IsMo-GAN: Adversarial Learning for Monocular Non-Rigid 3D Reconstruction

Soshi Shimada 1,2, Vladislav Golyanik 2,3, Christian Theobalt 3 , and Didier Stricker 1,2

1. Augmented Vision, DFKI 2. University of Kaiserslautern 3. MPI for Informatics

Motivation

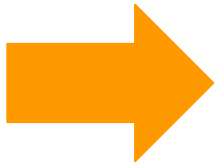
Motivation

- 3D reconstruction of a deformable object from monocular 2D image sequences is still a challenging problem

Motivation

- 3D reconstruction of a deformable object from monocular 2D image sequences is still a challenging problem

2D RGB image



3D point clouds

Related works

Related works

- Non Rigid Structure from Motion (NRSfM)

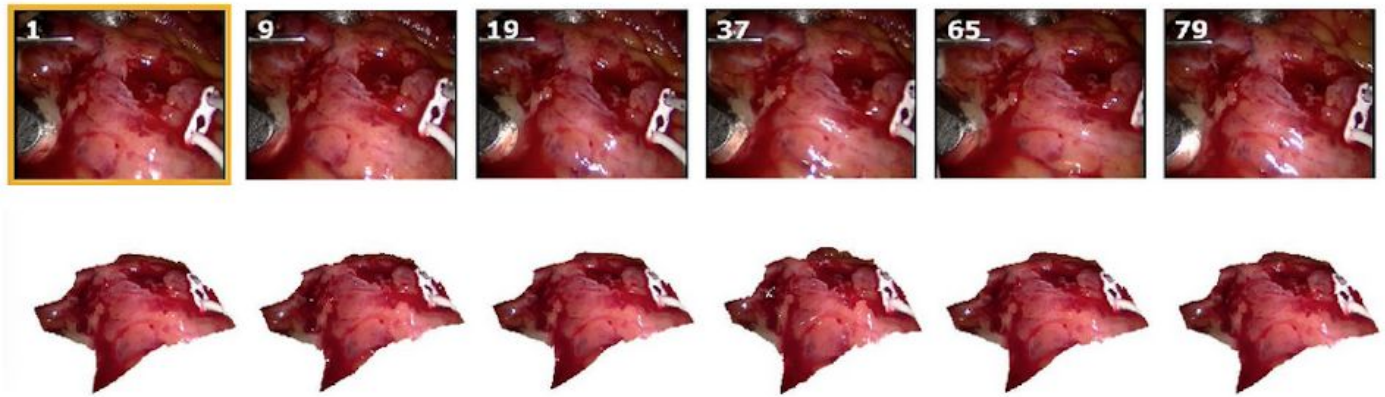


Figure1. NRSfM technique (Golyanik et al. , 2017)

Related works

- Non Rigid Structure from Motion (NRSfM)
  - input: point tracks on 2D frames

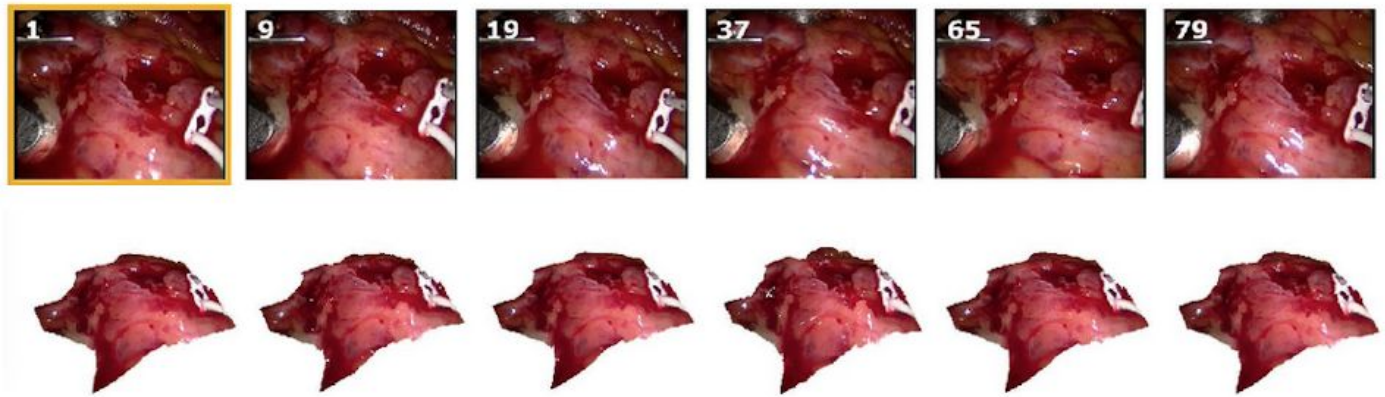


Figure1. NRSfM technique (Golyanik et al. , 2017)

Related works

- Non Rigid Structure from Motion (NRSfM)
  - input: point tracks on 2D frames
  - basically no limitation regarding target objects

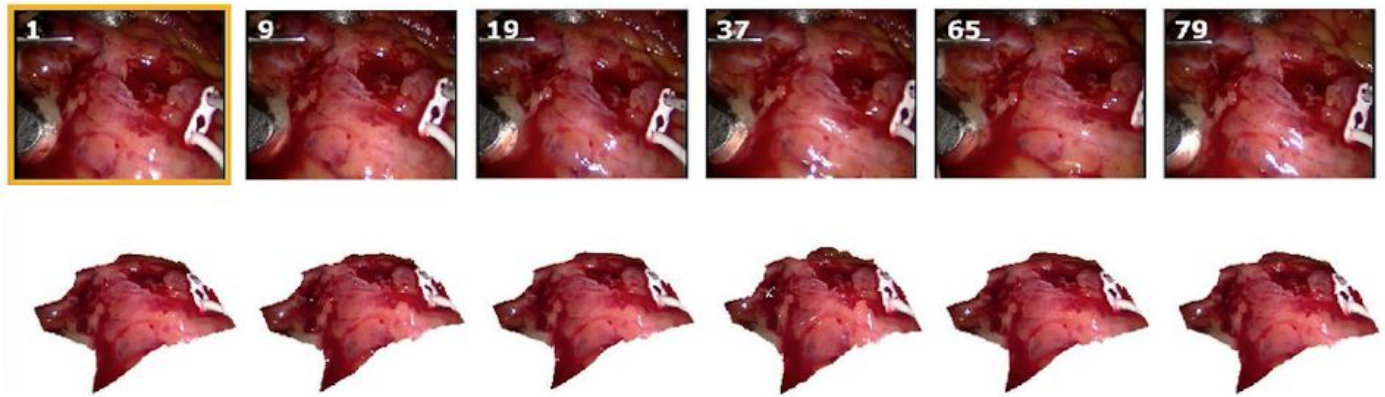


Figure1. NRSfM technique (Golyanik et al. , 2017)

Related works

- Non Rigid Structure from Motion (NRSfM)
  - input: point tracks on 2D frames
  - basically no limitation regarding target objects
  - multiple frames are required

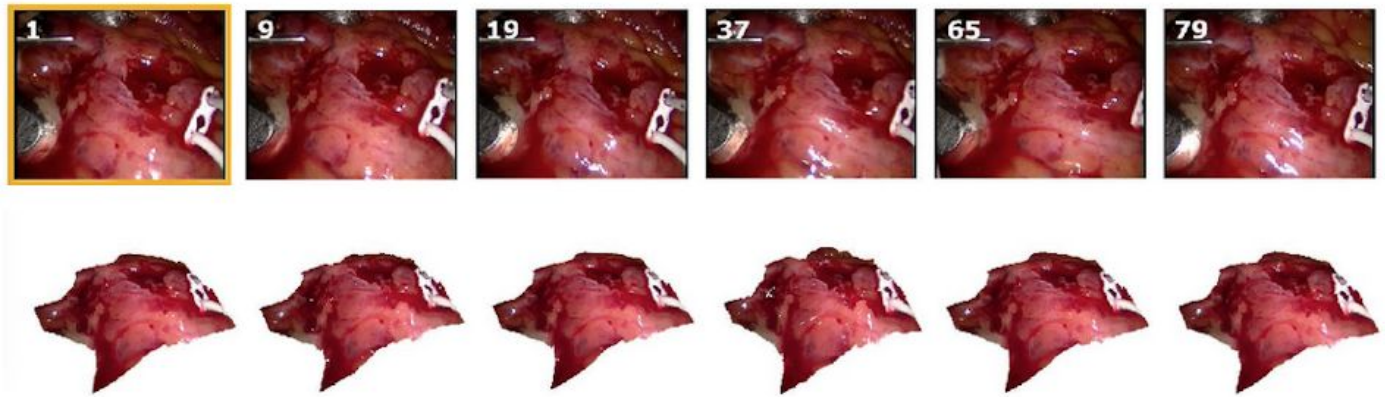


Figure1. NRSfM technique (Golyanik et al. , 2017)

Related works

- Non Rigid Structure from Motion (NRSfM)
  - input: point tracks on 2D frames
  - basically no limitation regarding target objects
  - multiple frames are required
  - difficulty to apply on non-textured objects

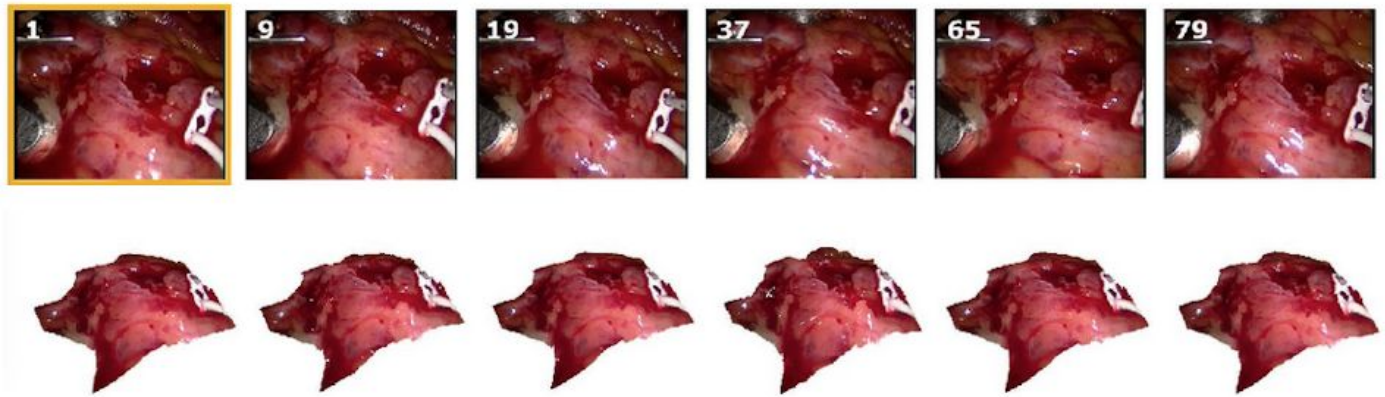


Figure1. NRSfM technique (Golyanik et al. , 2017)

Related works

Related works

- Template based

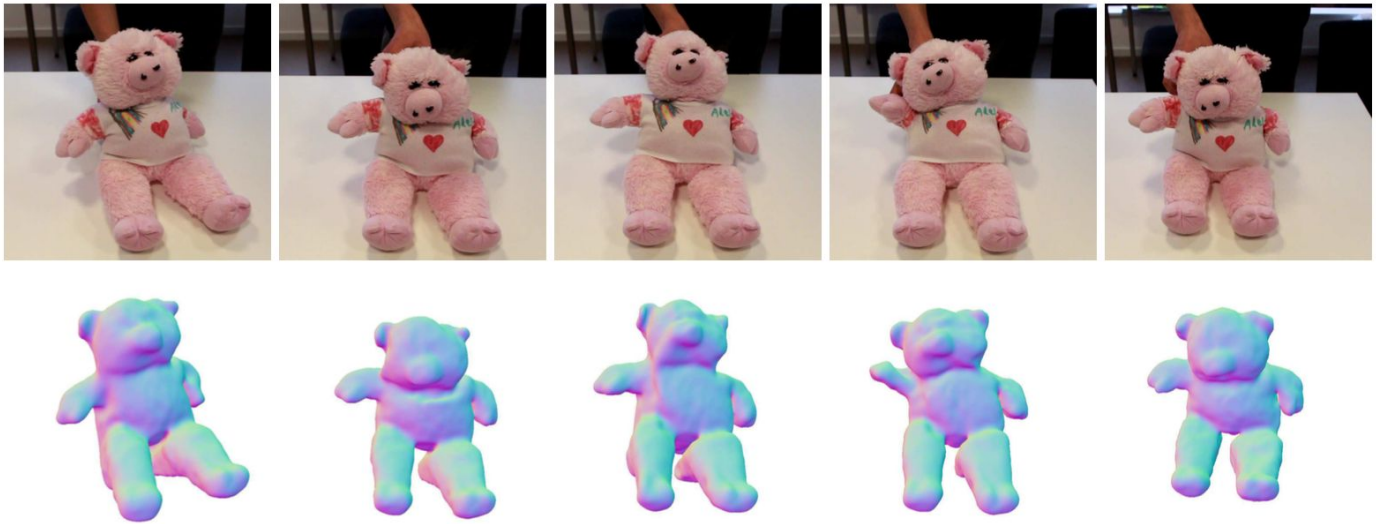


Figure2. 3D reconstruction from a sequence of images (Yu et al,2015)

Related works

- Template based
  - input: 3D template & 2D images

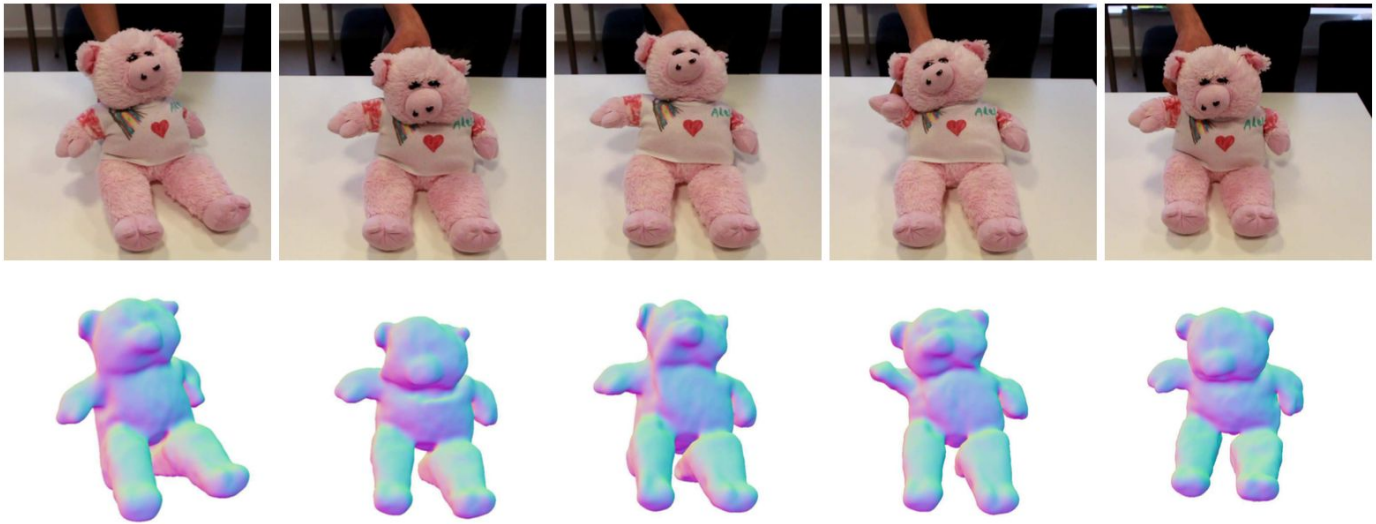


Figure2. 3D reconstruction from a sequence of images (Yu et al,2015)

Related works

- Neural network based

Related works

- Neural network based
  - Input: a single/sequence of images

Related works

- Neural network based
  - Input: a single/sequence of images
  - Output: 3D geometry (Voxel/Point Set/Mesh)

Related works

- Neural network based
  - Input: a single/sequence of images
  - Output: 3D geometry (Voxel/Point Set/Mesh)
  - E.g. HDM-Net

Related works

- Neural network based
  - Input: a single/sequence of images
  - Output: 3D geometry (Voxel/Point Set/Mesh)
  - E.g. HDM-Net

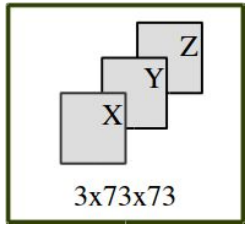
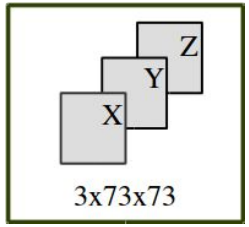


Figure3. HDM-Net (Golyanik, 2018)

Related works

- Neural network based
  - Input: a single/sequence of images
  - Output: 3D geometry (Voxel/Point Set/Mesh)
  - E.g. HDM-Net



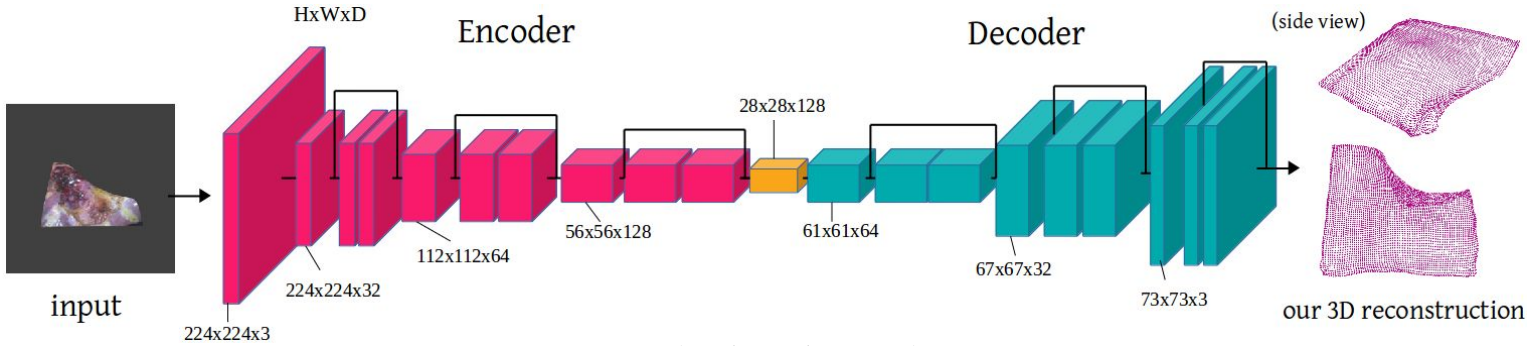
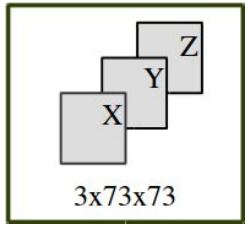


Figure3. HDM-Net (Golyanik, 2018)

  - 3D Reconstruction from a single RGB image

Related works

- Neural network based
  - Input: a single/sequence of images
  - Output: 3D geometry (Voxel/Point Set/Mesh)
  - E.g. HDM-Net



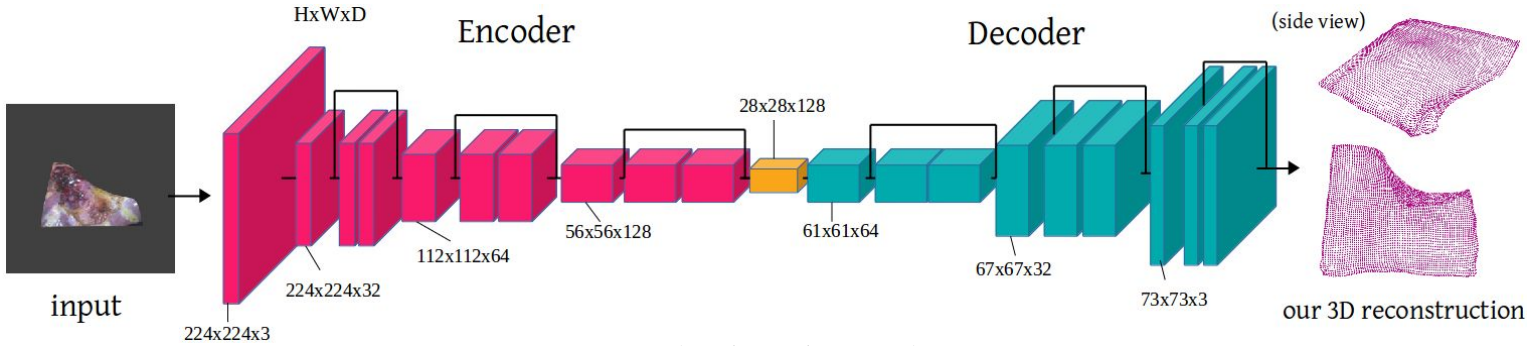
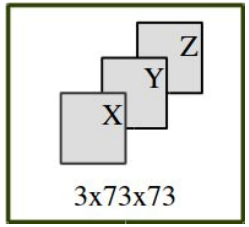


Figure3. HDM-Net (Golyanik, 2018)

  - 3D Reconstruction from a single RGB image
  - Regress 3D coordinates (xyz geometry)

Related works

- Neural network based
  - Input: a single/sequence of images
  - Output: 3D geometry (Voxel/Point Set/Mesh)
  - E.g. HDM-Net



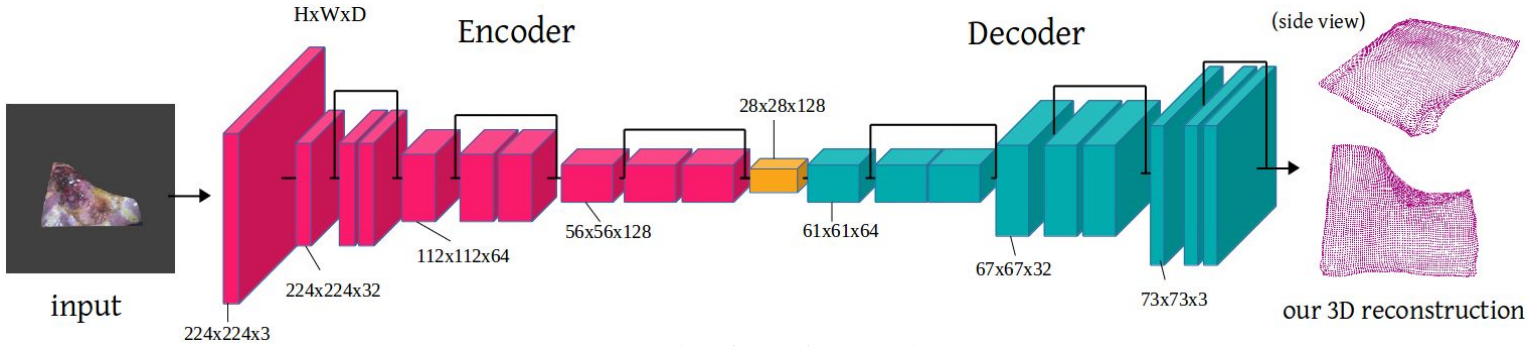


Figure3. HDM-Net (Golyanik, 2018)

  - 3D Reconstruction from a single RGB image
  - Regress 3D coordinates (xyz geometry)
  - Apply 2D conv. not 3D conv.

Limitations of HDM-Net

Limitations of HDM-Net

- In a real-world scenario, our architecture has difficulty to reconstruct geometries especially when

Limitations of HDM-Net

- In a real-world scenario, our architecture has difficulty to reconstruct geometries especially when

  1) the deformation states in the scene is quite different from the ones in the training dataset

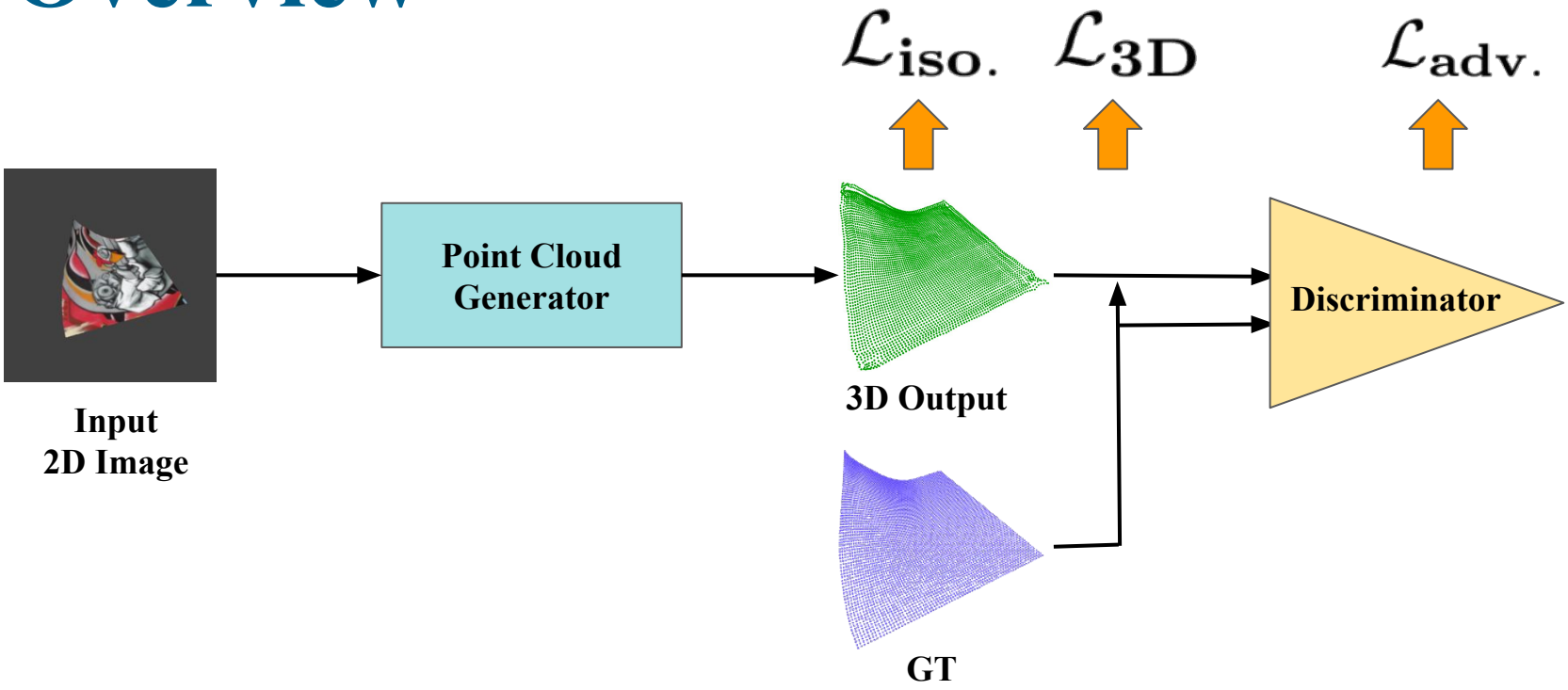
Limitations of HDM-Net

- In a real-world scenario, our architecture has difficulty to reconstruct geometries especially when

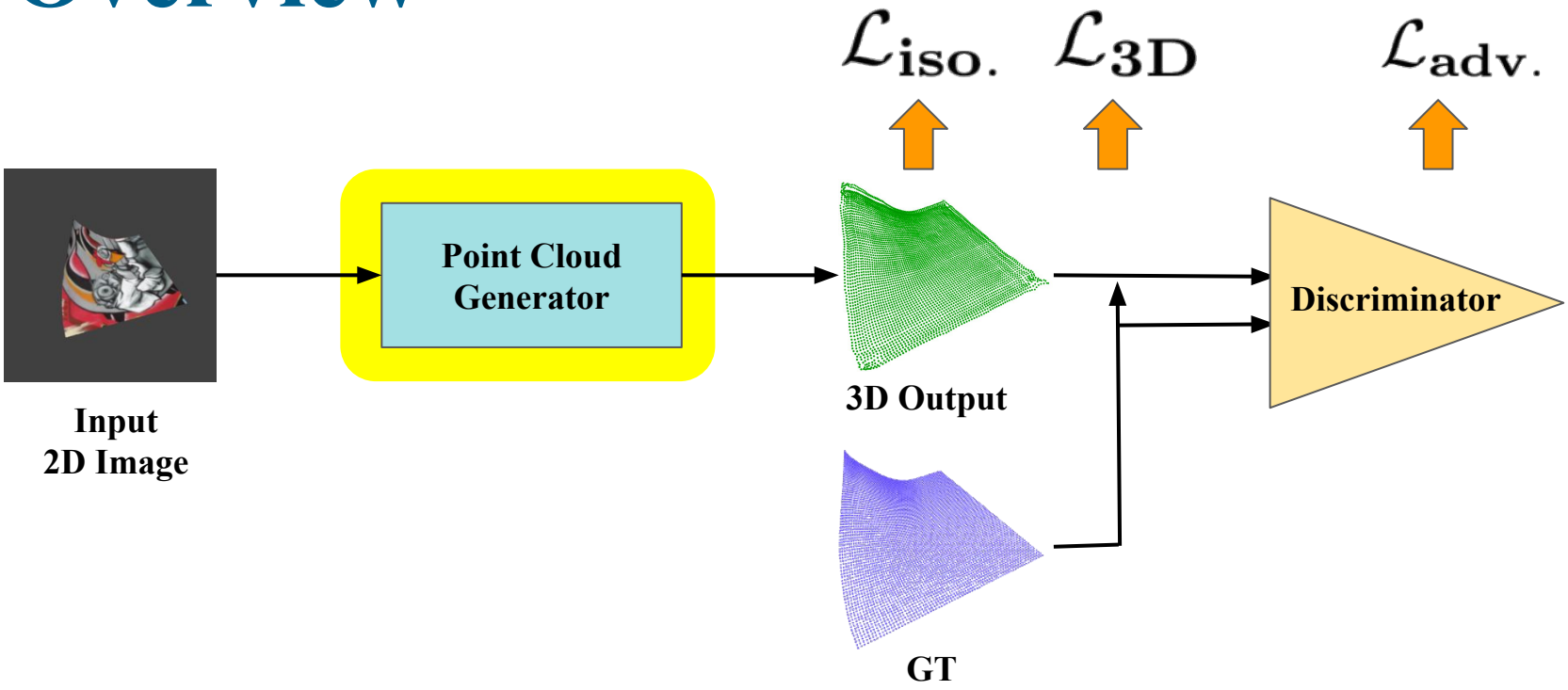
  1) the deformation states in the scene is quite different from the ones in the training dataset
  2) the scene has a complicated background

Overview

Overview


$\mathcal{L}_{iso.}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{adv.}$
Point Cloud Generator
3D Output
Discriminator
Input 2D Image
GT

Overview


$\mathcal{L}_{iso.}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{adv.}$
Point Cloud Generator
Discriminator
3D Output
GT
Input 2D Image

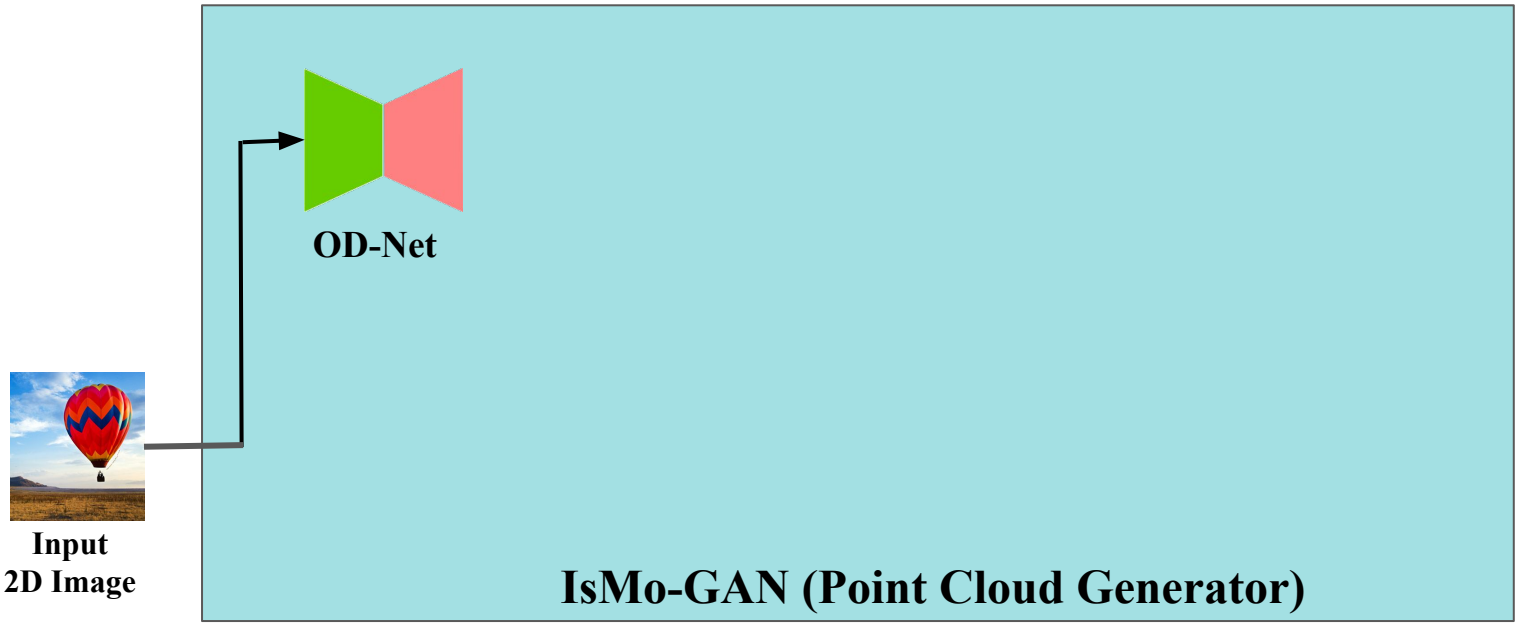
Overview

Overview

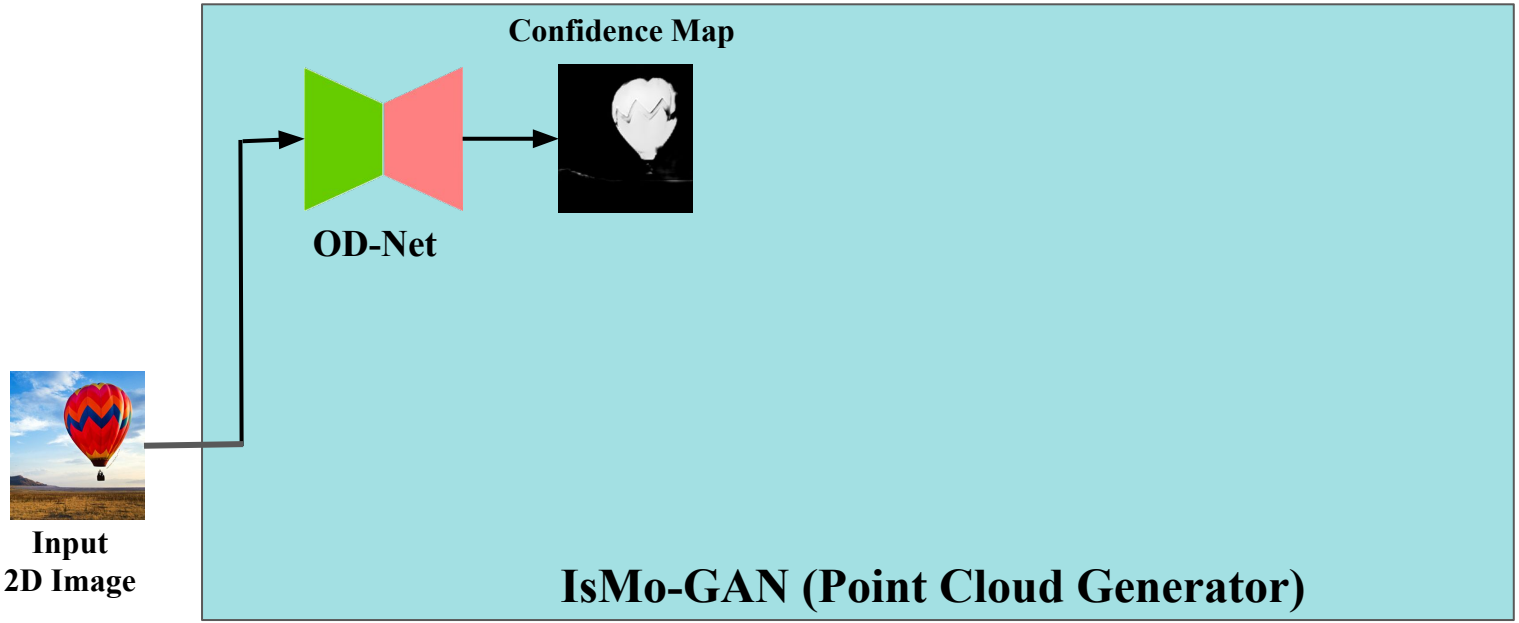
Input
2D Image

IsMo-GAN (Point Cloud Generator)

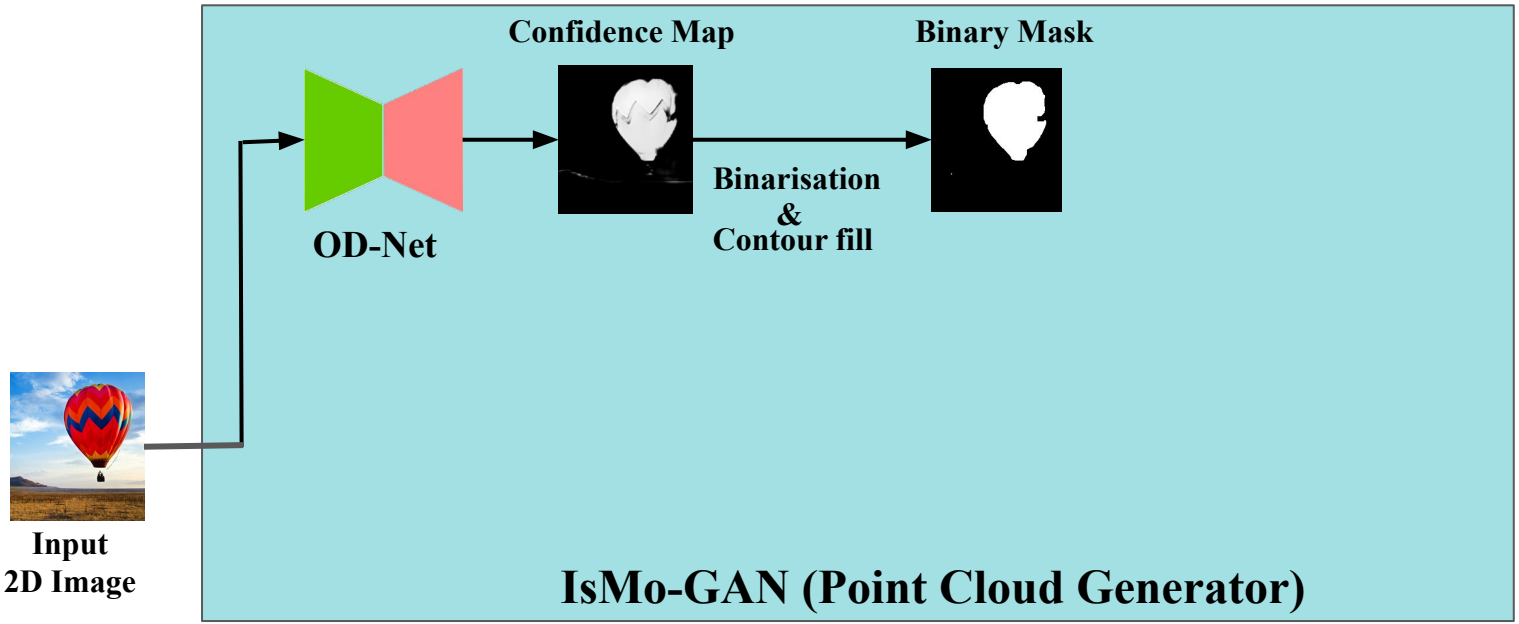
Overview



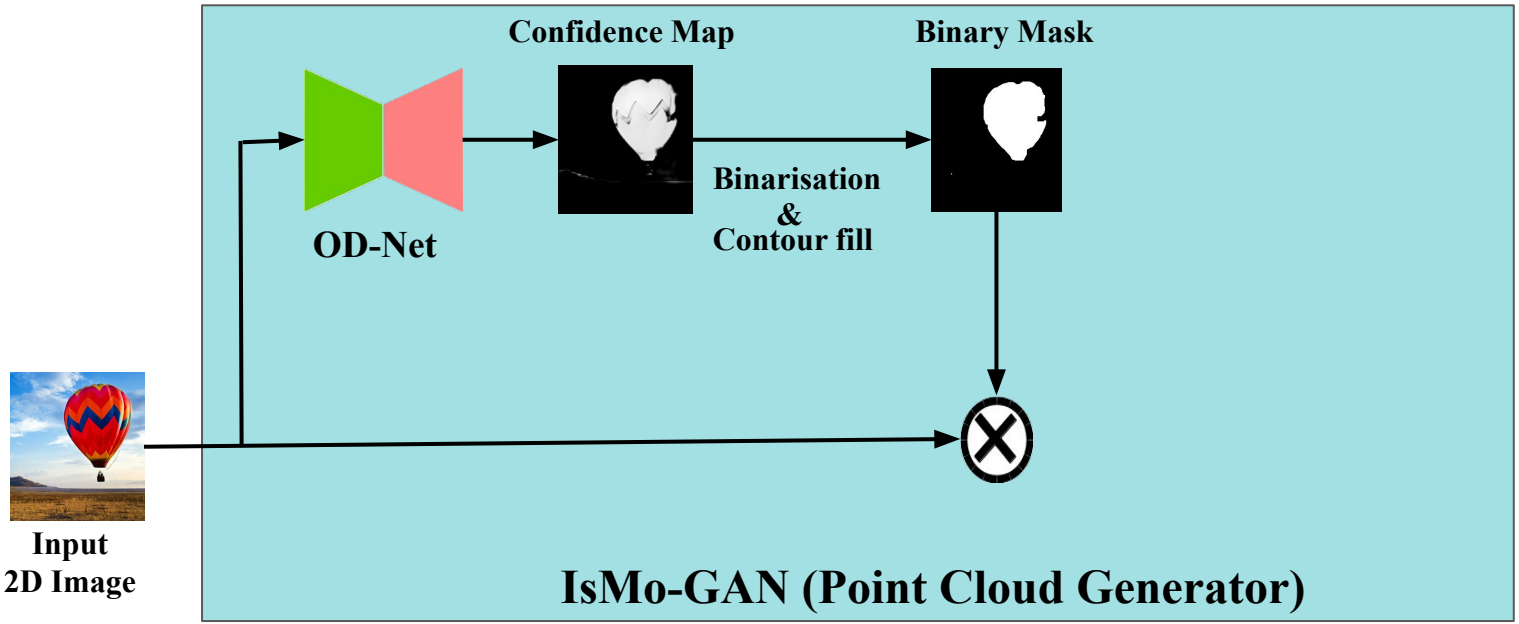
Overview


Confidence Map
OD-Net
Input
2D Image
IsMo-GAN (Point Cloud Generator)

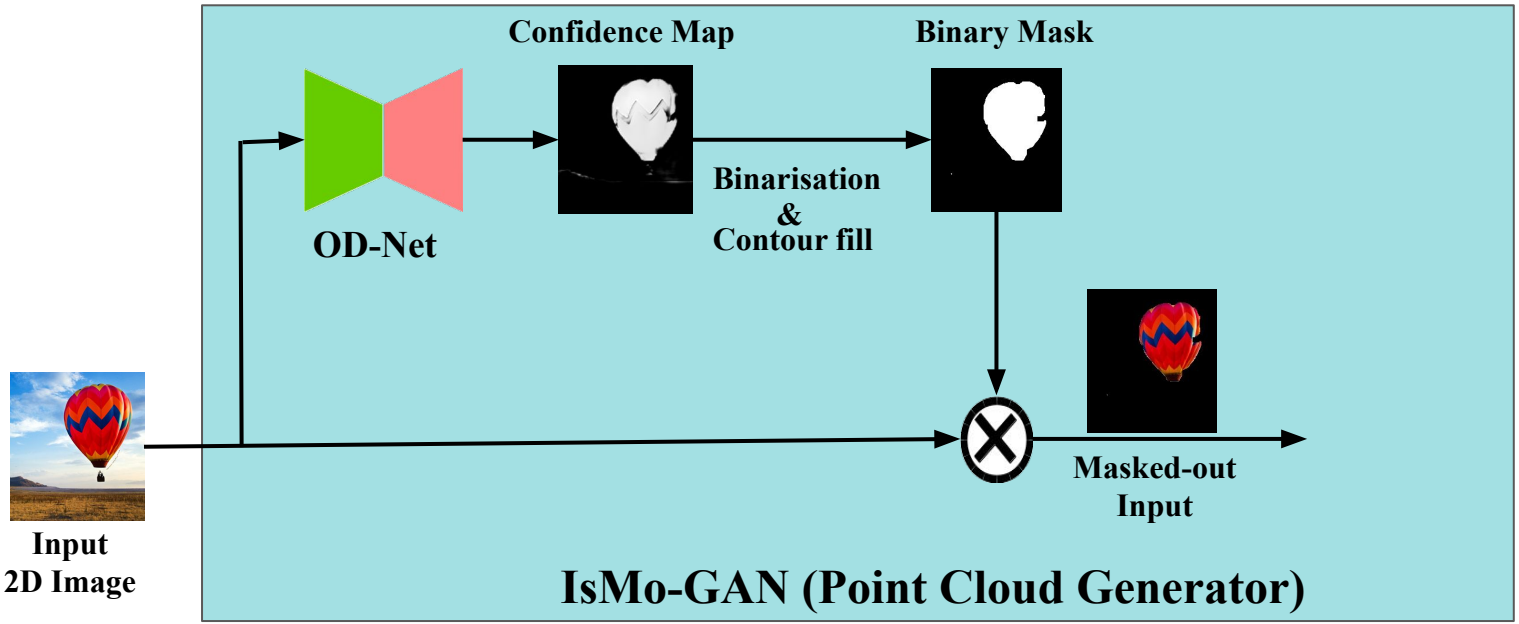
Overview


Confidence Map
Binary Mask
OD-Net
Binarisation
&
Contour fill
Input
2D Image
IsMo-GAN (Point Cloud Generator)

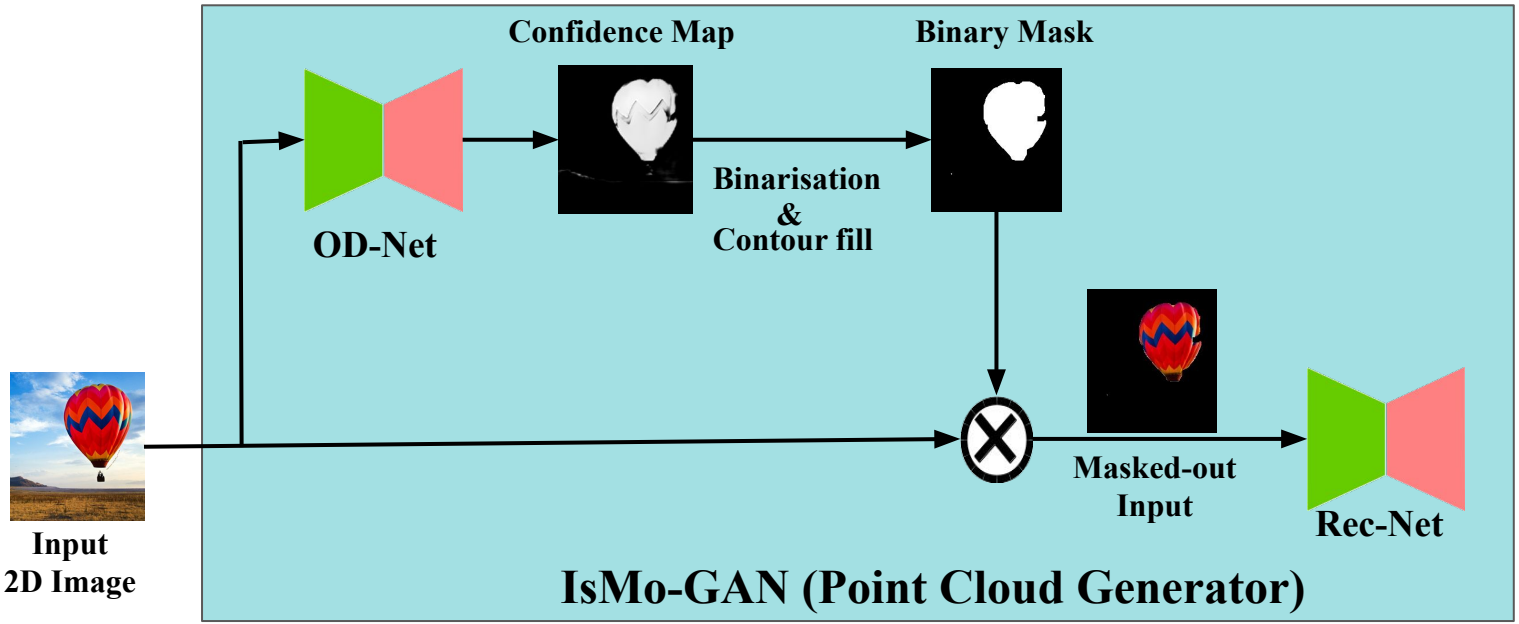
Overview


Confidence Map
Binary Mask
OD-Net
Binarisation
&
Contour fill
Input
2D Image
IsMo-GAN (Point Cloud Generator)

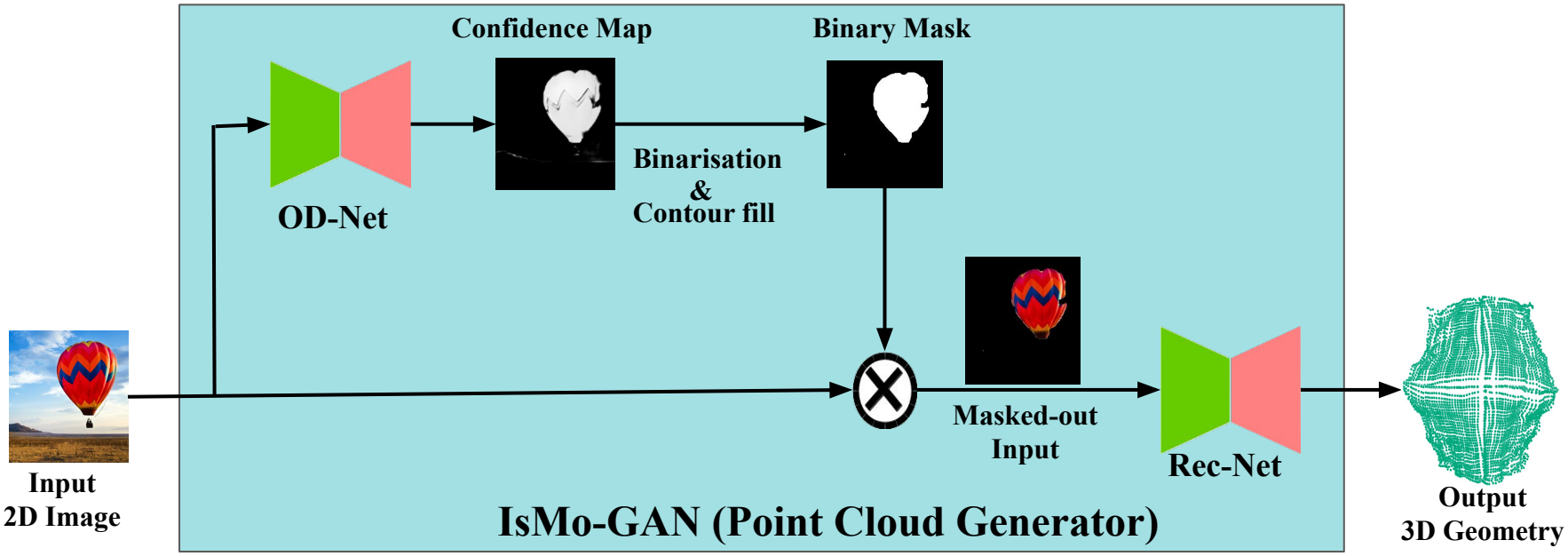
Overview


Confidence Map
Binary Mask
OD-Net
Binarisation
&
Contour fill
Masked-out
Input
Input
2D Image
IsMo-GAN (Point Cloud Generator)

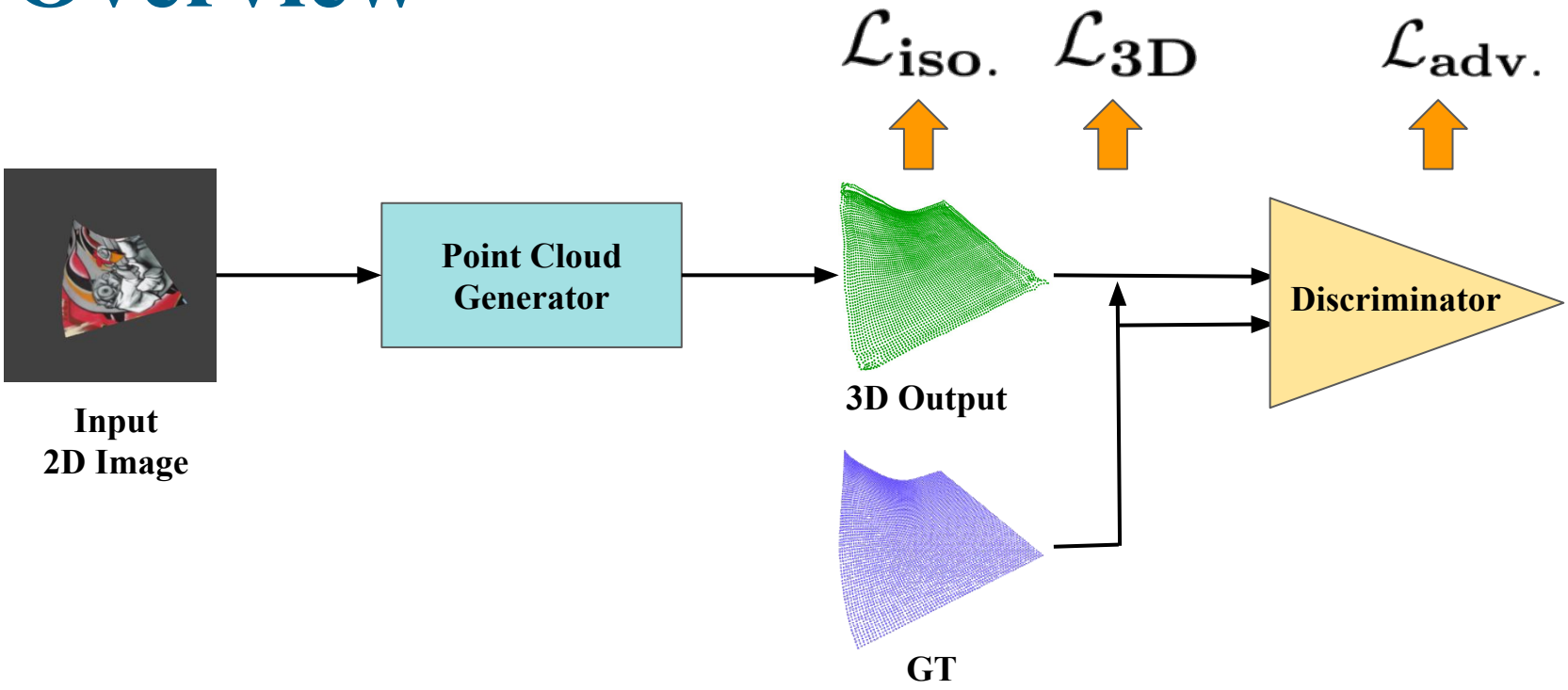
Overview


Confidence Map
Binary Mask
OD-Net
Binarisation
&
Contour fill
Masked-out
Input
Rec-Net
Input
2D Image
IsMo-GAN (Point Cloud Generator)

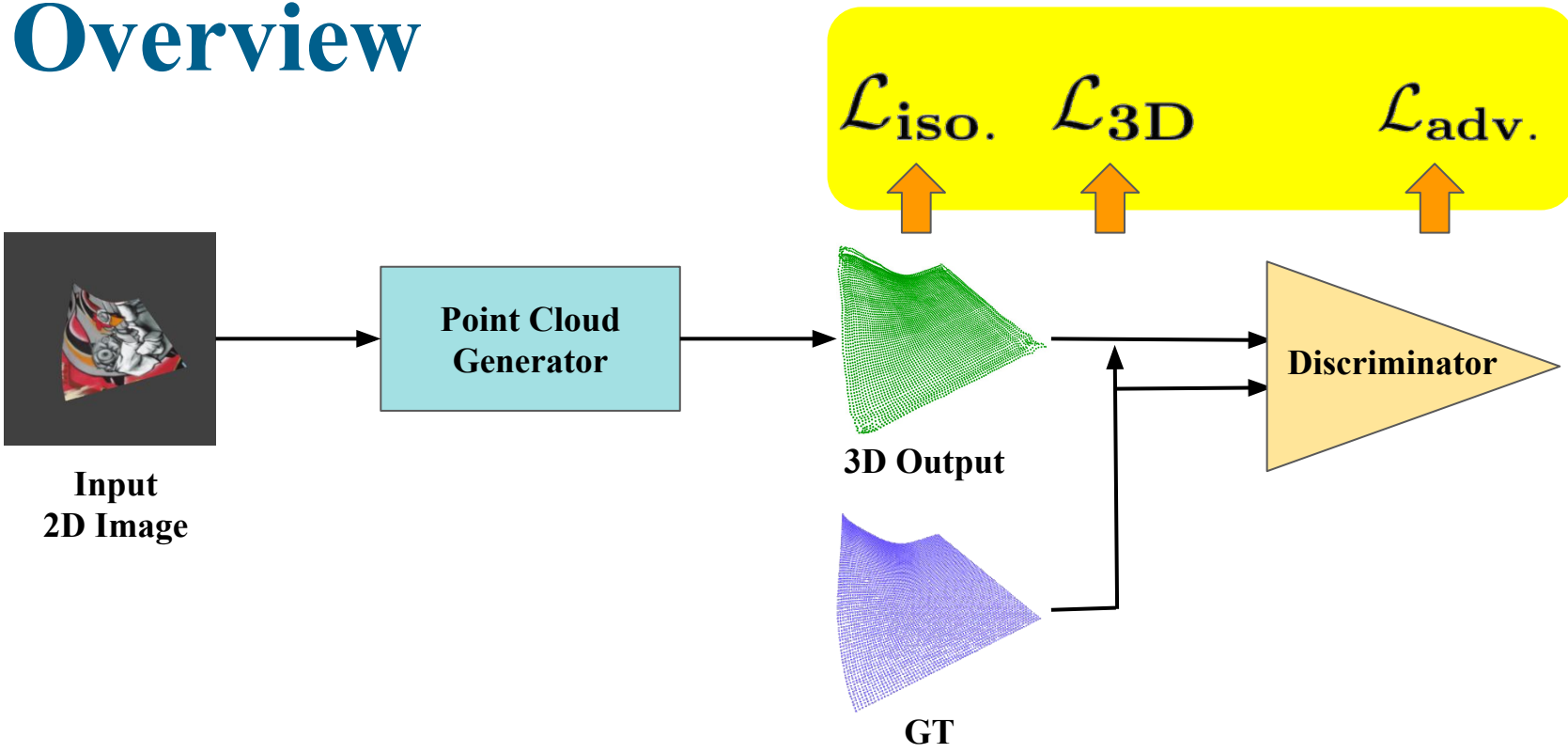
Overview


Confidence Map
Binary Mask
OD-Net
Binarisation
&
Contour fill
Masked-out
Input
Rec-Net
Input
2D Image
IsMo-GAN (Point Cloud Generator)
Output
3D Geometry

Overview


$\mathcal{L}_{iso.}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{adv.}$
Point Cloud Generator
3D Output
Discriminator
Input 2D Image
GT

Overview


$\mathcal{L}_{iso.}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{adv.}$
Point Cloud Generator
3D Output
Discriminator
Input 2D Image
GT

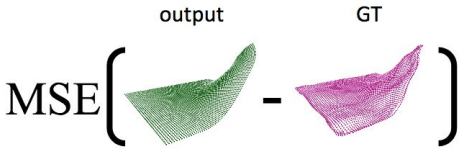
Loss functions

Loss functions

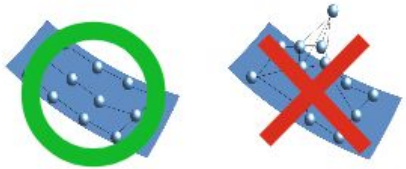
- In order to penalize the network output critically, three kinds of loss functions were incorporated.

Loss functions

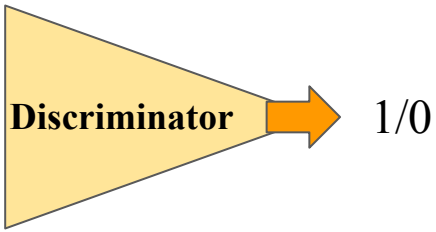
- In order to penalize the network output critically, three kinds of loss functions were incorporated.



1) 3D error



2) Isometry prior



3) Adversarial loss

Loss functions

1) 3D error

Loss functions

1) 3D error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{X}_i - X_i\right)^2$$

- Main loss component for 3D coordinates regression

Loss functions

1) 3D error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{X}_i - X_i\right)^2$$

- Main loss component for 3D coordinates regression
- Penalize difference between 3D coordinates of output and GT

Loss functions

2) Isometry prior

Loss functions

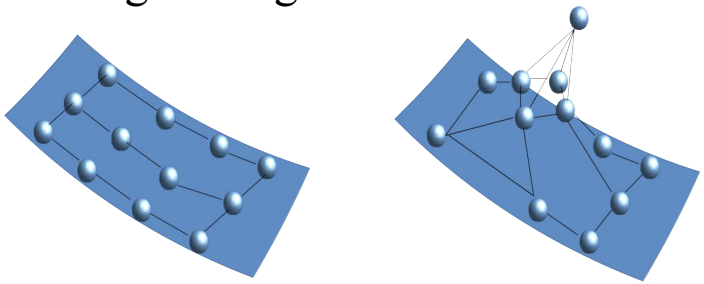
2) Isometry prior

- Idea: since we assume our target object is isometric, a vertex position has to be close to neighboring vertices.

Loss functions

2) Isometry prior

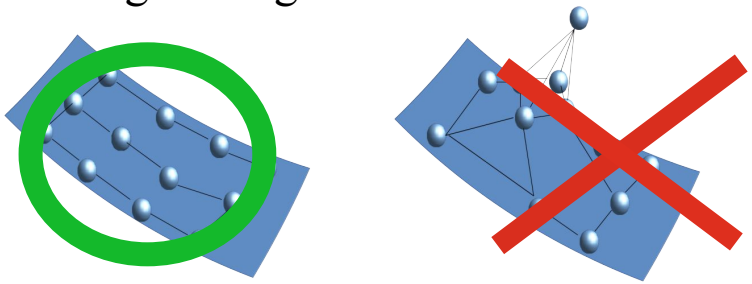
- Idea: since we assume our target object is isometric, a vertex position has to be close to neighboring vertices.



Loss functions

2) Isometry prior

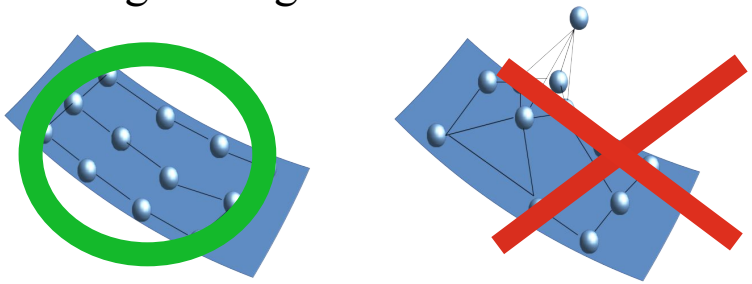
- Idea: since we assume our target object is isometric, a vertex position has to be close to neighboring vertices.



Loss functions

2) Isometry prior

- Idea: since we assume our target object is isometric, a vertex position has to be close to neighboring vertices.

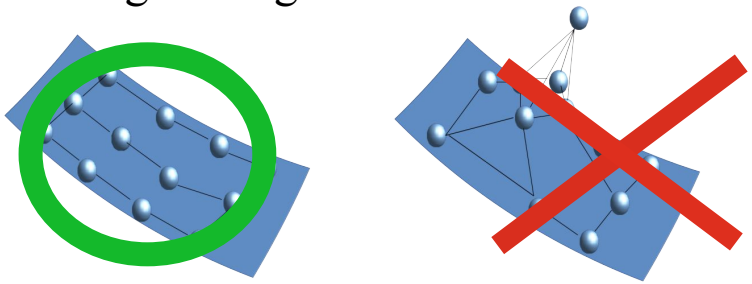


- Apply gaussian smoothing on the output and compute the difference between X_G and X.

Loss functions

2) Isometry prior

- Idea: since we assume our target object is isometric, a vertex position has to be close to neighboring vertices.



- Apply gaussian smoothing on the output and compute the difference between X_G and X.

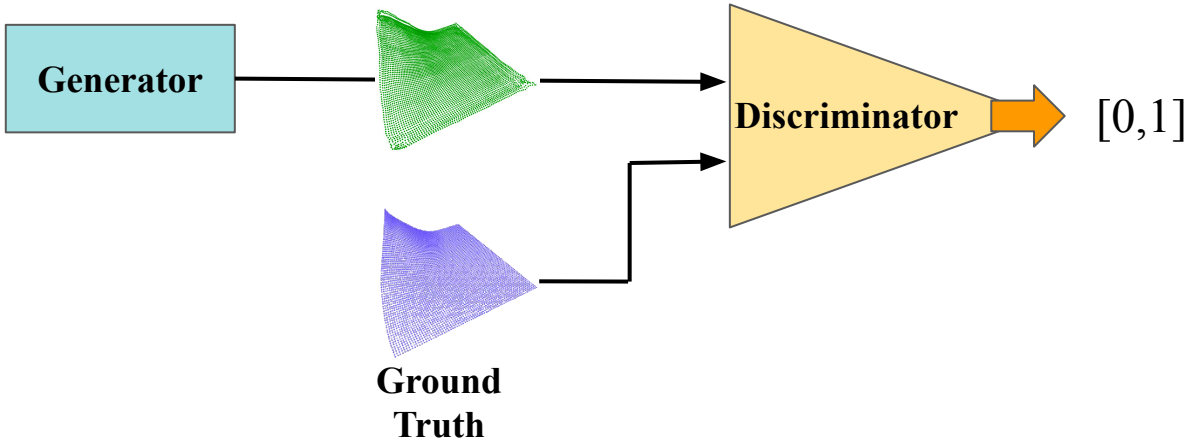
$$X_G = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) * X \quad \Rightarrow \quad SAD = \sum_{i=1}^{n} |X_G - X_i|$$

Loss functions

3) Adversarial loss

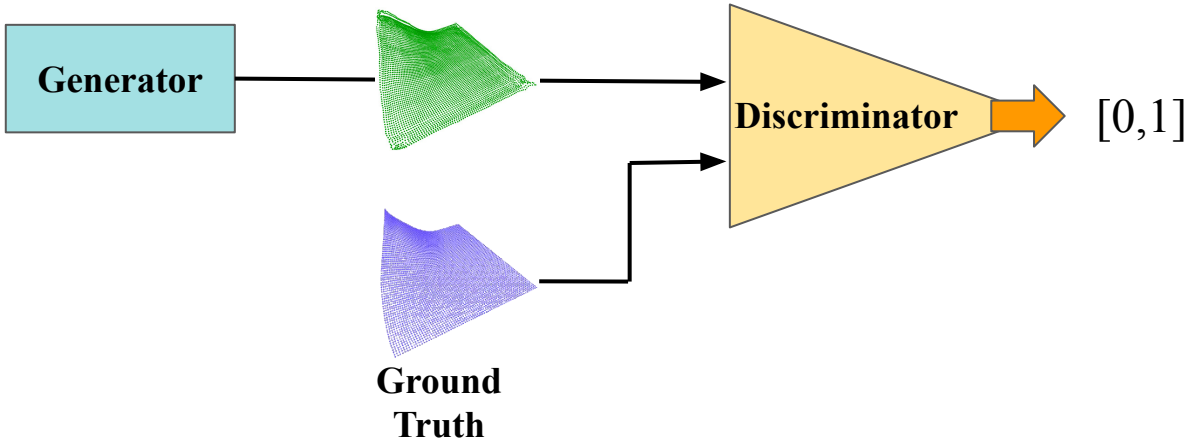
Loss functions

3) Adversarial loss



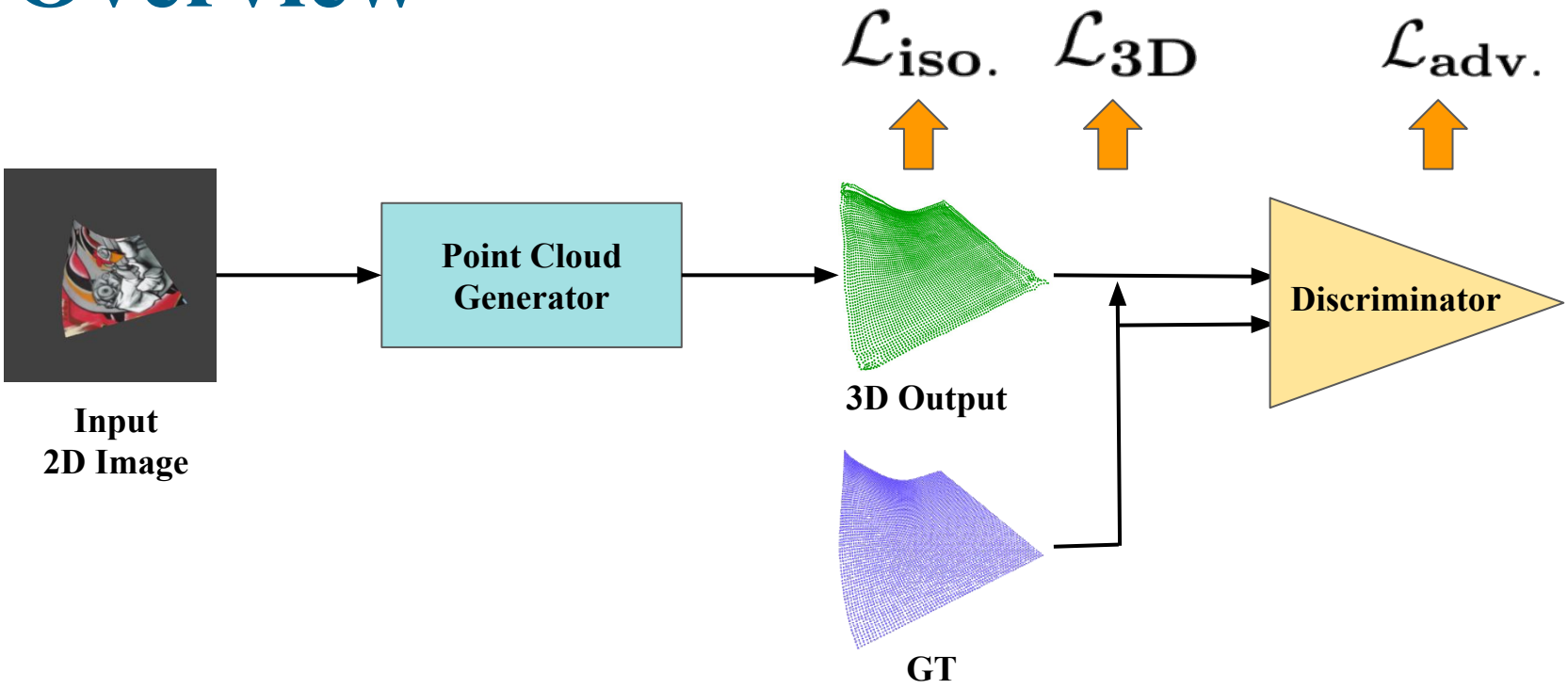
Loss functions

3) Adversarial loss

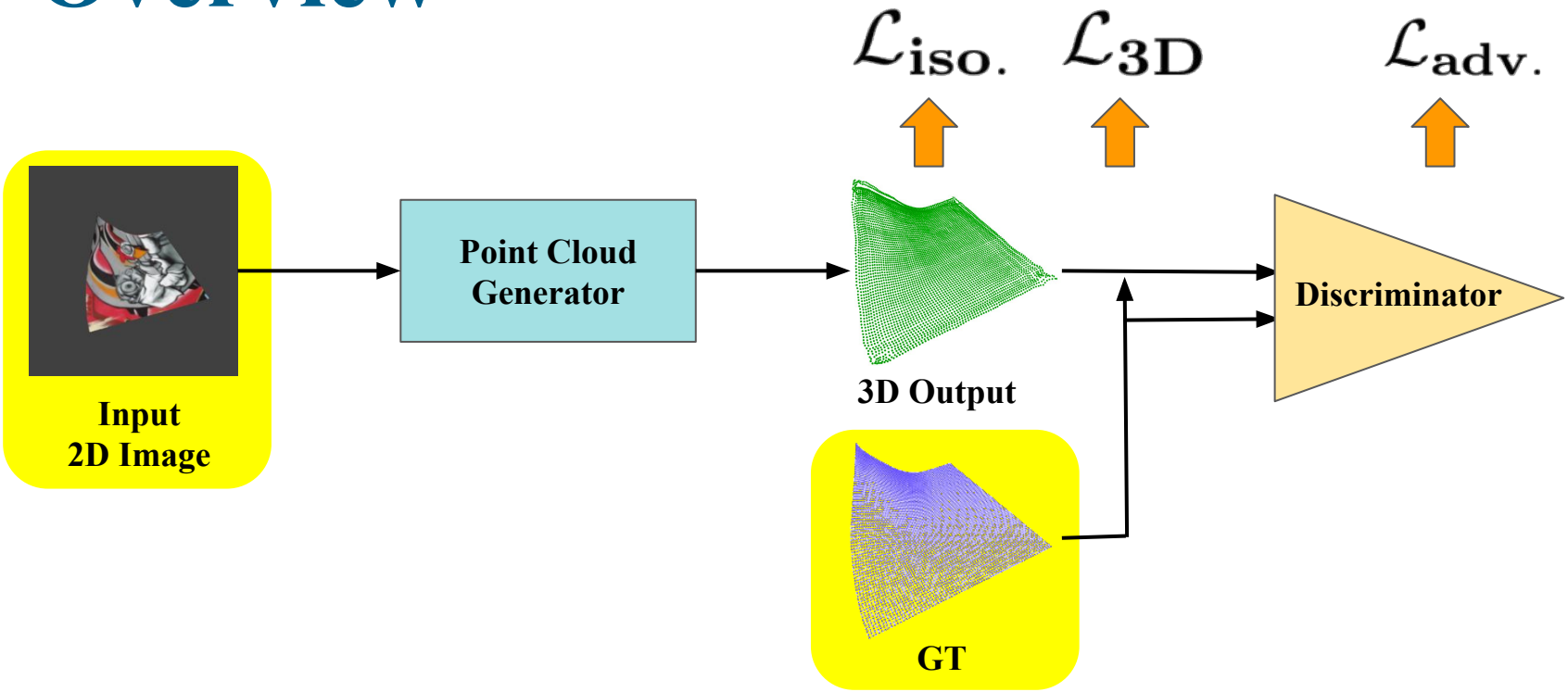


- For further generalisability, the network is trained in an adversarial manner

Overview


$\mathcal{L}_{iso.}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{adv.}$
Point Cloud Generator
3D Output
Discriminator
Input 2D Image
GT

Overview


$\mathcal{L}_{iso.}$
$\mathcal{L}_{3D}$
$\mathcal{L}_{adv.}$
Input
2D Image
Point Cloud
Generator
3D Output
Discriminator
GT

Datasets

Datasets

- Generated 4648 deformation states on blender game engine

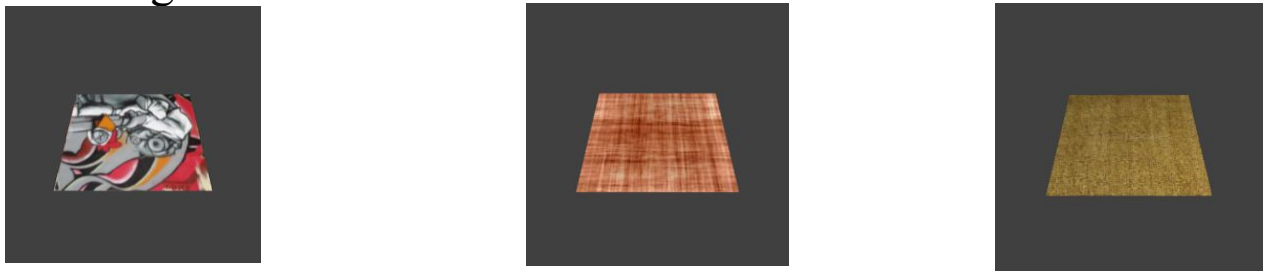
Datasets

- Generated 4648 deformation states on blender game engine
- Took 5 images for each state.

Datasets

state0 ... state1282 ... state3426 ... state4648

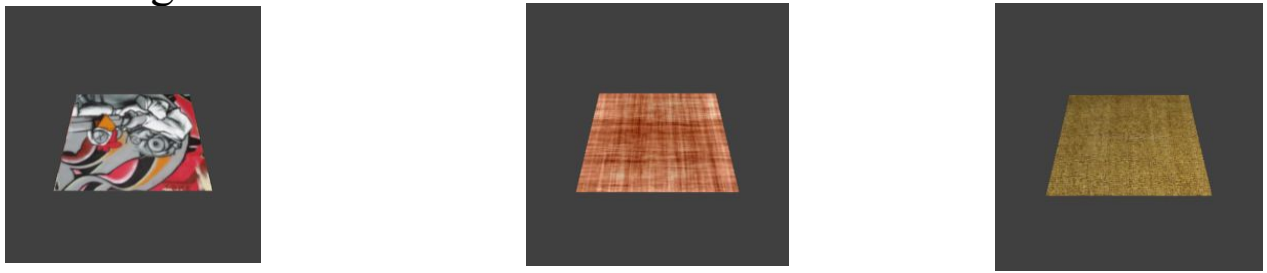
- Generated 4648 deformation states on blender game engine
- Took 5 images for each state.



- 4 different textures(Organ, Flag, Cloth, Carpet)

Datasets

- Generated 4648 deformation states on blender game engine
- Took 5 images for each state.

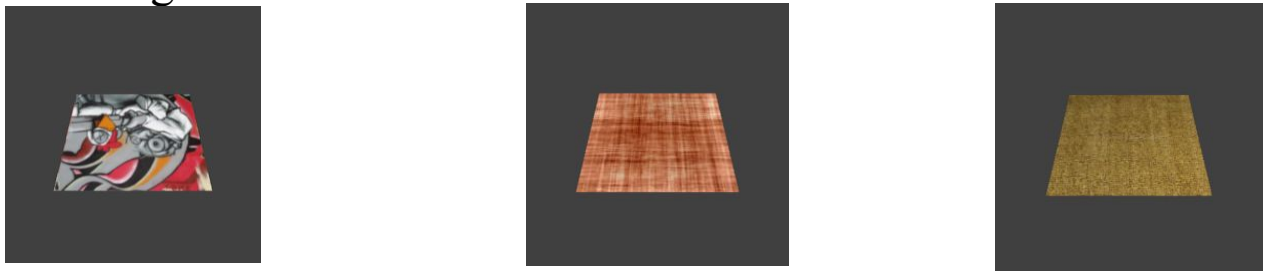


- 4 different textures(Organ, Flag, Cloth, Carpet)
- 5 different illumination positions

Datasets

state0 ... state1282 ... state3426 ... state4648

- Generated 4648 deformation states on blender game engine
- Took 5 images for each state.



- 4 different textures(Organ, Flag, Cloth, Carpet)
- 5 different illumination positions
- 4648 ply file and 330K images in total(4648x5x4x3 + 4648x5 + 4648x5)

Evaluation and Visualization

Quantitative Results

	Yu *et al.* [71]	Liu-Yin *et al.* [38]	AMP [18]	VA [15]	HDM-Net [17]	IsMo-GAN
t, *sec.*	3.305	5.42	0.035	0.39	0.005	**0.004**
e_{3D}	1.3258	1.0049	1.6189	0.46	0.0251	**0.0175**
σ	**0.007**	0.0176	1.23	0.0334	0.03	0.01

Table 1: Reconstruction times per frame t in *seconds*, e_{3D} and standard deviation σ for Yu *et al.* [71], Liu-Yin *et al.* [38], AMP [18], VA [15], HDM-Net [17] and our IsMo-GAN method, for the test interval of 400 frames.

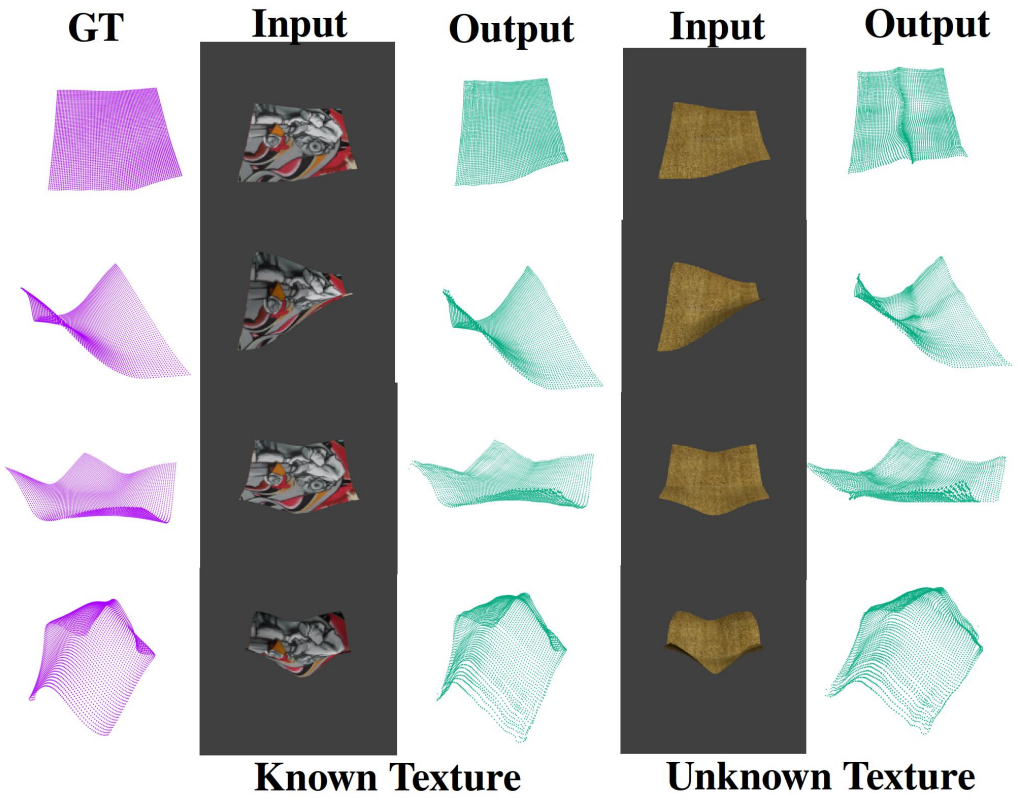
		illum. 1	*illum. 2*	*illum. 3*	*illum. 4*	*illum. 5*
HDM-Net [17]	e_{3D}	0.07952	0.0801	0.07942	0.07845	0.07827
	σ	0.0525	0.0742	0.0888	0.1009	0.1123
IsMo-GAN	e_{3D}	**0.06803**	**0.06908**	**0.06737**	**0.06754**	**0.06685**
	σ	**0.0499**	**0.0696**	**0.0824**	**0.093**	**0.102**

Table 2: Comparison of 3D error for different illuminations. The *illuminations* 1-4 are known, and the *illumination* 5 is unknown.

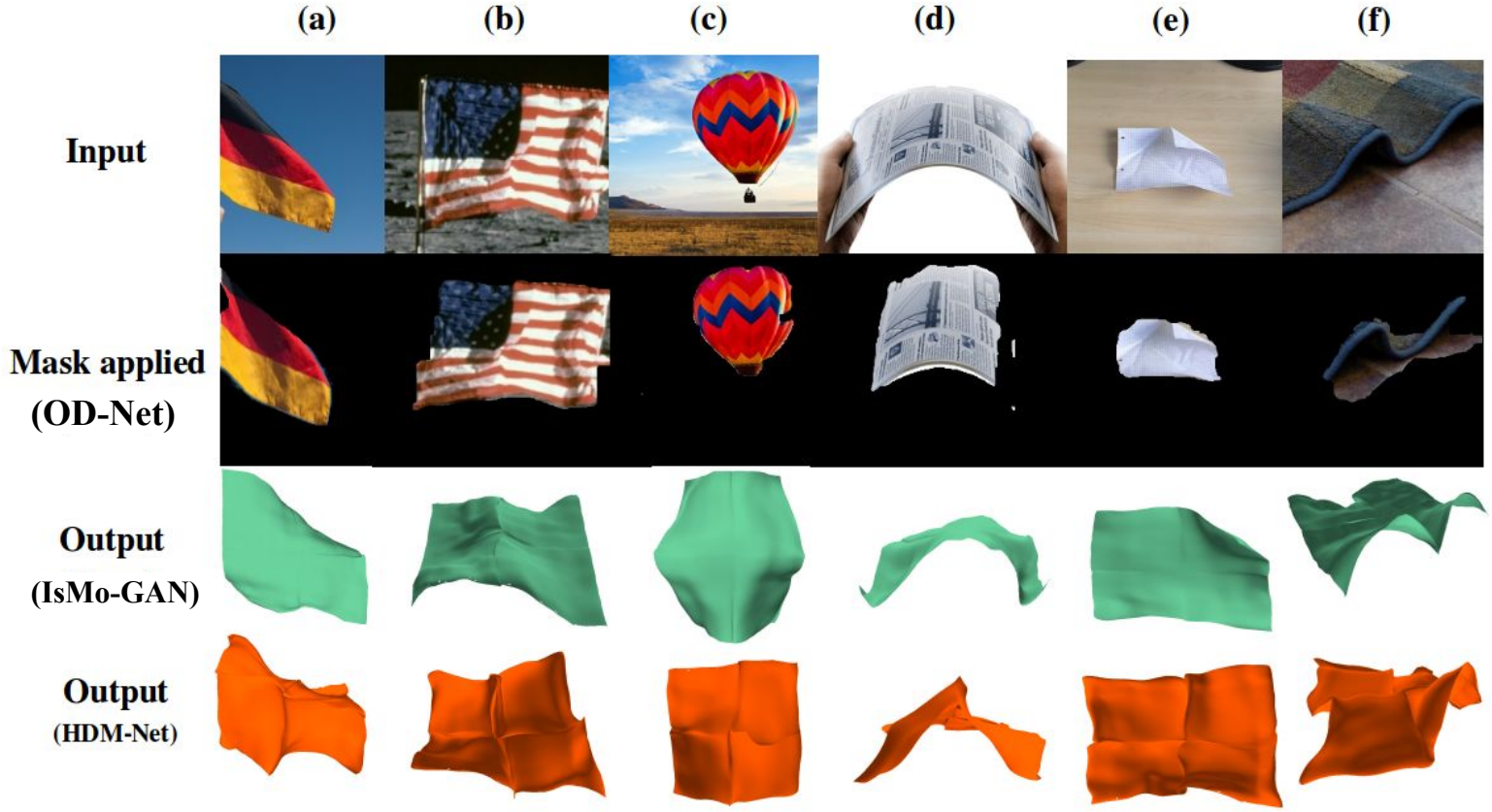
Different textures

		endoscopy	*graffiti*	*clothes*	*carpet*
HDM-Net [17]	e_{3D}	0.0485	0.0499	0.0489	0.1442
	σ	**0.0135**	0.022	0.0264	0.0269
IsMo-GAN	e_{3D}	**0.0336**	**0.0333**	**0.0353**	**0.1105**
	σ	0.0148	**0.0208**	**0.0242**	**0.0268**

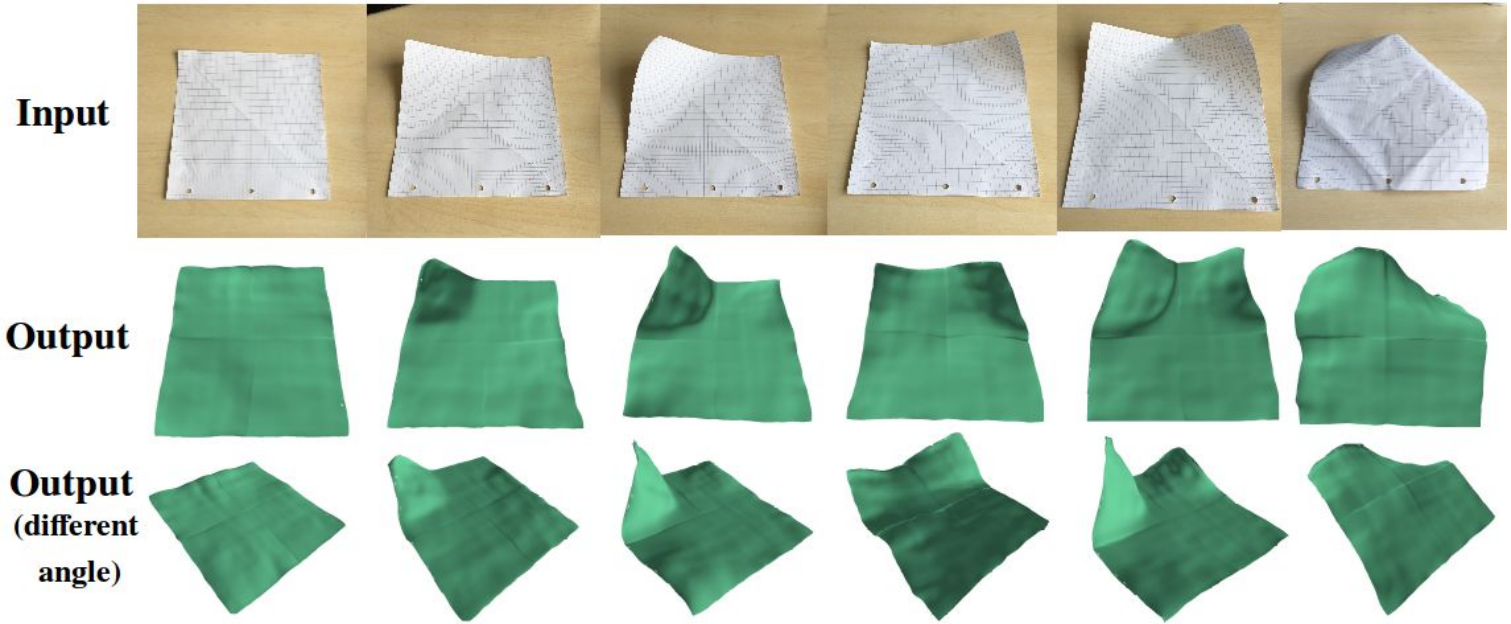
Table 3: e_{3D} comparison for differently textured surfaces under the same illumination (*illumination* 1).



Real-world images

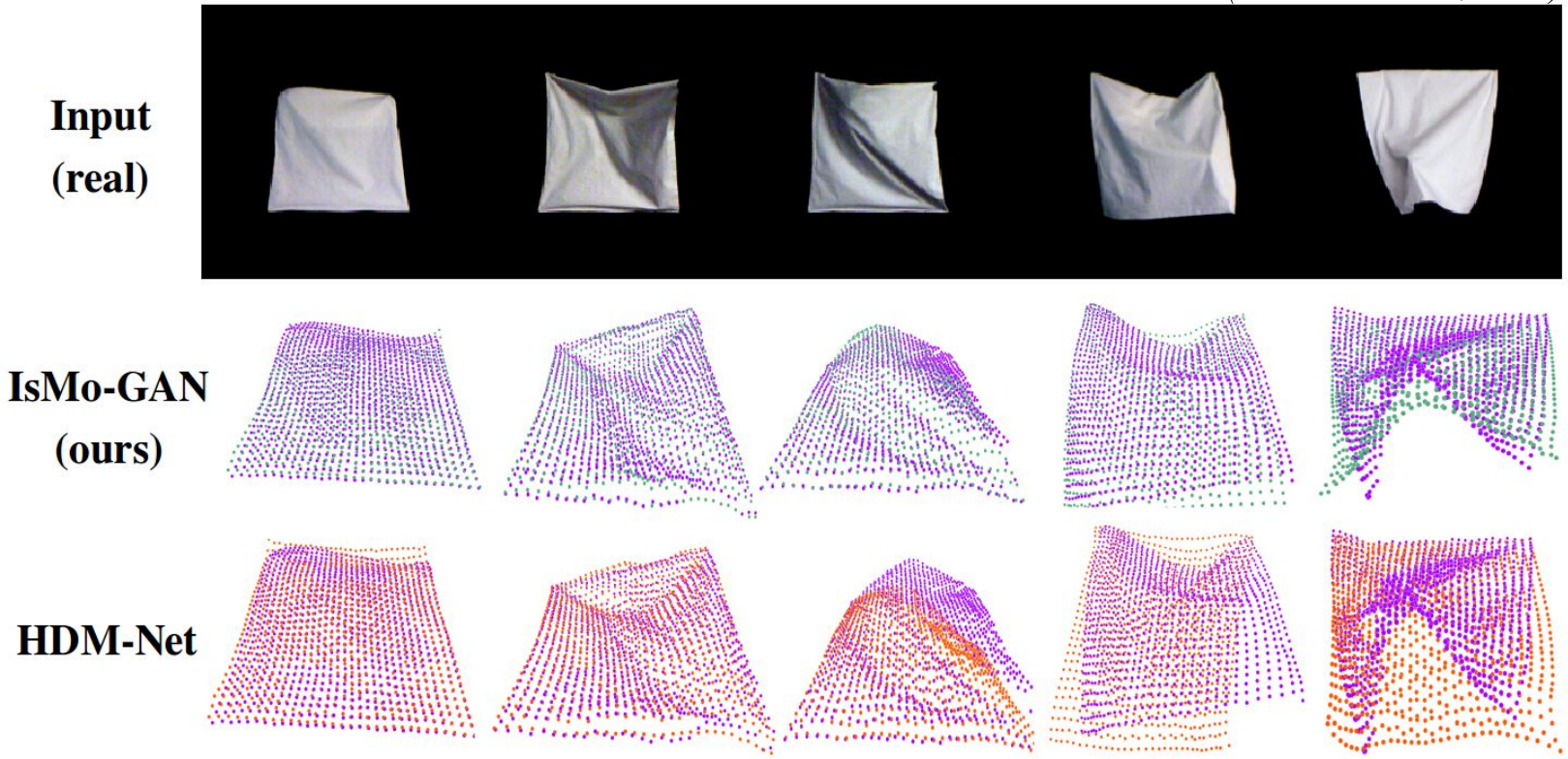


Origami sequences

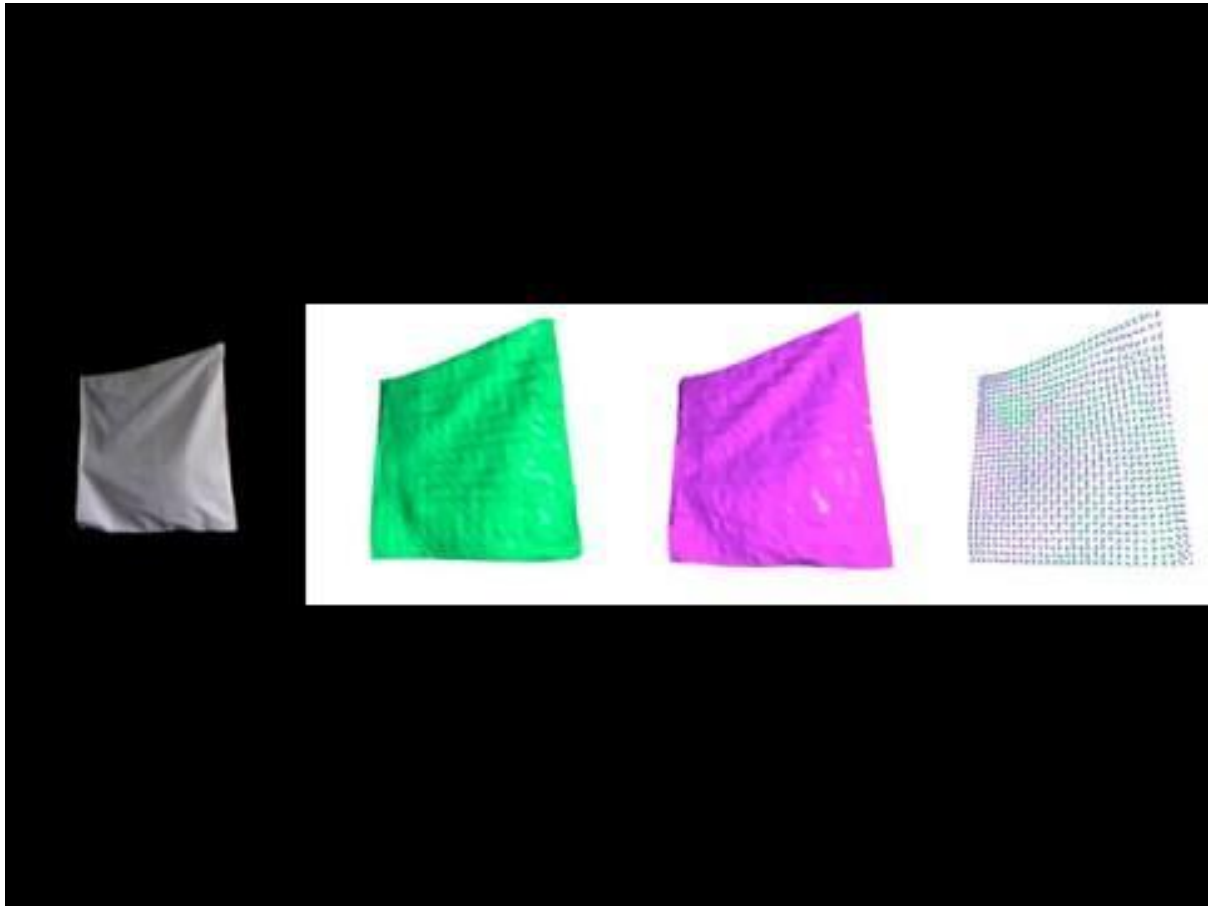


Real texture-less cloth

Texture less dataset (Bednarik et al. , 2018)



Real texture-less cloth



Conclusion

Conclusion

- IsMo-GAN excels other model-based approaches in accuracy and inference time (250 HZ)

Conclusion

- IsMo-GAN excels other model-based approaches in accuracy and inference time (250 HZ)
- Robust to illumination position changes

Conclusion

- IsMo-GAN excels other model-based approaches in accuracy and inference time (250 HZ)
- Robust to illumination position changes
- Thanks to OD-Net, IsMo-GAN shows better generalizability in a texture-less and real-world scenario comparing with HDM-Net

Conclusion

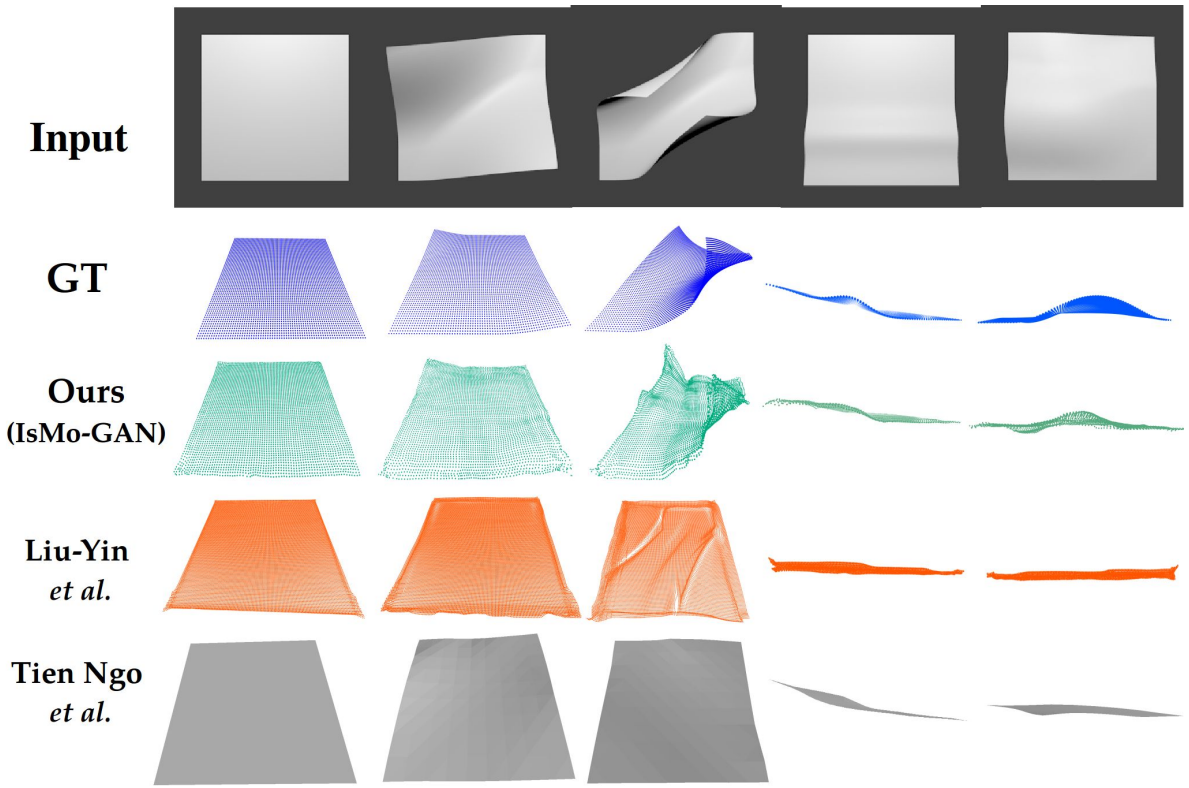
- IsMo-GAN excels other model-based approaches in accuracy and inference time (250 HZ).
- Robust to illumination position changes.
- Thanks to OD-Net, IsMo-GAN shows better generalizability in a texture-less and real-world scenario comparing with HDM-Net.
- (Limitation) Training data (deformation state) is limited.

References

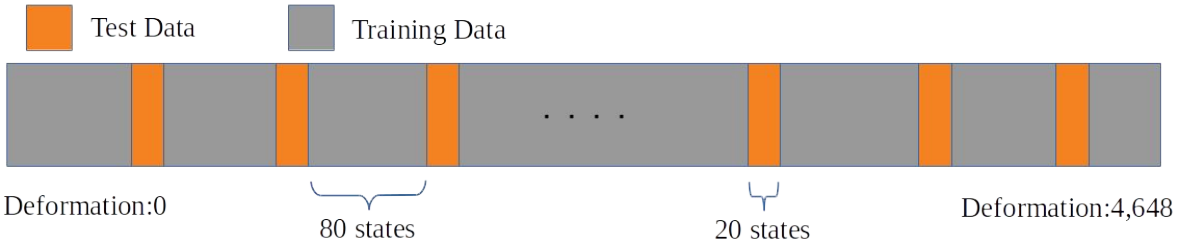
1. Bednarik, J., & Fua, P., & Salzmann, M. (2018). Learning to Reconstruct Texture-Less Deformable Surfaces from a Single View. In *International Conference on 3D Vision* (pp. 606-615).
2. Garg, R., Roussos, A., & Agapito, L. (2013). Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1272-1279).
3. Golyanik, V., Shimada, S., Varanasi, K., & Stricker, D. (2018, October). HDM-Net: Monocular Non-rigid 3D Reconstruction with Learned Deformation Model. In *International Conference on Virtual Reality and Augmented Reality* (pp. 51-72). Springer, Cham.
4. Golyanik, V., & Stricker, D. (2017, March). Dense batch non-rigid structure from motion in a second. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 254-263). IEEE.
5. Liu-Yin, Q., Yu, R., Agapito, L., Fitzgibbon, A., & Russell, C. (2016). Better together: Joint reasoning for non-rigid 3d reconstruction with specularities and shading. In *Proceedings of the British Machine Vision Conference* (pp. 42.1-42.12.).
6. Yu, R., Russell, C., Campbell, N. D., & Agapito, L. (2015). Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 918-926).

Thank you for your attention!

Texture-less dataset



Datasets



- Extract 20 sequential deformation from each 100 states as a test dataset
- Training:Test = 8 : 2

External Occlusion