

3D Computer Vision - WT 2004

Singular Value Decomposition

Darko Zikic

CAMP - Chair for Computer Aided Medical Procedures

November 4, 2004

- 1 Singular Value Decomposition (SVD)
- 2 Optimization
- 3 Examples
- 4 SVD Computation
- 5 Summary and References

Singular Value Decomposition (SVD)

- For any given matrix $A \in \mathbf{R}^{m \times n}$ there exists a decomposition

$$A = UDV^T$$

such that

- U is an $m \times n$ matrix with orthogonal columns
- D is a $n \times n$ diagonal matrix with non-negative entries
- V^T is an $n \times n$ orthogonal matrix

SVD - Visualized

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \begin{bmatrix} D \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix}$$

- A is an $m \times n$ matrix
- U is an $m \times m$ matrix with orthogonal columns
- D is a $n \times n$ diagonal matrix with non-negative entries
- V^T is an $n \times n$ orthogonal matrix

Singular Value Decomposition (SVD)

- The diagonal values of D are called **Singular Values** of A
- The column vectors of U are the **Left Singular Vectors** of A
- The column vectors of V are the **Right Singular Vectors** of A .

Properties

- The SVD can be performed s.t. the diagonal values of D are descending i.e. $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$.
- We will assume that the SVD is always performed in that way
- The diagonal values of D are the square roots of the Eigenvalues of $A^T A$ and AA^T
(Hence the non-negativity of the elements of D)

Some More Properties

- It holds for the left singular vectors u_i :

$$A^T A u_i = d_i^2 u_i$$

- It holds for the right singular vectors v_i :

$$A A^T v_i = d_i^2 v_i$$

- The left singular vectors u_i are eigenvectors of $A^T A$
- The right singular vectors v_i are eigenvectors of $A A^T$

Even More Properties

- SVD explicitly constructs orthonormal bases for the null-space and the range of a matrix
- columns of U corresponding to non-zero elements of D span the range
- columns of V corresponding to zero elements of D span the null-space

Properties Galore

- SVD allows a rank decision:

$\text{rank}(A)$ is the largest r s.t. $d_r > 0$

- there are $m - r$ left singular vectors corresponding to the singular value 0.
- there are $n - r$ right singular vectors corresponding to the singular value 0.

Linear Optimization

SVD can be used for linear optimization by using the following property

- Let v_n be the right singular vector corresponding to d_n (the smallest element of D)
- The product Ax with $\|x\|_2 = 1$ has the minimal value for $x = v_n$.

Minimization by SVD

The minimizing property of the last right singular vector v_n can be used to solve the following minimization task

- Given the linear function $f = Ax, f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ to be minimized (in most applications $m \gg n$)
- With the constraint that the solution x is not trivial ($x \neq 0$)
(We will assume that $\|x\|_2 = 1$)

Minimization by SVD II

- The minimization problem is thus

$$\text{minimize } \|Ax\|_2 \quad \text{s.t. } \|x\|_2 = 1$$

- It can be shown that the solution is the right singular vector $x = v_n$ corresponding to the smallest singular value d_n .

Proof

- Problem: minimize $\|Ax\|_2$ subject to $\|x\|_2 = 1$
- Because of the orthogonality of U and V we have
 - $\|Ax\|_2 = \|UDV^T x\|_2 = \|DV^T x\|_2$
 - $\|x\|_2 = \|V^T x\|_2$
- Hence have to minimize $\|DV^T x\|_2$ subject to $\|V^T x\|_2 = 1$
- With $V^T x = y$ we have: minimize $\|Dy\|_2$ subject to $\|y\|_2 = 1$
- Since D diagonal with descending entries we get
 $y = (0, 0, \dots, 0, 1)^T$
- Since $V^T x = y \Leftrightarrow x = Vy$ we get $x = v_n$

What's next...?

So in order to solve a linear minimization problem by SVD we have to do two things

- 1 State it in the form

$$\text{minimize } Ax \quad \text{s.t.} \quad \|x\|_2 = 1$$

- 2 Compute the SVD $A = UDV^T$ and take the last right singular vector v_n as the solution

Fitting Lines

- Task: Given a set of n noisy points $\{p_i\}$, find the line l that goes through them
- We know: In homogeneous coordinates we have $p_i^T l = 0$ if the point lies on the line
- For all points we get:

$$\begin{bmatrix} p^1 \\ p^2 \\ \vdots \\ p^3 \end{bmatrix} l = 0$$

Fitting Lines II

- Since the points are noisy we can't satisfy the equation
- The best we can do is to get the minimal solution
- Since we are not interested in the trivial solution $l = 0$ we set $\|l\|_2 = 1$

Fitting Lines III

- So the problem is

$$\text{minimize } Pl \quad \text{s.t. } \|l\|_2 = 1$$

with

$$P = \begin{bmatrix} p^1 \\ p^2 \\ \vdots \\ p^3 \end{bmatrix}$$

- We solve it by applying the SVD-method

DLT Algorithm

- Computing the homography H between two images (next lecture)
- Solution by applying the SVD
- Only trick: State the problem the right way i.e.:
 - Represent the homography as a vector $h \neq 0$ ($\|h\|_2 = 1$)
 - Find a linear function A s.t. minimizing Ah does exactly what you want

Linear Triangulation

- Reconstructing the real world structure from two or more images (coming soon in the lecture)
- Boiled down that means finding the world point X by back projecting two image points x and x'

Notes

- Minimization by SVD is extremely simple. The tricky part is to state the problem the right way.
- The error minimized by the SVD is called **Algebraic Error** or **Algebraic Distance**
- A drawback of the algebraic error is that it is geometrically meaningless, so that minimizing it can lead to completely meaningless results

SVD Black Box

- Only thing left is to show how an SVD of a matrix A can be computed
- But we won't do that.
- We'll use SVD as a black box
- Computation involves QR procedure and Householder reduction
- Original algorithm by Golub and Reinsch

Computation Properties

- Algorithm is extremely stable
- Computation time for SVD of an $m \times n$ A :
 - Computation of U , V and D : $4m^2n + 8mn^2 + 9n^3$
 - Computation of V and D : $4mn^2 + 8n^3$
- Keep in mind that in most cases $m \gg n$

Implementation in Matlab

- Matlab has the command $[U,S,V] = \text{SVD}(X)$
- Attention: The command returns V and not V^T
- Hence it holds that $X = U*S*V'$

Summary

- Properties of the SVD
- Linear minimization using SVD

References

- Hartley and Zisserman. *Computer Vision*
- Walter Gander. *Fitting Data by Least Squares - Algorithms and Examples*
- Gander and Hrebicek. *Solving Problems in Scientific Computing using Maple and Matlab*
- Vachenauer. *Höhere Mathematik 1*
- Press et. al. *Numerical Recipes in C*
- Golub and van Loan. *Matrix Computations*