Computer Vision — Szeliski
Chapter 1 — Introduction

1 Reading

1.1. What Is Computer Vision?

In computer vision, we want to give the computer the ability to interpret visual imagery in a general sense. As it stands, even a two-year old human is much better at this kind of thing, and can automatically perform depth estimation, segmentation, and tracking on a wide range of inputs.

Computer vision is an inverse problem. We need to reconstruct a lot of potentially high-level unknowns (shape, semantics) from only the final result – the pixels or other sensory information. By contrast, physics and computer graphics help make sense of the forward problem, and are near-perfect in this regard: given enough time and computing power, we can model and render scenes at an almost perfect level of realism.

It is much easier to *create* [given a spec, at least] than it is to *infer* that spec from a creation.

1.2. A Brief History

1960s

- infer a 3D "blocks world" from topological structure of edges in an image

1970s

- model 3D non-polyhedral objects using generalized cylinders
- pictorial structures (meaning *generalized cylinders arranged elastically into parts*)
- stereo correspondence algorithms based on features
- structure from motion (SfM)
- Marr's three levels of visual processing, from high to low:
  (1) computational theory, (2) representation / algorithms, (3) hardware

1980s

- intensity-based optical flow algorithms
- image pyramids and multi-scale processing (replaced by wavelets in some cases)
- shape from shading, texture, focus
- better edge and contour detection (including Canny edge detection)
- integration of algorithms with optimization (/regularization) techniques
- 3D range data processing (acquisition, merging, modeling, recognition)

1990s

- many previously mentioned topics were further explored during this time
- physics-based vision (using detailed measurements and accurate physical models)
- new methods for optical flow and dense stereo correspondence (e.g. global optimization using graph cut techniques)
- multi-view stereo for 3D reconstruction
- large improvements in tracking algorithms, e.g. contour tracking using active contours
- image segmentation using, e.g., minimum energy and/or normalized cuts
- statistical learning, e.g. eigenfaces for facial recognition / linear dynamical systems for curve tracking
- increased interplay with computer graphics, especially in *image-based modeling and rendering* (computational photography)

2000s

- many, many computational photography techniques (e.g. texture synthesis, quilting, and inpainting)
- learning and feature-based techniques for recognition tasks (often *interest point* features, aka patch-based features)
- more efficient algorithms for global optimization problems
- data-driven ML techniques for computer vision problems

References

[1] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London, 2010.