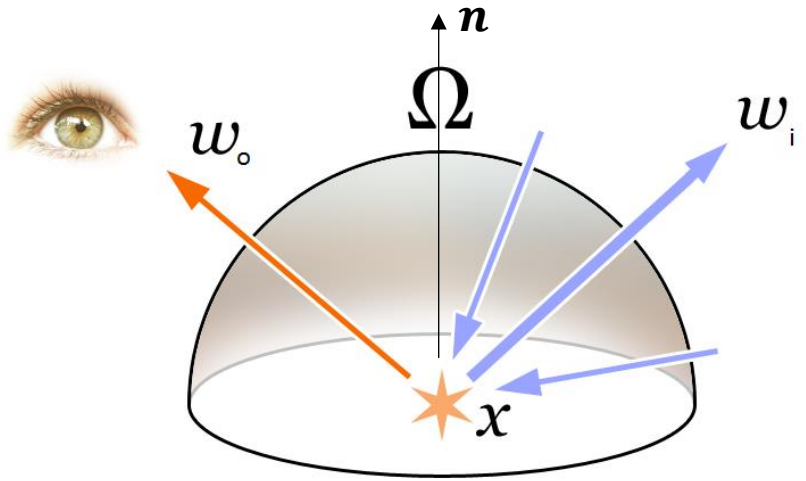
1986: The Rendering Equation

incoming radiance at x from direction ω_i

$$L_r(x, \omega_o, \lambda, t) = \int_\Omega f_i(x, \omega_i, \omega_o, \lambda, t) L_i(x, \omega_i, \lambda, t)(\omega_i \cdot \boldsymbol{n}) d\omega_i$$

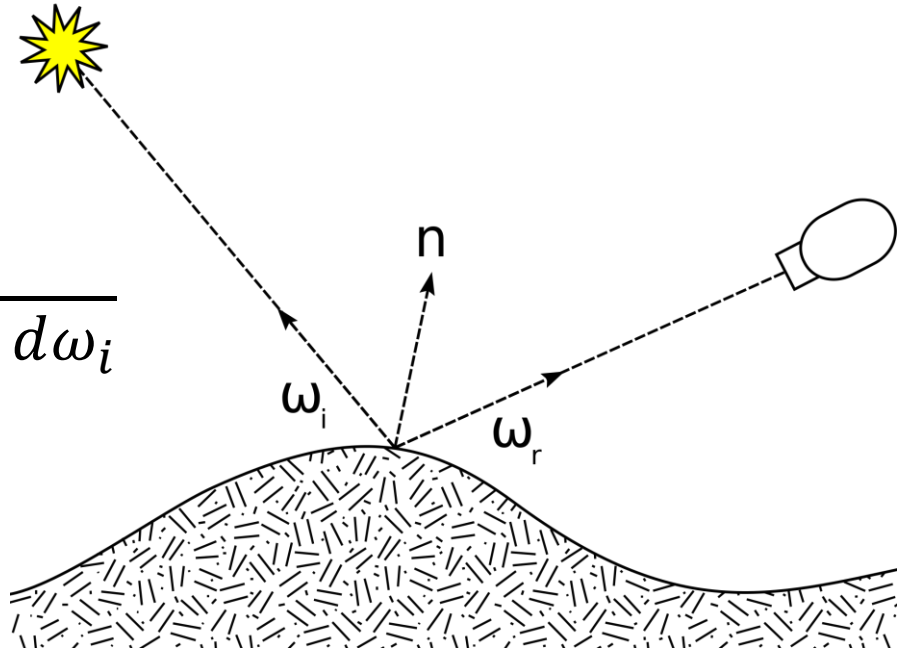
bidirectional reflectance distribution function (BRDF)

surface normal



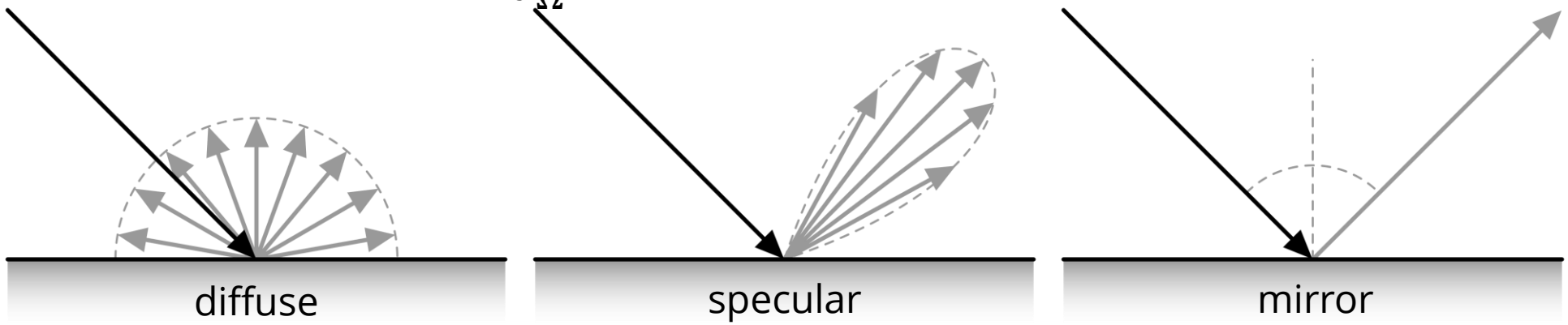
1965: The BRDF

$$f_r(\omega_i, \omega_r) = \frac{dL_r(\omega_r)}{L_i(\omega_i)\,(\omega_i \cdot n)\,d\omega_i}$$



- Positivity: $f_r(\omega_i, \omega_r) > 0$
- Reciprocity: $f_r(\omega_i, \omega_r) = f_r(\omega_r, \omega_i)$
- Energy conservation:

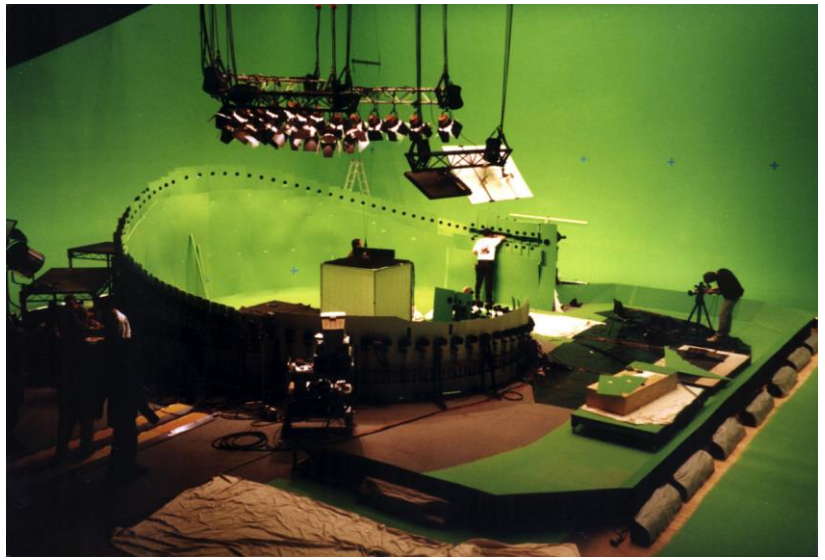
$$\forall \omega_i, \int_\Omega f_r(\omega_i, \omega_r)(\omega_r \cdot n) d\omega_r \leq 1$$



Nicodemus, Fred (1965). "*Directional reflectance and emissivity of an opaque surface*". Applied Optics

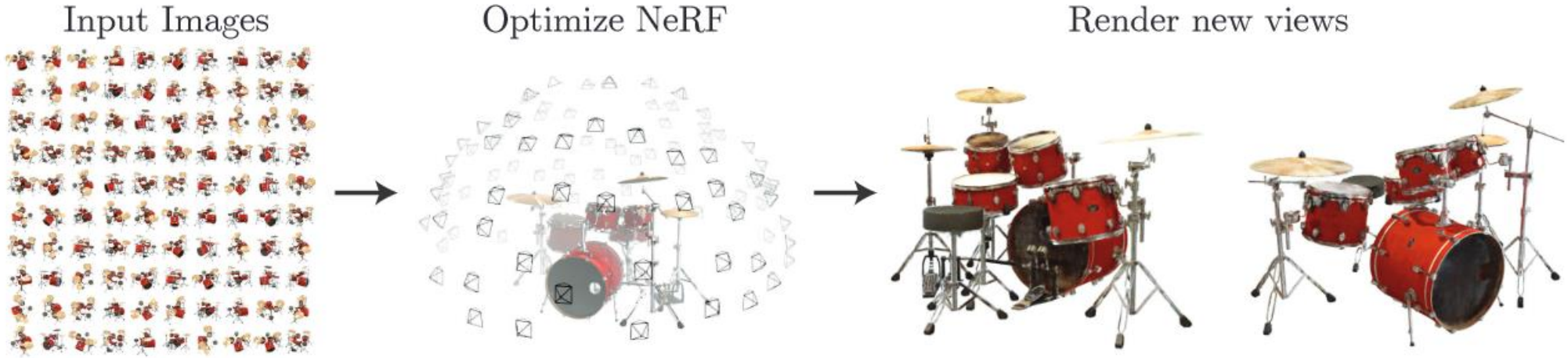
2000s: Lightfield camera arrays

- Use many synchronized cameras to capture a scene from many angle simlutaneously
- Film use: The Matrix (1999)



2020: Neural Radiance Fields

- Input: Image collection
- Learning: mapping coordinates (x,y,z) to color and occupancy
- Output: rendering from novel viewpoints



Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis.“, ECCV 2020

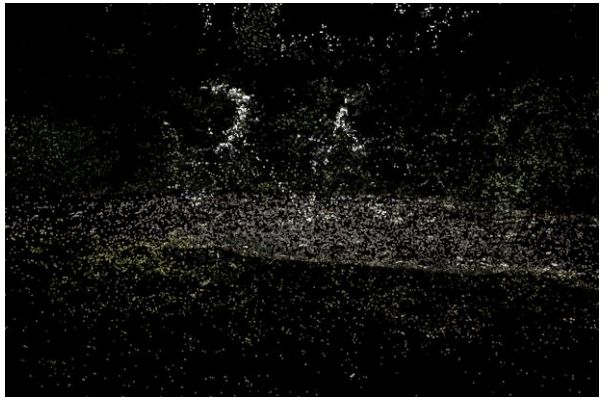
NeRF now(-ish)

- Improvements in generalisation, speed, quality, etc.
- Dynamic scenes remain difficult – triangulation is ambiguous

Li, Zhengqi, et al. "Dynibar: Neural dynamic image-based rendering." *CVPR*. 2023 (Best Paper Honourable Mention)

Gaussian Splatting

- Intuition: ray-casting through every pixel is wasteful (mostly empty space)
- Represent the scene as a collection of points with size: 3D Gaussians (mean & covariance)
- Projecting 3D Gaussians to the image plane results in approx. 2D Gaussians (Zwicker et al.,2002)
- Advantage: only spend time/memory on the surface of objects



Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*

Summary

- Computer vision is usually harder than we think.
- There are many open problems.
- “just throw a big network at a lot of data” usually does not work – we need to be smart about it.
- Current trend: old ideas can improve things quite a bit.