Computer Vision Group
Institut für Informatik
Technische Universität München

Exercise Sheet 3

Room: 02.09.023

Tue, 13.05.2014, 13:45-15:15

Submission deadline: Mon, 12.05.2014, 23:59 to windheus@in.tum.de

Mathematics: Differential Geometry

Exercise 1 (One point). *Let $g : \mathbb{R}^2 \to \mathbb{R}$ be a smooth function. We consider its graph*

$$f : \mathbb{R}^2 \to \mathbb{R}^3,\ (u,v) \mapsto (u,v,g(u,v))$$

1. *Show that f is regular and therefore a parametrized surface element.*
2. *Derive the first fundamental form of f.*
3. *In what cases is the parametrization f orthogonal (conformal, isometric)?*

Exercise 2 (One point). *Let $g : I \to \mathbb{R}^+$ be a smooth function. Consider*

$$f : I \times \mathbb{R} \to \mathbb{R}^3,\ (u,v) \mapsto (u, \cos(v)g(u), \sin(v)g(u)).$$

1. *Give an interpretation of f.*
2. *Show that f is regular and derive its first fundamental form.*
3. *Calculate the area $A(f|_{I\times(0,2\pi)})$.*
4. *Calculate the surface area of a sphere with radius r.*

Exercise 3 (One point). *Find a regular parametrization $f : \mathbb{R}^2 \to \mathbb{R}^3$ of the torus*

$$T = \left\{ (x,y,z) \in \mathbb{R}^3 | \left(\sqrt{x^2+y^2} - a \right)^2 + z^2 = r^2 \right\}$$

with $a > r > 0$.

Programming: Quadratic Assignment Problem

Exercise 4 (One point)**.** *Download and expand the file* `exercise3.zip` *from the lecture website. Modifiy the files* `distortioncostmatrix.m`*,* `spectralrelax.m`*, and* `gametheoretic.m` *to implement the functions as explained below. You can run the script* `exercise.m` *to test and visualize your solutions.*

Given two finite sets X, Y, where $|X| = |Y| = n$, equipped with metrics d_X, d_Y, we can represent a correspondence between X, Y as a binary matrix $R \in \{0,1\}^{n \times n}$, such that $R\mathbf{1} = \mathbf{1}$ and $R^\top \mathbf{1} = \mathbf{1}$. The lecture introduced a relaxation to the Gromov-Hausdorff distortion measure by

$$distortion^p(R) = \sum_{i,j,k,l} C^p_{(il)(jk)} R_{ij} R_{kl}, \text{ where } C^p \in \mathbb{R}^{n^2 \times n^2} \text{ and} \tag{1}$$

$$C^p_{(il)(jk)} = |d_X(x_i, x_j) - d_Y(y_l, y_k)|^p. \tag{2}$$

This leads to the Quadratic Assignment Problem*:*

$$\begin{aligned} \min_{R \in \{0,1\}^{n \times n}} \; & vec(R)^\top \; C \; vec(R) \\ s.t. \; & R \; \mathbf{1} = \mathbf{1} \\ & R^\top \; \mathbf{1} = \mathbf{1}. \end{aligned} \tag{3}$$

1. *Implement function* `distortioncostmatrix` *that returns matrix $C^p \in \mathbb{R}^{n^2 \times n^2}$ given metrics d_X, d_Y, and $p \in \mathbb{R}$.*

   Be careful about the indexing. Since R is reshaped into $vec(R)$ by stacking the columns of R it holds $vec(R)_{i+nj} = R_{i,j}$. Analogously $C^p_{(il)(jk)} \widehat{=} C^p_{i+nl,j+nk}$.

2. *Implement the* power iteration *scheme introduced in the lecture to compute the optimum of the* spectral relaxation*:*

$$\begin{aligned} \max_{R \in [0,1]^{n \times n}} \; & vec(R)^\top \; \hat{C} \; vec(R) \\ s.t. \; & \|R\|_2 = 1, \end{aligned} \tag{4}$$

   Where $\hat{C} = \max(C) - C$. (Note that the conversion to a maximization problem makes the optimization problem solveable to the power iteration scheme.) Place the implementation into function `spectralrelax`*. Function* `spectralrelax` *has four arguments,* `C`*,* $\mathtt{x}_0$*,* `maxI`*, and* `epsilon`*, and returns* `R`*. The function solves optimization problem* (4) *specified by* `C` *and returns the* arg min *as* `R`*.* $\mathtt{x}_0$ *specifies the starting point of the iterations. The iterations should be stopped if $i >$* `maxI` *or $|R^i - R^{i-1}| <$* `epsilon`*.*

3. *Implement the iterative scheme introduced in the lecture to compute an (locally optimal) solution of the* game-theoretic relaxation*:*

$$\begin{aligned} \max_{R \in [0,1]^{n \times n}} \; & vec(R)^\top \; \hat{C} \; vec(R) \\ s.t. \; & \|R\|_1 = 1. \end{aligned} \tag{5}$$

Place the implementation into function `gametheoretic`*. Function* `gametheoretic` *has four arguments,* `C`, `x`$_0$, `maxI`, *and* `epsilon`*, and returns* `R`*. The function solves optimization problem* (5) *specified by* `C` *and returns the* $\arg\max$ *as* `R`. `x`$_0$ *specifies the starting point of the iterations. The iterations should be stopped if* $i >$ `maxI` *or* $|R^i - R^{i-1}| <$ `epsilon`.