TRANSFORMATION OF 2D IMAGES INTO 3D BY THE DEEP-LEARNING

Adel ABDELHADI[1] and Ouahab KADRI[2]

[1] Laboratory of Automation & Production Engineering, Department of Computer Science and Math, University of BATNA2, Algeria
A.Abdelhadi@univ-batna2.dz
[2] Laboratory of Automation & Production Engineering Department of Computer Science and Math, University of BATNA2, Algeria
O.kadri@univ-batna2.dz

***ABSTRACT**: Neural networks are a set of algorithms whose operation is inspired by biological neurons, these networks have been developed to solve problems: control, recognition of shapes or words, decision, and memorization. In this work, we tried to make an implementation that combines the advantages of the compact point cloud representation but uses the traditional 2D ConvNet to learn the prior knowledge about the shapes. And by combining the 3 modules together, the convolution structure generator 2D and the merge and pseudo-rendering modules, we have obtained an end-to-end model that learns to generate a compact point cloud representation from a single 2D image, using only a convolution structure generator 2D. And at the end we got as final result: from a single RBG image → 3D point cloud*

***KEYWORDS**: Artificial Intelligence, Deep Learning, Machine Learning, Neural Network, CNN (Convolutional Neuronal Network) and Tensor flow.*

1 INTRODUCTION

The Currently, the image represents a key element in several areas, especially with the availability of instruments allowing the acquisition and/or recording of different types of images with very variable qualities depending on the applications or the use of this information.

Image processing is an essential step in the visualization, analysis, Interpretation, storage and transmission of images. Indeed, all scientific sectors (physics, biology, medicine, astronomy, automation, etc.) call on this discipline. The image can be represented in two different planes: the spatial plane and the plane Frequency. In the spatial plane, the image is described by a set of points called pixels resulting in a matrix mathematical representation. However, in the frequency plane, the image is transported in this plane through the discrete Fourier transform (Manuel, 2016).

In machine learning, a convolutional neural network or Convolutional Neural Networks (CNN) is a type of artificial neural network, in which the connection pattern between neurons is inspired by the visual cortex of animals. Neurons in this region of the brain are arranged so that they correspond to overlapping regions when tiling the visual field. Their operation is inspired by biological processes, they consist of a multilayer stack of perceptrons, the purpose of which is to pre-process small amounts of information. Convolutional neural networks have wide applications in image and video recognition, recommender systems, and natural language processing.

In this work, we have chosen to aim to: Implement an application to transform 2D images into 3D images based mainly on Deep learning and Convolutional Neural Networks (CNN). For this, we used the open source library TENSORFLOW as well as other tools like Anaconda, pycharm and the Python programming language.

2. IMAGE PROCESSING

Image processing refers to a discipline of applied mathematics that studies digital images and their transformations, with the aim of improving their quality or extracting information from them.

In the context of artificial vision, image processing takes place after the acquisition and digitization stages, ensuring the image transformations and the calculation part allowing to go towards an interpretation of the processed images. This interpretation phase is also increasingly integrated into image processing, in particular by calling on artificial intelligence to manipulate knowledge, mainly on the information available about what the images represent. Processed (domain knowledge).

Understanding image processing begins with understanding what an image is. The mode and conditions of acquisition and digitization of the processed images largely condition the operations that will have to be carried out to extract information (Manuel, 2016).

The purpose of image processing is the study, design and production of image processing systems considered as vehicles of information. Its ultimate goal is to extract the informational content (or relevant information) from images for decision-making or action.

Much research has been carried out with the aim of extracting and exploiting this information, which can be at two different levels: The "low level" processing to which is assigned the task of extracting the relevant primitives from the image, the aim of which is to reduce the amount of information contained in the image.

The "high level" processing intended for the interpretation of the content of the image for the purpose of recognition and understanding.

In practice, these two levels are strongly interdependent and cooperate. Indeed, the extraction of primitives only has meaning and interest in relation to the interpretation given by the user.

2.1 Modeling phase

3D modeling is the step in 3D computer graphics which consists in creating, by 3D modeling software, a three-dimensional object; by adding, subtracting and modifying its constituents.

Assisted by specialized software, the designer describes and places the objects manually in the volume of the scene. It uses an appropriate formalism to describe the geometric (shapes) and photometric (colors, flat textures) characteristics of the objects in the scene. Each of these objects is then dressed in a color (in the simplest case) or a texture that represents the material of this object, or even its roughness and relief. Then, the different lights illuminating the scene and the cameras from which the views are taken are defined.

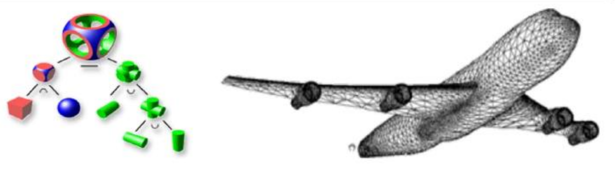


Fig. 1 Two methods for 3D modeling

We call digital model; the model of the object thus constituted (computer representation from geometric information). The most classic method of representation consists of breaking the object down into "Facets", or polygons, which - placed end to end - make it possible to account for the outer envelope of a solid. This representation is purely geometric in that it does not highlight the optical characteristics of the object. The more polygons a mockup has, the more accurate the resulting image.

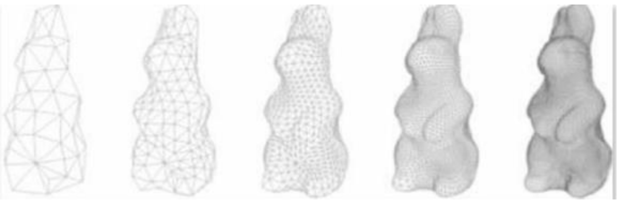


Fig. 2 Effects of number of polygons on the effect of model accuracy

2.2 Render phase

The rendering calculation makes it possible to truly produce the computer-generated image. The goal is to transform the 3D description of the scene into a 2D image depending on the different elements of the scene and from the point of view from which it is viewed. Images of maximum realism (with respect to the model) are generated by the image synthesis system, and this by applying the physical laws of propagation of light, taking into account reflections, refractions, diffusions, and interactions with the materials composing the scene.

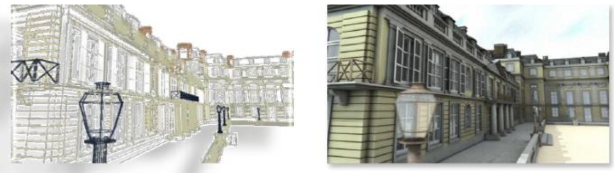


Fig. 3 Rendering of the modeled 3D scene

3. PROBLEMATIC

For crafted items by man, the classic modeling methods give very satisfactory and high quality results (photographic and cinematographic).

But, they turn out to be inapplicable to real scenes (vast and complex) which contain a large number of details (case of the forest), or quite simply too large a volume of data (case of the city), difficult to manage in practice.

The appearance of increasingly realistic and increasingly complex synthesis techniques has reduced the synthetic and artificial aspects of the objects produced by providing excellent photorealism, but this poses several types of problems:

- Traditional image synthesis methods are computationally intensive. If the scene is too complex, synthesizing a single image can already require several minutes of calculation on specialized machines. This can be seen as a secondary problem, as the computing power of machines grows exponentially over time.
- A complex scene requires tedious modeling, which can amount to man-years. Thus, it is out of the question to model every building

and every street of an entire city by hand, or to model the branches of a bare tree in winter.

- Another observation relates to the long and difficult collection of data. If we wish, for example, to model an existing building, we will have to use metric readings taken in the field in order to remain as faithful as possible to reality. Obtaining this data and entering it can be daunting work.
- In addition; the use of these tools often requires a rather long learning phase on the part of the user, and this one does not always master all the power of the system, because of the great diversity of functions and options that 'he proposes. The modeling phase is therefore, most of the time, entrusted to a specialist in 3D modeling.

Even if the problems encountered are generally different, the methods used to solve them are often shared between several disciplines, and 3D reconstruction is one of the most widely used approaches:

- In Photogrammetry, we use stereoscopic photography techniques, which are techniques that appeared before computers. Overlapping images are used to determine the shapes. The basic idea is triangulation: If we have two images from different points of view of the same 3D point, its 2D projection is at the intersection of the two rays formed by the center of projection and this 3D point. The major problem lies in the large amount of data that needs to be processed (particularly point matches) and the need to use precisely calibrated4 devices.
- Computer Vision and Robotics make 3D reconstruction one of their main fields of research. The methods used are structure inference from motion, stereovision, depth maps, etc. It is indeed crucial in robotics to be able to reconstruct in real time a 3D environment for the navigation of a machine. And there is no need, in general in these applications, for a detailed reconstruction because the model is not intended to be visualized.
- The interest in computer graphics is to be able to create more realistic models more quickly that correspond to objects of which we have photographs, plans, drawings, etc. Modeling systems are often sophisticated, complex and time-consuming to master, hence a recent interest in reconstruction from images. In addition, applications require more and more realism (immersive environments, augmented reality, special effects, ...) and traditional models are often too (clean), or (simple). In these approaches, the general aim is to make the system an aid to the designer who plays an important role in the reconstruction process.

Manual methods have proven to be very difficult to apply for seemingly very simple tasks such as recognizing objects in images or speech recognition. Real-world data—samples of sound or pixels of an image—is complex, variable, and tainted with noise.

For a machine, an image is an array of numbers indicating the brightness (or color) of each pixel, and a sound signal is a series of numbers indicating the air pressure at each instant.

How can a machine transcribe the sequence of numbers from a sound signal into a series of words while ignoring ambient noise, the speaker's accent and the particularities of his voice?

How can a machine identify a dog or a chair in the number array of a picture when the appearance of a dog or a chair and the objects around them can vary infinitely?

It is virtually impossible to write a program that will work robustly in all situations. This is where Machine Learning (also known as machine learning) comes in. It is learning that drives the systems of all major internet companies. They have long used it to filter unwanted content, order search responses, make recommendations, or select information of interest to each user.

4. DEEP LEARNING

Deep Learning (DL) or deep learning is a new area of research in Machine Learning, which was introduced with the aim of bringing it closer to its main objective: Artificial Intelligence.

It concerns algorithms inspired by the structure and functioning of the brain. They can learn several levels of representation in order to model complex relationships between data.

Deep Learning is based on the idea of artificial neural networks and is tailored to handle large amounts of data by adding layers to the network.

A Deep Learning model has the ability to extract features from raw data through multiple layers of processing consisting of multiple linear and nonlinear transformations and learn about those features step by step through each layer with minimal human intervention.

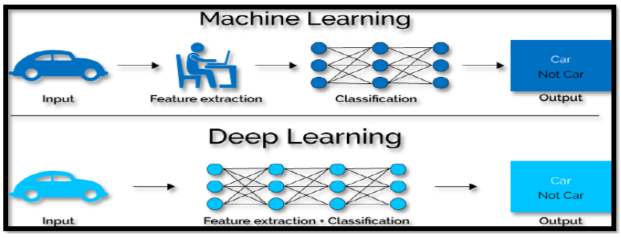


Fig. 4 From ML to Deep Learning

Over the last five years, Deep Learning has gone from a niche market where only a handful of researchers were interested in it to the field most precise by researchers (Durand, 2018).

The term "Deep Learning" was first introduced to ML by Dechter (1986), and to artificial neural networks by Aizenberg et al (2000). (Durand, 2018)

4.1 Why Deep Learning?

ML algorithms work well for a wide variety of problems. However, they failed to solve some major AI problems such as voice recognition and object recognition. (Durand, 2018)

The development of deep learning was driven in part by the failure of traditional algorithms in such AI tasks. But it was only after larger amounts of data became available, thanks in particular to Big Data and connected objects, and calculation machines became more powerful, that we were able to understand the real potential of Deep Learning. (Durand, 2018).

4.2 The convolutional neural network

We call convolutional neural network, (CNN for Convolutional Neural network) a type of artificial neural network used in image recognition and processing, and specially designed for pixel analysis.

Convolutional networks are a powerful application of artificial intelligence (AI) to image processing, which leverages deep learning to perform descriptive and generative tasks. They often exploit machine vision, including video and image recognition, recommendation systems, and automatic natural language processing.

4.3 General principle of convolutional neural networks

Convolutional neural networks are very similar to neural networks. They nevertheless exploit one of the important characteristics of the images, namely the spatial distribution of the sampling. (Baakek, 2020)

They consist successively of layers of convolutions, layers of groupings, and connected layers. The term deep learning deep learning refers to the many layers that must be learned during training. (Baakek, 2020)

Convolutional neural networks are not directly inspired by biology and rely on learning algorithms that may fundamentally differ from those of biological brains.

However, they learn internal representations that strongly resemble the ideas one imagines from representations of the visual cortex.

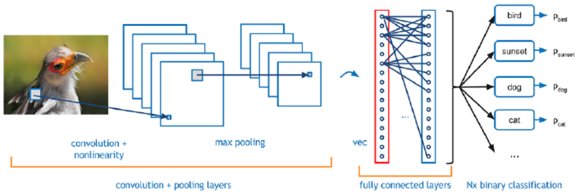


Fig. 5 CNN which receives a 2D image as input and which is composed of a convolutional layer, a nonlinear activation function, a MAX pooling layer and finally a multilayer perceptron.

4.4 The main types of layers:

4.4.1 Convolutional Layers: Convolutional Layers

Convolution layers are a set of filters that are learned during training. The size and number of these filters are defined a priori.

During forward propagation, each filter slips (more precisely is convolved) and the dot products between the inputs (e.g. the pixel values of an image) and these filters are computed. As the filter moves across the image an activation map is produced.

It represents the response of this filter at each spatial position. These activation maps are concatenated along the dimension n+1 where n represents the number of initial dimensions of the image to form a new tensor.

4.4.2 Grouping layers: Pooling Layers

It is common to periodically insert pooling layers after the convolution layers. These functions are predefined and reduce the number of parameters to learn for later layers while widening the receptive fields.

They operate independently at the different depths of the network and do not require weights to be driven. One of the classic operations operated is the maximum function, where in the vicinity of N pixels only the maximum is retained in the grouping layer. (Baakek, 2020)

4.4.3 Fully connected layers: Fully connected layers

The neurons in these layers are all connected to the set of neurons from the previous activation maps. Thus, if the output of a convolution or clustering layer produces an activation map of 512 7x7 filters

and the FCL layer contains 4096 neurons, there will be:
512x7x7x4096 weight to train.
The FCL is the last layer and makes it possible to make a transition in order to transform the maps of activations into probabilities. So if we try to predict 10 categories, the FCL will contain 10 neurons. This is allowed thanks to the properties and the links between the sigmoid activation functions classically used on these layers and the laws of probability (in particular logistic). (Baakek, 2020)

4.5 TensorFlow

TensorFlow is an open source machine learning tool developed by Google. The source code was opened on November 9, 2015 by Google and released under the Apache license.
It is based on the DistBelief framework, initiated by Google in 2011, and has an interface for Python and Julia. (Mimoune, 2019)
TensorFlow is one of the most widely used AI tools in the field of machine learning.

TensorFlow is today particularly used for Deep Learning, and therefore neural networks. Its name is particularly inspired by the fact that common operations on neural networks are mainly done via multi-dimensional data tables, called Tensors (Tensors). A two-dimensional Tensor is the equivalent of a matrix.

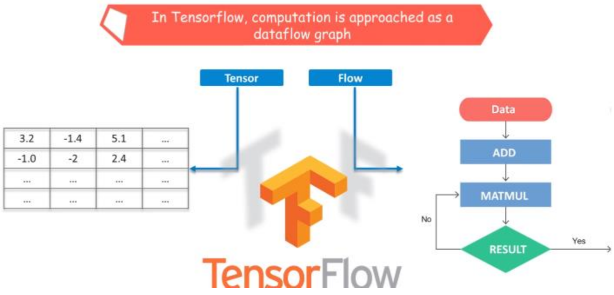


Fig. 6 The main constituents of Tensorflow

5. THE TASK TO BE ACCOMPLISHED

My work is an attempt to use Deep Learning in the single image 3D reconstruction task.

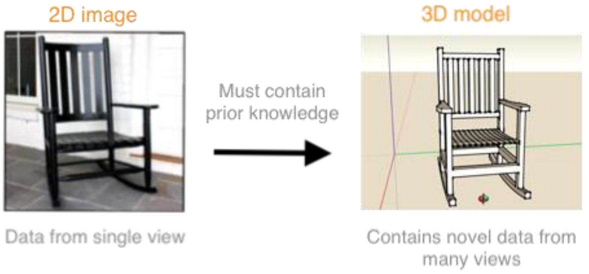

2D image
3D model
Must contain prior knowledge
Data from single view
Contains novel data from many views

Fig. 7 Transformation of 2D images into 3D by the deep-learning

A single image is just a projection of a 3D object into a 2D plane, so some data from the higher dimensional space must be lost in the lower dimensional representation. Therefore, from a single-view 2D image, there will never be enough data to build its 3D component.
A method for creating 3D perception from a single 2D image therefore requires prior knowledge of the 3D shape itself.
In 2D Deep Learning, a convolutional AutoEncoder is a very efficient method to learn a compressed representation of input images. Extending this architecture to learning compact shape knowledge is the most promising way to apply Deep Learning to 3D data.

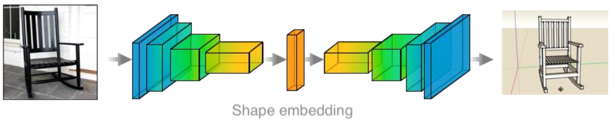
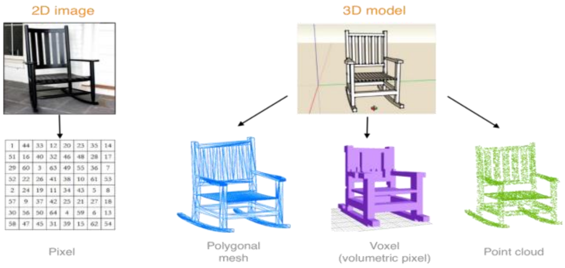


Fig. 8 CNN encodes prior knowledge of deep forms.

5.1 Representation of 3D data

Unlike a 2D image which has only one universal representation in computer format (pixel), there are many ways to represent 3D data in digital format. They have their own advantages and disadvantages, so the choice of data representation directly affected the approach that can be used.

Fig. 8 Different representations of 3D data



5.2 The Raster Shape (The Voxel Grids)

Voxel, short for volumetric pixel, is the direct extension of spatial grid pixels into volume grid voxels. The locality of each voxel together defines the unique structure of this volumetric data, so ConvNet's locality assumption is always true in the volumetric format.

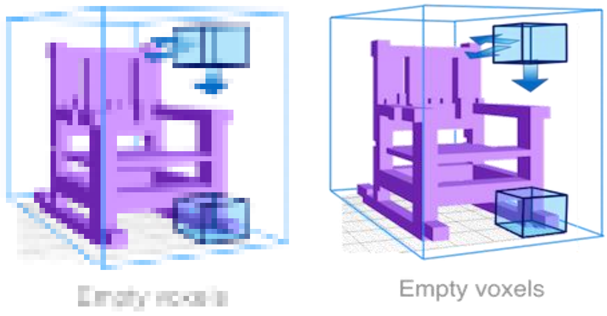


Fig. 9 Each blue box is a single voxel, most voxels are empty.

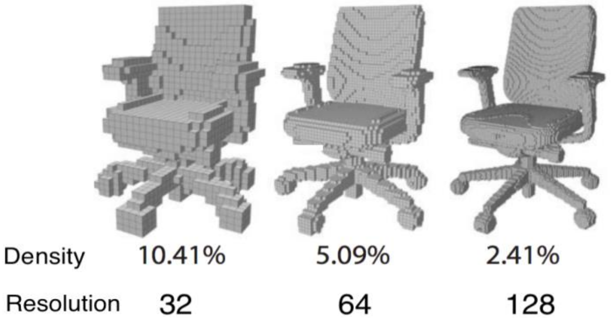


Fig. 10 Low density of voxel representation.

However, this representation is useless. The density of useful voxels decreases as the resolution increases.

- **Advantage:** can directly apply the CNN of the 2D representation to the 3D.
- **Disadvantage**: useless representation, high compromise between details and resources (calculation, memory).

5.3 2D structure generator

We will build a standard 2D CNN structure generator that learns prior knowledge of an object's shape. Voxel approach is not desired

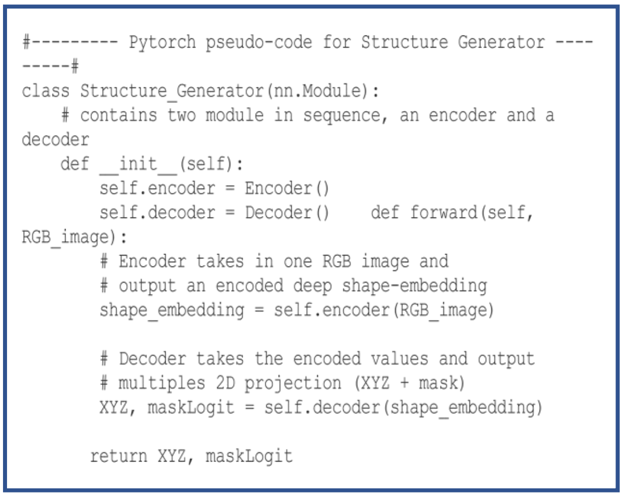
because it is inefficient and it is not possible to directly learn a scatter plot with CNN.

Therefore, we will instead learn the mapping of a single image to multiple 2D projections of a point cloud, with a 2D projection at a viewpoint defined as: 2D

```
Projection == 3D coordinates (x,y,z) +

binary mask (m)
```

- **Input:** single RGB image
- **Output:** 2D projections at predetermined viewpoints.



```
#--------- Pytorch pseudo-code for Structure Generator ----
-----#
class Structure_Generator(nn.Module):
    # contains two module in sequence, an encoder and a
decoder
    def __init__(self):
        self.encoder = Encoder()
        self.decoder = Decoder()     def forward(self,
RGB_image):
        # Encoder takes in one RGB image and
        # output an encoded deep shape-embedding
        shape_embedding = self.encoder(RGB_image)

        # Decoder takes the encoded values and output
        # multiples 2D projection (XYZ + mask)
        XYZ, maskLogit = self.decoder(shape_embedding)

       return XYZ, maskLogit
```

5.4 Point cloud fusion

The predicted 2D projections are fused into native 3D point cloud data. This is possible because the viewpoints of these predictions are fixed and known in advance.

- **Input:** 2D projections at predetermined viewpoints.
- **Output:** point cloud

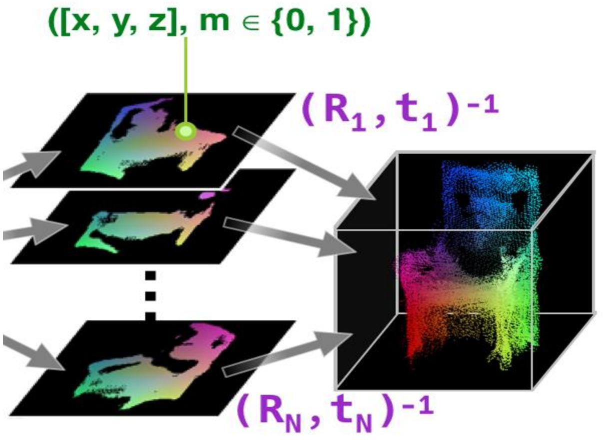
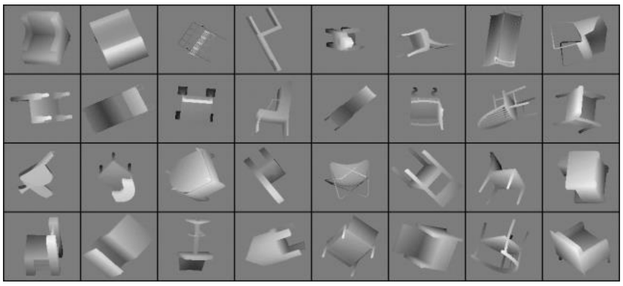


Fig. 11 Point cloud fusion.

5.5 Pseudo-rendering

It is believed that, if the merged point cloud from the predicted 2D projections is good, then if one rendered different 2D projections from new viewpoints, it should also look like the projections from the ground truth 3D model.

- **Input:** point cloud
- **Output:** depth images at new vantage points.



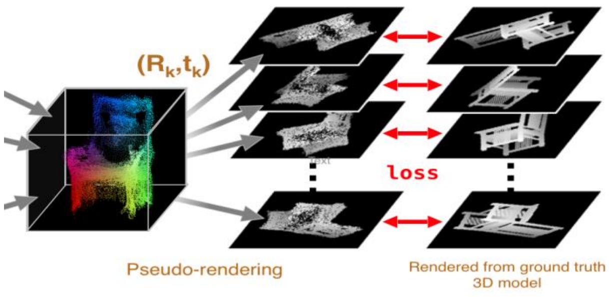


Fig. 12 Pseudo-rendering.

5.6 Training dynamics:

Combining the 3 modules together resulted in an end-to-end model that learns to generate a compact point cloud representation from a single 2D image, using only a 2D convolution structure generator.

The trick of this model is to make the merging + pseudo-rendering modules purely differentiable, geometric reasoning:

- Geometric algebra means there are no learnable parameters and makes the model size smaller and easier to train.
- Differentiable means that one can back-propagate gradients through it, which makes it possible to use the loss of 2D projections to learn how to generate a 3D point cloud.

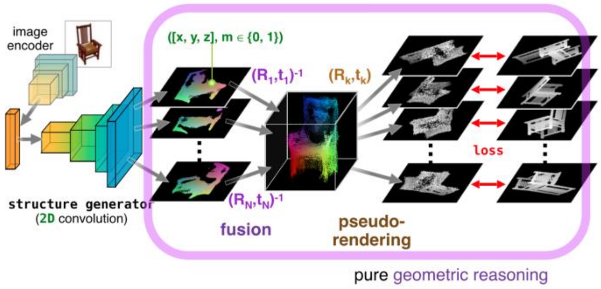
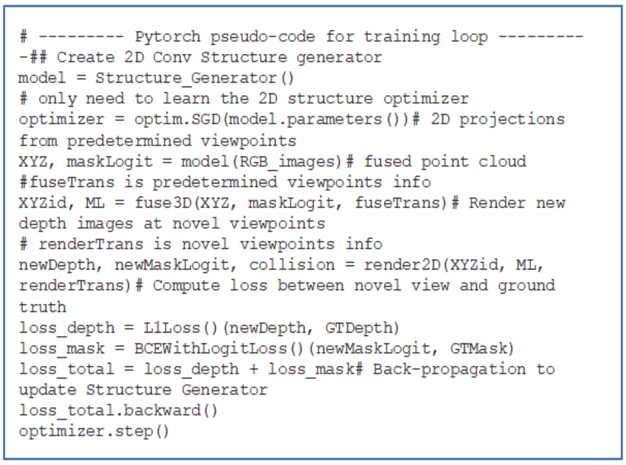


Fig. 12 Complete architecture from 2D convolution structure generator and fusion and pseudo-rendering modules.



```
# --------- Pytorch pseudo-code for training loop ---------
-## Create 2D Conv Structure generator
model = Structure_Generator()
# only need to learn the 2D structure optimizer
optimizer = optim.SGD(model.parameters())# 2D projections
from predetermined viewpoints
XYZ, maskLogit = model(RGB_images)# fused point cloud
#fuseTrans is predetermined viewpoints info
XYZid, ML = fuse3D(XYZ, maskLogit, fuseTrans)# Render new
depth images at novel viewpoints
# renderTrans is novel viewpoints info
newDepth, newMaskLogit, collision = render2D(XYZid, ML,
renderTrans)# Compute loss between novel view and ground
truth
loss_depth = L1Loss()(newDepth, GTDepth)
loss_mask = BCEWithLogitLoss()(newMaskLogit, GTMask)
loss_total = loss_depth + loss_mask# Back-propagation to
update Structure Generator
loss_total.backward()
optimizer.step()
```

5.7 Results

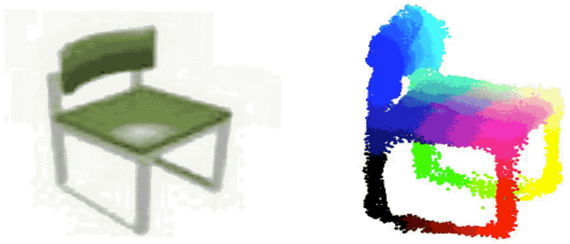
Comparison of a new depth image from the 3D ground truth model and the rendered depth image from the learned point cloud model.

Fig. 12 Final result: from a single RBG image → 3D point cloud

With the detailed representation of the point cloud, it is possible to use MeshLab to convert it to other representations such as voxel or polygonal mesh that are compatible with 3D printers.

Fig. 13 Use MeshLab to convert it to other representations such as voxel or polygonal

6. CONCLUSION



In this paper, we try to expose the implementation we used to obtain an end-to-end model that learns to generate a compact point cloud representation from a single 2D image, using only a generator of 2D convolution structure. And this was possible by combining the 3 modules together, the 2D convolution structure generator and the merging and pseudo-rendering modules.

And in the end, we obtained as final result the following: from a single RBG image → 3D point cloud

7. REFERENCES

Manuel, A. (2016), *2D/3D semantic annotation of spatialized images for the documentation and analysis of heritage objects*, Doctoral thesis, Paris Tech University, France.

Durand, P.B. (2018), *Convolutional neural networks in nuclear medicine: Applications to the automatic segmentation of glial tumors and to attenuation correction in PET/MRI*, Doctorate thesis, Paris Descartes University, France.

Allal, M.A. (2017), *Use of deep learning in cognitive radio*, Master's Memory, Abou Bekr Belkaid University, Tlemcen, Algeria.

Mimoune, Z. (2019), Development of an Architecture Based on Deep Learning (Deep Learning) for Intrusion Detection in Networks, Master's Memory, Ahmed Draia University, Adrar, Algeria.

Moualek, D.Y. (2017), *Deep Learning for image classification*, Master's Memory, Abou Bekr Belkaid University, Tlemcen, Algeria.

Boufar. N, Taghribet. A. (2016), *The use of self-organized systems to analyze medical images*, Master's Memory, Larbi ben M'hidi Oum El Bouaghi University, Algeria.

Rouabhia, M. D. (2011), *A MULTI-VIEW METHOD FOR 3D RECONSTRUCTION*, Magister's Memory, 2011, Mohamed Khider University, Biskra, Algeria.

Baakek, T. (2020), *Three-dimensional (3D) Segmentation of Medical Images*, Magister's Memory, Abou Bekr Belkaid University, Tlemcen, Algeria.

Bentata, R. (2011), *Segmentation of tomographic images by positron emission*, Magister's Memory, University of es-Senia Oran, Algeria.

Ghennam, B, Smara, S. (2019), *Convolutional neural networks (CNN) for the classification of images associated with parking spaces in a vehicle fleet*, Master's Memory, University of es-Senia Oran, Algeria. .

Boughaba, M, Boukhris, B. (2017), *Deep Learning for Image Classification and Search by Content*, Professional Master's Memory, Kasdi Merbah Ouargla University, Algeria.