Variational Methods in Imaging

Yvain QUÉAU

CNRS researcher at GREYC

ENSICAEN - Université de Caen Normandie

March 2023

Chapter 1

Introduction to Variational Methods and Computer Vision

Variational Methods in Imaging

March 2023

Yvain QUÉAU
GREYC-CNRS
ENSICAEN - Université de Caen Normandie

Today's lecture

1. **Introduction to Computer Vision**
2. **Two Different Paradigms for Computer Vision**
3. **Introduction to Variational Methods**
4. **A Bit of History**

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, PDEs

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.

Lectures at TU Munich from Daniel CREMERS:
`https://www.youtube.com/playlist?list=PLTBdjV_4f-EJ7A2iIH5L5ztqqrWYjP2RI`

1 Introduction to Computer Vision

2 Two Different Paradigms for Computer Vision

3 Introduction to Variational Methods

4 A Bit of History

Computer vision tools: Sensors

Camera

Movie camera

Depth sensor

Infrared sensor

Ultrasound sensor

X-ray scanner

- Sensors capture **images** (of different types) of the world
- Computer vision aims at high-level analysis (i.e., “understanding”) these visual signals

Computer vision: What for?

Autonomous driving

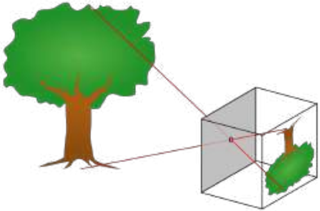
Augmented reality

Robotics

And also...

- Computer-assisted medical diagnostic
- Videosurveillance
- Surface inspection
- After effects
- Earth monitoring
- ...

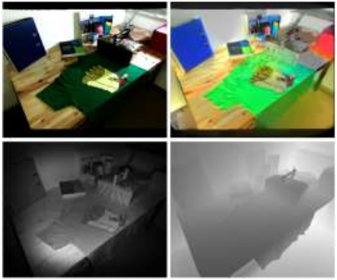
Different types of images: Cameras



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

- Greylevel image $=$ function u associating to each pixel $(x, y) \in \Omega \subset \mathbb{R}^2$ a float value:
  $u: \Omega \to \mathbb{R}; (x, y) \mapsto u(x, y)$
- RGB cameras associate three float values to each pixel:
  $u: \Omega \to \mathbb{R}^3; (x, y) \mapsto [u_R(x, y), u_G(x, y), u_B(x, y)]^\top$
- Movie RGB cameras associate three float values to each (pixel, time):
  $u: \Omega \times \mathbb{R} \to \mathbb{R}^3; (x, y, t) \mapsto$
  $[u_R(x, y, t), u_G(x, y, t), u_B(x, y, t)]^\top$

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

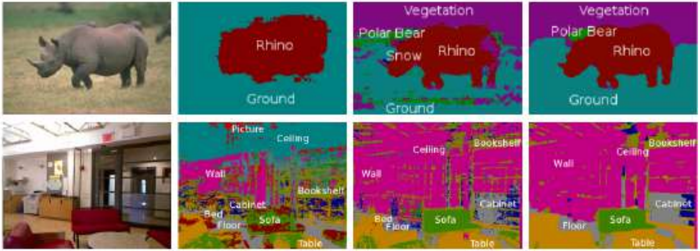
- IR image = greylevel image:
  $u : \Omega \to \mathbb{R}; (x, y) \mapsto u(x, y)$
- Depth image:
  $u : \Omega \to \mathbb{R}; (x, y) \mapsto u(x, y)$

Different types of images: X-ray Scanners

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

X-ray image = sinogram:
$u : [0, 2\pi] \times [0, 1] \to \mathbb{R}; (\theta, \rho) \mapsto u(\theta, \rho)$

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 - see also video for more recent results)

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**)

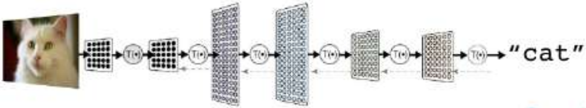
(image source: copyme 3D by J. STURM et al., GCPR 2013 – see video)

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

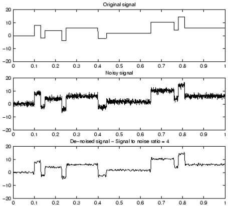
Provide the machine with annotated data;
Let it "learn" what a cat is



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	Variational methods
• AI-oriented	• Mathematics-oriented
• Not clear why it works	• Guarantee of optimality
• Human tells the computer the solution	• Human tells the computer how to solve
• Requires heavy computational power	• Usually much more efficient
• Natural framework for classification	• Natural framework for inverse problems
• Community growing since 2012	• Community reducing since 2012

This lecture: **variational methods**

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

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:

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

Key issues

- How to choose $\mathcal{L}$?
- Is there a solution? Unique?
- How to discretize and solve the optimization problem?

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)

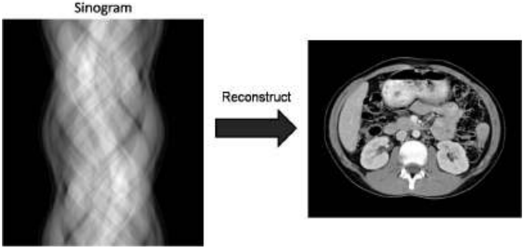
Find an image $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ “close to” the compressed image $u_0 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ on $\overline{\Omega} \subset \Omega$, but “smooth elsewhere”:

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

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

(image source: normal integration by Y. QUÉAU et al., JMIV 2018)

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 $u_0 : [0,1] \times [0,2\pi] \to \mathbb{R}$

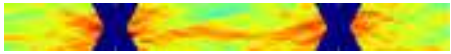
$$\min_{u:\,\Omega\subset\mathbb{R}^2\to\mathbb{R}} \iint_{(x,y)\in\Omega} \underbrace{|R(u)(x,y) - u_0(x,y)|^2}_{\text{"matches sinogram"}} + \lambda \underbrace{\|\nabla u(x,y)\|^2}_{\text{"smooth"}} \, \mathrm{d}x\mathrm{d}y$$

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

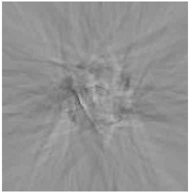
All these tools can be combined in a big variational problem if needed. E.g., joint reconstruction, inpainting and segmentation for Synchrotron X-ray tomography:

Max IV synchrotron

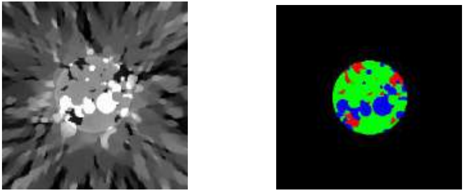
Acquisition device



Sinogram



Reconstruction only

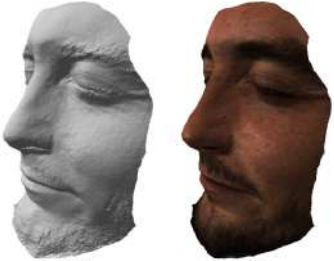


Reconstruction + Segmentation + Inpainting

(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)



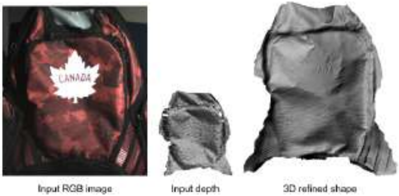
(d) (e)

(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 $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$:

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

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



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

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

$$\min_{u:\,\Omega\subset\mathbb{R}^2\to\mathbb{R}} \underbrace{\iint_{(x,y)\in\Omega_{LR}} |Ku(x,y) - u_0(x,y)|^2 \,\mathrm{d}x\mathrm{d}y}_{\text{"close to"}}$$
$$+ \lambda \underbrace{\iint_{(x,y)\in\Omega_{HR}} \|\mathbf{a}(x,y)\cdot\nabla u(x,y) - I(x,y)\|^2}_{\text{"matches"}}$$

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:

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

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?

1. Introduction to Computer Vision

2. Two Different Paradigms for Computer Vision

3. Introduction to Variational Methods

4. **A Bit of History**

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 ($\Rightarrow$ Euler-Lagrange equation in 1766) :

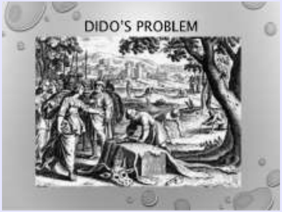
$$\frac{\partial \mathcal{L}}{\partial u} - \frac{d}{dx} \left(\frac{\partial \mathcal{L}}{\partial u'} \right) = 0$$

- 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** :

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

  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,...
- 1980-... : Imaging problems revisited by mathematicians