Computer Vision Group
Institut für Informatik
Technische Universität München

Exercise Sheet 4

Room: 02.09.023

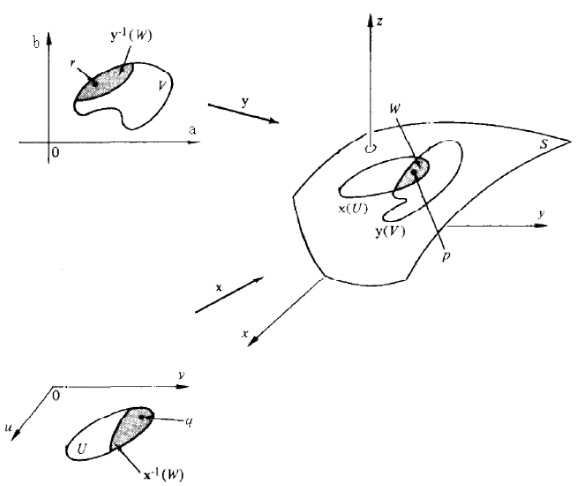
Tue, 27.05.2014, 13:45-15:15

Submission deadline: Mon, 26.05.2014, 23:59 to windheus@in.tum.de

Mathematics: The gradient

Exercise 1 (First fundamentalform).

1. *Show that the first fundamental form is positive definit if and only if the parametrization is regular.*
2. *Let* $\mathbf{x} : U \to S$ *and* $\mathbf{y} : V \to S$ *be two parametrizations of a surface* S *and let* $h = \mathbf{x}^{-1} \circ \mathbf{y} : V \to U$ *be the diffeomorphic change of coordinates. Find an expression for the first fundamental form* $g_y = d\mathbf{y}^T d\mathbf{y}$ *in terms of* h *and* $\mathbf{x}$*. (For simplicity assume that* $\mathbf{x}(U) = \mathbf{y}(V)$*)*



Exercise 2 (The gradient). *Let* S *be a regular surface and* $f : S \to \mathbb{R}$ *a differentiable function.*
For a given point $q = \mathbf{x}(p) \in S$ *find the vector* $v \in T_pS$ *such that...*

1. *... an infenitesimal step in the direction of* v *increases* f *the most.*
2. *... an infenitesimal step in the direction of* v *does not change the value of* f*.*

Programming: Multi-Dimensional Scaling

Exercise 3 (Two points)**.** *Download and expand the file* `exercise4.zip` *from the lecture website. Modifiy the files* `mds.m` *and* `alignpoints.m` *to implement the functions as explained below. You can run the script* `exercise.m` *to test and visualize your solutions.*

`mds.m` *Implement the* multi-dimensional scaling *method introduced in the lecture. The function accepts a metric given by matrix* $D \in \mathbb{R}^{n \times n}$ *and a dimension* $m \in \mathbb{N}$. *It should return the coordinates of each of the* n *points embedded into* $\mathbb{R}^m$ *as matrix* $Z \in \mathbb{R}^{n \times m}$.

Parameters `alpha`, `epsilon` *and* `maxI` *control the gradient descent's behaviour and are the stepsize, minimum relative progress and maximum number of iterations, respectively.*

`alignpoints.m` *Write a function that aligns two point clouds. Given two set of points* $Z_1, Z_2 \subset \mathbb{R}^m$ *embedded into* $\mathbb{R}^m$ *by the MDS method find two rigid transformations that are represented by* $(R_1, \mathbf{t}_1), (R_2, \mathbf{t}_2) \in \mathbb{R}^{m \times m} \times \mathbb{R}^m$. *The transformed point sets* $\hat{Z}_1 = \{R_1(\mathbf{z} + \mathbf{t}_1) | \mathbf{z} \in Z_1\}, \hat{Z}_2 = \{R_2(\mathbf{z} + \mathbf{t}_2) | \mathbf{z} \in Z_2\}$ *should be aligned to each other.*

Translations $\mathbf{t}_1, \mathbf{t}_2$ *can be found by computing the point clouds' mean. There are several ways to find good rotation matrices. We suggest to align the principal axis* $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m \in \mathbb{R}^m$ *of each point cloud with the standard Euclidean axis* $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m \in \mathbb{R}^m$. *Note that* $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ *form an orthonormal basis of* $\mathbb{R}^m$ *and can be computed by the matlab function* `pca`.

The matlab function `alignpoints` *should accept two point clouds by arguments* `Z1` *and* `Z2` *and should return the transformed poitns as* `Zhat1` *and* `Zhat2`. *The rotations and translations are not needed as return values.*

Note that since the signs of the principal axis are not uniquely determined, the visualization code in `exercise.m` *includes sign parameters that can be altered.*