TSBB06 Multi-dimensional Signal Processing

Lecture 2C
Subspace Bases and Normalised Convolution

Subspaces

Recapitulation:

- Given a vector space *V*, a set *U* is a **subspace** if all $\mathbf{u} \in U$ satisfy the properties of a vector space
  - assumes that *U* has the same scalar field as *V*
- Since we assume that *V* is a scalar product space, *U* inherits this property from *V*
  - *U* uses the same scalar product as *V*
- *U* is *a proper subspace* of *V* if *U* is a proper subset of *V* (i.e., there exists $\mathbf{v} \in V$ such that $\mathbf{v} \notin U$)
- *V* is sometimes called the **embedding space** or **ambient space** of *U*

Orthogonal complement

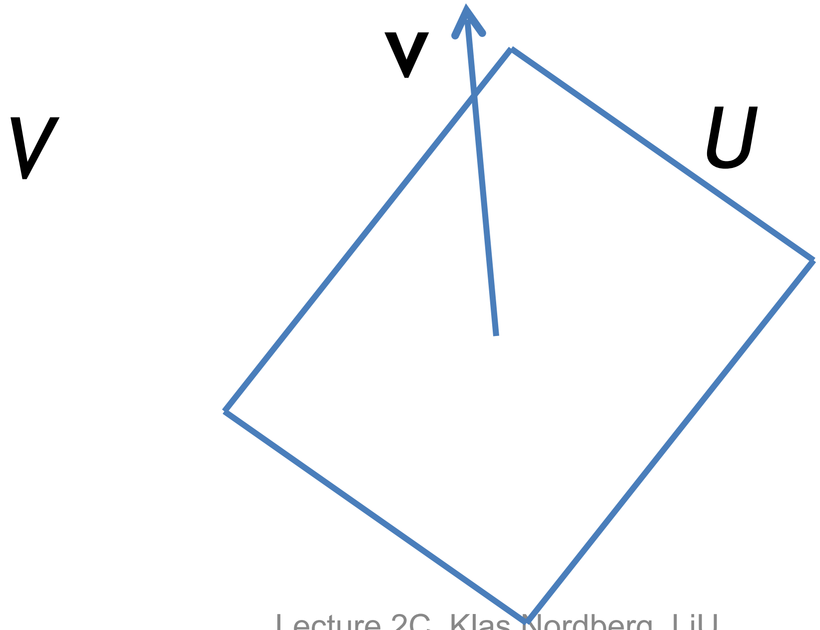
- Let *U* be a subspace of *V*
- We define $U_\perp$ as the set of all $\mathbf{v} \in V$ that are orthogonal to all $\mathbf{u} \in U$
- $U_\perp$ is itself a subspace of *V* (why?)

Subspace basis

- Let *U* be an *M*-dimensional subspace of *N*-dimensional vector space *V*, $M \leq N$
- Let $\mathbf{b}_k$, $k = 1, \ldots, M$, be a basis of *U*
- We refer to $\mathbf{b}_k$ as a *subspace basis*
- Since *U* is a scalar product vector space and we have a basis for *U*, all that is said about coordinate computations based on dual bases are valid also for *U*
  - However, this gives us coordinates of $\mathbf{u} \in U$, and not $\mathbf{v} \in V$, if U is a proper subspace

A general question

- Let $\mathbf{b}_k$ be a basis of a proper subspace $U \subset V$
- Let **v** be a vector in *V* (perhaps not in *U*)
- What can be said about **v** in this case?



A general observation

- $\mathbf{v} \in V$ can always be decomposed as

$$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_0$$

where $\mathbf{v}_1 \in U$ and $\mathbf{v}_0 \in U_\perp$

- This decomposition is unique (why?)

A general answer

In context of the previous question:

- We should be able to determine $\mathbf{v}_1$ and its coordinates relative to the subspace basis
- However, it may not be obvious how to determine $\mathbf{v}_1$, since we only know $\mathbf{v}$ at this point

A least squares problem

- Let **B** be the basis matrix of the subspace (**B** is $N \times M$ and known)
- Let **c** be a column vector of the coordinates of $\mathbf{v}_1$ (M-dimensional and not known)

$$\mathbf{v}_1 = \mathbf{B}\,\mathbf{c}$$

- We want to determine **c** such that

$$\epsilon(\mathbf{c}) = \|\mathbf{v} - \mathbf{B}\mathbf{c}\|^2$$

is minimised

This is a least squares problem

Solving least squares problems

- First we expand the norm using the scalar product

$$\|\mathbf{v} - \mathbf{Bc}\|^2 = \langle \mathbf{v} - \mathbf{Bc} | \mathbf{v} - \mathbf{Bc} \rangle$$

- Let $\mathbf{G}_0$ be the metric matrix

$$\|\mathbf{v} - \mathbf{Bc}\|^2 = (\mathbf{v} - \mathbf{Bc})^* \mathbf{G}_0 (\mathbf{v} - \mathbf{Bc})$$

Solving least squares problems

- Take the derivative wrt. **c** and set equal to zero to find the stationary point. This gives us the expression:

$$\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ \mathbf{c} = \mathbf{B}^*\mathbf{G}_0\mathbf{v}$$

(verify this!)

from which we get

$$\mathbf{c} = (\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ )^{-1}\mathbf{B}^*\mathbf{G}_0\mathbf{v}$$

This gives us coordinates c of v_1 relative to the subspace basis **B**

Determining v_0

- This gives us

$$\mathbf{v}_1 = \mathbf{B}\ \mathbf{c} = \mathbf{B}\ (\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ )^{-1}\mathbf{B}^*\mathbf{G}_0\mathbf{v}$$

and

$$\mathbf{v}_0 = \mathbf{v} - \mathbf{v}_1 = \mathbf{v} - \mathbf{B}\ (\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ )^{-1}\mathbf{B}^*\mathbf{G}_0\mathbf{v}$$

Determining $\mathbf{v}_0$

- If $\mathbf{v}_1$ is the solution we seek, it must be the case that $\mathbf{v}_0 \in U_\perp$
  $\Rightarrow$ it must be the case that $\mathbf{B}^*\mathbf{G}_0\mathbf{v}_0 = \mathbf{0}$

This is the scalar product between **v** and all basis vectors

- Check:
  $\mathbf{B}^*\mathbf{G}_0(\mathbf{v} - \mathbf{B}\ (\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ )^{-1}\mathbf{B}^*\mathbf{G}_0\mathbf{v}) =$
  $\mathbf{B}^*\mathbf{G}_0\mathbf{v} - \mathbf{B}^*\mathbf{G}_0\mathbf{B}\ (\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ )^{-1}\mathbf{B}^*\mathbf{G}_0\mathbf{v} =$
  $\mathbf{B}^*\mathbf{G}_0\mathbf{v} - \mathbf{B}^*\mathbf{G}_0\mathbf{v} = \mathbf{0}$ OK!

Summary so far

For a general $\mathbf{v} \in V$:

- We can uniquely decompose **v** as $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_0$ with $\mathbf{v}_1 \in U$ and $\mathbf{v}_0 \in U_\perp$
- With $\mathbf{v}_1 = \mathbf{B}\,\mathbf{c}$,

  c is given by $\mathbf{c} = (\mathbf{B}^*\mathbf{G}_0\mathbf{B}\ )^{-1}\mathbf{B}^*\mathbf{G}_0\mathbf{v}$
- $\mathbf{v}_1$ is the *orthogonal projection* of **v** onto *U*
- $\mathbf{B}^*\mathbf{G}_0\mathbf{v} = \mathbf{B}^*\mathbf{G}_0\mathbf{v}_1$ (why?)
- Given **v** and *U*, $\mathbf{v}_1$ does not depend on **B** (why?), but it does depend on $\mathbf{G}_0$ (why?)

An observation

- $\mathbf{B}^*\mathbf{G}_0\mathbf{v} = \mathbf{B}^*\mathbf{G}_0\mathbf{v}_1$ is the scalar product between $\mathbf{v}_1$ and all the subspace basis vectors
  $\Rightarrow \mathbf{B}^*\mathbf{G}_0\mathbf{v}$ are the dual coordinates $\tilde{\mathbf{c}}$ of $\mathbf{v}_1$
- $\mathbf{B}^*\mathbf{G}_0\mathbf{B}$ contains all the scalar products between the basis vectors = the metric $\mathbf{G}$
- This gives us $\mathbf{c} = \mathbf{G}^{-1}\tilde{\mathbf{c}}$
- This is consistent with the previous lecture:
  - We get dual coordinates by scalar products with the basis
  - We can transform dual to ”standard” coordinates by means of $\mathbf{G}^{-1}$

Convolution

- Given a signal *g* and a filter *f* (both discrete), and the convolution $h = g * f$
- We have already seen that we can interpret *h* as computing scalar products, one for each element of *h*, for example as

$$h[k] = \langle g[k+n] | f^*[-n] \rangle$$

Summation in the scalar product is here made over *n*

Convolution

- In practice, the filter *f* is often an FIR filter (what is that?). Assume it has *N* taps.
- The summation over *n* is made over *N* integer values
- Consequently, each time we evaluate the scalar product for *h*[*k*] we can take the summation over a finite interval from signal *g*
  - centred on *k*

  together with the reversal (and complex conjugate) of the filter *f*

Convolutions

- We can of course convolve g with several filters f_m, where m = 1, …, M

- This gives us M filter responses:

$$h_m[k] = \langle g[k+n] | f_m^*[-n] \rangle \ , \quad m = 1, \ldots, M$$

$$h_m[k] = \langle g[k+n] | b_m[n] \rangle \ , \quad b_m[n] = f_m^*[-n]$$

Toward normalised convolution

- Given this picture of convolving g with multiple filters, in the context of the previous subspace theory, it is possible to reason like this:
  - We consider the functions b_m as a basis **B** of some subspace, $m = 1, \dots, M$
  - The filter responses are the dual coordinates $\tilde{\mathbf{c}}$ of the **local signal**, the signal region that is covered by the filter when $h[k]$ is computed
    - More precisely: the dual coordinates of $\mathbf{v}_1$, the *projection* of the local signal onto the subspace spanned by **B**

Toward normalised convolution

- The subspace basis defines an $N \times M$ basis matrix **B**
- This gives the metric $\mathbf{G} = \mathbf{B}^* \mathbf{G}_0 \mathbf{B}$, where initially we set $\mathbf{G}_0 = \mathbf{I}$
- From $\tilde{\mathbf{c}}$ we get the coordinates of the signal region as $\mathbf{c} = \mathbf{G}^{-1} \tilde{\mathbf{c}}$
- Conclusion: we can determine the coordinates of the local signal relative to the basis given by the filters

Taking the next step

- Until now we have identified the ”filter functions” $b_m[n]$ as the basis functions
- Let us instead choose the basis functions rather freely, without thinking too much about if they are suitable as filters or not
- The filters are instead defined by multiplying the basis functions $b_m[n]$ by a suitable ”localising” function $a[n]$

The applicability function

- The filters are now defined as

$$f_m[n] = a[-n]\, b_m^*[-n], \quad m = 1, \ldots, M$$

The same a for all m

- a is a real-valued, positive, and (often) symmetric function called the *applicability function*
- a is chosen such that the resulting filters are localised, e.g., as a Gaussian function

Putting things together

- With this type of filters, we get

$$h_m[k] = \sum_n g[k+n] f_m[-n]$$

The summation is made over all filter coefficients

$$h_m[k] = \sum_n g[k+n] a[n] b_m^*[n]$$

Generalising the scalar product

- We choose *a* such that $a[n] > 0$ for all n
- This allows us to interpret $h[k]$ directly as the **scalar product** between the local signal $g[k+n]$ and the basis function $b_m[n]$
- This means that we change from the scalar product given as a simple product sum of elements to a *weighted product sum*
  - The weighting is done by *a*, the applicability function

Summary so far

- We choose a set of M basis functions b_m
- We choose an applicability function a
- This results in M filter functions

  $$f_m[n] = a[-n]\, b_m^*[-n]$$

  that are applied to the signal
- convolution with these filters gives us h_m
- We can interpret the filter responses $h_m[k]$ as the scalar products between the local signal and the basis functions b_m
- In this case: $\mathbf{G}_0 = \text{diag}(a)$
  - Assumes $a[k] > 0$ (why?)

Summary so far

- We now have: $\mathbf{G} = \mathbf{B}^* \mathbf{G}_0 \mathbf{B} = \mathbf{B}^* \, \text{diag}(a) \, \mathbf{B}$
- The coordinates of the projection of the local signal onto the subspace spanned by **B** are given by

$$\mathbf{c} = \mathbf{G}^{-1} \tilde{\mathbf{c}} = (\mathbf{B}^* \, \text{diag}(a) \, \mathbf{B})^{-1} \, [h_m]$$

For a fixed k
For $m = 1, \ldots, M$

Normalised Convolution

- This technique is called *Normalised Convolution*
  - Knutsson & Westin (SCIA 1993), Farnebäck (PhD 2002)
- It is a general technique for managing incomplete or uncertain data/signals:
  - Filtering of incomplete signals
  - Extracting local features such as gradients
  - Normalised averaging
- Alternatively: for making a local analysis of the signal in terms of some suitable basis
  - For example: polynomials

Local polynomial expansion

- One application of this result is *local polynomial expansion* of the signal
- At each point of the signal, we can approximate the local region around the point as a low order polynomial in the signal variables (typically of order two)
- Developed by Farnebäck in his PhD thesis

Local polynomial expansion

Motivation

- Let *g*(*x*) be a function (signal) of $x \in \mathbb{R}$.
- We can then (when?) make a *Taylor expansion* of *g* around *x*:

$$g(x+\tau) = g(x) + g'(x)\tau + \frac{1}{2}g''(x)\tau^2 + \ldots$$

- **Interpretation:** the local signal around *x* (a function of τ) is a linear combination of the basis functions $\{1, \tau, \frac{1}{2}\tau^2, \ldots\}$
- The coordinates of the local function in this basis are the derivatives of *g* of orders {0, 1, 2, ... } at *x*

Local polynomial expansion

- For a discrete signals, derivatives can now be computed in (at least) two ways:
  - **Standard:** Use a filter that is a ramp function in the Fourier domain
    - not suitable as a discrete filter function (why?)
    - we must weight the ramp in the *Fourier domain*
  - **NC:** As the coordinate that belongs to the basis function τ
    - We may have to use more than one basis function, e.g. $\{1, \tau, \frac{1}{2}\tau^2\}$ to describe the local signal
    - We weight the basis functions in the *signal domain*, using the applicability function

Local polynomial expansion

Applications for 2D signals

- We use (typically) basis functions: $\{1, x, y, x^2, y^2, xy\}$
- Applicability: (for example) a Gaussian function
- The corresponding 2D convolutions can be made separable $\Rightarrow$ efficient computations (why?)
- The corresponding coordinates c_1, c_x, c_y, c_{x^2}, c_{y^2}, c_{xy} give
  - Local mean (average) of the signal
  - First and second order derivatives of the signal
- The local signal g is expanded as

$$g(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + c_1$$

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} c_x^2 & c_{xy}/2 \\ c_{xy}/2 & c_y^2 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} c_x \\ c_y \end{pmatrix}$$

Local polynomial expansion

- The local polynomial expansion provides an efficient method for estimating local displacement (motion) between two images

$$g_1(\mathbf{x}) = \mathbf{x}^T\mathbf{A}_1\mathbf{x} + \mathbf{b}_1^T\mathbf{x} + c_1$$

$$g_2(\mathbf{x}) = \mathbf{x}^T\mathbf{A}_2\mathbf{x} + \mathbf{b}_2^T\mathbf{x} + c_2$$

- Assuming $g_2(\mathbf{x}) = g_1(\mathbf{x} + \mathbf{d})$, where $\mathbf{d}$ is the local displacement, we get (why?)

$$\mathbf{A}_1 = \mathbf{A}_2$$

$$\mathbf{b}_1 + 2\mathbf{A}\mathbf{d} = \mathbf{b}_2$$

$$\mathbf{d}^T\mathbf{A}\mathbf{d} - \mathbf{b}^T\mathbf{d} + c_1 = c_2$$

⇒

$$\mathbf{d} = \frac{1}{2}\mathbf{A}^{-1}(\mathbf{b}_2 - \mathbf{b}_1)$$

Displacement estimation

Summary

- We make a local polynomial expansion of both images at each point using Norm. Conv.
- Assuming that there is a position dependent displacement **d** between each local region of the two images
  - **d** can be estimated as $\mathbf{d} = \frac{1}{2}\mathbf{A}^{-1}(\mathbf{b}_2 - \mathbf{b}_1)$
  - Requires that **A** is of full rank (when is this true?)

Signal certainty

- So far we let the scalar product between the basis functions and the local signal be controlled by the applicability *a*
- *a* is a fixed function, typically a Gaussian, that controls the localisation of the coordinate computation process
- *a* is mainly related to properties of the basis functions
  - To make them "localized"
- We can go one step further and allow the scalar product to be controlled also by a signal dependent weight function: the *signal certainty*, denoted *c*
- It allows us to apply coordinate estimation also for signals with missing data

Signal certainty

Examples of incomplete or uncertain signals:

- Laser range data
- Local motion estimation
  - (the aperture problem)
- Inconsistent measurements of local features
- Outside the edges of an image
- Dead pixels in a camera
- Bayer patterns in 1-chip color cameras

Signal certainty

- Until now we have described the filter responses as

$$h_m[k] = \sum_n g[k+n]a[n]b_m^*[n]$$

- Instead we now use

c is the signal certainty function

$$h_m[k] = \sum_n g[k+n]c[k+n]a[n]b_m^*[n]$$

Modified scalar product

- We can still interpret $h_m[n]$ as the scalar product between the local signal and the basis functions:

$$h_m[k] = \langle g[k+n] | b_m[n] \rangle$$

where the scalar product now includes both the **applicability function**, a, and the **signal certainty**, c

Signal certainty

- c is the signal dependent certainty function
- $c[n]$ describes how much we can trust the signal value $g[n]$
- For example: we can assume: $c \in \{0, 1\}$
  1 if signal value is known, 0 if unknown
- Another common choice is $c = 1/\sigma^2$
  where σ^2 is the variance (i.e. uncertainty) this is called *inverse variance weighting*

Signal certainty

- We must assume that *c* is a known function
- In the case that $c[n] = 0$,
  we interpret this as: $g[n]$ is not known for this n
- Furthermore, we still interpret $h[k]$ as a scalar product between the local signal at point k and the basis functions b_m
  - The scalar product is constructed from both the fixed applicability function a and the position varying signal certainty function c
- $\mathbf{G}_0 = \mathrm{diag}(a \cdot c)$

G_0 becomes position dependent!

Implementation

- In the case $c \in \{0, 1\}$, normalised convolution can be implemented by first setting all unknown signal values $g[n] = 0$
- Convolve this g with the filter functions f_m
- The filter responses $h_m[k]$ become the dual coordinates of the projection of the local signal onto the subspace spanned by **B**
  - Same as before

Implementation

- We also need to determine the position dependent scalar product
  $$\mathbf{G}_0[k] = \mathrm{diag}(a[n] \cdot c[k+n])$$
  from which we get the position dependent metric
  $$\mathbf{G}_{ij}[k] = \langle \mathbf{b}_j | \mathbf{b}_i \rangle = \mathbf{b}_i^* \mathbf{G}_0[k] \mathbf{b}_j =$$
  $$= \sum_n b_j[n] c[k+n] a[n] b_i^*[n]$$
  and then transform the dual coordinates to "standard" coordinates by means of $\mathbf{G}^{-1}[k]$
- Produces useful results, but with more computations than if c is constant = 1

Normalised averaging

- A simple example of normalised convolution on uncertain data uses only the single basis function = 1 and some suitable applicability function a (e.g., a Gaussian) in combination with the signal certainty c
- In this case $\mathbf{G}_0[k] = \mathrm{diag}(a[n] \cdot c[k+n])$
- In this case $\mathbf{G}[k] = \mathrm{sum}_n(a[n] \cdot c[k+n])$ (why?)
  - We can write this as $\mathbf{G} = a_{rev} * c$ (why?)

    where $a_{rev}[n] = a[-n]$

a symmetric
$\Rightarrow a_{rev} = a$

Normalised averaging

- Furthermore, we now have

$$h[k] = \sum_n g[k+n]\, c[k+n]\, a[n]$$

$$= \sum_n g[k-n] c[k-n] a[-n]$$

which can be written as $h = (g \cdot c) * a_{\text{rev}}$

- In summary: the local coordinate of "1" is

$$\frac{(g \cdot c) * a_{\text{rev}}}{c * a_{\text{rev}}}$$

Both numerator and denominator are functions of position

Normalised averaging

- Normalised averaging can be implemented as two convolutions
  - $(g \cdot c) * a_{\mathrm{rev}}$
  - $c * a_{\mathrm{rev}}$
- The resulting functions are then divided point-wise
- The result is the coordinate of the ”complete” signal *g* (without missing data) projected onto the basis function ”1”
  – The way the projection is done depends on *c*, therefore it is position dependent

What you should know includes

- Definition of a subspace basis
- subspace coordinate computation in terms of basis and metric
- Application: normalised convolution, where metric $\mathbf{G}_0$ is local and defined from
  - Applicability
  - Signal certainty

  and where convolution is used to compute scalar products = dual coordinates of the local signal relative some chosen basis
- Application: local polynomial expansion (1D)
- Application: normalized averaging (2D)