


グラフィックス対話処理技術の
数学的基礎

オー・エル・エム・デジタル
安生健一

R15 9/26(水) 16:40~18:00 

背景

～グラフィックス対話処理の重要性～

- 重要性は増大の一途
- オフラインアニメーションの制作にも必須の機能
 - 制作効率への寄与
 - 最終アウトプットの質の向上
 - 映像そのものの意味、制作体制を変革する鍵

2

グラフィックス対話処理のための技術

- インタラクティブアニメーションの生成
 - 直感的かつシンプルな入力手段
Ex. 数値入力よりマウス操作
 - 高速処理
Ex. 高速プレビュー、ハードウェアレンダリング
- 優れたGUI
 - クールデザイン
 - かゆいところに手が届きめ細かな機能
- 新しい表現を作り出す
 - 高効率化がもたらす、新手法取り込みの加速

3

本セッションの概要

- インタラクティブ処理が可能ないくつかのモデリング手法を紹介
 - 事例: 最近の我々の研究紹介
 - 学習論的な手法
 - 補間手法
- これらの技術を支える数学的基礎の概説
 - 将来の新技术開発への布石として

4

数学のおさらい

- 最小自乗法^(*)
- Lagrangeの未定乗数法

(*) F.Pighin, J.P.Lewis: Practical Least-Squares for Computer Graphics (SIGGRAPH07 Course Notes No. 11), also available at <http://graphics.stanford.edu/~jplewis/lscourse>

最小自乗法(least-square methods)

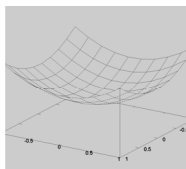
- 多次元線形回帰問題 (multi-dimensional linear regression)
 - 数ベクトル b が何らかの意味で d 個の変数 a_1, a_2, \dots, a_d の関数として表せるとしてその関数を1次式で近似出来るかを考える。
 - N 個の b の測定値 $\{b_1, b_2, \dots, b_N\}$ が与えられて、これらの各値をとるときの変数 a_1, a_2, \dots, a_d の値も分かっているとす。つまり $\{b_i, \{a_{i,j}\}\}$ が、測定値であるとする。
 - $\hat{b} = x_1 a_1 + \dots + x_d a_d$ とおき、 b と \hat{b} との誤差 e の値を最小にする $x = (x_1, \dots, x_d)$ を求める。

$$e = \sum_{i=1}^N (b_i - \hat{b}_i)^2 = \sum_{i=1}^N (b_i - \sum_{j=1}^d x_j a_{i,j})^2$$

$$a_i = \begin{pmatrix} a_{i,1} \\ \vdots \\ a_{i,d} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix} \text{ と表せば } e = \left\| b - \sum_{j=1}^d x_j a_j \right\|^2 = \|b - Ax\|^2 \text{ である。}$$

誤差関数 e を最小にする x を求める

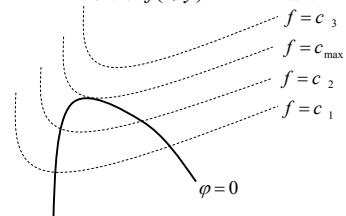
- e を x の関数とみて x について微分する
 $\frac{\partial e}{\partial x} = \nabla e = 2A^T Ax - 2A^T b$
- もう一度微分すると
 $H_e(x) = 2A^T A$
 を得る。これは半正値行列
 ($v^T H_e(x) v \geq 0$, for any vector v .)
- $A^T A$ が正則なら、 x は $\frac{\partial e}{\partial x} = 0$ より一意に決まる。
- $A^T A$ が非正則ならば、制約を付加しつつ、解を絞る。



2次元の場合の半正値な様子

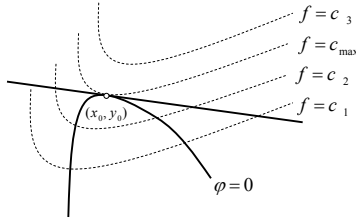
制約問題を解くための基本的な考え方 - Lagrangeの未定乗数法 -

- 条件付き極値問題
 例: $(x, y) \in \mathbb{R}^2$ が $\varphi(x, y) = 0$ を満たすとき、関数 $f(x, y)$ の極値を求めよ。



Lagrangeの未定乗数法 条件付き極値問題の“直感的”解法

- $(x_0, y_0) \in \mathbf{R}^2$ が極値を取る点なら、この点を共有する $f(x, y) = c_{max}$ と $\varphi(x, y) = 0$ は、この点 (x_0, y_0) で接線の傾きが等しい。

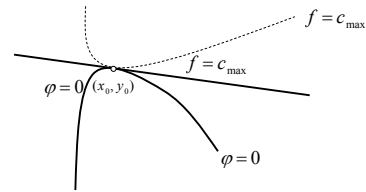


9

Lagrangeの未定乗数法 条件付き極値問題の“直感的”解法

- $(x_0, y_0) \in \mathbf{R}^2$ が極値を取る点なら、この点での法線ベクトルで考えれば:

$$\exists \lambda \in \mathbf{R} [(f_x(x_0, y_0), f_y(x_0, y_0)) = \lambda(\varphi_x(x_0, y_0), \varphi_y(x_0, y_0))]$$



10

Lagrangeの未定乗数法

- 定理

$(x, y) \in \mathbf{R}^2$ が $\varphi(x, y) = 0$ を満たしつつ変化するとき、関数 $f(x, y)$ の極値を取る点 (x_0, y_0) で $grad[\varphi] \neq 0$ ならば、以下の条件を満たす実数 λ が存在する:

$$\begin{cases} f_x(x_0, y_0) - \lambda \varphi_x(x_0, y_0) = 0 \\ f_y(x_0, y_0) - \lambda \varphi_y(x_0, y_0) = 0 \\ \varphi(x_0, y_0) = 0 \end{cases}$$

11

Lagrangeの未定乗数法

- 一般化(その1): 変数を増やす

$x = (x_1, x_2, \dots, x_n) \in \mathbf{R}^n$ が $\varphi(x) = 0$ を満たしつつ変化するとき、関数 $f(x)$ が極値を取る点 $x = x_0$ において $grad[\varphi] \neq 0$ ならば、以下の条件を満たす実数 λ が存在する:

$$\begin{cases} f_{x_1}(x_0) - \lambda \varphi_{x_1}(x_0) = 0 \\ \vdots \\ f_{x_n}(x_0) - \lambda \varphi_{x_n}(x_0) = 0 \\ \varphi(x_0) = 0 \end{cases}$$

12

Lagrangeの未定乗数法

- 一般化(その2): 制約条件を増やす

$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ が k 個の制約条件
 $\varphi_1(x) = \varphi_2(x) = \dots = \varphi_k(x) = 0$

を満たしつつ変化するとき、関数 $f(x)$ が極値を取る点 $x = x_0$ にて

$$\text{rank} \left(\frac{\partial \varphi_j}{\partial x_i} \right)_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}} = k$$

であるとする。このとき $F(x) = f(x) - \sum_{j=1}^k \lambda_j \varphi_j(x)$ とおけば、

以下の $(n+k)$ 個の条件を満たす k 個の実数 $\lambda_1, \lambda_2, \dots, \lambda_k$ が存在する:

$$\begin{cases} \frac{\partial F}{\partial x_1}(x_0) = 0, \frac{\partial F}{\partial x_2}(x_0) = 0, \dots, \frac{\partial F}{\partial x_n}(x_0) = 0, \\ \varphi_1(x_0) = 0, \varphi_2(x_0) = 0, \dots, \varphi_k(x_0) = 0 \end{cases}$$

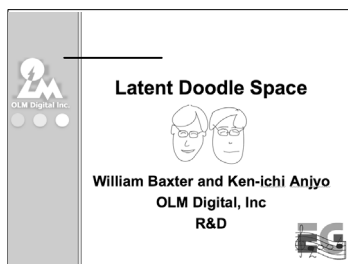
13

参考書

- 金谷健一 著:
これなら分かる応用数学教室—最小二乗法からウェーブレットまで (共立出版 2003)
- 金谷健一 著:
これなら分かる最適化数学—基礎原理から計算手法まで (共立出版 2005)

14

事例1: 「落書き(Doodle)」を学習する



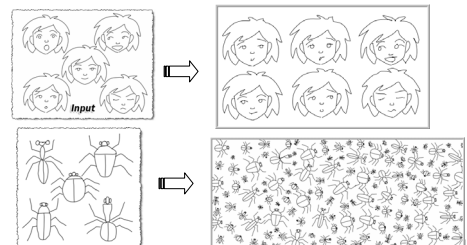
In: *Computer Graphics Forum* Vol. 25 No.3
 (Proc. Eurographics 2006)

15

Doodle Sketch



- 目的: 小数の類似した手描き線画を入力し、それらと似た大量のコピー線画を作り出す:



16

Doodle Handwriting

- 手書き文字の入力データの「癖」を学ぶ

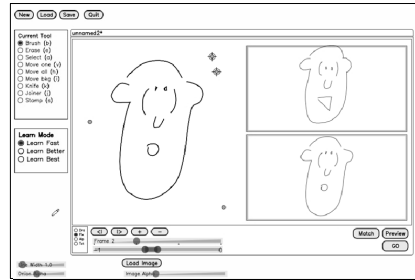
DEAR MOM
HOW ARE YOU DOING
EUROPE IS GREAT THE PEOPLE ARE
SUPER NICE AND THE COFFEE IS
DELICIOUS
THE WEATHER IS GREAT TOO
WISH YOU WERE HERE
WITH LOVE JENNIFER

DEAR MOM
HOW ARE YOU DOING
EUROPE IS GREAT THE PEOPLE ARE
SUPER NICE AND THE COFFEE IS
DELICIOUS
THE WEATHER IS GREAT TOO
WISH YOU WERE HERE
WITH LOVE YOUR SON

この手法で自動生成された筆跡の例

17

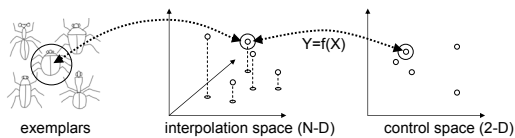
Demo:



18

Our Solution

- 高次元空間での補間 + 低次元空間での制御(パラメータコントロール)
 - 手書き情報を高次元空間の「点」と考える
 - 低次元空間から高次元空間への写像 $Y=f(X)$ を入力データから「学習」して構築する.

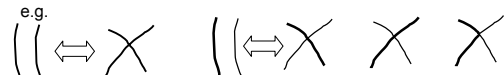


- この高次元補間空間を“Latent Doodle Space”と呼ぶ

19

Latent Doodle Space をどうやって作る？:

- 線画同士の対応付けをする
 - 一つの「線画」はストロークの集まりと定義し、各ストロークは(向きづけられた)点列からなるとする.
 - 完璧な対応付け法は存在しない



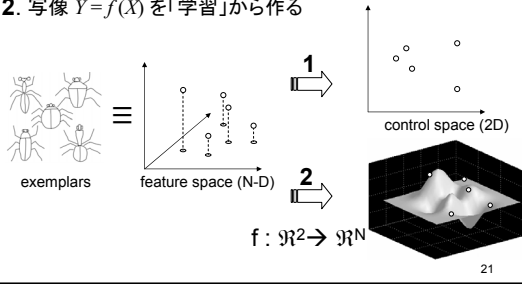
⇒【方針】best possible な解法+それを補う UI

- 類似した線画の生成
 - 「学習」方法の適用
 - ⇒【方針】最新の手法を適用して評価する

20

Latent Space の構築ステップ

- 次の二つがメインタスク:
 1. 潜在する(制御)座標系を探し出す
 2. 写像 $Y=f(X)$ を「学習」から作る

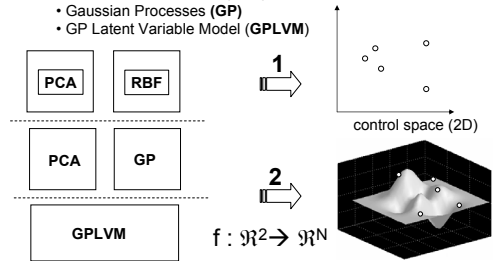


21

Latent Space の構築手法

- いくつかの手法が考えられる

- 主な道具立て:
- Principle Component Analysis (PCA)
 - Radial Basis Functions (RBF)
 - Gaussian Processes (GP)
 - GP Latent Variable Model (GPLVM)



22

主成分分析法

PCA :Principal Component Analysis

参考文献: 加納学「主成分分析」
www-pse.chem.kyoto-u.ac.jp/~kano/document/text-PCA.pdf

24

PCA

PCAとは？

- 多次元データに対して、情報を極力損なわぬように低次元化(“可視化”)する方法
- d 次元のデータのN個のサンプルについて考えると

	d 個の「変数」の並び			
第1番目のサンプル	x_{11}	x_{12}	\cdots	x_{1d}
第2番目のサンプル	x_{21}	x_{22}	\cdots	x_{2d}
\vdots	\vdots	\vdots	\ddots	\vdots
第N番目のサンプル	x_{N1}	x_{N2}	\cdots	x_{Nd}

- 多次元変数を低次元変数へ変換する
- $\phi: (x_1, x_2, \dots, x_d) \rightarrow (z_1, z_2, \dots, z_M) \quad (M \leq d)$

25

PCA

PCAの基本方針

- 【方針】 1) ϕ は1次式 (→「変数」はベクトル空間で考える)
- 2) $(z_m)_{m=1, \dots, M}$ は互いに「独立」
- d次元のデータのN個のサンプルは行列で表す:

$$\begin{matrix} \text{第1番目のサンプル} \rightarrow & \begin{pmatrix} x_{11}^* & x_{12}^* & \cdots & x_{1d}^* \\ \text{第2番目のサンプル} \rightarrow & x_{21}^* & x_{22}^* & \cdots & x_{2d}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \text{第N番目のサンプル} \rightarrow & x_{N1}^* & x_{N2}^* & \cdots & x_{Nd}^* \end{pmatrix} & = & X \end{matrix}$$
- $\phi: (x_1, x_2, \dots, x_d) \rightarrow (z_1, z_2, \dots, z_M)$ $\{w_{im}\}$: 結合係数

$$z_m = \sum_{i=1}^d w_{im} x_i, \quad \sum_{i=1}^d w_{im}^2 = 1 \quad (m=1, 2, \dots, M)$$

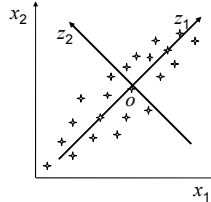
26

PCA

PCAの基本方針

- 【方針】 1) ϕ は1次式
- 2) $(z_m)_{m=1, \dots, M}$ は互いに「独立」
- 3) 次元を下げても得られる情報を最大限に
- 「データのばらつきを最もよく表す方向」を情報の主成分 z_1 と考える
- ばらつきを調べるために、各変数 x_i についての測定値の平均値との差を考える:

$$x_{ni} = x_{ni}^* - \bar{x}_i$$
- ここで x_{ni}^* はサンプル測定値、 \bar{x}_i は変数 x_i についてのサンプル平均値
- データ行列 X は $X = (x_{ni})$ と考える



27

PCA

固有値解析による低次元化

- 第1主成分 z_1 の結合係数を求める:

z_1 の結合係数 $w_1 = \begin{pmatrix} w_{11} \\ w_{21} \\ \vdots \\ w_{d1} \end{pmatrix}$, i番目のサンプル $x_i = (x_{i1} \ x_{i2} \ \cdots \ x_{id})$

$t_1 = \begin{pmatrix} t_{11} \\ t_{21} \\ \vdots \\ t_{N1} \end{pmatrix}$ とおけば $\bar{t}_1 = 0$. かつ z_1 の分散 $\sigma_{z_1}^2 = \frac{1}{N} t_1^T t_1$ と表せる.

ここで共分散行列 V を $V = \frac{1}{N} X^T X$ と定義すれば、 $\sigma_{z_1}^2 = w_1^T V w_1$

28

PCA

固有値解析による低次元化

- 第1主成分 z_1 の結合係数を求める (続):

z_1 の分散 $\sigma_{z_1}^2 = w_1^T V w_1$ が最大となる結合係数 w_1 を求めるため、Lagrange乗数 λ を導入する.

次の J_1 の値を最大にする w_1 と λ を求めることになる:

$$J_1 = w_1^T V w_1 - \lambda (w_1^T w_1 - 1)$$

$0 = \frac{\partial J_1}{\partial w_1} = \begin{pmatrix} \frac{\partial J_1}{\partial w_{11}} \\ \frac{\partial J_1}{\partial w_{21}} \\ \vdots \\ \frac{\partial J_1}{\partial w_{d1}} \end{pmatrix}$ を計算すると、 $(V - \lambda I) w_1 = 0$ を得る.

すなわち、共分散行列 V についての固有値が λ であり、 w_1 はその固有ベクトル. このとき $\sigma_{z_1}^2 = w_1^T V w_1 = w_1^T \lambda w_1 = \lambda w_1^T w_1 = \lambda$ であるから、最大固有値が z_1 の分散の最大値を与えることがわかる.

29

PCA

固有値解析による低次元化

- 第 m 主成分 z_m の結合係数を求める:
 z_m の分散 $\sigma_{z_m}^2$ は、再び共分散行列の固有値となることを、帰納的に示すことができる。その際 $\{z_i\}_{i=1}^m$ が他と独立であるという条件も満たす:
 $w_i^T w_j = \delta_{ij}$
- X の特異値分解と、 $\{\lambda_k\}$ との関係:

$$X = U \Sigma P^T$$

$$\Sigma = \begin{pmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \sqrt{\lambda_r} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{pmatrix} = \begin{pmatrix} O_{r,d-r} \\ O_{N-r,r} \\ O_{N-r,d-r} \end{pmatrix}$$
 ここで U と P は直交行列で

30

事例1: 「落書き(Doodle)」を学習する

Latent Doodle Space をどうやって作る?:

Step 1: 入力された例の各ストロークを一対一に対応付けする。
 Step 2: 対応づけられたストロークのサンプル点を同じ数にする。
 Step 3: 点対点の対応としてうまく対応づけるために、必要ならストローク点列の並ぶ順序を逆にする。
 Step 4: 以上によりデータ行列 X を構成し、「学習」方法を適用して Latent Space を構築する:

$$\begin{matrix} \text{2次元座標値 } (x, y) & \text{ストローク1} & \text{ストローク2} & \dots & \text{ストローク}m \\ \text{線画例1} \rightarrow & a^1_0 & a^1_1 & \dots & a^1_{n_1} & b^1_0 & \dots & b^1_{n_2} & \dots & \\ \text{線画例2} \rightarrow & a^2_0 & a^2_1 & \dots & a^2_{n_1} & b^2_0 & \dots & b^2_{n_2} & \dots & \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \end{matrix} = X$$

各ストロークのサンプル点数 n_i は Step 2 で自動的に決められる。

Step 5: 必要ならGUIにてストロークの対応付けを修正し、データ行列を再構築することで、より良い Latent space を作る。

31

事例1: 「落書き(Doodle)」を学習する

Latent Doodle Space : PCA+RBFアプローチ

Step 1: 入力された例の各ストロークを一対一に対応付けする。
 Step 2: 対応づけられたストロークのサンプル点を同じ数にする。
 Step 3: 点対点の対応としてうまく対応づけるために、必要ならストローク点列の並ぶ順序を逆にする。
 Step 4: 以上によりデータ行列 X を構成し、以下の処理を施す:

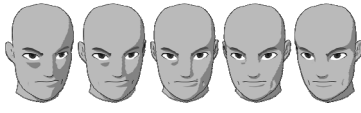
- PCAを適用して、多次元座標空間を2変数パラメータで記述する:

$$\begin{matrix} & \text{ストローク1} & \text{ストローク2} & \dots & \text{ストローク}m \\ \text{線画例1} \rightarrow & a^1_0 & a^1_1 & \dots & a^1_{n_1} & b^1_0 & \dots & b^1_{n_2} & \dots & \\ \text{線画例2} \rightarrow & a^2_0 & a^2_1 & \dots & a^2_{n_1} & b^2_0 & \dots & b^2_{n_2} & \dots & \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots & \end{matrix} = X$$
- 各座標値は2変数関数となるので、2次元版RBFを用いて補間する

Step 5: 必要ならGUIにてストロークの対応付けを修正し、データ行列を再構築することで、より良い Latent space を作る。

32

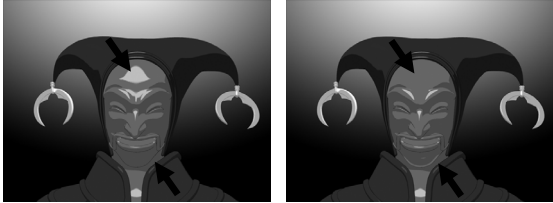
事例2: ペイントブラシによる演出可能なトゥーンシェーダー



藤堂英樹、安生健一、William Baxter、五十嵐健夫
 "Locally Controllable Stylized Shading"
 In: ACM Transactions on Graphics Vol. 26 No.3 (Proc. SIGGRAPH 2007)

51

トゥーンシェーダーの課題:
「こうしたい」ライティングができない



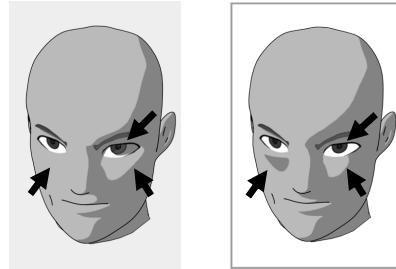
Images: Courtesy of YOUN IN-WAN, YANG KYUNG-IL/Shin Angyo Project 2004

通常の toon shader

望ましい結果(例)

52

トゥーンシェーダーの課題:
もっと自由に「演出」したい



通常の toon shader

望ましい結果(例)

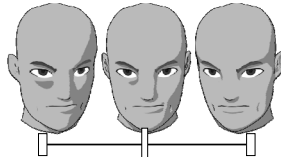
53

提案手法と効果

- ペイントブラシで直接陰影を付加する
- キーフレームアニメーションとしてダイナミックに表現出来る
- ペイント結果は 3Dライティングとシームレスに繋げる
- GPU応用でリアルタイムプレビューが可能



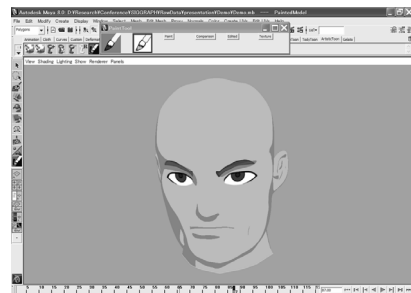
Paint brush metaphor



Key-framing

54

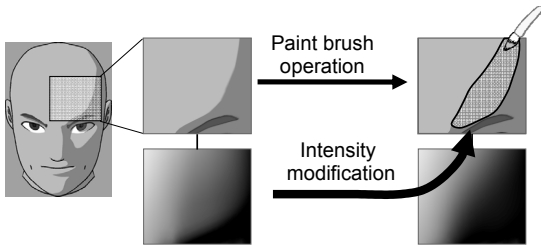
Demo (Maya plug-in)



55

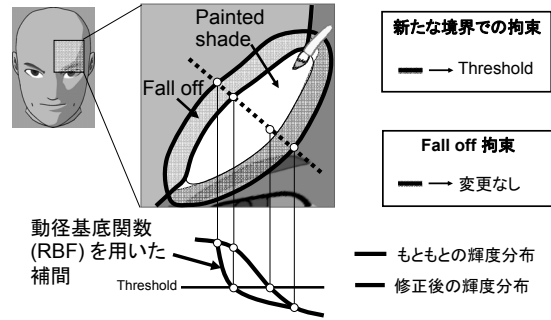
キーアイデア

- ペイントストロークにもとづいて輝度分布を修正する



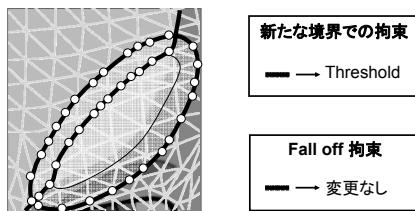
56

ペイントストロークにマッチした輝度補正



57

RBFによる輝度値の補間



- RBFについての制約条件は、メッシュ上の点 (○ や ○) について付与される

58

3次元空間内のRBFによる定式化

- 今回は、3次元空間内に定義されたRBFで次の形のものを用いた:

$$f(x) = \sum_{i=1}^l w_i \phi(x - c_i) + P(x)$$

ここで $x = (x_1, x_2, x_3)$ 、 $\phi(x) = \frac{1}{\|x\|} = \frac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}}$ である。 $P(x)$ は x_1, x_2, x_3 に関する1次式であり、 c_i は制約点 (○ とか ○) である。 l は制約点の総数となる。

先の制約条件から求めるべきは、上記の重み係数 $\{w_i\}$ と1次多項式 $P(x)$ の係数である。

- これら諸係数を求めるために、次の連立一次方程式を解く: 与えられた $h_j (1 \leq j \leq h)$ に対する l 個の1次方程式

$$\sum_{i=1}^l w_i \phi(c_j - c_i) + P(c_j) = h_j, \quad \text{for } 1 \leq j \leq l.$$

これは $l + 4$ 個の未知数に対する方程式。ここにさらに次の条件 ♦ を課す:

♦ 任意の1次式 $Q(x)$ に対し、 $\sum_{i=1}^l w_i Q(c_i) = 0$.

59

RBFを決定する連立一次方程式

- 1次の多項式がなすベクトル空間 \mathcal{P}_1 は、 $1, x_1, x_2, x_3$ という4つの関数を基底としてもつから、先の条件♦は、次の4つの条件と同値:

$$\sum_{i=1}^4 w_i = 0 \text{ および } \sum_{i=1}^4 w_i c_{ij} = 0, \quad (j=1, 2, 3)$$

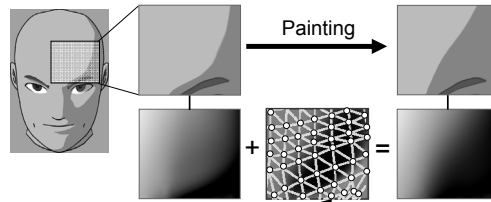
- $P(x) = p_0 + p_1x_1 + p_2x_2 + p_3x_3$ とし $\phi_{ij} = \phi(c_i - c_j)$ とおけば、求める諸係数は、次の連立方程式の解として求まる:

$$\begin{pmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1l} & 1 & c_{11} & c_{12} & c_{13} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2l} & 1 & c_{21} & c_{22} & c_{23} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{l1} & \phi_{l2} & \dots & \phi_{ll} & 1 & c_{l1} & c_{l2} & c_{l3} \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ c_{11} & \dots & \dots & c_{1l} & \vdots & \ddots & \vdots & p_1 \\ c_{12} & \dots & \dots & c_{12} & \vdots & \ddots & \vdots & p_2 \\ c_{13} & \dots & \dots & c_{13} & 0 & \dots & \dots & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_l \\ p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} h_1 \\ \vdots \\ h_l \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

60

オフセットデータの構成

- モデル表面メッシュの各頂点ごとに、修正後の輝度値ともとの輝度値の差分(offset)を保持しておく



$$\text{Modified Intensity} = \text{Original Intensity} + \text{Offset}$$

61

RBF の数学的基礎

—主にスプライン平滑化としての側面から—

62

補間と平滑化

補間とは?

- 関数 f の既知の値 $f(x_1), f(x_2), \dots, f(x_n)$ をもとに、これらの点 $(x_k, f(x_k))$ ($k=1, 2, \dots, n$)を通る別の関数の値で代用すること
- このときの「代用」関数を補間式(interpolation formula)と呼ぶ。 $\{x_k\}$ は補間点(または分点)と呼ばれる。
 - 求める補間式を $\min\{x_k\} \leq x \leq \max\{x_k\}$ の範囲で x を考えると内挿、それ以外の場合外挿という。

63

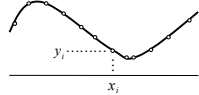
さまざまな補間方式

- 多項式補間(Lagrange補間)
 - n 個の点を通る n-1次多項式(補間多項式)は一意に定まる
- Hermite 補間
 - $f(x_i)$ と $f'(x_i)$ を用いて、 $2n-1$ 次多項式で補間する
- スプライン補間
 - 区間 $[a, b]$ の分割 $\Delta: a = t_0 < t_1 < \dots < t_N = b$ に対して、 N 個の小区間 $[t_p, t_{p+1}]$ にて m 次多項式となる関数を、区分的多項式と呼び、全体で C^m 級となるとき、 m 次スプライン関数と呼ぶ。
 - B スプライン、3次スプライン
- 有理補間(rational interpolation)
 - 有理式 $R_n^m(x) = P_n(x)/Q_m(x)$ (ここで $P_n(x), Q_m(x)$ はそれぞれが高々 m 次、 n 次多項式) を用いた補間を有理補間という。
 - NURBS (非一様有理Bスプライン Non-Uniform Rational B-Spline)

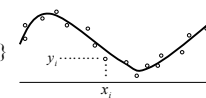
スプライン補間/平滑化

以下で P は微分作用素とする:

- スプライン補間

$$\min_f \int |Pf(x)|^2 dx \quad \text{subject to } f(x_i) = y_i$$


- スプライン平滑化

$$\min_f \left\{ \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int |Pf(x)|^2 dx \right\}$$


これらの「最適化」問題を何処で解くか?

関数空間

—最適化問題を解く舞台として—

- 前記の最適化問題の解を求める空間として、どんな関数空間を考えるべきか?
 - 関数空間とは、関数を要素として考える関数の集合体
 - Ex. 1) $\mathcal{P}_m := \{P(x) | P(x) \text{ は高々 } m \text{ 次多項式} \}$
 - 2) $C^m(\Omega)$ は \mathbb{R}^n 内の領域 Ω 上で定義された m 階連続微分可能な関数全体
 - 3) $L^p(\Omega) := \{f: \Omega \rightarrow \mathbb{R} \cup \{\pm\infty\} \mid \int |f(x)|^p d\mu(x) < \infty\}$
 は、 Ω 上で p 乗可積分な関数全体、と呼ばれる ($p > 0$)
- 関数空間は通常線形位相空間として扱われる
 - ベクトル空間であり、関数列の収束が定義されている

ヒルベルト空間

【定義】実ベクトル空間 H で内積 $\langle \cdot, \cdot \rangle$ が定義されている空間を内積空間、またはプレ・ヒルベルト空間(pre-Hilbert space) という:

- (1) $\langle \alpha f + \beta g, h \rangle = \alpha \langle f, h \rangle + \beta \langle g, h \rangle$, for any $f, g, h \in H, \alpha, \beta \in \mathbb{R}$
- (2) $\langle f, g \rangle = \langle g, f \rangle$ for any $f, g \in H$
- (3) $\langle f, f \rangle \geq 0$ for any $f \in H$; $\langle f, f \rangle = 0 \Leftrightarrow f = 0$.

- 内積空間は、ノルム(norm) $\|f\|_H = \sqrt{\langle f, f \rangle}$ によって線形位相空間となる
 - $\lim_{n \rightarrow \infty} f_n = f \Leftrightarrow \lim_{n \rightarrow \infty} \|f_n - f\|_H = 0$ により H の要素からなる関数列 $\{f_n\}_{n=1}^{\infty}$ の収束が定義される。

【定義】内積空間 H がノルムにつき完備(complete)であるとき、ヒルベルト空間(Hilbert space)である、という

- H における任意のCauchy列 $\{c_n\}_{n=1}^{\infty}$ (i.e., $\lim_{k, m \rightarrow \infty} \|c_k - c_m\|_H = 0$) は収束し、 $\lim_{n \rightarrow \infty} c_n \in H$.

内積空間の例

- n 次元実数空間 \mathbf{R}^n .
 - $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n) \in \mathbf{R}^n$ に対し $\langle x, y \rangle := \sum_{i=1}^n x_i y_i$ により内積を定義すれば \mathbf{R}^n はヒルベルト空間となる.
 - \mathbf{Q} を有理数全体のなす空間で、内積を通常の乗算とすれば、 \mathbf{Q} は内積空間. そのノルムは通常の絶対値である:

$$\|q\|_0 = \sqrt{\langle q, q \rangle} = |q| \text{ for } q \in \mathbf{Q}.$$
 しかし \mathbf{Q} は完備ではない.
 - L^2 空間 $L^2(\Omega) := \{f: \Omega \rightarrow \mathbf{R} \cup \{\pm\infty\} \mid \int |f(x)|^2 d\mu(x) < \infty\}$ は内積 $\langle f, g \rangle := \int f(x)g(x)d\mu(x)$ によりヒルベルト空間である.
- 【参考】任意の内積空間は、あるヒルベルト空間に稠密に埋め込むことができる. このときのヒルベルト空間をもとの空間の完備化という.

再生核ヒルベルト空間 RKHS: Reproducing Kernel Hilbert Space

- $H = H(\Omega)$ を、領域 Ω 上に定義された関数から構成されるヒルベルト空間とする. いま $\Omega \times \Omega$ 上で定義され、次の性質を持つ関数 $K(x, y)$ が存在すると仮定する:
 - (i) 任意の点 $y \in \Omega$ を固定するとき、 $K(x, y)$ は x を変数とする関数として H に属す.
 - (ii) 任意の $f \in H$ に対して $f(y) = \langle f(x), K(x, y) \rangle_x$ が成立する.
 ($\langle \cdot, \cdot \rangle_x$ とは、内積の対象となる関数を x の関数とみて計算することを示す).
- 【定義】上記の関数 $K(x, y)$ を、ヒルベルト空間 H の再生核と呼ぶ. またこのとき H を再生核ヒルベルト空間と呼ぶ.
- $K(x, y)$ を再生核とすると次が成立する
 - $K(y, z) = \langle K(x, y), K(x, z) \rangle_x$
 - $K(x, y)$ は対称かつ半正値関数. 即ち任意の $a = (a_1, a_2, \dots, a_n) \in \mathbf{R}^n$ に対して $\sum_{i,j=1}^n a_i a_j K(x_i, x_j) \geq 0$ が成立する.

再生核 K と RKHS の同値性

- 次の定理は RKHS についての基本的な定理で、これをもとに実用上のさまざまな手法の理論的根拠が導かれる:
- 【定理】 Ω を \mathbf{R}^n 内の領域とし、 $\Omega \times \Omega$ 上の実数値関数 $K(x, y)$ が対称かつ半正値であったとする.
- このとき Ω 上の関数空間 H で K を再生核とするような再生核ヒルベルト空間 (RKHS) が存在する.
- 【証明のスケッチ】与えられた K をもとに、次式で定義される関数空間 F を考える:
- $$F := \left\{ \sum_{j=1}^n \alpha_j K(x, x_j) \mid l \in \mathbf{N}, \alpha_l \in \mathbf{R}, x_j \in \Omega \right\}$$
- これは明らかに Ω 上の関数空間であり、ベクトル空間をなす. さらに次式により F の任意の要素 f と g に対し、内積 $\langle f, g \rangle$ を定義できる:
- $$f = \sum_{i=1}^m \alpha_i K(x, x_i), \quad g = \sum_{j=1}^n \beta_j K(x, x_j) \in F \Rightarrow \langle f, g \rangle = \sum_{i=1}^m \sum_{j=1}^n \alpha_i \beta_j K(x_i, x_j) \in \mathbf{R}$$
- この定義が内積の条件の一つ $\langle f, f \rangle \geq 0$ ($\forall f \in F$) を満たすことは K の半正値性による. こうして F は pre-Hilbert space となる. K が再生核の条件(i), (ii) を満たすことも定義からすぐわかる. F の完備化を H とすればよい. もちろん K は H でも条件(i), (ii) を満たす.

Native Hilbert Space

- RKHSの要素 $f(x)$ は各点 x ごとに有限の値が定まっているという意味で一般のHilbert空間の関数よりも“普通”の関数である
- 【参考】例えば L^2 空間には無限大の値をとる関数も含まれる. また一方で Dirac の δ 関数は L^2 にも属さない.
- 前記定理の証明にあるように、 $\Omega \times \Omega$ 上の実数値、対称半正値関数 $K(x, y)$ が与えられると、この関数を再生核とするRKHSは

$$F := \left\{ \sum_{j=1}^n \alpha_j K(x, x_j) \mid l \in \mathbf{N}, \alpha_l \in \mathbf{R}, x_j \in \Omega \right\}$$
 を稠密な部分空間として持つ: $H(\Omega) = \overline{F}$ (\overline{F} は F の閉包).
 - $H(\Omega) = \overline{F}$ とは、任意の $H(\Omega)$ の要素は F の要素で近似できることを示している. しかも F の要素は K により具体的に表せる(次スライド).
 - $H(\Omega) = \overline{F}$ は native Hilbert space (10) とも呼ばれている.

RKHSにおける平滑化問題

- 再生核ヒルベルト空間 H の中で次の平滑化問題を考える:

$$\min_f \left\{ \sum_{i=1}^N (y_i - f(x_i))^2 + \Psi(\|f\|_H) \right\}$$

ここで「正則化」の項にある $\Psi(t)$ は $t \geq 0$ で定義された単調増加関数。

【定理】(Representer Theorem) $\dagger\dagger$ 再生核ヒルベルト空間 H の要素 $\{K(\cdot, x_m)\}_{m=1}^N$ が張る N 次元部分空間を F_N とする:

$$F_N = \text{span} \{K(\cdot, x_1), \dots, K(\cdot, x_N)\}$$

このとき平滑化の問題の解は F_N の中に存在する。

- この定理により、無限次元空間 H での最適化問題が有限次元での最適化問題に置き換えられることになる。
- しかもその最適解の形は $f(x) = \sum_{m=1}^N \alpha_m K(\cdot, x_m)$ で求めてよい。

$\dagger\dagger$ 福水健次「カーネル法の基礎」
http://www.sim.ac.jp/~fukumizu/ISM_lecture_2006/Lecture2006_kernel_theory.pdf 72

正則化問題 (regularization problem)

- 以下では簡単のため $\Omega = \mathbb{R}^n$ とする。次の正則化問題を考える:

$$\min_f \left\{ \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda J_m^a(f) \right\}$$

ここで正則化の項 $J_m^a(f)$ は「解」の滑らかさを規定するのに使われる。

$$J_m^a(f) = \sum_{\alpha_1 + \dots + \alpha_n = m} \frac{m!}{\alpha_1! \dots \alpha_n!} \|D^\alpha f\|_{L^2}^2$$

$$D^\alpha f = \frac{\partial^m f}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}$$

- $\mathbf{B}_m^a := \{f: \mathbb{R}^n \rightarrow \mathbb{R} \mid D^\alpha f \in L^2 (\forall \alpha, |\alpha|=m)\}$ とおく。これが上記の問題を解くための RKHS となれば話は簡単だが、そうはいかない。

【参考】上記の $J_m^a(f)$ の代わりに $\sum_{m \geq 0} a_m J_m^a(f)$ を考え (a_m は全て定数)、 $a_0 \neq 0$ である場合には、事情が異なる(後述)。

\dagger 福水健次「カーネル法の基礎」
http://www.iem.ac.jp/~fukumizu/ISM_lecture_2006/Lecture2006_kernel_theory.pdf 73

正則化問題 に対するRKHS

- $m = n = 2$ の場合は Thin Plate Spline:

$$J_2^2(f) = \iint \left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \frac{\partial^2 f}{\partial x_1 \partial x_2} \frac{\partial^2 f}{\partial x_2 \partial x_1} + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 dx_1 dx_2$$

- 一般の場合に戻る。次の点に注意:
 - 零化空間の存在: $J_m^a(f) = 0 \Leftrightarrow f \in \mathcal{P}_{m-1}$ ($m-1$ 次以下の多項式全体)
 - $J_m^a(f) = (-1)^m (f, \Delta^m f)_{L^2}$ ($\int_{\Omega} D^\alpha f(x) = 0$ に注意して形式的に部分積分)
- そこで $\mathbf{B}_m^a = \mathcal{P}_{m-1} \oplus H_m^a$ と直和分解し H_m^a で正則化問題を考える
 - \mathcal{P}_{m-1} は有限次元ベクトル空間で $\dim \mathcal{P}_{m-1} = m + \dots + 1, C_n = M$ と以下では表記)
 - 射影 $\pi: \mathbf{B}_m^a \rightarrow \mathcal{P}_{m-1}$ は以下のように構成する:
 - 次の条件を満たす M 個の点 $a_1, a_2, \dots, a_M \in \mathbb{R}^n$ を求める (これは常に可能):
 $\forall b_1, b_2, \dots, b_M \in \mathbb{R} \exists p(x) \in \mathcal{P}_{m-1}$ s.t. $p(a_i) = b_i (1 \leq i \leq M)$
 - \mathcal{P}_{m-1} の基底 ϕ_1, \dots, ϕ_M s.t. $\phi_j(a_i) = \delta_{ij} (1 \leq j, k \leq M)$ をとる
 - $f \in \mathbf{B}_m^a$ に対し $\pi(f) = \sum_{j=1}^M f(a_j) \phi_j$ と定義する

正則化問題 に対するRKHS

【定理】(Meinguet 1979^{*)} $m > n/2$ のとき、 H_m^a は次の $\langle f, g \rangle_{H_m^a}$ を内積とする再生核ヒルベルト空間である:

$$\langle f, g \rangle_{H_m^a} := \sum_{\alpha_1 + \dots + \alpha_n = m} \frac{m!}{\alpha_1! \dots \alpha_n!} \langle D^\alpha f, D^\alpha g \rangle_{L^2} = \langle (-1)^m \Delta^m f, g \rangle_{L^2}$$

特に $\|f\|_{H_m^a}^2 = J_m^a(f)$ である。

- \mathbf{B}_m^a から H_m^a への射影 $\theta := 1 - \pi$ を考慮すれば、上記の定理から次は同値となる(証明略):

$$\begin{aligned} & \min_f \left\{ \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda J_m^a(f) \right\} \\ \iff & \min_f \left\{ \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \|\theta(f)\|_{H_m^a}^2 \right\} \\ \iff & \min_{g \in H_m^a, p \in \mathcal{P}_{m-1}} \left\{ \sum_{i=1}^N (y_i - (g(x_i) + p(x_i)))^2 + \lambda \|g\|_{H_m^a}^2 \right\} \end{aligned}$$

(*) Meinguet, J. (1979) Multivariate Interpolation at Arbitrary Points Made Simple. *J. Applied Mathematics and Physics (ZAMP)* 30, 292–304. 75

正則化問題の解の形

- $\langle f, g \rangle_{H_m^n}$ の定義に登場した偏微分作用素 Δ^m に対する Green 関数は $\Delta^m G(x) = \delta(x)$ を満たすが具体的には以下のように表せる:

$$G(x) = \begin{cases} \beta_{m,n} |x|^{2m-n} \log|x| & (2m-n \text{ が正の偶数のとき}) \\ \gamma_{m,n} |x|^{2m-n} & (\text{上記以外}) \end{cases}$$

- H_m^n の再生核 K はこの Green 関数をもとに表現される. Representer Theorem を経由して、結局最適解は次の形に表現できる (証明略):

$$f(x) = \sum_{m=1}^N a_m G(x - x_m) + \phi_p(x)$$

ここで $\phi_p(x)$ は $m-1$ 次以下の多項式.

[例] Latent Doodle Space のときは、 $m = n = 2$ の場合に相当する。 $2m - n = 2$ となるので、 $G(x) = |x|^2 \log|x|$ の形となり、かつ $\phi_p(x)$ は 1 次式である。

正則化問題とRBF

- $J_m^n(f)$ の場合の RKHS に付随して 偏微分作用素 Δ^m の Green 関数が登場し、それが 動径基底関数(RBF)であることがわかった:

$$G(x) = \begin{cases} \beta_{m,n} |x|^{2m-n} \log|x| & (2m-n \text{ が正の偶数のとき}) \\ \gamma_{m,n} |x|^{2m-n} & (\text{上記以外}) \end{cases}$$

【参考】上記の $J_m^n(f)$ の代わりに $\sum_{m=0}^N a_m J_m^n(f)$ (a_m は定数、 $a_0 \neq 0$) で正則化問題を考えると、 $\sum_{m=0}^N (-1)^m a_m \Delta^m$ に対する Green 関数 G が Fourier 変換により求められ、最適解の形は $f(x) = \sum_{m=1}^N c_m G(x - x_m)$ となる。例えば、 $a_m = \frac{\sigma^{2m}}{m! 2^m}$ ($\sigma > 0$) であるなら、 $G(x) = c \exp(-\frac{\|x\|_2^2}{2\sigma^2})$ が登場する(*)。

(*) 福水健次「カーネル法の基礎」
http://www.iam.ac.jp/~fukumizu/ISM_Lecture_2006/Lecture2006_kernel_theory.pdf 77

おわりに

数学的基礎の重要性と面白さ

- 今回はPCAやRBFなど、普段使っているアルゴリズムの背景について紹介しました
- プログラミングに直接有効な情報は少ないかもしれませんが
- アルゴリズムの背景を知ることによりその理論の奥深さに触れることができます
- 未知の問題を解決しようとするときに、役立ちます
- 理論と実践を強固につなぐための仲間作りをしてはどうでしょうか？