Image decomposition

Overview:

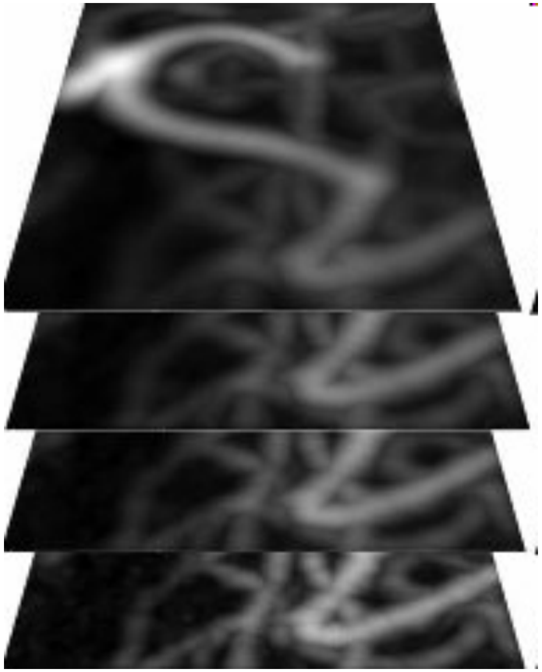
- Scale-space
- Unitary transform definition
- Generic transforms (methods)
- PCA: Domain-specific transforms

Scale Space

Scale space: goal

Scenes contain information at different levels of detail

1. Develop hierarchical descriptions
2. Increase efficiency by working on lower resolutions

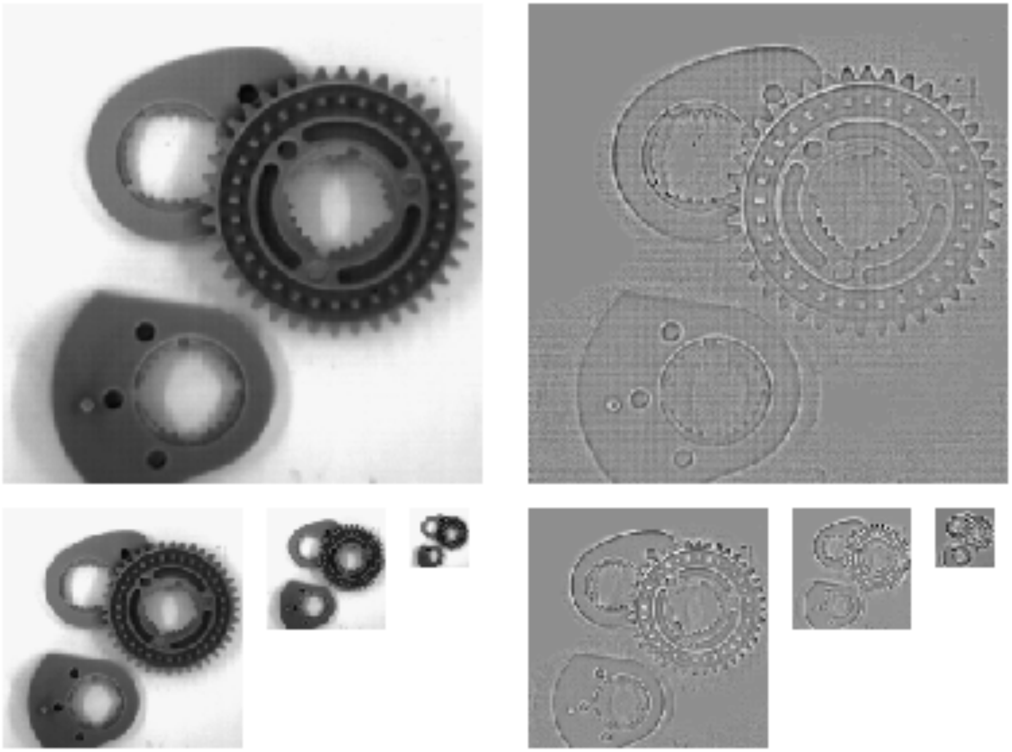


Psychophysical and neurophysiological relevance

Computer Vision

Scale space: pyramids

Gaussian-Laplacian pyramid:



Remark spatial coincidence at all scales of important edges

Scale space : pyramids

For image I_i

1. Smooth I_i (with Gaussian) => S_i
2. Take difference image: (DoG ~ Laplacian)

$$L_i = I_i - S_i$$

3. Reduce size of smoothed image

$$I_{i+1} = \text{down-sample}(S_i)$$

The 3rd step is allowed following the Nyquist theorem (i.e., given sufficient smoothing)

Zero-crossings of the Laplacian yield edges, thus interesting information in the Laplacian pyramid

Scale space: discrete

Discrete approximations of the Gaussian filters

Spurious structures might emerge!

e.g. small smoothing filter with positive coefficients c_{-1}, c_0, c_1

$$\Downarrow$$

make sure that $c_0^2 \geq 4c_{-1}c_1$

Thus, [1,2,1] is a valid scale space filter, whereas [1,1,1] is not.

Unitary Transforms

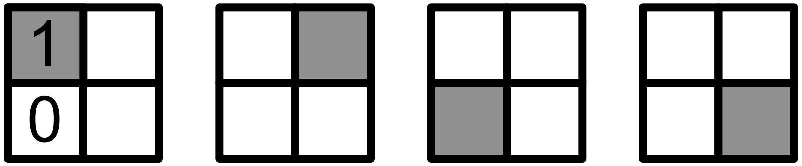
Unitary image transforms

Image decomposition into a family of orthonormal *basis images*

Decomposition as linear combination of basis vectors/images

2 examples so far:

1. *Pixelwise decomposition:* 1 Dirac impulse at the corresponding pixel in each basis image
   (perfect localization in image space, none in frequency)



Example: For 2x2 images

2. *Fourier decomposition:* 1 oriented cosine/sine pattern in each basis image
   (perfect localization in frequency domain, none in space)

Unitary transforms

Unitary operators:
"preserving the inner product", i.e. U*U=UU*=I
For real funcs, only possible (iff) columns of U are orthonormal
(orthonormal: inner-product of all components with self =1, others =0)

- Fourier transform (follows from Parseval's theorem)

- Rotations are unitary (does not change vector lengths)

- Pixelwise/Fourier have orthonormal basis images

Unitary transforms

Properties:

- Concentrate energy in a few components, i.e. only few basis images that can faithfully represent
- Compromise localization in space/frequency

(other examples of decompositions given later for more balanced localizations in different spaces)

Image independent rotations

(rotations, because new axes also orthonormal
+ Euclidean distance preserved)
(image independent transforms are generic
but suboptimal, as opposed to PCA that we will see later)

E.g.: decomposition as Dirac impulses or Fourier domain is decided without knowing type/content of images

Computer Vision

Basis images: Orthonormal

Orthonormal basis images B conform:

$$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} B_i(x,y)\ \ B_j^*(x,y) = \delta_{ij}$$

with * indicating the complete conjugate

necessary and sufficient

- We do want basis images linearly independent of each other ➔ orthogonal: Bi Bj=0
- We do not want an all zero basis B, which would generate zero under any linear combination, thus be useless in representing anything
- In fact, better to have a unit length B ➔ thus Bi Bi=1

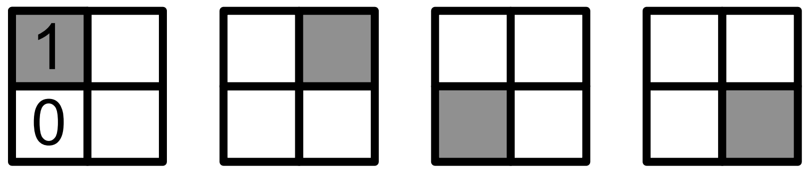
Let's check if these for the basis images:

- Dirac impulses in pixelwise
- cos/sin in Fourier

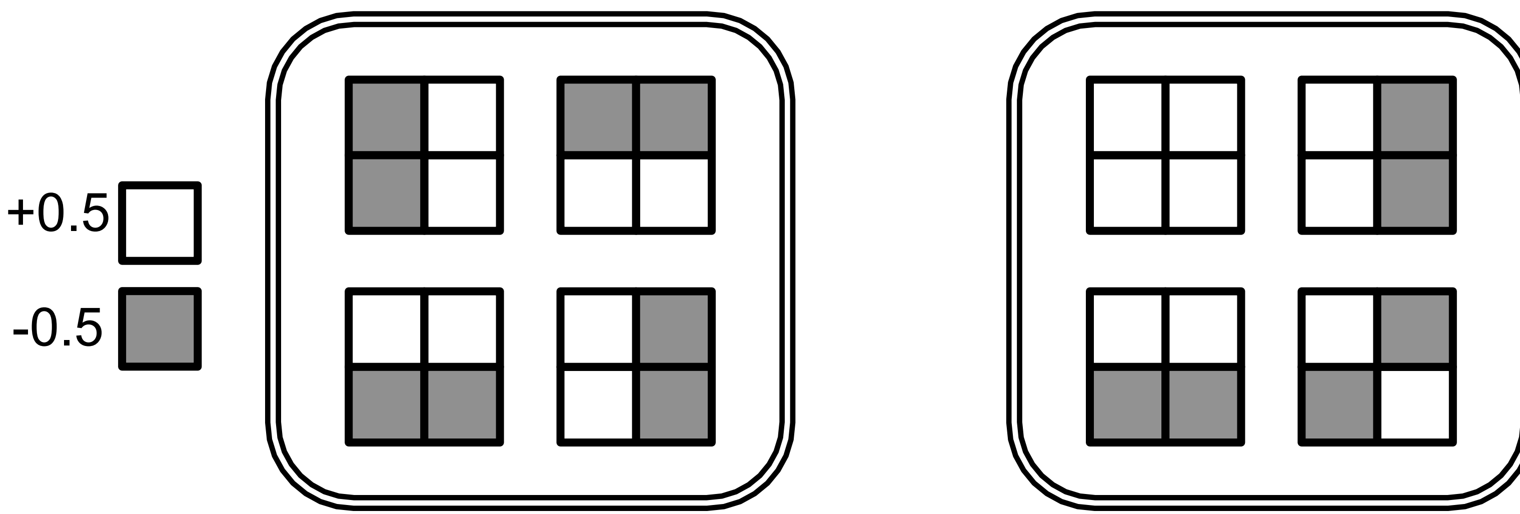
Basis images: Orthonormal

$$\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} B_i(x,y)\ \ B_j^*(x,y) = \delta_{ij}$$

See it holds for Dirac impulses in pixelwise:



Can these be unitary decompositions?



(A) (B)

Orthogonality of functions

Example: period $P = \frac{2\pi}{\omega}$ of $\cos m\omega x$ for $m = 1,2,...$

$$\int_0^p \cos m\omega x \ \cos n\omega x \ dx = \delta_{mn} \frac{P}{2}$$

$$\int_0^p \cos m\omega x \ \sin n\omega x \ dx = 0$$

$$\int_0^p \sin m\omega x \ \sin n\omega x \ dx = \delta_{mn} \frac{P}{2}$$

For all positive values of m=1,2,… a countable set of orthogonal functions is generated

Generalization of vector calculus towards infinite dimensional space (Hilbert spaces)

Orthogonality of functions

Example: period $P = \frac{2\pi}{\omega}$ of $\cos m\omega x$ for $m = 1,2,\ldots$

$$\int_0^p \cos m\omega x \ \cos n\omega x \ dx = \delta_{mn} \frac{P}{2}$$

$$\int_0^p \cos m\omega x \ \sin n\omega x \ dx = 0$$

$$\int_0^p \sin m\omega x \ \sin n\omega x \ dx = \delta_{mn} \frac{P}{2}$$

Problems with infinite dimensions: representation need not be unique (e.g. aliased freqs) & may not be complete (even funcs)

These problem disappear with discretization

Completeness condition

Arbitrary square-integrable functions characterized by their correlations with the basis set of orthonormal functions

(N x 1) sample vectors ⇨ any N orthogonal bases will be complete ⇨ need for finding them

In discrete, problem is how to find **sufficient** number of orthogonal basis functions. Example with 16 samples:

- Cos set is all **orthogonal, BUT** they repeat (9 & 7 are identical)
- To no surprise, **odd funcs** cannot be represented by cos set
- Sine can represent odds, thus Fourier basis funcs is a complete set
- This yields **16 orthogonal complex** trigonometric basis funcs

E.g. $\cos\frac{2\pi}{16}ux \quad x = 0,1,2\ldots15$, and $u = 0,1,\ldots,8$

other *u*'s identical, but signs reversed; e.g. u=7 & u=9 identical

$\sin\frac{2\pi}{16}ux \quad u = 1\ldots7$ functions with *u=0* and *u=8* vanish

Hence, 16 Fourier basis funcs of form: $\frac{1}{N}e^{-2\pi i\frac{ux}{N}}$

Basis images: Separable

1-D → higher dimensions

$$B_{ij}(x,y) = \phi_i(x)\ \psi_j(y)$$

Or, equivalently

preferred

$$B_{ij} = \phi_i \psi^t_j$$

(can be decomposed into products of 1D functions)

Many basis functions are not separable, but pixelwise (Dirac) is, i.e. abscissa and ordinate).
We will consider *separable* basis images (with which image analysis operation can be run faster)

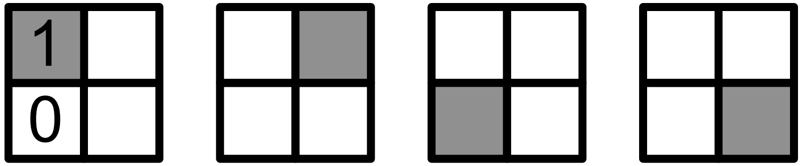
In case of a transition from an orthonormal set to another orthonormal set :

unitary transform matrices $A^{-1} = A^{*\,t}$

Computer Vision

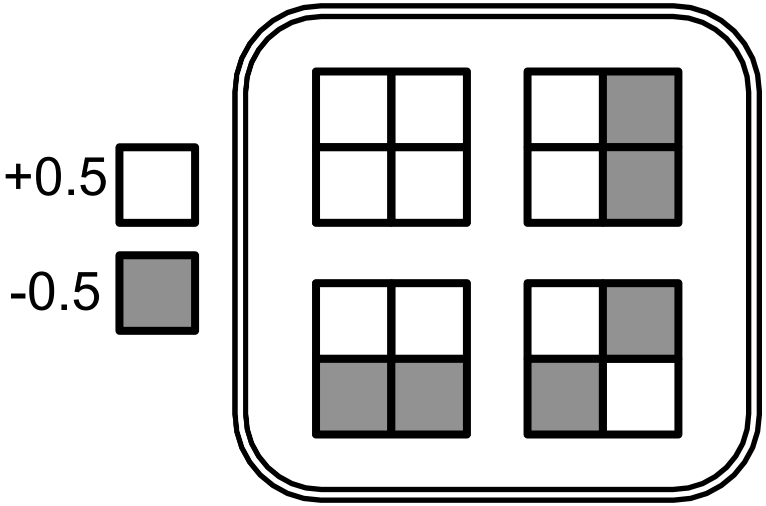
Orthonormal

Pixelwise:



φ_i^t : [1 0], [0 1]

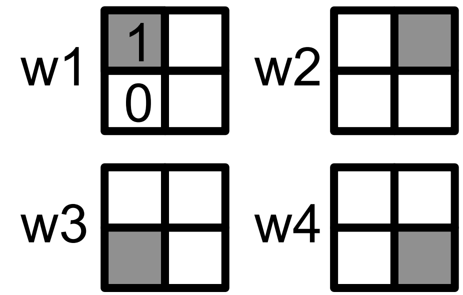
Is this (unitary) decomposition separable?
If so, what are φ_i^t ?



Decomposition of images

Now we decided B, but how to find basis weights w_{uv} to represent a given image:

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} w_{uv} B_{uv}(x,y)$$



For a given basis $B_{u'v'}$ in order to find the weight $w_{u'v'}$

Multiply and sum both sides, then use orthonormality:

$$\begin{aligned}
&\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) B^{*}_{u'v'}(x,y)\\
&=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} w_{uv} B_{uv}(x,y)\right) B^{*}_{u'v'}(x,y)\\
&=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} w_{uv}\left(\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} B_{uv}(x,y) B^{*}_{u'v'}(x,y)\right)\\
&=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} w_{uv}\delta_{u'v'}\\
&=w_{u'v'}
\end{aligned}$$

Computer Vision

Decomposition of images (2)

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} w_{uv} B_{uv}(x,y)$$

cf. projection of vector onto basis vectors or as correlation with reference patterns

Transformed image: $F(u,v) = w_{uv}$

Forward transform:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) B^{*}_{uv}(x,y)$$

Backward transform:

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v) B_{uv}(x,y)$$

Optimal truncation property

GOAL: Find truncated decomposition

$$\hat{f}(x,y) = \sum_{u=0}^{M'-1} \sum_{v=0}^{N'-1} c_{uv} B_{uv}(x,y)$$

Find a smaller number of basis funcs:
Which weights to use if not all retained

with $M' < M$ and $N' < N$ that minimizes

$$e_{M'N'} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(f(x,y) - \hat{f}(x,y) \right)^2$$

Minimize the approximation error

Optimal truncation property

THEOREM:

The weights w_{uv} that minimize $e_{M'N'}$ are given by

$$w_{uv} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) B_{uv}^{*}(x, y)$$

Show that these weights are indeed the ones from the original decomposition

Optimal truncation property

Proof : Show that other weights c_{uv} → larger $e_{M'N'}$

$$c_{uv} = w_{uv} - (w_{uv} - c_{uv})$$

$$e_{M'N'} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(f(x,y) - \hat{f}(x,y)\right)^2$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left| f(x,y) - \sum_{u=0}^{M'-1}\sum_{v=0}^{N'-1} c_{uv}B_{uv}(x,y)\right|^2$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left| f(x,y) - \sum_{u=0}^{M'-1}\sum_{v=0}^{N'-1} w_{uv}B_{uv}(x,y) + \sum_{u=0}^{M'-1}\sum_{v=0}^{N'-1}(w_{uv} - c_{uv})B_{uv}(x,y)\right|^2$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left| \sum_{u=M'}^{M-1}\sum_{v=N'}^{N-1} w_{uv}B_{uv}(x,y)\right|^2 + \sum_{u=0}^{M'-1}\sum_{v=0}^{N'-1}\left|w_{uv} - c_{uv}\right|^2$$

Last term is positive and is minimized for $c_{uv} = w_{uv}$

Optimal truncation property

This theorem underlies the use of unitary transforms for *image compression* applications

Energy in images tends to be concentrated in lower frequencies
taking more terms always improves the result:
for $c_{uv} = w_{uv}$:

$$e_{M'N'}$$

$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| \sum_{u=M'}^{M-1} \sum_{v=N'}^{N-1} w_{uv} B_{uv}(x,y) \right|^2$$

$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(\sum_{u=M'}^{M-1} \sum_{v=N'}^{N-1} \left| w_{uv} \right|^2 \right)$$

Examples of unitary transforms

Assuming square images

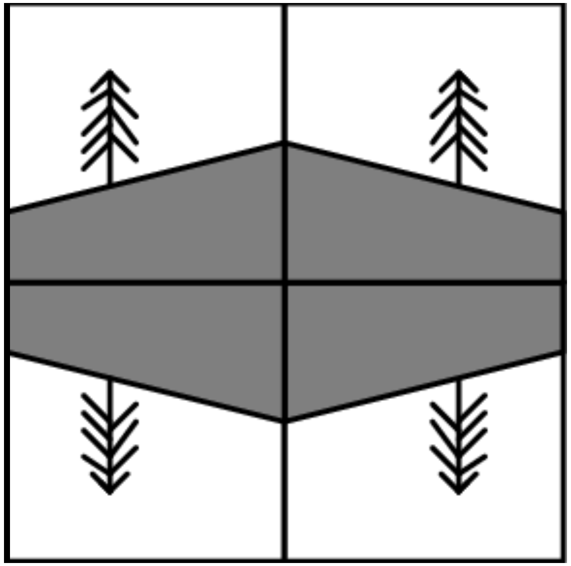
- 1. Cosine transform
- 2. Sine transform
- 3. Hadamard transform
- 4. Haar transform
- 5. Slant transform

Generally, we seek decompositions with strong compaction; driven by practical experience and implementation efficiency

Cosine transform gives best decorrelation

The cosine transform

Turning Fourier into real transform and suppression of spurious high frequencies:



The extended image is even

The cosine transform

DFT of the extended image:

$$F_e(u,v) =$$

$$\frac{1}{4N^2}\sum_{x=-N}^{N-1}\sum_{y=-N}^{N-1} f_e(x,y)e^{-2\pi i\left(\frac{u(x+1/2)}{2N}+\frac{v(y+1/2)}{2N}\right)}$$

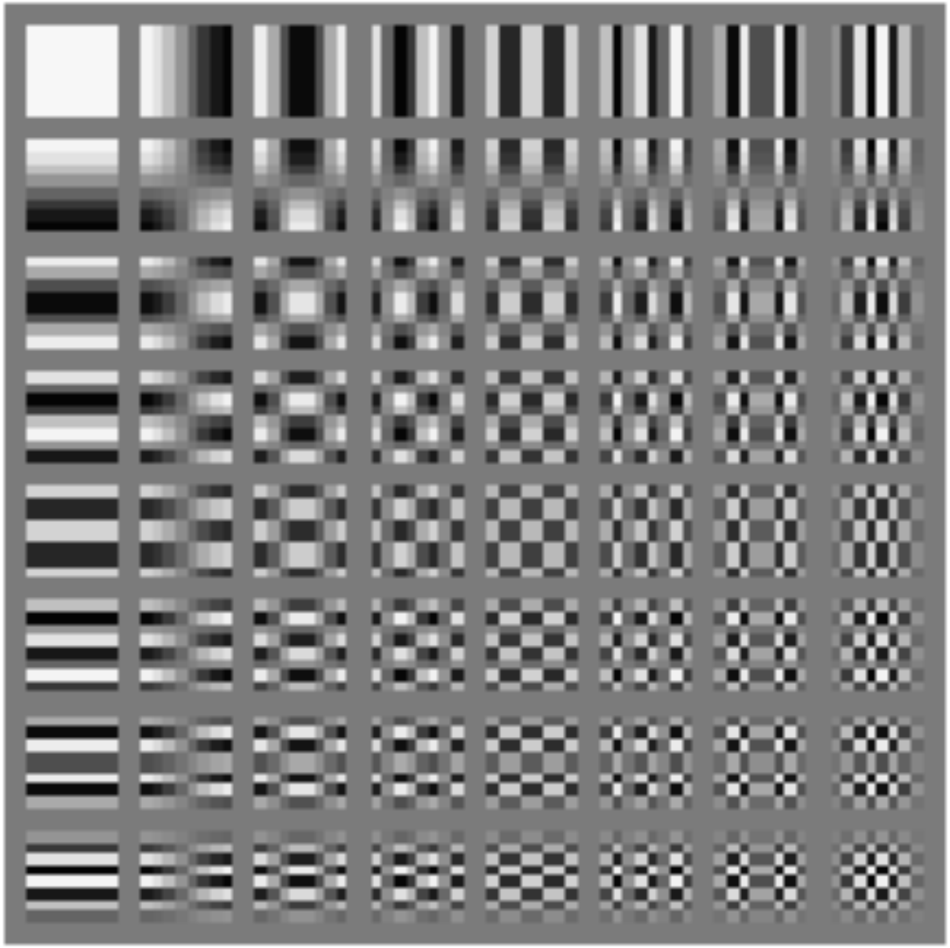
Domain [-N .. N], normalized by $4N^2$
Because $f_e(x,y)$ is even, sines disappear:

$$\frac{1}{N^2}\sum_{x=-N}^{N-1}\sum_{y=-N}^{N-1} f_e(x,y)\cos\left(\frac{\pi}{N}u(x+1/2)\right)\cos\left(\frac{\pi}{N}v(y+1/2)\right)$$

separable

The cosine transform

8 x 8 basis images:



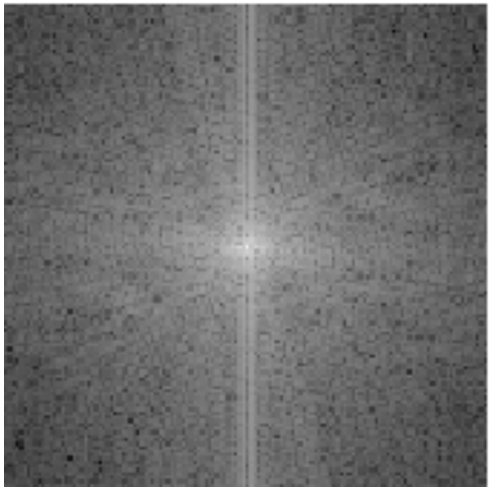
The cosine transform

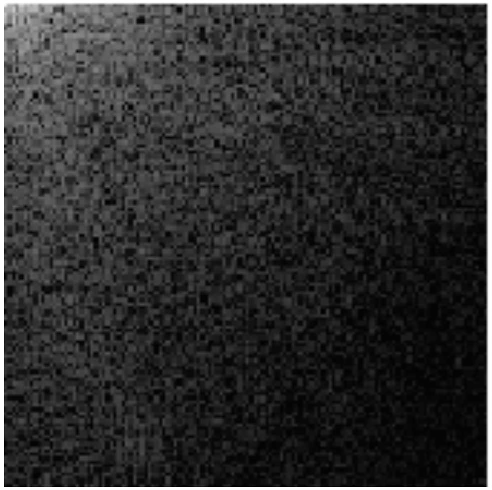
Remarks on the DCT:

1. Eliminates the boundary discontinuities
2. Components are well decorrelated
3. Has $\mathcal{O}(n \log n)$ implementations
4. Requires real computations only
5. DCT chips are available
6. Was long time the most popular compression basis

The cosine transform

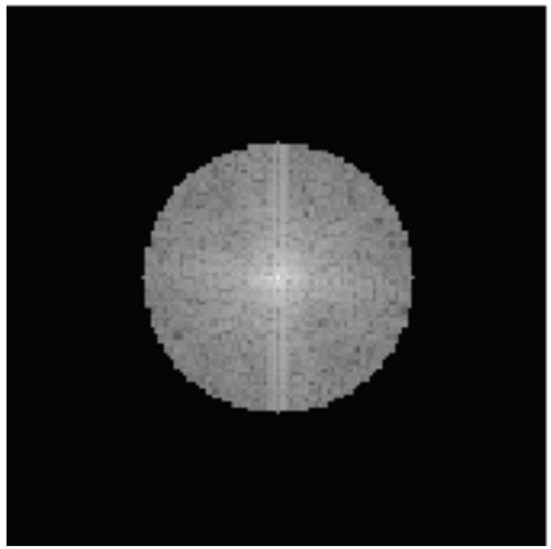
Left : DFT, right : DCT

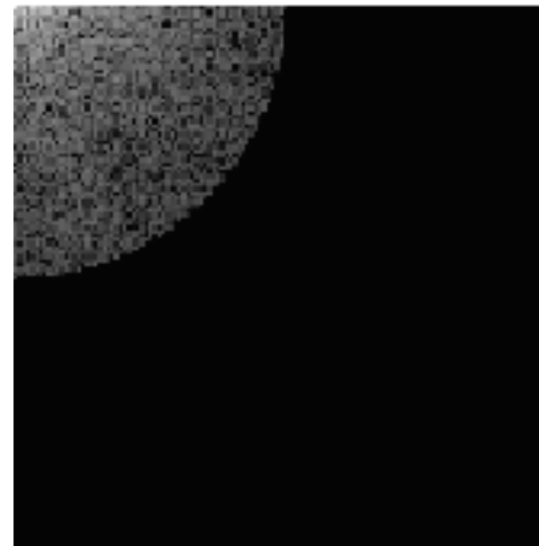




The cosine transform

Zonal truncations:





When the same number of samples are retained in both cases (i.e., same compression ratio)

The cosine transform

DFT

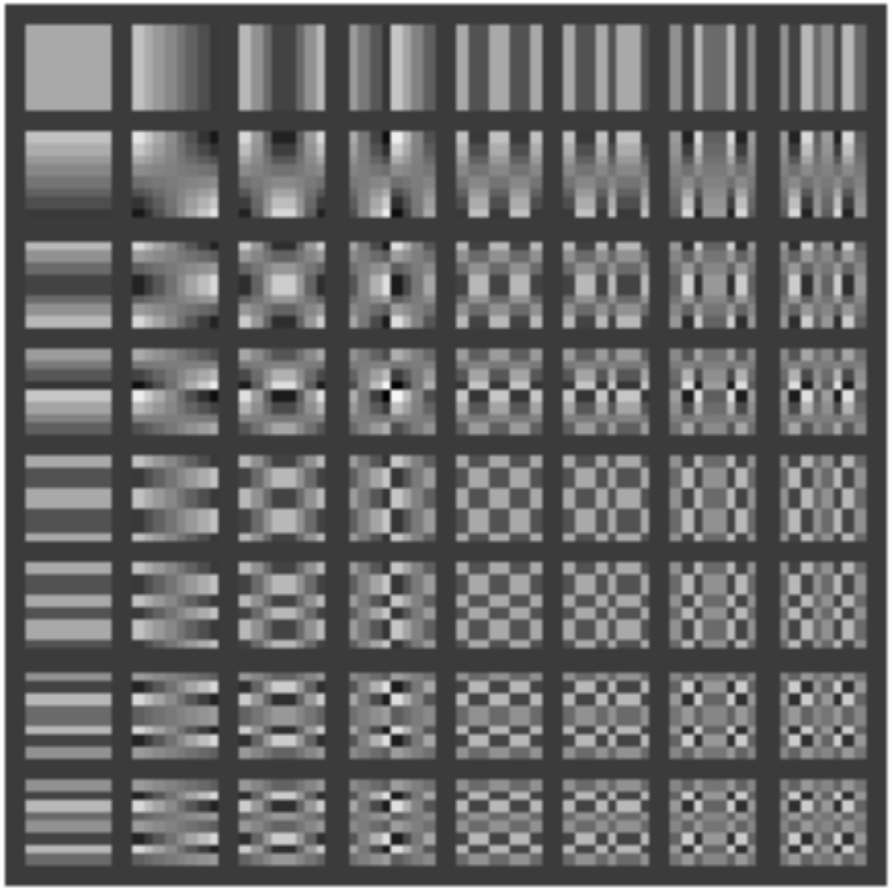
Horizontal top/bottom ripple, spurious high frequencies

DCT

The slant transform

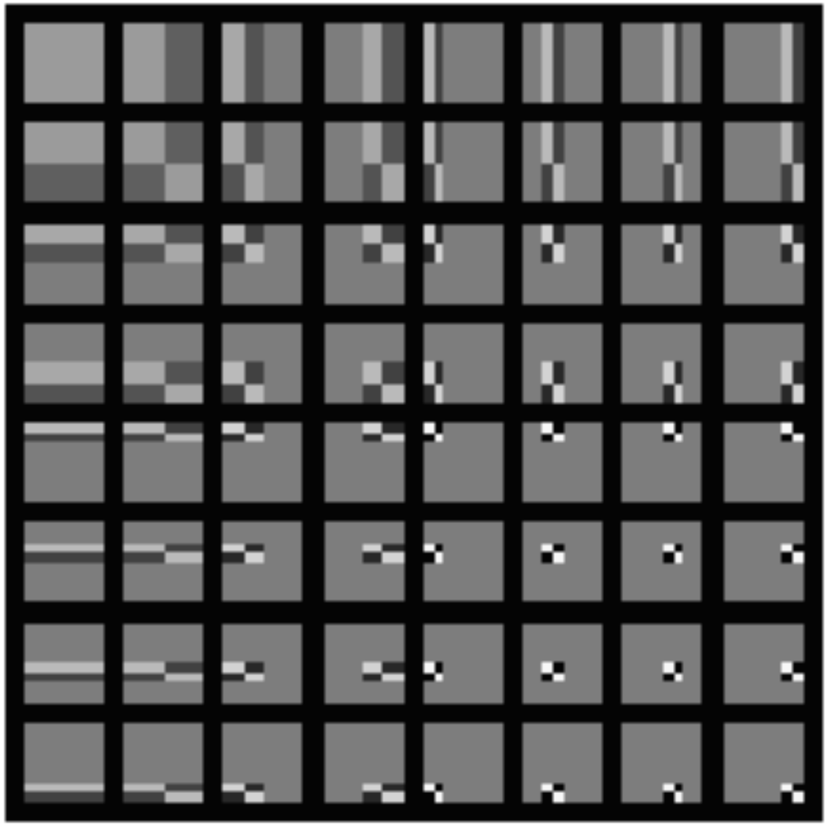
on the basis of slant matrices

e.g. basis images for 8 x 8 :



discrete sawtooth-like basis vectors which efficiently represent linear brightness variations along an image line

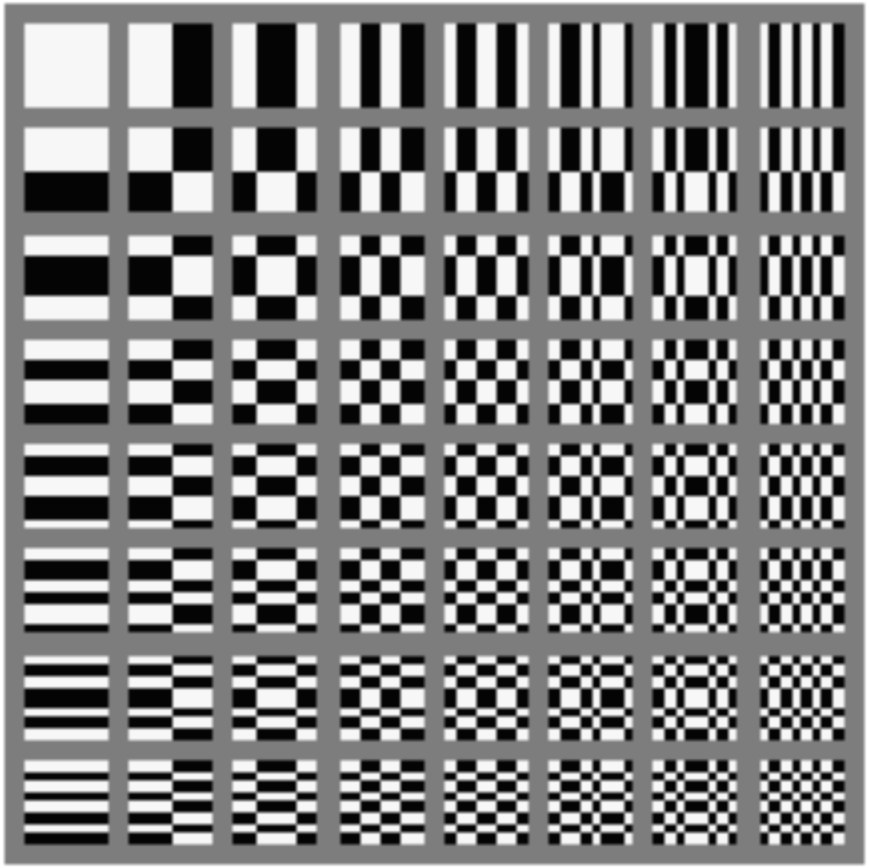
The Haar transform



- Is an example of a wavelet transform
- Note how the analysis is localised both in space and in terms of frequency
- Note also that for higher frequencies, the spatial extent gets smaller, a typical feature of wavelets

Computer Vision

The Hadamard transform



- Only 1s and -1s, therefore no multiplication needed: one of the first for HW implementation
- Recursive operation of [1 1; 1 -1]
- Generates minimally correlated binary blocks
- Binary ➔ efficient ➔ barcode reading
- All examples had same orthogonal set for rows&cols, BUT need not be so, e.g. Haar X Hadamard possible

Principal Component Analysis

Principal component analysis: goals 1

Decorrelation of data

E.g. image independent transforms suboptimal

⇓

Karhunen-Loève Transform (KLT)
extract statistics from images for a customized orthogonal basis set with uncorrelated weights

PCA: technique based on eigenvectors of the covariance matrix

PCA: goals 2

Central idea:

Reduce the dimensionality of data consisting of many interrelated variables, while retaining as much as possible of the variation

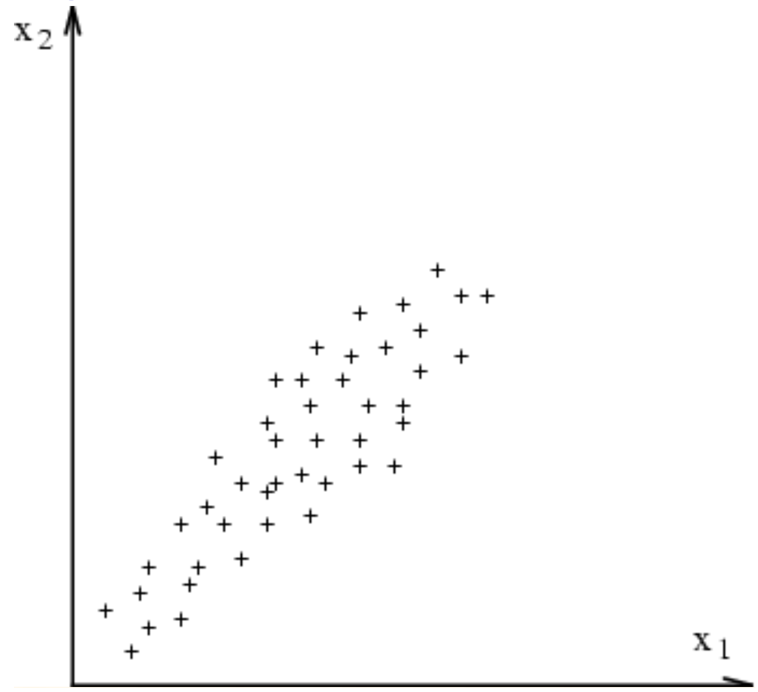
Achieved by transforming to new, uncorrelated variables, the principal components, which are ordered so that the first few retain most of the variation

Remarks:

- For a diverse set, PCA will resemble DCT (optimal decorr)
- Receiver needs the bases
- Uses, e.g. feature selection for classification or inspection

Computer Vision

Highlighting the essence via decorrelation

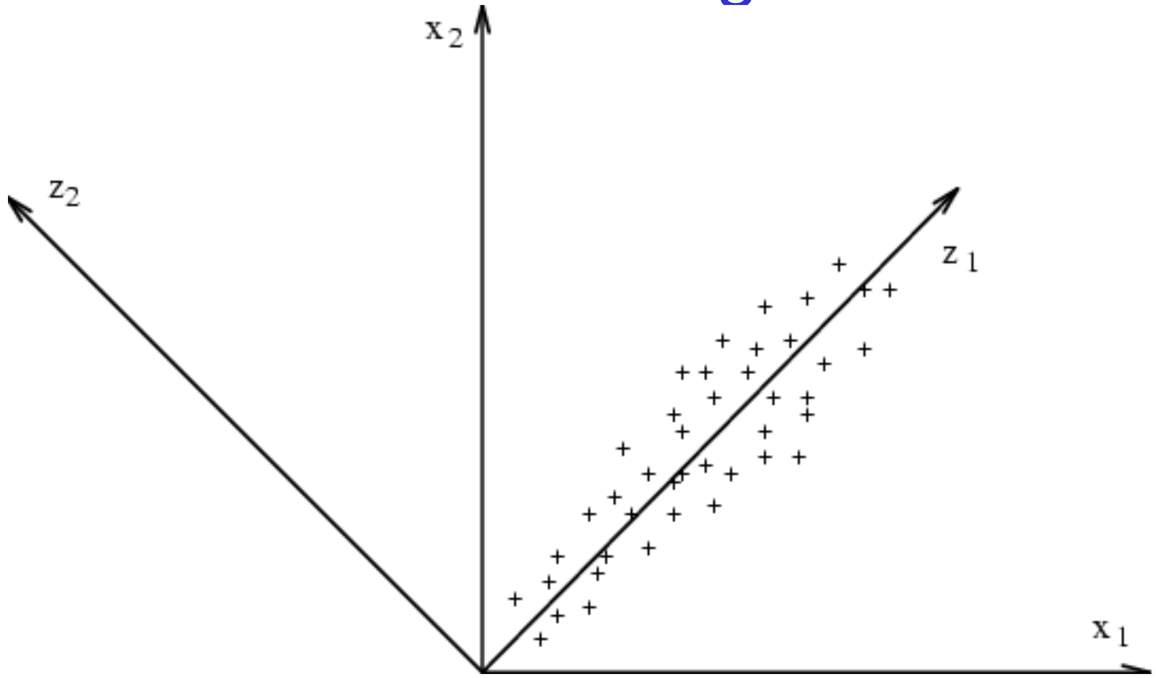


- Observations with two highly-correlated variables: e.g. grey-value at neighbouring pixels OR length&weight of growing children
- Highly correlated values: x1 has info on x2
- Instead of storing 2 variables, we can store only 1

knowledge about correlation helps in compression, inspection, and classification

Computer Vision

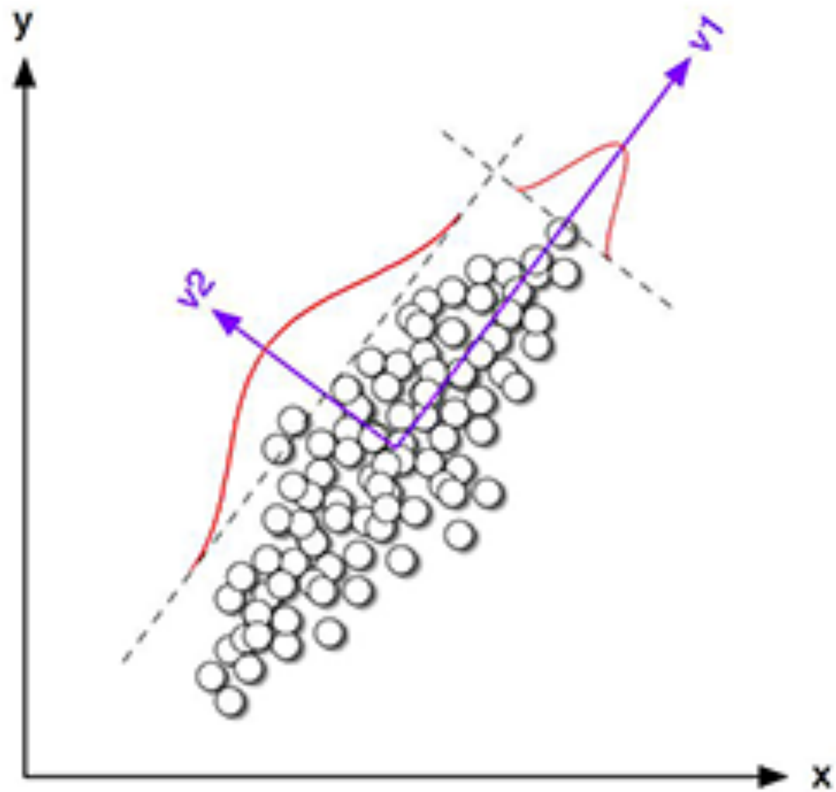
Decorrelation through rotation



- Using correlation, rotate frame axes: Variation in 1st component max, in 2nd min
- (Can potentially drop z_2 now)

principle behind *unitary transforms* :
rotation in high dimensional spaces

Decorrelation through rotation



We will work around the mean

- So, we apply a rotation (e.g., from ellipse fitting) about the mean of the distribution
- Extends to hyperellipsoids in higher dimensions, where visual inspection is not possible

Decorrelation through rotation

Sum of variances do not change with rotations:

$$\sum_{i=1}^{p} \sigma_i^2 = \sum_{j=1}^{p} \tilde{\sigma}_j^2$$

With σ_i^2 variance in x_i and $\tilde{\sigma}_j^2$ variance in z_j

result of invariance of center of gravity and distance under rotation

Parseval equation

redistribution of energy / variance

We want as much variance in as few coordinates

PCA: introduction

In high dimensional spaces, an optimal rotation no longer clear upon visual inspection

Some statistics needed: covariance matrix
(note: underlying assumption of Gaussian distr.!)

Intuitive: fit hyperellipsoid to cluster
subsequent PCs correspond to axes from the longest to the shortest

PCA: method

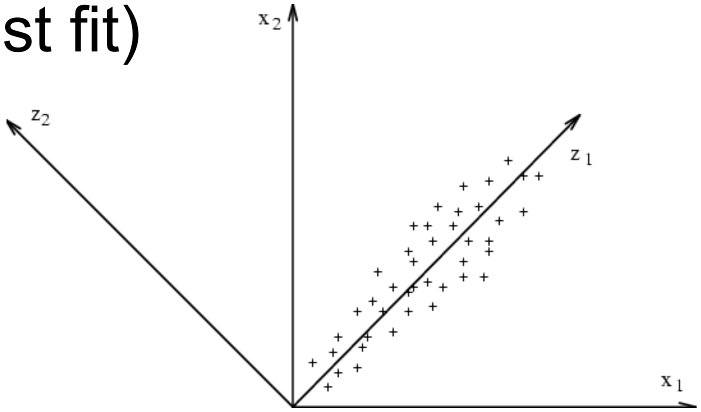
Suppose x is a vector of p random variables

(can extend to points in space & images with pixels)

first step: look for a linear combination $c_1^T x$ which has maximum variance (fitting a line in R^N)

second step: look for a linear combination, $c_2^T x$ uncorrelated (orthogonal) with $c_1^T x$ and with maximum variance (best fit)

third step: repeat...



Computer Vision

Algorithm : Find PCA basis formally

1. Calculate the covariance matrix C, *suppose the data are centered around the mean*

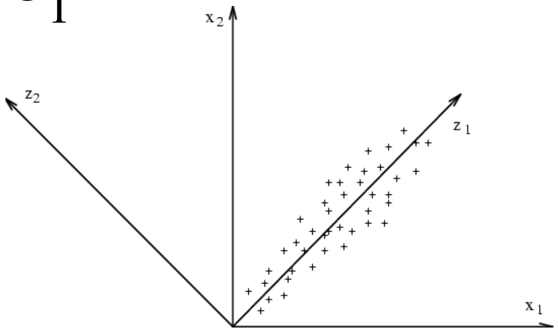
2. Consider $c_1^T x$ with c_1 and maximize its variance
   var [$c_1^T x$] =

$$\sum c_1^T x \, (c_1^T x)^T = \sum c_1^T x \, x^T c_1 \; = \; c_1^T \sum (x \, x^T) c_1$$

$$= \; c_1^T C \, c_1 \quad \text{is maximized,}$$

but

3. Normalize to find a finite c_1 : $c_1^T c_1 = 1$



PCA algorithm: c_1

Using Lagrange multipliers we maximize

$$c_1^T C c_1 - \lambda\left(c_1^T c_1 - 1\right)$$

Differentiation w.r.t. c_1 gives

$$C c_1 - \lambda c_1 = 0$$

$$\left(C - \lambda\, I_p\right) c_1 = 0$$

where $\boldsymbol{I_p}$ is the ($\boldsymbol{p}$ x $\boldsymbol{p}$) identity matrix

Thus, λ is an eigenvalue of C

$\boldsymbol{c_1}$ is the corresponding eigenvector

PCA algorithm: c_1

Which of the ***p*** eigenvectors?

$$c_1^T C c_1 = c_1^T \lambda c_1 = \lambda c_1^T c_1 = \lambda$$

So λ must be as large as possible

Thus, c_1 is the eigenvector with the largest eigenvalue

The k^{th} PC is the eigenvector with the k^{th} largest eigenvalue

Computer Vision

PCA algorithm: c_2

Proof for $\boldsymbol{k}$ = 2

Maximize $c_2^T C c_2$ while uncorrelated with $c_1^T x$

$$\operatorname{cov}\left[c_1^T x, c_2^T x\right] =$$

$$c_1^T C c_2 = c_2^T C c_1 = c_2^T \lambda_1 c_1 = \lambda_1 c_2^T c_1 = \lambda_1 c_1^T c_2$$

Thus uncorrelatedness becomes

$$c_1^T C\, c_2 = 0,\ c_2^T C\, c_1 = 0,\ c_1^T c_2 = 0,\ c_2^T c_1 = 0$$

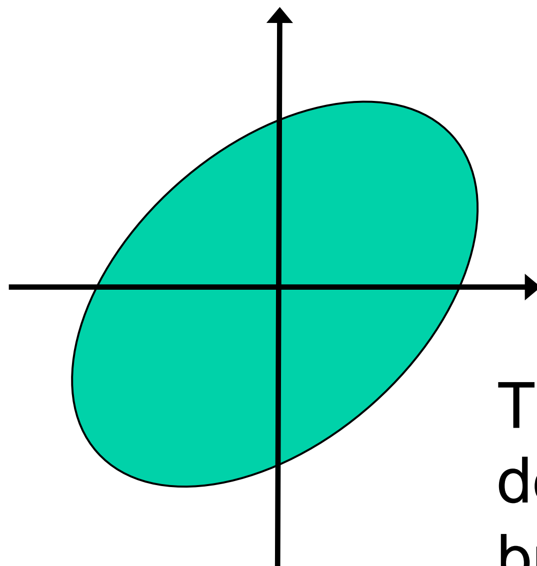
PCA algorithm: c_2

decorrelation vs. orthogonality

$$c_1^T C\, c_2 = 0,\ \ c_2^T C\, c_1 = 0,\ \ c_1^T c_2 = 0,\ \ c_2^T c_1 = 0$$

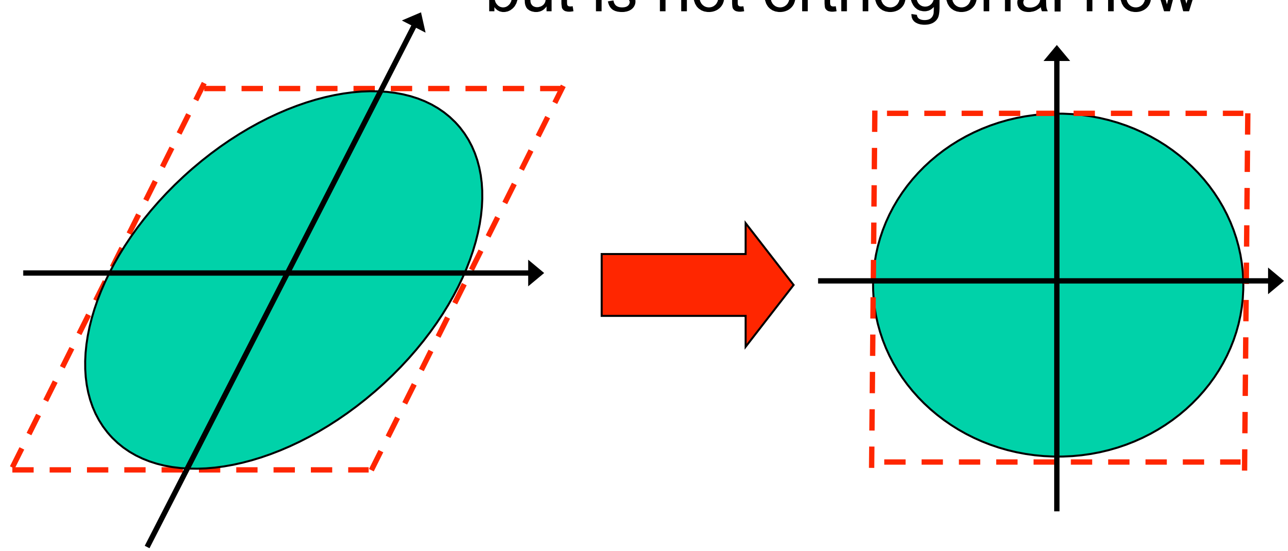
go hand in hand only for main axes of the ellipsoid defined by the covariance matrix !

PCA: Decorralation vs. orthogonality example



The given axes are orthogonal,
But this 2D distribution has
correlated projections on them

The conjugate axis yields
decorrelation,
but is not orthogonal now



Thus, we should satisfy both decorrelation and orthogonality

Computer Vision

PCA algorithm: c_2

Then, using λ, ϕ as Lagrange multipliers

$$c_2^T C c_2 - \lambda\left(c_2^T c_2 - 1\right) - \phi\, c_2^T c_1$$

Differentiation w.r.t. c_2 gives

$$Cc_2 - \lambda c_2 - \phi c_1 = 0$$

Multiplication on the left by c_1^T gives

$$c_1^T C c_2 - \lambda c_1^T c_2 - \phi\, c_1^T c_1 = 0$$

Thus ϕ = 0

Therefore, $Cc_2 - \lambda\, c_2 = 0$, i.e. $\left(C - \lambda I_p\right) c_2 = 0$

Again, maximize $c_2^T C c_2 = \lambda$, so select 2nd largest λ_2

PCA: interpretation

Similarly, the other PCs can be shown to be eigenvectors of *C* corresponding to the subsequently next largest eigenvalues

Because *C* is a real, symmetric matrix, we know all its eigenvectors will be orthogonal

We therefore can interpret PCA as a coordinate rotation/reflection in a higher dimensional space (orthogonal transformation)

Decorrelation through rotation

Principal Component Analysis (PCA):
collects maximum variance in subsequent uncorrelated components.

In that sense, it is the optimal rotation.

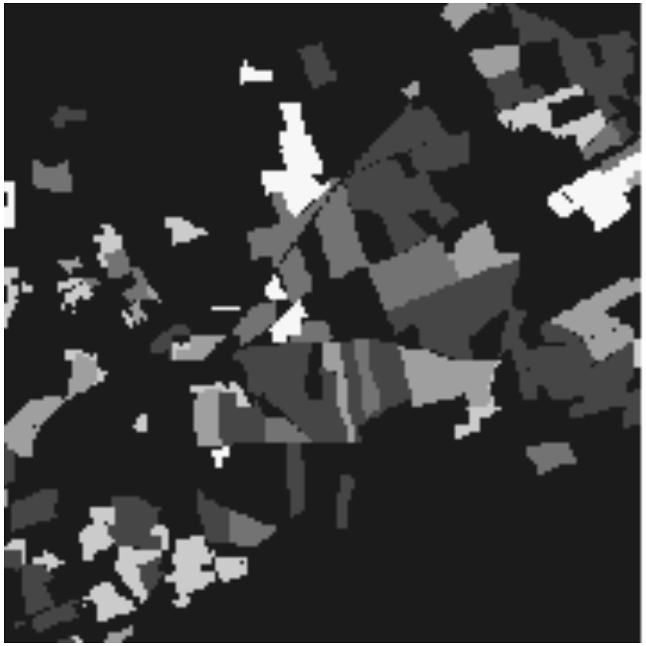
PCs can be interpreted as linear combinations of original variables.

Strongly correlated data $\Rightarrow$ first PCs contain most of the variance
information loss is minimal if only retaining these

Classification example with PCA: satellite images

Example: classification of 5 crop types

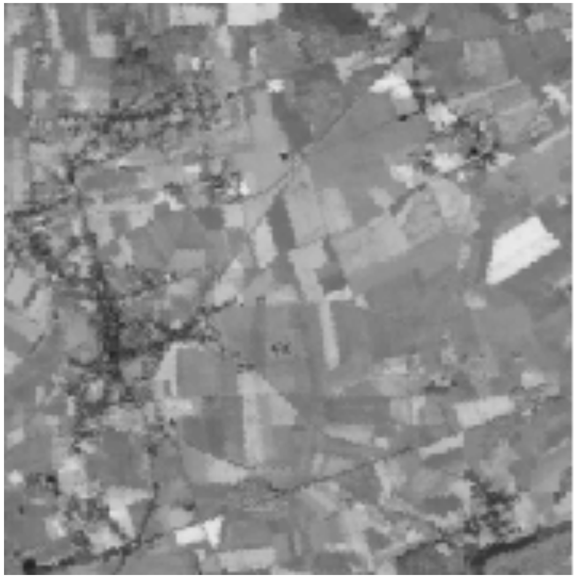
Input: 3 spectral bands from SPOT satellite
Near-infrared (N), Red (R), and Green (G)
each pixel = 20 m x 20 m



Comparison of 2 PCs vs. 3 original bands

Computer Vision

Classification example : satellite images



Computer Vision

Classification example : satellite images

Evident: correlation between R and G
N seems uncorrelated

Corroborated by covariance matrix:

$$C = \begin{pmatrix} 127.2447 & 13.3062 & -5.9095 \\ 13.3062 & 34.2264 & 39.2092 \\ -5.9095 & 39.2092 & 54.8805 \end{pmatrix}$$

Correlation coefficient $\sigma_{ij} = \dfrac{c_{ij}}{\sigma_i \sigma_j}$

$\sigma_{NR} = 0.2016$ and $\sigma_{RG} = 0.9047$

Classification example : satellite images

PCs:

$$\begin{pmatrix} 0.9907 \\ 0.1360 \\ -0.0070 \end{pmatrix} \quad \begin{pmatrix} -0.0765 \\ 0.5980 \\ 0.7978 \end{pmatrix} \quad \begin{pmatrix} -0.1127 \\ 0.7899 \\ -0.6028 \end{pmatrix}$$

Eigenvalues 129.1135, 84.8359, and 2.4022
1st PC ≈ near-infrared input N
notice low variance of 3rd PC

Classification results in this toy example:

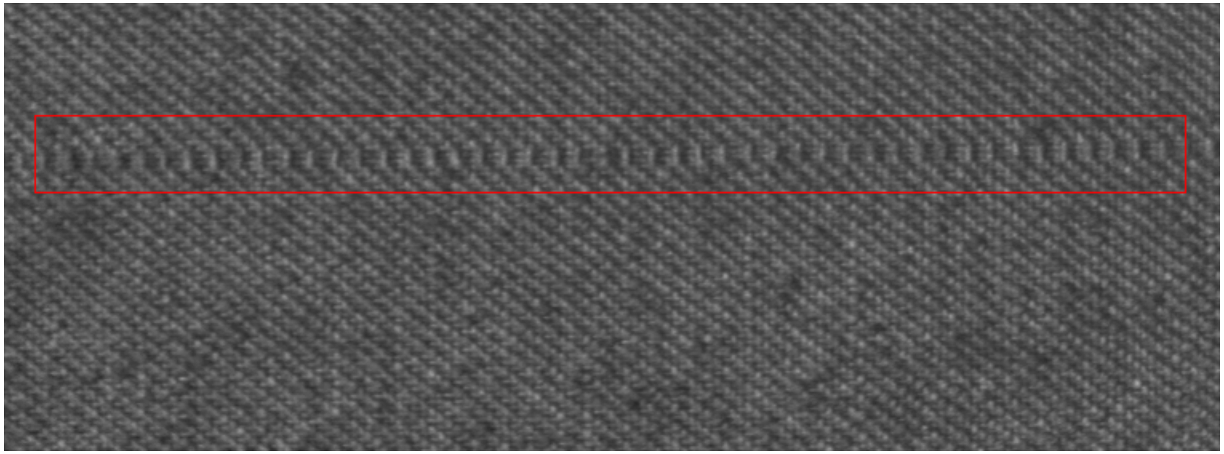
3 original bands: 76.3 % accuracy

2 first PCs: 73.5 % accuracy

(comp. : R-G: 60 %)

Inspection ex.: eigenfilters for textile

Example applications: textile inspection



Filters with size of one period [8,6]
(period found as peak in autocorrelation)

Inspection ex.: eigenfilters for textile

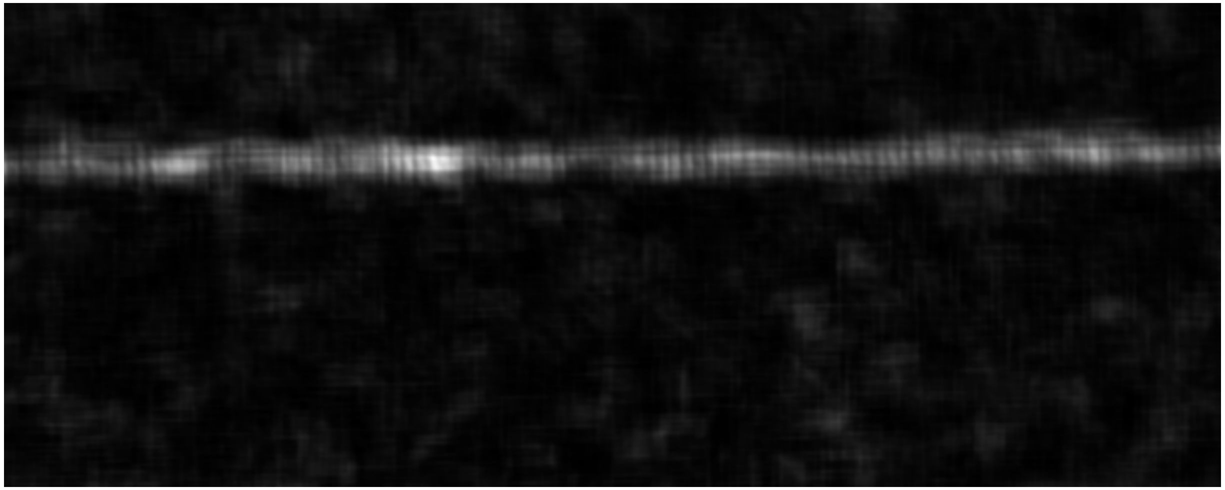
Further details will be given when we discuss texture analysis.

As we will see, PCA allows for the design of dedicated convolution filters, ordered by the variance in their output when applied across the image.

Flaws which won’t follow the typical pattern may then express itself in the low-variance components (as outlier values)

Inspection ex.: eigenfilters for textile

Mahalanobis distance of filter energies:



Flaw region found by thresholding:

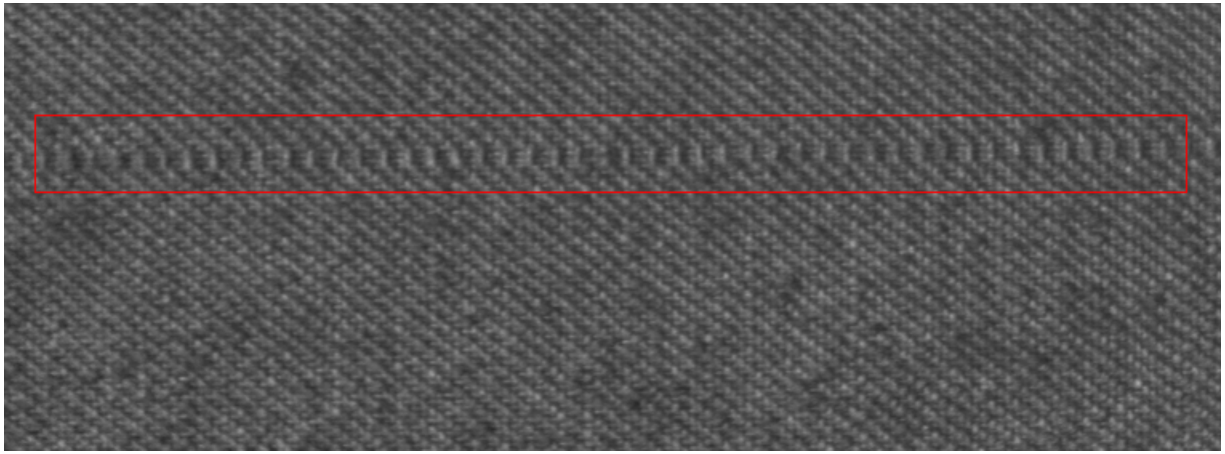


Image compression ex.: eigenfaces

Averaging of input faces

“Mean” face

Computer Vision

Image compression ex.: eigenfaces

Neighbouring pixel intensities are highly correlated

Consider image as large intensity vector

Eigenvectors: *"eigenimages"*

Computational problems :
N^2 x N^2 covariance matrices!

Specifying image statistics: which exemplary set?

Image dependence: eigenimages needed!

Image compression ex.: eigenfaces

Karhunen-Loève transform = PCA on images

Redistributes variance over a few components most efficiently

Best approximation: Minimal least-square error for truncated approximations

Dimensionality problem can be remedied: formulation as eigenvalue problem in space of dimension equal to number of sample images

Computer Vision

Dimension in number of samples/images

n samples, p-dimensional space, $n < p$

Consider the $(p \times n)$-matrix X with samples as columns

(p x p) covariance matrix

$$C = \frac{1}{n} X \; X^T$$

Much smaller (n x n) matrix

$$S = X^T X$$

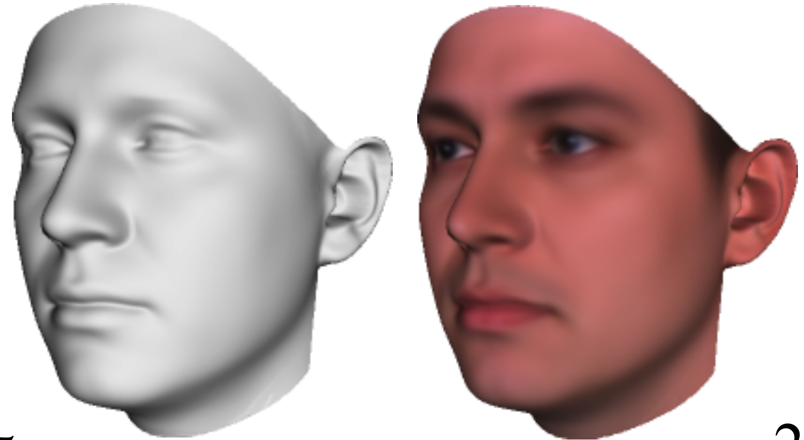
$$X^T X \; c_i = \lambda_i c_i$$

$$X \; X^T X \; c_i = X \; \lambda_i c_i$$

$$\left(\frac{1}{n} X \; X^T \right) \left(X \; c_i \right) = \left(\frac{\lambda_i}{n} \right) \left(X \; c_i \right)$$

Eigenvectors of a (n x n)-matrix need to be found

Image compression ex.: eigenfaces



-2σ +2σ

$$\mathbf{S} = \overline{\mathbf{S}} + \sum_{j=1}^{m} \alpha_j \mathbf{S}_j$$

$$\mathbf{T} = \overline{\mathbf{T}} + \sum_{j=1}^{m} \beta_j \mathbf{T}_j$$

-2σ +2σ