



# M2 - Optimization and inference techniques for CV

Team 7

José Manuel Lopez Camuñas, Marcos V. Conde, Alex Martin Martinez



# INDEX

- What is optimization in Computer Vision?
- How can we solve Computer Vision problems with optimization?
- Optimization-based Image Segmentation - W3
- 3D Shape Modeling and Reconstruction
- Discussion and Conclusions

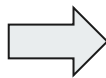
# What is optimization in CV?

Optimization in mathematics:

Optimization in Computer Vision:

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq b_i, \quad i = 1, \dots, m \end{aligned}$$

- $x = (x_1, \dots, x_n)$ : optimization variables
- $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ : objective function
- $f_i : \mathbf{R}^n \rightarrow \mathbf{R}, i = 1, \dots, m$ : constraint functions



Given the image  $f \in L^\infty(\Omega)$  solve

$$u = \arg \min_{u \in W^{1,2}(\Omega)} \left\{ \int_{\Omega} |\nabla u|^2 dx dy + \frac{1}{2\lambda} \int_{\Omega} |u - f|^2 dx dy \right\}$$

$$\min_u \int_{\Omega} |Du| + \frac{\lambda}{2} \|k * u - f\|_2^2$$



(a) Degraded image  $f$



(b) Reconstructed image  $u$

**solution** or **optimal point**  $x^*$  has smallest value of  $f_0$  among all vectors that satisfy the constraints



## What is optimization in CV?

- A method to find the best possible solution to a problem with a well-defined objective and criterias (continuity, smoothness, size, similarity, etc).
- Optimization appears in many computer vision and image processing problems such as **image restoration** (denoising, inpainting, compressed sensing), **multi-view reconstruction**, shape from X, object detection, **image segmentation**, optical flow, matching, and **network training**. While there are formulations allowing for global optimal optimization (e.g. using convex objectives), many problems in computer vision and image processing require efficient approximation methods.



# Solving Computer Vision problems with optimization

1. Define (and understand) what we want to solve: denoise an image? inpaint? segmentation contour? deblur an image?
2. Define the input and output: do we want to obtain an image from an image? This will define our **domain**.
3. Decide the **criteria** to solve the problem. How is the solution? constraints? We need to express the criteria with maths!
4. Depending on our criteria and functions (inputs), we decide **the method** to find the optimal solution. Is a convex problem? non-convex? non-differentiable?
5. Implement the method, experiment and pray to find the best possible solution.

# Week 3: Image segmentation - The problem

Segmentation is a very active research area. The complexity of this problem is the fact that it is an “ill-posed” problem as it can have more than one final solution. For this reason, it is complicated to compare methods. Most methods are evaluated by comparing their results with ground-truth human annotations, however, different persons might segment an image in different ways.



Published in Image Processing On Line on 2012-09-08.  
Submitted on 2012-00-00, accepted on 2012-00-00.  
ISSN 2105-1232 © 2012 IPOL & the authors CC-BY-NC-SA  
This article is available online with supplementary materials,  
software, datasets and online demo at  
<http://dx.doi.org/10.5201/ipol.2012.g-cv>

## Active Image Segmentation Propagation

Suyog Dutt Jain      Kristen Grauman  
University of Texas at Austin

## Laplacian Coordinates for Seeded Image Segmentation

Wallace Casaca<sup>1</sup>,      Luis Gustavo Nonato<sup>1</sup>,      Gabriel Taubin<sup>2</sup>  
<sup>1</sup>ICMC, University of São Paulo, São Carlos, Brazil  
<sup>2</sup>School of Engineering, Brown University, Providence, United States

## Chan–Vese Segmentation

Pascal Getreuer

## Fully Convolutional Networks for Semantic Segmentation

Jonathan Long\*      Evan Shelhamer\*      Trevor Darrell  
UC Berkeley

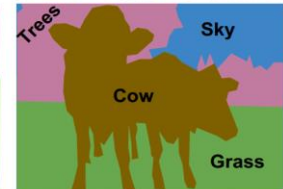
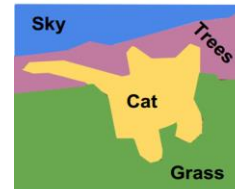
## Superpixel Segmentation using Linear Spectral Clustering

Zhengqin Li      Jiansheng Chen  
Department of Electronic Engineering, Tsinghua University, Beijing, China

## Week 3: Image segmentation - The problem

Image Segmentation is one of the main problems in Computer Vision with applications such as scene understanding, medical image analysis, robotic perception, among many others.

It involves partitioning the image into multiple segments (objects), and each one is meaningful.





## Week 3: Image segmentation - Criteria

1. *The result of the segmentation is an image*

$$f \rightarrow u$$

1. *The segmentation image is similar to the original*

$$\int_{\Omega} (f(x) - u(x))^2 dx$$

1. *The segmented regions are homogeneous*

$$\int_{\Omega \setminus C} |\nabla u(x)|^2 dx$$

1. *The regions have smooth boundaries*

$$\arg \min_{u, C} \mu \text{Length}(C)$$

1. *The segmentation image has two regions*

$$u(x) = \begin{cases} c_1 & \text{where } x \text{ is inside } C, \\ c_2 & \text{where } x \text{ is outside } C, \end{cases}$$



## Week 3: Image segmentation - Criteria

These 4 criteria constitute the **Simplified Mumford-Shah model**

$$\arg \min_{u, C} \mu \text{Length}(C) + \lambda \int_{\Omega} (f(x) - u(x))^2 dx + \int_{\Omega \setminus C} |\nabla u(x)|^2 dx,$$

In this minimization problem The Mumford-Shah approximation suggests selecting this edge set  $C$  as the segmentation boundary. This minimization problem is non-convex [1].

$f$



Mumford-Shah  
piecewise-smooth approximation



[1] Pascal Getreuer, "Chan-Vese Segmentation", IPOL 2012

## Week 3: Image segmentation - Energy function

- Adding up all the different criteria for the segmentation we obtain an **energy function**.
- We have the hyper-parameters  $\mu$  (*length penalization*),  $\nu$  (*area inside C penalization*),  $\lambda_1$  and  $\lambda_2$  (*fidelity weight terms*).

$$\begin{aligned} \arg \min_{c_1, c_2, C} & \mu \text{Length}(C) + \nu \text{Area}(\text{inside}(C)) \\ & + \lambda_1 \int_{\text{inside}(C)} |f(x) - c_1|^2 dx + \lambda_2 \int_{\text{outside}(C)} |f(x) - c_2|^2 dx. \end{aligned}$$

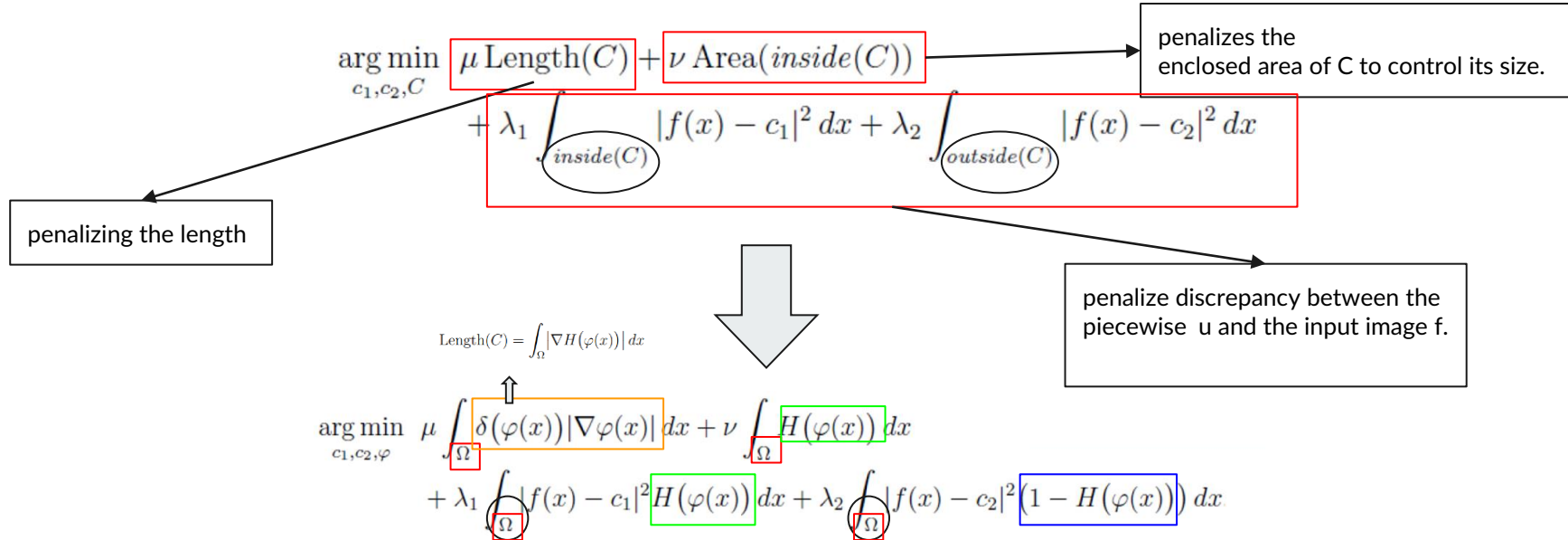
## Week 3: Image segmentation - Energy function

To unify the regions the **Heaviside function** will be added to change the regions of the integral to  $\Omega$ , as the function will be 1 in the inside of the contour and zero in the outside.

$$\arg \min_{c_1, c_2, C} (\mu \text{Length}(C) + \nu \text{Area}(\text{inside}(C))) \\ + \lambda_1 \int_{\text{inside}(C)} |f(x) - c_1|^2 dx + \lambda_2 \int_{\text{outside}(C)} |f(x) - c_2|^2 dx.$$

$$H(t) = \begin{cases} 1 & t \geq 0, \\ 0 & t < 0, \end{cases} \quad \delta(t) = \frac{d}{dt} H(t)$$

# Week 3: Image segmentation - Energy function



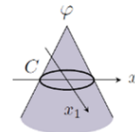
## Week 3: Image segmentation - Level set function

In the previous expression  $\varphi$  is the level set function. Instead of operating with the contour itself the contour is represented as the zero-crossing of the level set function and the inside and outside of the contour  $C$  will be determined by the sign of  $\varphi$  in the region. This allows us to not be restricted by single connected contour, since we can have many contours.

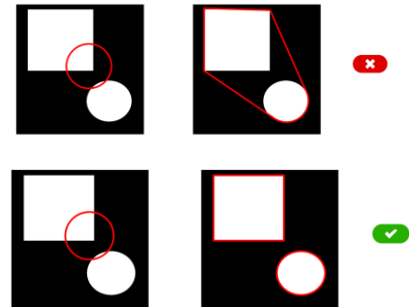
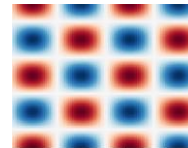
$$C = \{x \in \Omega : \varphi(x) = 0\}.$$

Example of level set functions

$$\varphi(x) = r - \sqrt{x_1^2 + x_2^2}$$



$$\varphi(x) = \sin\left(\frac{\pi}{5}x_1\right) \sin\left(\frac{\pi}{5}y\right)$$



## Week 3: Image segmentation - Implementation

We optimize the energy function using **gradient descent**:

$$\frac{\partial \varphi_{i,j}}{\partial t} = \delta_\epsilon(\varphi_{i,j}) \left[ \mu \left( \nabla_x^- \frac{\nabla_x^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{i,j})^2 + (\nabla_y^0 \varphi_{i,j})^2}} + \nabla_y^- \frac{\nabla_y^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{i,j})^2 + (\nabla_y^+ \varphi_{i,j})^2}} \right) - \nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2 \right], \quad i, j = 1, \dots, M-1,$$

Ghost boundaries:  $\varphi_{i,-1} = \varphi_{i,0}, \quad \varphi_{i,M} = \varphi_{i,M-1}, \quad \varphi_{-1,j} = \varphi_{0,j}, \quad \varphi_{M,j} = \varphi_{M-1,j}$

This iterative process will then be stopped when the tolerance is bigger than the difference between the  $\varphi^n$  and  $\varphi^{n+1}$  or when the maximum number of iterations is surpassed.

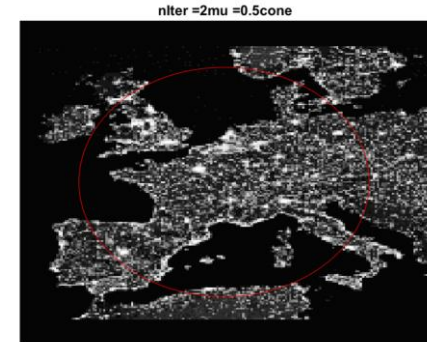
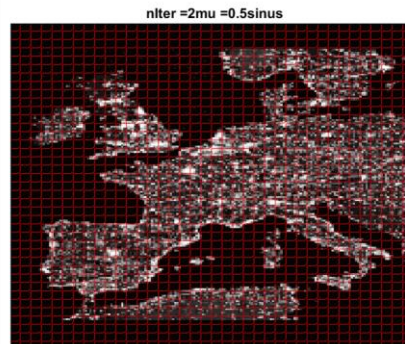
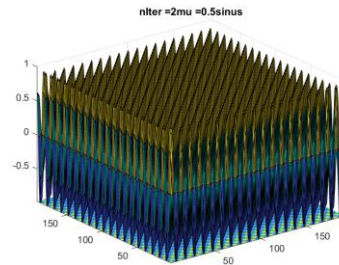
**if**  $\|\varphi^{n+1} - \varphi^n\|_2/|\Omega| \leq tol$  **then stop**  
(Optional) If  $n$  is divisible by  $N$ , reinitialize  $\varphi$

## Week 3: Image segmentation - Experiments

**Variation of the initialization function:** Different  $\phi$  functions resulted in faster/slower segmentations. In our experiments, the checkerboard function is “always” faster than the cone function, this is because it covers most of the space from the beginning. In other cases the cone function couldn’t converge into a solution while the checkerboard could.

Parameters used:

$\epsilon=1$   
 $\eta=10^{-2}$   
 $dt=0.5$   
 $tol=0.01$   
 $\mu=0.5$   
 $\nu=0$   
 $relni=100$



## Week 3: Image segmentation - Experiments

**Variation of  $\mu$ :** The parameter  $\mu$  is the most important, this adjusts the length penalty, which balances between fitting the input image more accurately (smaller  $\mu$ ) vs. producing a smoother boundary (larger  $\mu$ ).

$$\arg \min_{u, C} \boxed{\mu \text{Length}(C)} + \lambda \int_{\Omega} (f(x) - u(x))^2 dx + \int_{\Omega \setminus C} |\nabla u(x)|^2 dx,$$

Parameters used:

$\varepsilon=1$

$\eta=10^{-8}$

$dt=0.5$

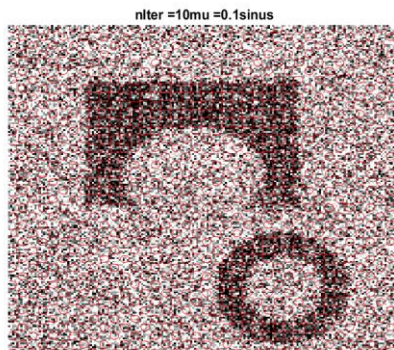
$tol=0.01$

**$\mu=0.1$**

$\nu=0$

$relni=100$

$\phi$ : Checkerboard



Parameters used:

$\varepsilon=1$

$\eta=10^{-8}$

$dt=0.5$

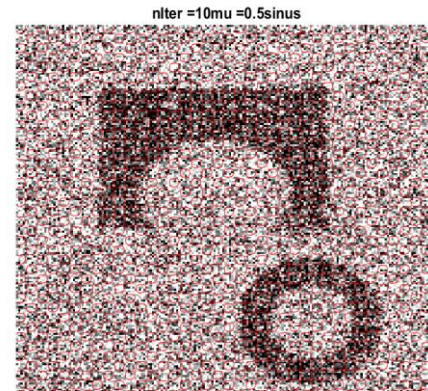
$tol=0.01$

**$\mu=0.5$**

$\nu=0$

$relni=100$

$\phi$ : Checkerboard

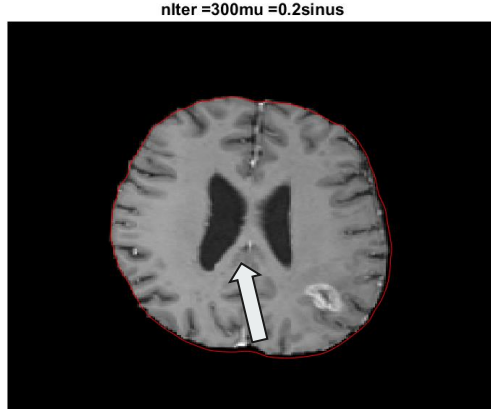




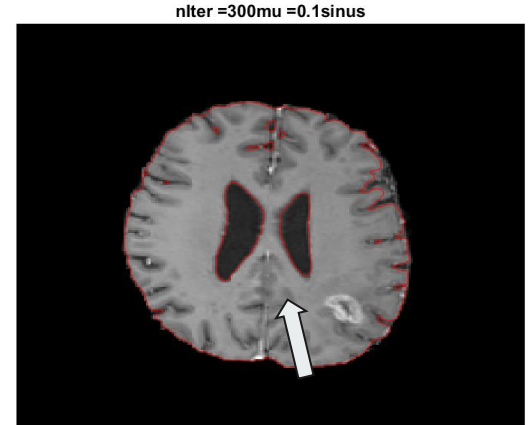
## Results: Variation of the $\lambda_1$ and $\mu$

Here we variated the  $\lambda_1$  and  $\mu$ . Decreasing the  $\mu$  and increasing the  $\lambda_1$  allows to detect the inner holes of the brain and some wrinkles from the exterior.

Parameters used:  
 $\varepsilon=1$   
 $\eta=10^{-2}$   
 $dt=0.5$   
 $tol=0.01$   
 **$\mu=0.2$**   
 **$\lambda_1=1$**   
 $\nu=0$   
 $relni=100$   
 $\varphi$ : Checkerboard



Parameters used:  
 $\varepsilon=1$   
 $\eta=10^{-2}$   
 $dt=0.5$   
 $tol=0.01$   
 **$\mu=0.1$**   
 **$\lambda_1=3$**   
 $\nu=0$   
 $relni=100$   
 $\varphi$ : Checkerboard





# 3D Shape Modeling and Reconstruction



This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.  
Except for this watermark, it is identical to the accepted version;  
the final published version of the proceedings is available on IEEE Xplore.

## Deep Optimized Priors for 3D Shape Modeling and Reconstruction

Mingyue Yang<sup>1\*</sup>, Yuxin Wen<sup>1\*</sup>, Weikai Chen<sup>2</sup>, Yongwei Chen<sup>1</sup>, Kui Jia<sup>134†</sup>

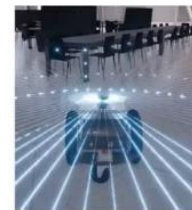
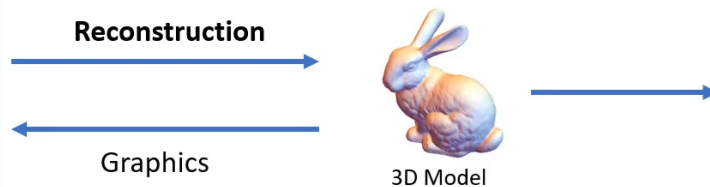
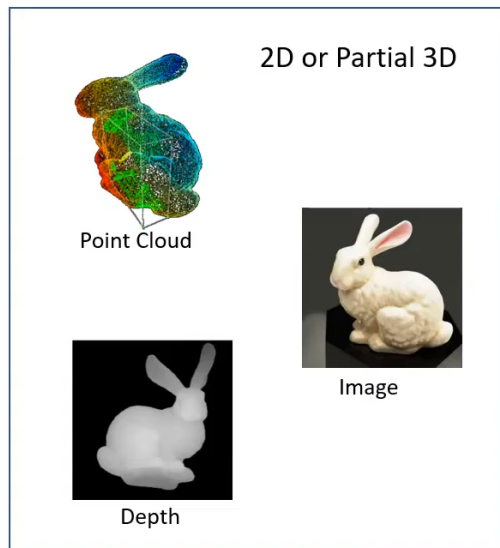
<sup>1</sup>South China University of Technology, <sup>2</sup>Tencent Game AI Research Center

<sup>3</sup>Pazhou Laboratory, <sup>4</sup>Peng Cheng Laboratory

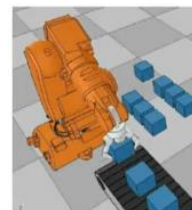
{eemingyueyang, wen.yuxin}@mail.scut.edu.cn,

chenwk891@gmail.com, eecyw@mail.scut.edu.cn, kuijia@scut.edu.cn

# 3D Shape Modeling and Reconstruction



Robotics



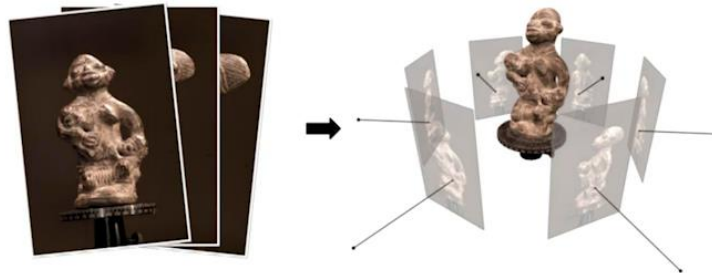
Simulation



Content Creation

# 3D Shape Modeling and Reconstruction

How to reconstruct a 3D surface from 3 views?  
(an ill-posed problem)




[Furukawa & Hernandez: Multi-View Stereo: A Tutorial]

## Task:

- ▶ Given a set of 2D images
- ▶ Reconstruct 3D shape of object/scene

# 3D Shape Modeling and Reconstruction

- Deep Learning models lack of the ability of representing fine surface details of unknown samples as they are training-dependant.
  - Our goal is to generate or reconstruct a faithful 3D surface  $O$  from the input physical observations by leveraging the priors encoded  $f_{\theta}(\mathbf{x}, \mathbf{z})$ .
  - query 3D point  $\mathbf{x}$  and  $\mathbf{z}$  latent code,  $f_{\theta}(\mathbf{x}, \mathbf{z})$ .  $\rightarrow$  target surface
  - Formally, a general 3D modeling problem, can be formulated as follows:
- 

$$\hat{\mathcal{O}}^* = \arg \min_{\hat{\mathcal{O}}} E(\hat{\mathcal{O}}; \mathcal{O}) + R(\hat{\mathcal{O}}), \quad (1)$$

$$\mathbf{z}^*, \boldsymbol{\theta}^* = \arg \min_{\mathbf{z}, \boldsymbol{\theta}} \boxed{E(f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}); \mathcal{O})} + \boxed{R(f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}))}, \quad (2)$$

# 3D Shape Modeling and Reconstruction

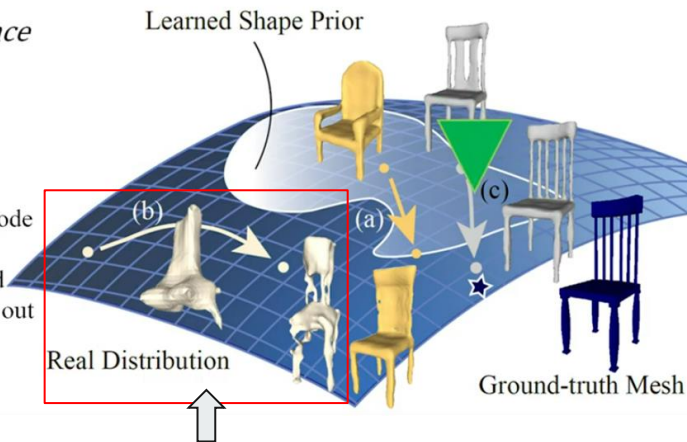
$$\hat{\mathcal{O}}^* = \arg \min_{\hat{\mathcal{O}}} E(\hat{\mathcal{O}}; \boxed{\mathcal{O}}) + R(\hat{\mathcal{O}})$$

$\mathcal{O}$ : 3D Surface

- Path - (c): Our method

Jointly optimize pre-trained prior and latent code according to input measurements.

→ effectively break the barriers of pre-trained prior and generalize to the unseen data that is out of the prior domain



(b) Optimizing a randomly initialized generator, on the other hand, is prone to be trapped in a local minimum due to the complex energy landscape.



# 3D Shape Modeling and Reconstruction

## Applications: Shape auto-encoding

Our goal is to generate an implicit field as a faithful approximation of the input surface  $S$ .

In this application, we represent 3D locations as a set of pairs  $\{(\mathbf{p}_i, s_i)\}_{i=1}^n$  where  $\mathbf{p}$  is the coordinates in the space and  $s$  the corresponding distance value. The reconstruction energy term is defined as:

$$E(f_{\theta}(\mathbf{z}); \mathcal{X}) = \sum_{i \in \{1, \dots, n\}} \|\hat{s}_i - s_i\|_1,$$

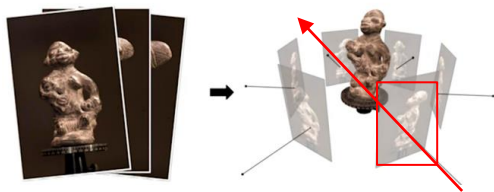
Where  $s_i$  is the ground-truth distance;  $\hat{s}_i$  denotes the estimated signed distance value for the point  $i$ -th predicted by the deep learning model  $f_{\theta}(\mathbf{z}, \mathbf{p}_i)$ . The regularization term will be the same as the one used in the multiview

# 3D Shape Modeling and Reconstruction

## Applications: Multi-view reconstruction

In this application, we have images from multiple views  $I$  and object silhouettes masks  $M$ . The aim of multi-view reconstruction is to recover the underlying object surface from these partial observations of  $n$  views. The energy term is formulated as:

$$E(f_{\theta}(z); \mathcal{X}) = \sum_{i=1}^n (\|\hat{I}_i - I_i\|_1 + \lambda_c \cdot \mathcal{L}_c(\hat{M}_i - M_i))$$



For the images from different views to be similar to the taken ones

Penalizes mismatched object silhouettes.



# 3D Shape Modeling and Reconstruction

## Applications: Multi-view reconstruction

In the presence of highly sparse views, the multi-view reconstruction task becomes a highly underdetermined problem. Hence, we further introduce additional regularizers on the neural network to ensure plausible results

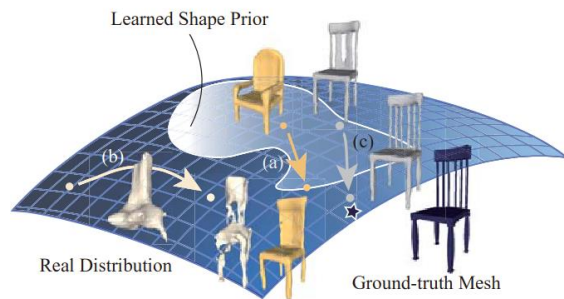
The regularizer term will ensure plausible results in the presence of highly sparse views and it will be defined in as:

$$R(f_{\theta}(z)) = \frac{1}{\sigma^2} \|z\|_2 + \lambda_{\theta} \cdot \|\theta - \theta_0\|_2,$$

Prevents bias in the latent space (zero-mean multivariate Gaussian.)

Prevents parameters from being very different than the originally learned ones

Then we can define the optimization problem as:  $\min_{\theta, z} L(\mathcal{X}) = E(f_{\theta}(z); \mathcal{X}) + \lambda \cdot R(f_{\theta}(z))$



## Optimization Process

Low-res, 3 images  
Same architecture



Path (a): "DeepSDF"



✓ Path (c): Our method

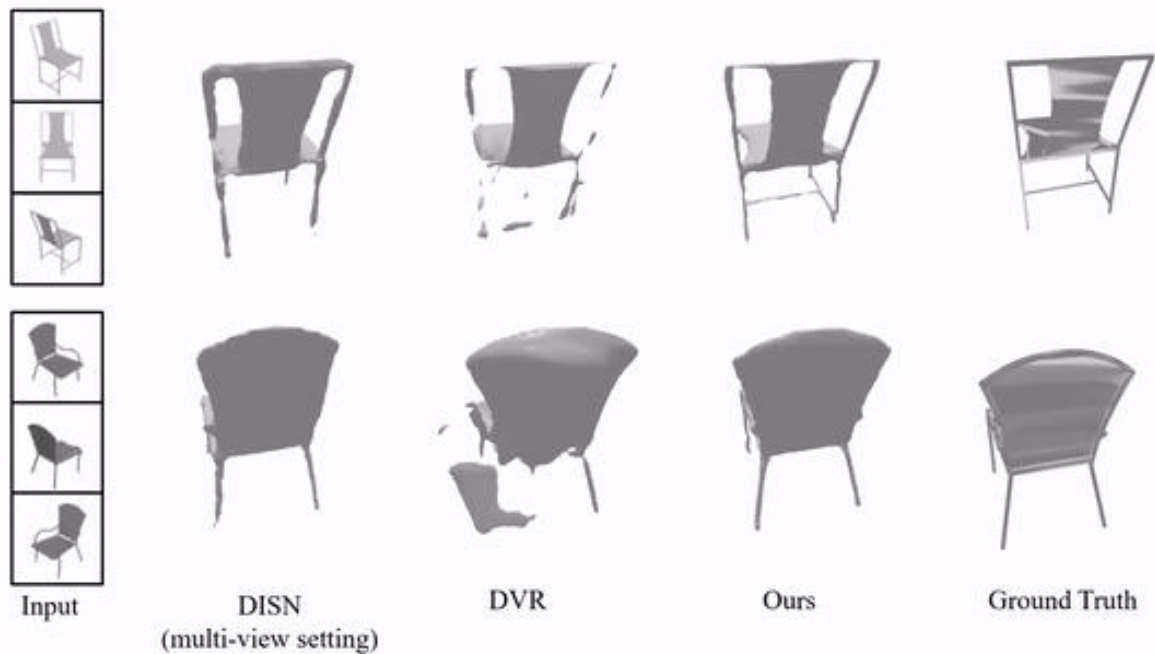


Path (b): "Fitting"



$$E(f_{\delta}^{\times}(\mathbf{z}); \mathcal{X}) = \sum_{i \in \{1, \dots, n\}} \|\hat{s}_i - s_i\|_1,$$

### Sparse multi-view reconstruction





# 3D Shape Modeling and Reconstruction

## Conclusions

- Shows benefits of combining deep learning and optimization (in this case at test time).
- This approach **can generalize significantly better to unseen data.**
- This method **is currently more expensive than alternatives.** It would be an interesting avenue to accelerate the optimization. Not optimal optimization?
- Still lacking a theoretical analysis of the working principle of our approach.



## Discussion and conclusions of optimization in CV

- + Despite Machine learning methods are the *state-of-the-art* in Computer Vision, Optimization methods can also provide high-quality results without the limitations of (i) creating a labeled dataset with ground-truth and (ii) design and train a complex ML model.
- + Optimization-based Computer Vision enables the definition and interpretation of the final results based on the selected criteria, which is something typically lost in the deep learning approaches.
- Optimization does not guarantee the “best” solution always! Moreover, the test-time might not be very competitive, and hyper-parameters are usually modified “empirically” after trial-error.
- + Optimization underlies Deep Learning hidden behind the “loss function to optimize” concept.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$