Lecture 1: Winter 2012

A.L. Yuille

Abstract

First (Introductory) Lecture: 9/Jan/2012.

1. Introduction to Vision

Note: each bullet point is intended to take about five minutes in a lecture.

1. What is vision? It is the task of extracting information about the external world from light rays imaged by a camera or an eye (”to know what is where by looking” Aristotle). It is part of a bigger enterprize – designing artificial intelligence systems. Many aspects of vision — ”high-level vision” – are closely related to other forms of intelligence – reasoning, causal learning, and natural language processing.

2. Challenge of vision. Vision is extremely difficult. This is due to enormous complexity and ambiguity of images (humans have only seen an infinitesimal fraction of all images that can occur). The difficulty of performing vision was first appreciated when scientists started trying to build computer vision systems with similar abilities to humans. They rapidly discovered that it was extremely difficult compared to other ”intelligent tasks” that seemed much harder (e.g., by 1995 there were automatic chess playing programs which could beat the world champion – but computer vision researchers were unable to detect faces in images).

3. Computer vision can be thought of as ”inverse computer graphics”. Computer graphics models how to generate images from a specification of the visual scene (e.g. objects, scene structures, light sources). But computer vision is faced with the much harder task of inverting this process to infer the structure of the world from the observed image(s).

4. Vision appears easy ho humans because we are ”vision machines” – roughly forty percent of neurons in our cortex are involved in visual processing (the cortex is where intelligence is performed), Moreover, most of this processing is performed unconsciously (Helmholts described vision as ”unconscious inference”) so we are unaware of how hard it is (by contrast, we are aware of the difficulty of performing some intelligence tasks – like doing mathematics – even though these involve much fewer neurons that doing vision).

5. Human vision is fallible. Vision has been described as controlled hallucination. There are many visual illusions which show that humans can fail to perceive the structure of visual scenes correctly. These illusions typically occur when the visual information is impoverished – and it is argued that human biases are due to using strategies will give the correct answers on natural images which contain richer cues. Moreover, humans can move and observe more images to disambiguate ambiguous images. But, when doing computer vision research, it is useful to know that human vision is not always correct – also what you perceive in an image is a function of processing by a large part of your brain and it does not correspond directly to the image that enters your eyes.

6. How can we build computer vision systems that can perform visual tasks given these enormous challenges of complexity? (And how can biological visual systems perform them?). We must understand and model the visual patterns that occur in images. We must exploit knowledge about the structure of images, the environment, and visual tasks. This is increasingly being done using benchmarked datasets. (see below).

7. How to organize a course on Computer Vision? Computer vision is extremely complex. There are three strategies: (I) Organize the course around big picture theories – e.g., focus on how to build a general purpose visual system

that can perform all visual tasks. (II) Organize a course around specific visual tasks and applications – i.e. how to build computer vision systems that are of practical use? (III) Organize a course around techniques – e.g., the 20-plus, or maybe, 80-plus, techniques which everybody doing computer vision should know. There are limitations to each of these strategies (most vision course are either type II or type III). Big picture theories are interesting and can be inspiring but do not, as yet, lead to practical real world systems. Techniques are vital, but focussing on them becomes boring and lacks motivation. Visual applications are increasingly impressive, but focussing on them misses the big picture and risks turning computer vision into a bag of tricks. This course tries to combine elements of each strategy.

8. *Big picture theories of vision I.* The standard big picture theory has a dichotomy of vision into low-,intermediate-, and high-level. Low-level vision performs low level visual tasks – e.g., finding salient structures like edges and grouping image pixels with have similar intensity properties (illustrate by a number-image), estimating the motion of images. It uses low-level knowledge about the statistics of natural images and the structures of the visual world (”natural” meaning the types of images that humans see every day – e.g., not medical images). Intermediate-level vision performs an intermediate range of visual tasks – including estimating depth, surface shape, and occlusion (where one surface partially hides another surface). There are many ways to estimate depth (sometimes called modules) – e.g., binocular stereo, shape from shading, shape from texture, structure from motion, depth from de-focus, depth from haze, and so on. Intermediate-level vision. Intermediate-level vision exploits knowledge about properties of the natural world – e.g., surfaces tend to be smooth, objects tend to move rigidly. High-level vision includes object recognition, scene understanding, action recognition, and many others. It exploits detailed knowledge about objects and scenes and so has considerably more knowledge about the world than the lower levels (the knowledge will be learnt – see below).

9. *Big picture theories of vision II.* Assuming the big picture theory, how are the low-,intermediate-, and high-level tasks performed? One strategy is feedforward, or bottom-up. This proceeds by performing the tasks in sequence – low-, intermediate-, and high-level – where the output of visual tasks at one level are used as inputs to the tasks at later levels. For example, to detect a face you first perform the low-level task of edge detection, then group the edges together to find coherent contours (intermediate-level vision), and then recognize that these contours include the boundary of a face and the boundaries of its salient features (e.g., eyes, mouth, nose). But this bottom-up approach does not work on most images (at least not in the simple form described here). The problem is that low-level vision is highly ambiguous – if you look at a small part of an image through an aperture it is very hard to detect edges, people seem to need bigger spatial context and high-level knowledge (e.g., recognizing that the object is a face and hence inferring where its edges are likely to be). Also the perception of shape is also influenced by high-level knowledge – if you look at the inside of a face mask you will see it as a normal (convex) face even though the local depth cues indicate a concave object. Hence to get good results on some low-level tasks you need access to high-level knowledge, which precludes a simple bottom-up strategy. A more sophisticated alternative is to use bottom-up processing to propagate forward multiple hypotheses (e.g., about the positions of edges) but without committing to a single hypothesis (unless the bottom-up evidence is overwhelming). The high-levels can resolve the ambiguities of these hypotheses by exploiting higher level knowledge and propagate down to remove false hypotheses. Debates about the roles of bottom-up and top-down processing are common in biological vision.

10. *Specific Visual Tasks and Applications*: In the last few years there have been several very impressive vision applications. These include Kinect, Face Detection and Recognition, Google Goggles, Build Rome in a Day, Automated Vehicles – and more examples keep arising (e.g., cosmetic surgery). All of these focus on specific tasks and work in specific domains (while human vision is more general purpose). There are a large range of visual tasks. Some of actively worked on – others, like material perception, are currently out of fashion. Nevertheless this potentially useful for robots – to detect whether a surface is slippery and whether an object can be grasped.

11. *Computer vision techniques.* Computer vision is a dynamic interdisciplinary field which has absorbed techniques from many disciplines (e.g., CS, Engineering, Mathematics, Statistics). A list of ”20 techniques that all computer vision researchers should know” has rapidly grown to over 80. Several of these techniques, however, are based on similar underlying principles. There are a core set of ideas that keep arising. The techniques include filtering, geometry, probabilities on graphs, inference algorithms, learning algorithms. This course will try to give examples of these core techniques.

12. Image Datasets and Learning. In recent years computer vision has increasingly relied on labeled datasets to evaluate and benchmark algorithms. These datasets also enable computer vision researchers to learn their models from data.

The datasets are fairly small compared to the enormous amount of real images that can occur. It is known that results of computer vision algorithms on some datasets do not generalize, or transfer, to give good performance on images outside the dataset.

13. Bayesian Approaches. This is an idealized way to solve vision tasks which treats vision as an inverse problem. Let $P(I|W)$ be the probability of generating the image I from the state W of the world (i.e. W is the representation discussed above). $P(W)$ is the prior on the world. Then $P(W|I) = P(I|W)P(W)/P(I)$ (Bayes Theorem) gives the posterior probability of the state W of the world conditioned on the image I. Computer graphics specifies $P(I|W)$ – i.e. how to generate an image I from a specification W of the world. But computer vision must face the harder task of estimating W from $P(W|I)$ exploiting prior knowledge $P(W)$ about the world to resolve ambiguities (or getting additional images).

14. Core Ideas: Probabilities and Representations. I.e. $P(I|W), P(W), P(W|I)$ and the representation W. These are the core ideas (in my opinion) underlying all of computer vision. They will be illustrated during the course. There is a growing amount of work – in many scientific domains – which can be expressed in these terms (Hidden Markov Models are one famous example of this type). These relate to tasks, if properties are represented then it is possible to perform tasks involving them (e.g., a model for a cat must have an explicit representation of ”paw” if it wants to detect the cat’s paws).