Computer Vision I: Variational Methods

Dr. Yvain Quéau, Christiane Sommer, Nikolaus Demmel
Chair for Computer Vision & Pattern Recognition
Technical University of Munich

Winter 2017/18

Chapter 0
Introduction to Variational Methods for Computer Vision

Computer Vision I: Variational Methods

Winter 2016/17

Dr. Yvain QUÉAU
Chair for Computer Vision and Pattern Recognition
Departments of Informatics & Mathematics
Technical University of Munich

Today's Lecture

1. **Module Organization**
2. **Introduction to Computer Vision**
3. **Two Different Paradigms for Computer Vision**
4. **Introduction to Variational Methods**
5. **A Bit of History**
6. **Overview of the Lecture**

1. **Module Organization**
2. Introduction to Computer Vision
3. Two Different Paradigms for Computer Vision
4. Introduction to Variational Methods
5. A Bit of History
6. Overview of the Lecture

Computer Vision Group in TUM

http://vision.in.tum.de

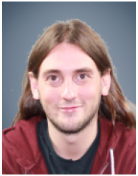
Organization

Contents

- ≈ 20 lectures + 10 tutorials, **3h + 3h** weekly
- Written exam at the end
- 8 ECTS

People

- Lectures: Dr. Yvain QUÉAU (based on material from previous years by Prof. Daniel CREMERS)
- Tutorials: Christiane Sommer, Nikolaus Demmel


Yvain


Christiane


Nikolaus


Daniel

Organization

Lectures

- Wednesday 10:15 - 11:45
  + Thursday 10:15 - 11:00
- Room: 09.02.023
- Slides online after the lecture

Tutorials

- Tuesday 16:00 - 18:15
- Room: 02.05.014
- Exercise sheet posted the week before the tutorial
- Solution discussed in class, then posted online

Check detailed agenda (date + topic) online:

`https://vision.in.tum.de/teaching/ws2017/vmcv2017`

(TUMOnline may be incorrect)

Organization

Online Resources:

https://vision.in.tum.de/teaching/ws2017/vmcv2017

- Agenda
- Slides
- Exercise sheets + solutions
- Recording of former lectures by Prof. Daniel CREMERS

Password: **vmcv_ws1718**

Contact

- Questions: cvvm-ws17@vision.in.tum.de
- Office hours: please ask for a meeting by email
- Yvain's office: 02.09.053
- Christiane's office: 02.09.037
- Nikolaus' office: 02.09.057

Goal of the lecture

- Give an overview of **computer vision**
- Describe major **inverse problems** in computer vision
- Provide a generic **mathematical approach** for solving them
- Show how to implement such solutions on CPU
- Discuss open problems and limits of the state-of-the-art

Required: basic analysis, linear algebra, statistics

Useful: optimization (Convex Optimization for Computer Vision and Machine Learning - IN2330)

Recommended readings

P. Kornprobst, G. Aubert, “Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations”, Springer 2006.

T. Chan, J. Shen, “Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods”, SIAM 2005.

J.-M. Morel, S. Solimini, “Variational Methods in Image Segmentation”, Birkhäuser 1995.

K. Bredies, D. Lorenz, “Mathematische Bildverarbeitung: Einführung in Grundlagen und moderne Theorie”, Vieweg & Teubner 2011.

What is computer vision?

1 Module Organization

2 **Introduction to Computer Vision**

3 Two Different Paradigms for Computer Vision

4 Introduction to Variational Methods

5 A Bit of History

6 Overview of the Lecture

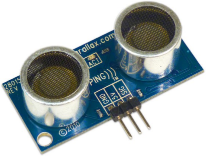
Computer vision tools: Sensors

Camera

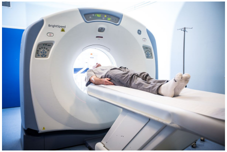
Movie camera

Depth sensor

Infrared sensor



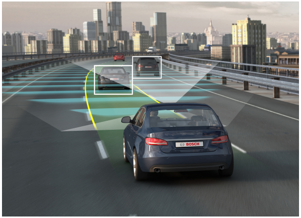
Ultrasound sensor



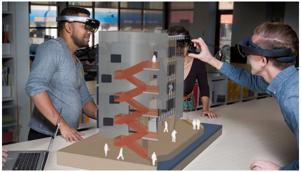
X-ray scanner

- Sensors capture **images** of the world
- Computer vision aims at analyzing / understanding these visual signals

Computer vision: What for?



Autonomous driving



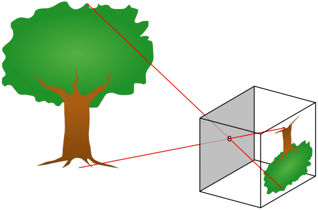
Augmented reality

Robotics

And also...

- Computer-assisted medical diagnostic
- Face recognition (surveillance)
- Surface inspection (quality control)
- Relighting (VFX)
- Earth monitoring
- ...

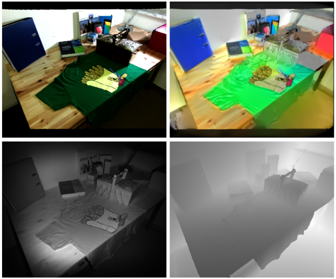
Different types of images: Cameras



Measures photons emitted (reflected) by the scene's surface

- Greylevel image = function u associating to each pixel (x, y) an integer value:
  $u: [1, N] \times [1, M] \to \{0, \ldots, 255\};\ (x, y) \mapsto u(x, y)$
- RGB image:
  $u: [1, N] \times [1, M] \to \{0, \ldots, 255\}^3;\ (x, y) \mapsto \mathbf{u}(x, y)$
- Movie camera:
  $u: [1, N] \times [1, M] \times [1, T] \to \{0, \ldots, 255\}^3;\ (x, y, t) \mapsto \mathbf{u}(x, y, t)$

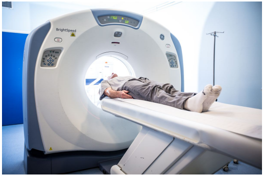
Different types of images: Depth sensors

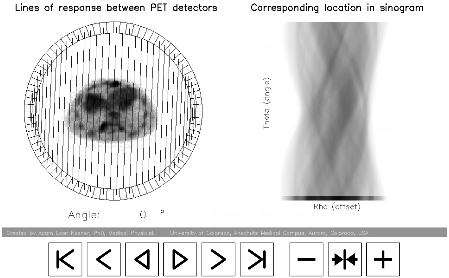


Measures distances to the scene's surface (based on triangulation or time-of-flight), sometimes also provides IR image

- IR image = greylevel image:
  $u: [1,N] \times [1,M] \to \{0,\dots,255\};\ (x,y) \mapsto u(x,y)$
- Depth image:
  $u: [1,N] \times [1,M] \to \mathbb{R};\ (x,y) \mapsto u(x,y)$

Different types of images: X-ray Scanners

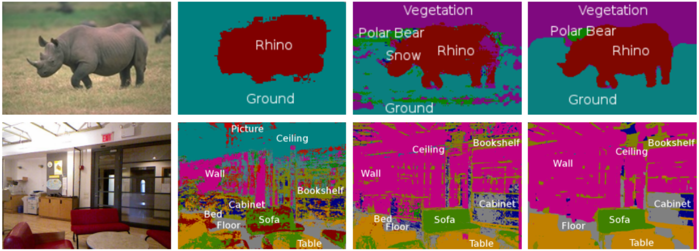




Measures attenuation of X-ray for a given time and angle

X-ray image = sinogram:
$u: [0, 2\pi] \times [0, 1] \rightarrow [0, 1]; (x, y) \mapsto u(x, y)$

From sensors to visual understanding: What is that?



a) Original image   b) Classification of [14]/[7]   c) Proposed Classification   d) Proposed Segmentation

- Raw measurements from a sensor are easily understood by humans, but not by computers
- Computer vision aims at making computers “understand” what they see

(image source: semantic segmentation by C. HARIZBAS et al., SSVM 2015)

From sensors to visual understanding: Where am I?

- Various information can be extracted from visual clues: location, map of the environment, etc.

(image source: stereo SLAM by R. WANG et al., ICCV 2017 – see video)

From sensors to visual understanding: Why do I see such images?

- Understanding the world requires understanding what led to the observed images, e.g. which 3D-shape could have produced a given set of RGB or depth images (**inverse problem**)

(image source: copyme 3D by J. STURM et al., GCPR 2013 – see video)

How to achieve this task?

Module Organization

Introduction to Computer Vision

Two Different Paradigms for Computer Vision

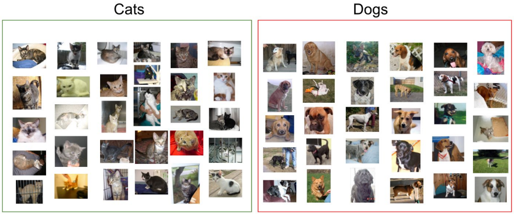
Introduction to Variational Methods

A Bit of History

Overview of the Lecture

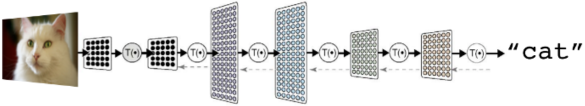
Paradigm 1: machine learning

Case 1: Humans can solve the problem, though they cannot explain why (e.g., recognition tasks): **machine learning**



Sample of cats & dogs images from Kaggle Dataset

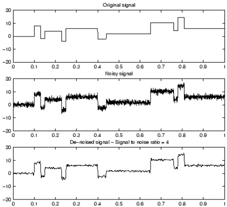
Provide the machine with annotated data;
Let it “learn” what a cat is



Based on the numerous examples it knows, machine can tell “this is a cat” when given a new image

Paradigm 2: variational methods

Case 2: Humans know how they would solve the problem (e.g., restoration tasks): **variational methods**



1) Model the signal acquisition process: $u_0(t) = u(t) + \mathcal{N}(0, \sigma^2)$, $t \in [0, 1]$ (u_0: observed signal, u: uncorrupted signal, $\mathcal{N}$: random Gaussian noise)

2) Invoke Bayesian inference to turn the problem into a **continuous optimization problem**:

$$\min_{u:\,[0,1]\to\mathbb{R}} \int_{t=0}^{1} |u(t) - u_0(t)|^2 + \lambda |u'(t)|^2 \,\mathrm{d}t$$

3) Turn the optimization problem into a differential equation (Euler-Lagrange):

$$\lambda u''(t) - u(t) = u_0(t), \quad t \in [0, 1]$$

4) Solve the differential equation with the computer

Machine learning VS Variational methods

Machine learning

- AI-oriented
- Not clear why it works
- Human tells the computer the solution
- Requires heavy computational power
- Broad range of applications
- Community growing since 2012

Variational methods

- Mathematics-oriented
- Guarantee of optimality
- Human tells the computer how to solve
- Usually much more efficient
- Restricted to problems we can model
- Community reducing since 2012

This lecture: **variational methods**

(in fact, these paradigms are much more complementary that it may seem)

What are variational methods?

A few classic inverse problems in computer vision: Denoising

Input image

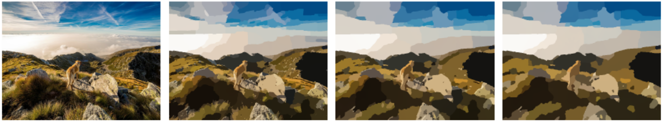
Piecewise smooth approximation

Find an image $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ “close to” the noisy data $u_0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, but “smoother”:

$$\min_{u:\, \Omega \subset \mathbb{R}^2 \to \mathbb{R}} \iint_{(x,y)\in\Omega} \underbrace{|u(x,y) - u_0(x,y)|^2}_{\text{“close to”}} + \lambda \underbrace{\|\nabla u(x,y)\|^2}_{\text{“smoother”}} \, \mathrm{d}x\mathrm{d}y$$

(image source: fast Mumford-Shah denoising by E. STREKALOVSKIY and D. CREMERS, ECCV 2014)

A few classic inverse problems in computer vision: Segmentation



Input image | Proposed, $\lambda = 0.2$ | Proposed, $\lambda = 0.4$ | Proposed, $\lambda = 0.6$

Find an image $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ “close to” the input image $u_0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, but “piecewise constant”:

$$\min_{u:\,\Omega\subset\mathbb{R}^2\to\mathbb{R}} \iint_{(x,y)\in\Omega} \underbrace{|u(x,y) - u_0(x,y)|^2}_{\text{“close to”}} + \lambda \underbrace{\delta \, \|\nabla u(x,y)\|}_{\text{“piecewise constant”}} \; \mathrm{d}x\mathrm{d}y$$

(image source: fast Mumford-Shah denoising by E. STREKALOVSKIY and D. CREMERS, ECCV 2014 – see video)

A few classic inverse problems in computer vision: Inpainting

(a) Original photograph (b) Inpainted photograph

Fig.1 Removing large objects from images.

Find an image $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ "close to" the input image $u_0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ on $\overline{\Omega} \subset \Omega$, but "smooth elsewhere":

$$\min_{u:\, \Omega \subset \mathbb{R}^2 \to \mathbb{R}} \underbrace{\iint_{(x,y)\in\overline{\Omega}} |u(x,y) - u_0(x,y)|^2 \, \mathrm{d}x\mathrm{d}y}_{\text{“close to on } \overline{\Omega}\text{”}}$$

$$+ \lambda \underbrace{\iint_{(x,y)\in\Omega\setminus\overline{\Omega}} \|\nabla u(x,y)\|^2 \, \mathrm{d}x\mathrm{d}y}_{\text{“smooth elsewhere”}}$$

A few classic inverse problems in computer vision: Data compression

(a) (b) (c)

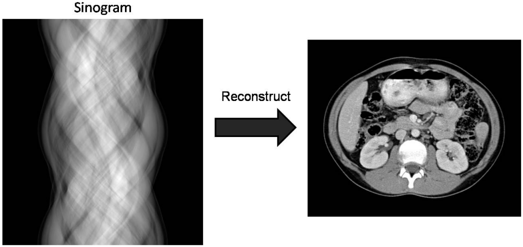
Find an image $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ "close to" the compressed image $u_0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ on $\overline{\Omega} \subset \Omega$, but "smooth elsewhere":

$$\min_{u:\,\Omega\subset\mathbb{R}^2\to\mathbb{R}} \underbrace{\iint_{(x,y)\in\overline{\Omega}} |u(x,y)-u_0(x,y)|^2 + \lambda \|\nabla u(x,y) - \nabla u_0(x,y)\|^2 \,\mathrm{d}x\mathrm{d}y}_{\text{"close to on } \overline{\Omega}\text{", at order 1}}$$

$$+ \mu \underbrace{\iint_{(x,y)\in\Omega\setminus\overline{\Omega}} \|\nabla u(x,y)\|^2 \,\mathrm{d}x\mathrm{d}y}_{\text{"smooth elsewhere"}}$$

(image source: normal integration by Y. QUÉAU et al., Arxiv 2017)

A few classic inverse problems in computer vision: 2D-reconstruction (tomography)

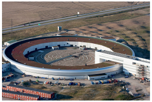


Find a "smooth" image $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ "whose Radon transform matches" the noisy sinogram $u_0 : [0,1] \times [0, 2\pi] \to \mathbb{R}$

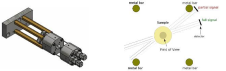
$$\min_{u:\,\Omega\subset\mathbb{R}^2\to\mathbb{R}} \iint_{(x,y)\in\Omega} \underbrace{|u(x,y) - R^{-1}(u_0)(x,y)|^2}_{\text{"matches sinogram"}} + \lambda \underbrace{\|\nabla u(x,y)\|^2}_{\text{"smooth"}} \,\mathrm{d}x\mathrm{d}y$$

A few classic inverse problems in computer vision: Combining several variational problems

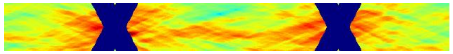
All these tools can be combined in a big variational problem if needed. E.g., joint reconstruction, inpainting and segmentation for Synchrotron X-ray tomography:



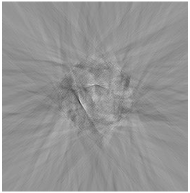
Max IV synchrotron



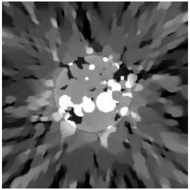
Acquisition device

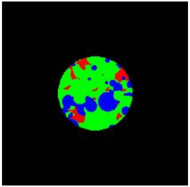


Sinogram



Reconstruction only

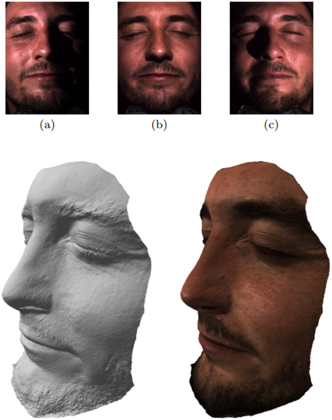




Reconstruction + Segmentation + Inpainting

(image source: CT reconstruction by F. LAUZE et al., SSVM 2017)

A few classic inverse problems in computer vision: Single-view 3D-reconstruction

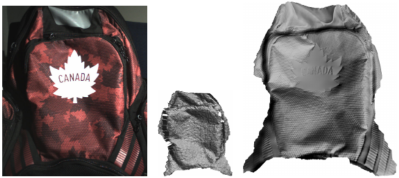

(a) (b) (c) (d) (e)

(image source: photometric stereo by Y. QUÉAU et al., JMIV 2017 – see video)

Find a depth map $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ “explaining” the image $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$:

$$\min_{u:\,\Omega\subset\mathbb{R}^2\to\mathbb{R}} \iint_{(x,y)\in\Omega} \left\| \mathbf{a}(x,y) \cdot \nabla u(x,y) - I(x,y) \right\|^2 \mathrm{d}x\mathrm{d}y$$

A few classic inverse problems in computer vision: shading-aware depth refinement



Input RGB image   Input depth   3D refined shape

(image source: depth super-resolution by S. PENG et al., ICCVW 2017)

Find a high-res depth map $u : \Omega_{HR} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ "close to" a low-res one $u_0 : \Omega_{LR} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ which "matches" a high-res image $I : \Omega_{HR} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$\min_{u:\,\Omega\subset\mathbb{R}^2\rightarrow\mathbb{R}} \underbrace{\iint_{(x,y)\in\Omega_{LR}} |Ku(x,y) - u_0(x,y)|^2 \,\mathrm{d}x\mathrm{d}y}_{\text{"close to"}}$$

$$+ \lambda \underbrace{\iint_{(x,y)\in\Omega_{HR}} \|\mathbf{a}(x,y)\cdot\nabla u(x,y) - I(x,y)\|^2}_{\text{"matches"}}$$

Variational Methods = a generic tool for inverse problems

Whatever the sensor:

- Camera
- Depth sensor
- X-ray sensor
- ...

Whatever the task:

- Restoration
- Reconstruction
- Segmentation
- ...

Recast the problem as an optimization problem:

$$\min_{u\colon \Omega\subset\mathbb{R}^n\to\mathbb{R}^d} \int_\Omega \mathcal{L}(x, u(x), \nabla u(x), \dots)\,\mathrm{d}x$$

Key issues

- What are Ω, n and d?
- How to choose $\mathcal{L}$?
- Is there a solution? Unique?
- How to discretize and solve the optimization problem?

Where do such ideas come from?

Historical motivation I

- 1657: Fermat's principle ("The path taken between two points by a ray of light is the path that can be traversed in the least time")
- **1744** (Euler) : first necessary condition to solve

$$\begin{cases} \min\limits_{u\colon [x_A, x_B] \to \mathbb{R}} \int_{x_A}^{x_B} \mathcal{L}(x, u(x), u'(x))\, \mathrm{d}x \\ u(x_A) = u_A \\ u(x_B) = u_B \end{cases}$$

- 1746: principle of least actions (Maupertuis): "Nature is thrifty in all its actions"
- 1755: reformulation by Lagrange of Euler's necessary condition (⇒ Euler-Lagrange equation in 1766) :

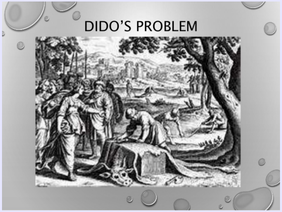
$$\frac{\partial \mathcal{L}}{\partial u} - \frac{d}{dx}\left(\frac{\partial \mathcal{L}}{\partial u'}\right) = 0$$

- 1786: extension to $\min\limits_{u} \int_{x_A}^{x_B} \mathcal{L}(x, u(x), u'(x), u''(x))\, \mathrm{d}x$ (Legendre)

Historical Motivation II: Before that...

Dido's problem

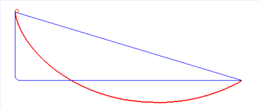
$\approx$ 800 BC: Queen Dido lands in Carthago...



What is the closed curve which has the maximum area for a given perimeter?

The brachistochrone

- 1638: first mention by Galileo
- 1696: challenge by Johann Bernoulli to his fellows
- 1697: solutions by Johann Bernoulli, Leibniz, Newton and... Jacob Bernoulli



Historical Motivation III: Hilbert

- 19th century: Dirichlet, Riemann, Weierstrass and Neumann study **Dirichlet's problem** :

$$\min_{:\,\Omega\to\mathbb{R}} \int_{\Omega} \|\nabla u(x)\|^2 \,\mathrm{d}x \tag{1}$$

  depending on boundary conditions, with $\Omega \subset \mathbb{R}$, $\mathbb{R}^2$ or $\mathbb{R}^3$
- 1900: Hilbert problems number 20 and 23
  – Number 20: Do all variational problems with certain boundary conditions have solutions?
  – Number 23: Further development of the calculus of variations
- 1900-... : Hilbert space theory, optimization,...

Conclusion on those historical landmarks

It is **natural** to formulate computer vision tasks as variational problems

Dr. Yvain QUÉAU

Overview of the Lecture

Chapter 0:	Introduction
Chapter 1:	Images and Image Filtering
Chapter 2:	Diffusion Filtering
Chapter 3:	Variational Calculus
Chapter 4:	Variational Image Restoration
Chapter 5:	Image Segmentation I – Basics
Chapter 6:	Image Segmentation II – Variational Approaches
Chapter 7:	Image Segmentation III – Bayesian Inference
Chapter 8:	Level Set Methods
Chapter 9:	Convex Relaxation Methods I – Segmentation
Chapter 10:	Motion Estimation & Optical Flow
Chapter 11:	Convex Relaxation Methods II – Multiview Reconstruction
Chapter 12:	Photometric Techniques