Computer Vision Systems Programming VO

Introduction

Christopher Pramerdorfer

Computer Vision Lab, Vienna University of Technology

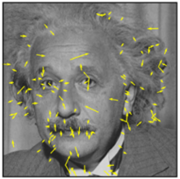
Topics

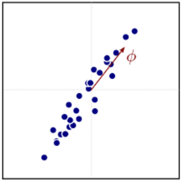
What is Computer Vision (CV) and why is it important?

CV past, present, future

Relation to other research fields

Brief image processing recap





Images from Prince 2012

What Is CV and Why Is It Important?

Let's hear what Fei-Fei Li has to say

Image from ted.com

What Is CV and Why Is It Important?

CV is about making computers understand images like humans do

Key to novel autonomous systems (cars, security, data analysis)

Tremendous progress in last decades, but still unsolved

CV Past, Present, Future

CV research started around 50 years ago

Let's take a look at a few examples

CV Past, Present, Future
1963: Pose Estimation

Edge-based pose estimation of polyhedra

Among first CV applications

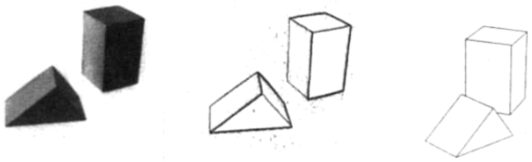


Image from Roberts 1963

CV Past, Present, Future
1973: Part-Based Object Detection

Object representation as parts connected by springs

Known as pictorial structures or constellation models

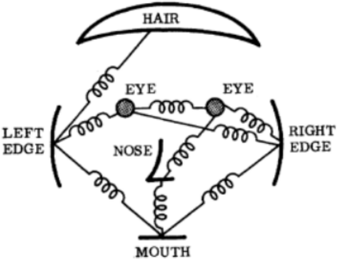


Image from Fischler and Elschlager 1973

Zip code recognition from images

Among first applications using convolutional neural networks

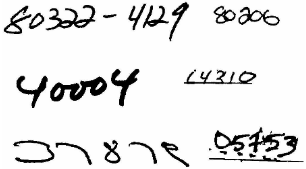


Image from LeCun et al. 1989

1989: OCR Using Convolutional Neural Networks

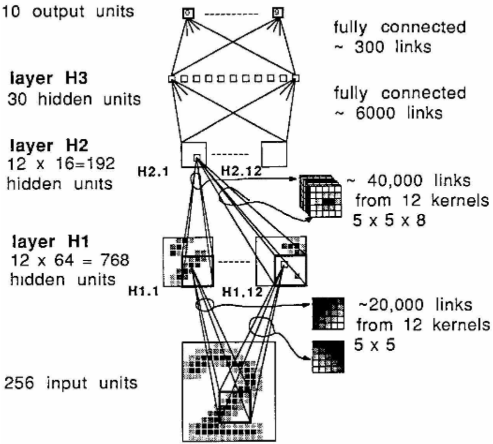


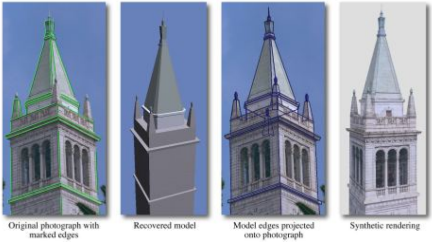
Image from LeCun et al. 1989

CV Past, Present, Future
1996: Image-Based Modeling

Generate a 3D model from a set of images

Use this model and input images to render new images

`https://www.youtube.com/watch?v=RPhGEiM_6lM`



Images from Debevec 1996

Fast object detection using Haar features and boosting

Similar technologies used in smart cameras for auto focus

Image from `olympus-europa.com`

3D reconstruction from photo collections

Structure from Motion (SIFT + bundle adjustment)

Image from Snavely, Seitz, and Szeliski 2006

CV Past, Present, Future
2006: Photo Tourism – Microsoft Photosynth

Image from photosynth.net

CV Past, Present, Future
2006: Photo Tourism – Building Rome in a Day

Image from https://www.youtube.com/watch?v=sQegEro5Bfo

CV Past, Present, Future
2011: Kinect

Depth estimation via active stereo

Real-time pose estimation of multiple players

Image from wikipedia.org

Image from Shotton et al. 2011

CV Past, Present, Future
2012: Deep Learning and Big Data

Deep Learning on huge datasets for object recognition

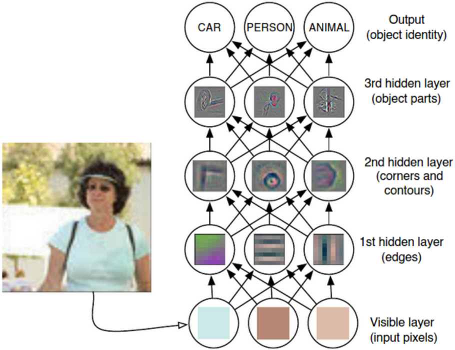


Image from Bengio, Goodfellow, and Courville 2015

CV Past, Present, Future
2012: Deep Learning and Big Data – Clarifai

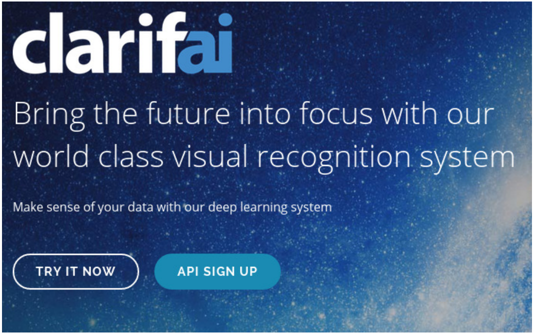


Image from `clarifai.com`

Image from ted.com

Object recognition without constraints

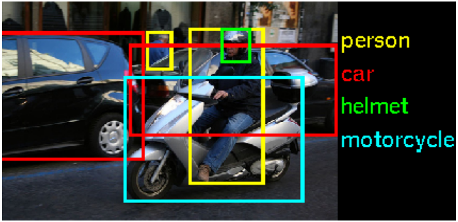


Image from image-net.org

CV Past, Present, Future
20xx: Autonomous Cars

Cars that drive autonomously

`https://www.youtube.com/watch?v=bDOnn0-4Nq8`

Image by Google

Segmentation, context, motion, emotions

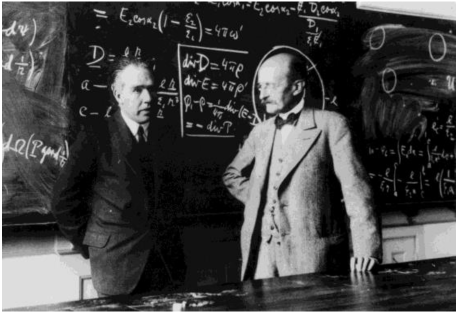


Image from Larry Zitnick's slides

CV Past, Present, Future
20xx: Human-Level Vision

Image from ted.com

Dead Sea, Jordan, 2014

CV and Related Fields

In other lectures you probably heard about

- Mathematics and statistics
- Image processing (e.g. linear filtering, SIFT)
- Machine learning (e.g. SVM)

Let's see how CV and these fields are related

CV is about making computers understand images like humans do

So in mathematical terms CV is about

- Inferring some world state (a scalar w or vector $\mathbf{w}$)
- From measurements $\mathbf{x}$ (a *feature vector*)

We use *image processing* to extract $\mathbf{x}$ from images

- Preprocessing step for CV
- Different problems favor different $\mathbf{x}$

Example: scene category classification

- $\mathbf{x}$: histogram of SIFT visual words
- w : scene class label (e.g. desert, jungle)

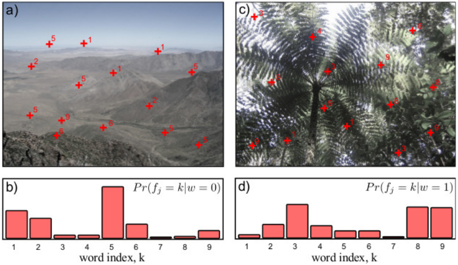


Image from Prince 2012

CV is about inferring some world state $\mathbf{w}$ from measurements $\mathbf{x}$

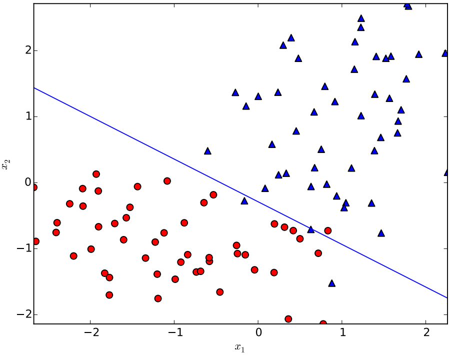
And thus about describing the relationship between $\mathbf{x}$ and $\mathbf{w}$

- This relationship is called *model*

Models are ideally statistical (probabilistic)

- Allow us to reason about uncertainty

Statistical analysis can help select a suitable model



A model usually has two kinds of *parameters*

- *Hyperparameters* that are set manually
- Parameters $\boldsymbol{\theta}$ that are *learned* from data

Learning involves finding a $\boldsymbol{\theta}$ that

- Minimizes the disagreement (*loss*) between $\dot{\mathbf{w}}$ and $\mathbf{w}$
- Given *training samples* $\{(\mathbf{x}, \dot{\mathbf{w}})\}$ and predictions $\mathbf{w} = \Gamma(\mathbf{x}; \boldsymbol{\theta})$

This is a *mathematical optimization* problem

Machine Learning (ML) studies techniques for learning from data

- Namely algorithms for learning and inference
- So any CV model that involves learning is a ML technique

CV often makes use of generic ML algorithms (e.g. SVM)

Strictly speaking, models and algorithms are not the same

- More on this later

Wadi Mujib, Jordan, 2014

Image Processing Recap

We use Image Processing (IP) to extract a suitable $\mathbf{x}$ from images

- IP has great influence on CV performance

Suitable means

- *Distinctive* features
- That are *invariant* and *robust*

Such features vary significantly (only) with w

- So different problems favor different $\mathbf{x}$

Image Processing Recap

More on feature selection later

Let's recap some generic IP methods for

- Gaining robustness to noise
- Detecting brightness changes
- Detecting interest points in images
- Describing image patches in an invariant way
- Dimensionality reduction

Gain robustness to noise via blurring

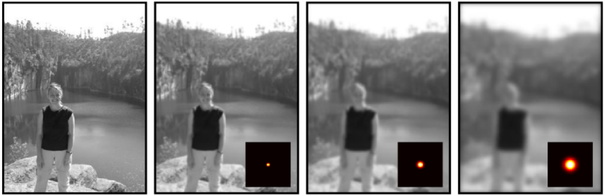
Often accomplished via *linear filtering*

- Pixel values linear combination of neighbor values
- Computed via *convolution* (or correlation) with kernel h

$$f'(x,y) = \sum_{i,j} f(x-i, y-j)h(i,j)$$

For blurring use a 2D Gaussian as kernel h:

$$h(i,j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right)$$



Images from Prince 2012

Brightness changes can be valuable information

- Object boundaries, textured regions

Use a Laplacian of Gaussian (LoG) filter as kernel h

- Gaussian for noise reduction
- Laplacian approximates $\nabla^2 = f_{xx} + f_{yy}$

LoG filters respond to intensity changes

- Regardless of direction
- At a frequency defined by σ of Gaussian

Detecting Brightness Changes – LoG Filter

Images from Prince 2012

Direction of brightness changes can be valuable information

- Texture information

Use a Gabor filter as kernel h, which consists of

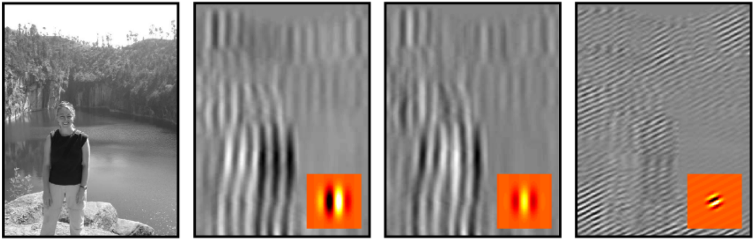
- A Gaussian for noise reduction
- A Sinusoid for change detection

Gabor filters respond to intensity changes at a

- Phase and orientation defined by the Sinusoid
- Frequency defined by the Gaussian and Sinusoid

Image Processing Recap

Detecting Brightness Changes – Gabor Filter



Images from Prince 2012

Interest points (keypoints) are

- Distinctive locations in images
- Invariant and robust to image transformations

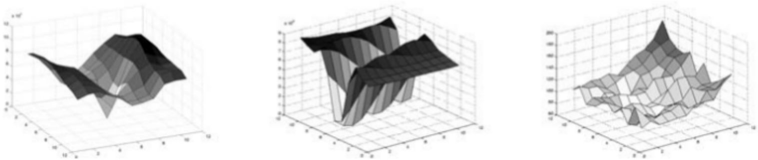
Can be detected reliably in multiple images of same object

- Used for object detection, structure from motion

Corners characterized by intensity change in multiple directions

Harris corner detector exploits this by

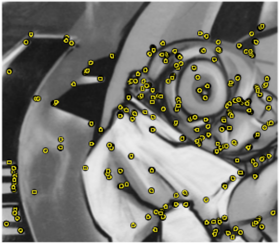
- Checking gradient distribution in local neighborhood
- Corner: gradient distribution has two large eigenvalues

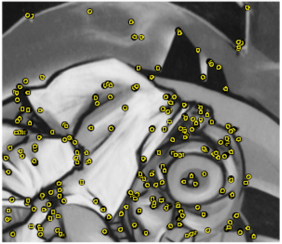


Images from Szeliski 2010

Harris interest points

- Are invariant to translation and rotation
- Stable under varying lighting conditions





Images from Tuytelaars and Mikolajczyk 2008

Scale invariant blob detector

- A blob is an image region with similar intensity

Blob detection accomplished via LoG filtering

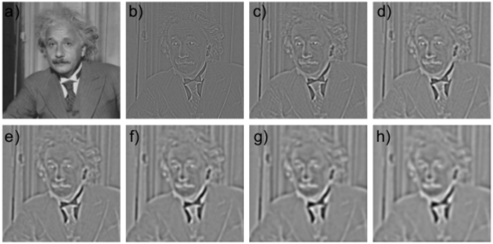
- LoG filter responds to blobs of size that depends on σ

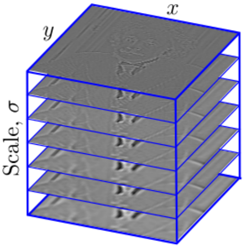
Scale invariance is achieved by

- Applying LoG filter with multiple σ
- Finding local maxima in resulting scale-space

Repeated LoG approximated by Differences of Gaussians (DoGs)

Image Processing Recap
Interest Point Detection – SIFT Scale Space





Images from Prince 2012

Local maxima are

- Localized to sub-voxel accuracy
- Discarded unless on corners
- Assigned an orientation via gradient histograms

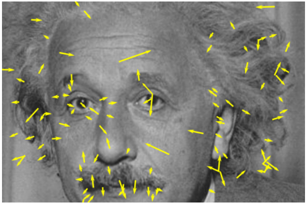


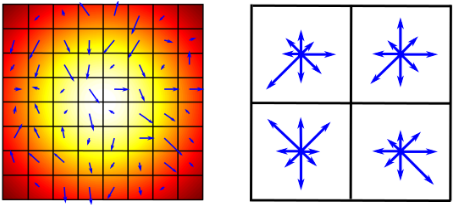
Image from Prince 2012

- Compact representations of contents of an image region
- Usually computed at interest point locations
- Invariant in conjunction with suitable interest points
- Pool information locally to achieve robustness

Computed from gradient histograms

Usually used together with SIFT interest points

- Compensate for scale, rotation



Images from Prince 2012

SIFT descriptors are

- Invariant to scale and rotation (interest points)
- Invariant to global intensity changes (gradients)
- Robust to small affine transformations (pooling)

Reduce the dimensionality of $\mathbf{x}$ by removing irrelevant features

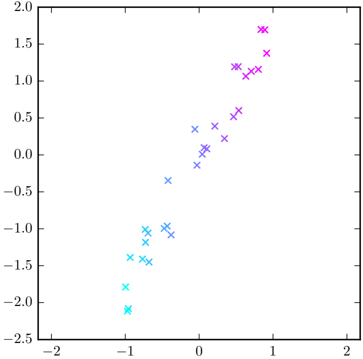
- Irrelevant means redundant or not discriminative (e.g. noise)

Advantages

- Makes learning and inference more efficient
- Can improve generalization performance
- Facilitates data visualization

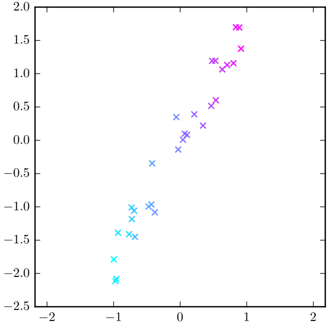
Assume the following data (30 samples $\mathbf{x}_1 \cdots \mathbf{x}_{30}$, $\dim(\mathbf{x}) = 2$)

- Features are highly correlated



We want to map the $\mathbf{x}_k$ to a linear subspace

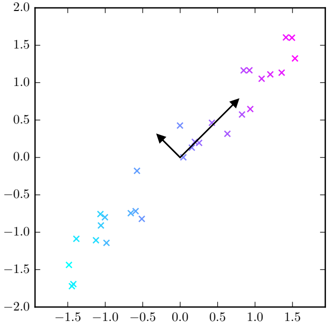
Spanned by directions of largest data variation



Standardize individual features (zero mean, unit stddev)

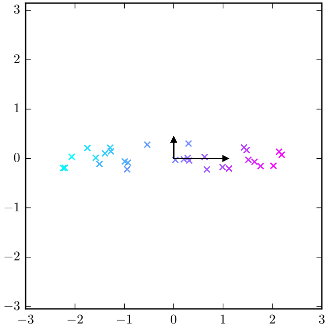
Compute covariance matrix Σ

- Eigenvectors $\mathbf{u}_1, \mathbf{u}_2$ of Σ are sought direction vectors



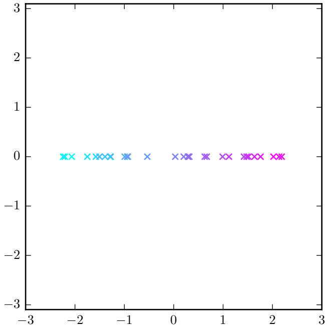
Represent $\mathbf{x}_k$ in the $U = (\mathbf{u}_1, \mathbf{u}_2)$ basis, $\mathbf{x}_k^r = U^\top \mathbf{x}_k$

- $\mathbf{u}_1, \mathbf{u}_2$ are orthogonal, so U is a rotation matrix

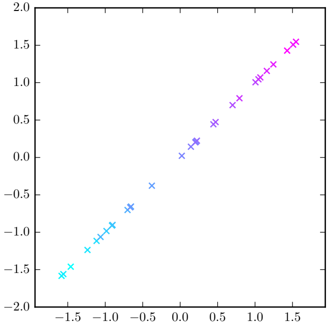


Now we can simply drop features that vary little

- Encoded by the corresponding eigenvalues λ_1, λ_2



If desired we can approximate $\mathbf{x}_k$ by multiplying with U



This method is called *Principal Component Analysis (PCA)*

Used to find features that retain e.g. 99% of variance

- Often leads to a significant dimensionality reduction

Used to perform *whitening*

- To obtain uncorrelated features with same variance
- Needed by some ML algorithms

PCA is unsupervised (no $\dot{\mathbf{w}}$ required) and linear

- There are more powerful supervised / non-linear methods

Bibliography I

Bengio, Yoshua, Ian Goodfellow, and Aaron Courville (2015). *Deep Learning (Draft)*. MIT Press.

Debevec, Paul E. (1996). *Modeling and Rendering Architecture from Photographs*. PhD thesis. Berkley.

Fischler, Martin A and Robert A Elschlager (1973). *The representation and matching of pictorial structures*. IEEE Transactions on Computers.

LeCun, Yann et al. (1989). *Backpropagation applied to handwritten zip code recognition*. Neural computation.

Prince, S.J.D. (2012). *Computer Vision: Models Learning and Inference*. Cambridge University Press.

Bibliography II

Roberts, Lawrence Gilman (1963). *Machine perception of three-dimensional solids*. PhD thesis. MIT.

Shotton, Jamie et al. (2011). *Real-Time Human Pose Recognition in Parts from a Single Depth Image*. CVPR.

Snavely, Noah, Steven M. Seitz, and Richard Szeliski (2006). *Photo tourism: Exploring photo collections in 3D*. SIGGRAPH.

Szeliski, Richard (2010). *Computer vision: algorithms and applications*. Springer.

Tuytelaars, Tinne and Krystian Mikolajczyk (2008). *Local invariant feature detectors: a survey*. Foundations and Trends in Computer Graphics and Vision.