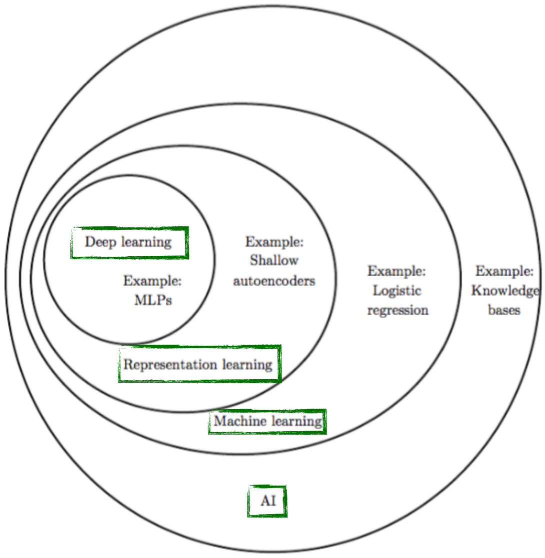
Machine Learning
.. and Graphics
A Personal View

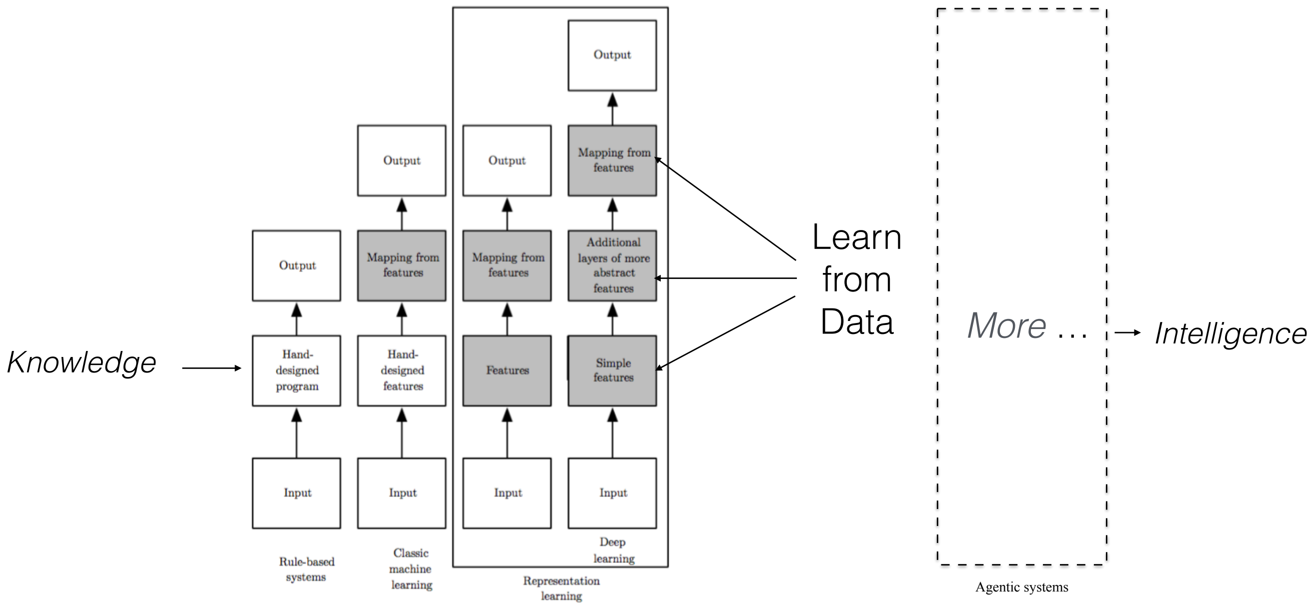
Luiz Velho
IMPA

Machine Learning
Fundamentals

Contextualization

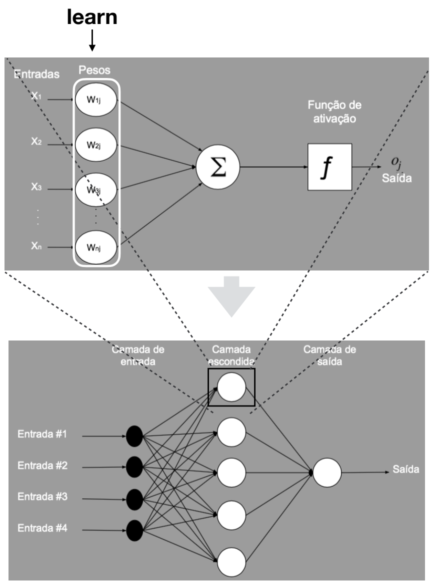


Historical Evolution



Neural Networks

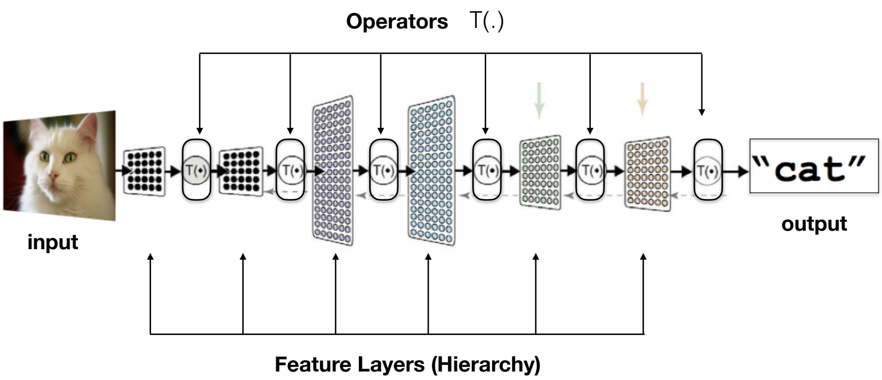
- Neuron (<u>operator</u>)
- Neural Nets (<u>composition</u>)



shallow

Deep Neural Networks

- Multi Layer Architecture



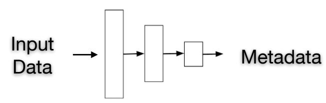
[input -> output]

Neural Network Models

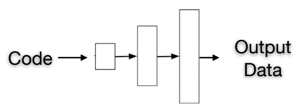
- Discriminative
  - Analysis
- Generative
  - Synthesis
- Descriptive
  - Representation
- Simulation
  - (Inter)Action

Neural Networks Types

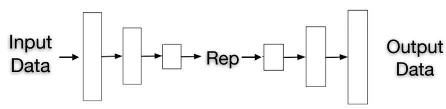
- Analysis



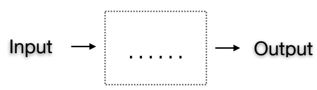
- Synthesis



- Full



- Composite

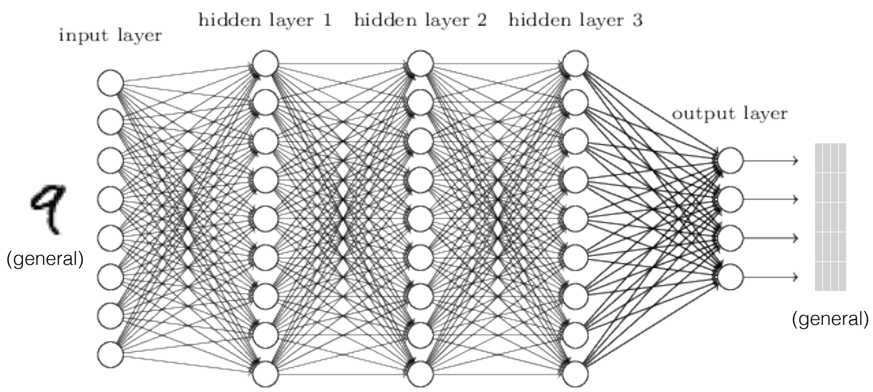


Classic Network Architectures

- Fully Connected MLP (Multi Layer Perceptron)
- Convolutional NN
- Auto-Encoder
- Recurrent NN
- Generative Adversarial Networks

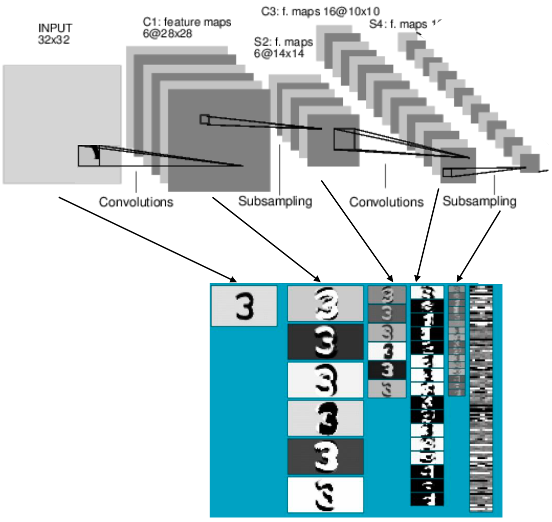
Fully Connected MLP

- Learn General Functions



Convolutional Networks

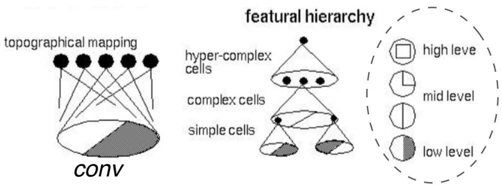
- Learn Image Features



Feature Maps

Discriminative CNNs

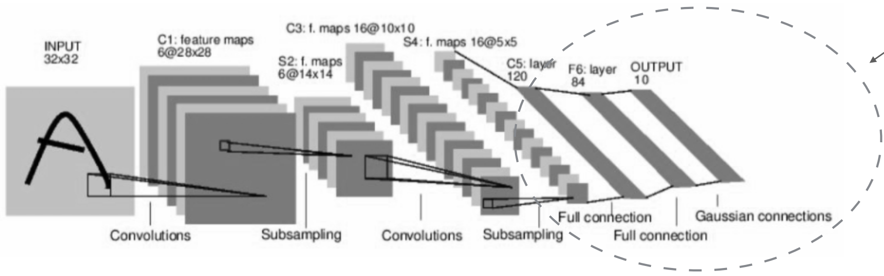
- Convolution Operators. (*human vision*)



learned features

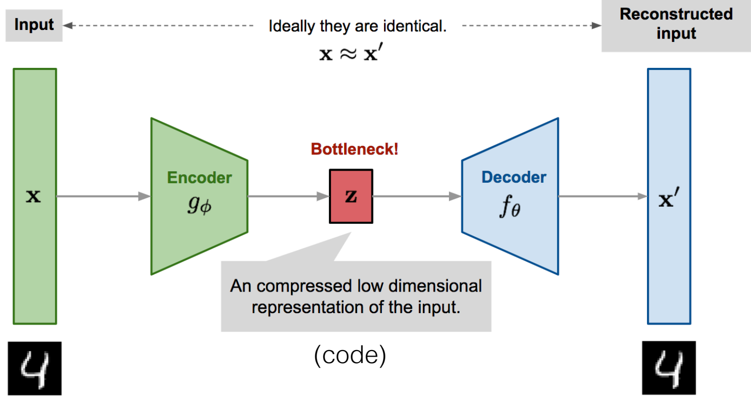
- Deep Convolutional Neural Networks

Discrimination



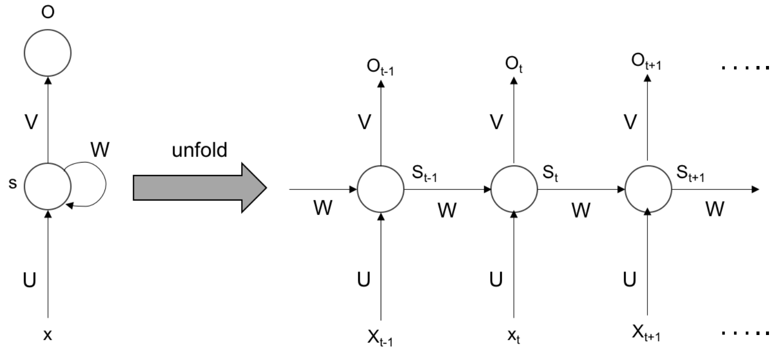
Variational Auto-Encoders

- Learn Representations



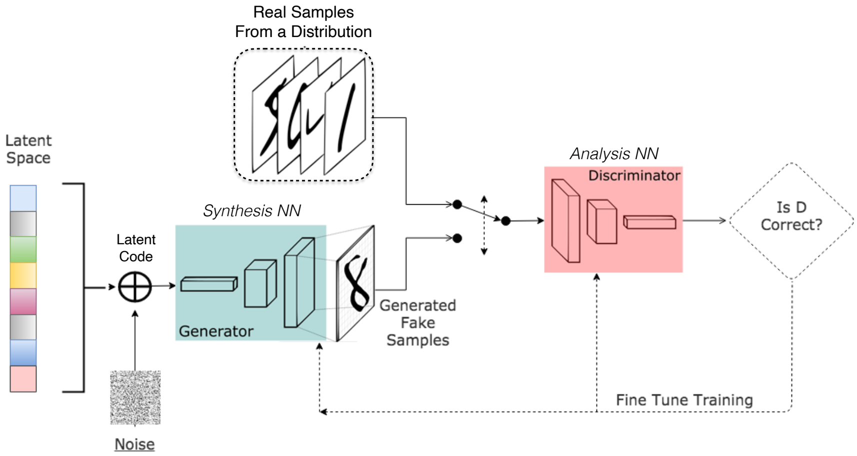
Recurrent Networks

- Learn a Sequence (time dependent process)



Generative Adversarial Networks

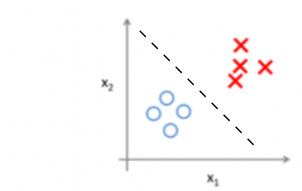
- Learn to Reproduce from a Distribution



Learning Methods

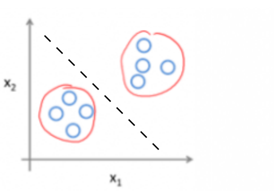
- Basic Methods

Supervised



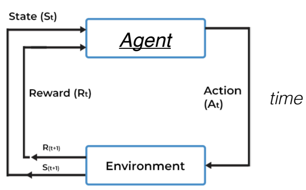
Labeled Data

Unsupervised



Unlabeled Data

Reinforcement



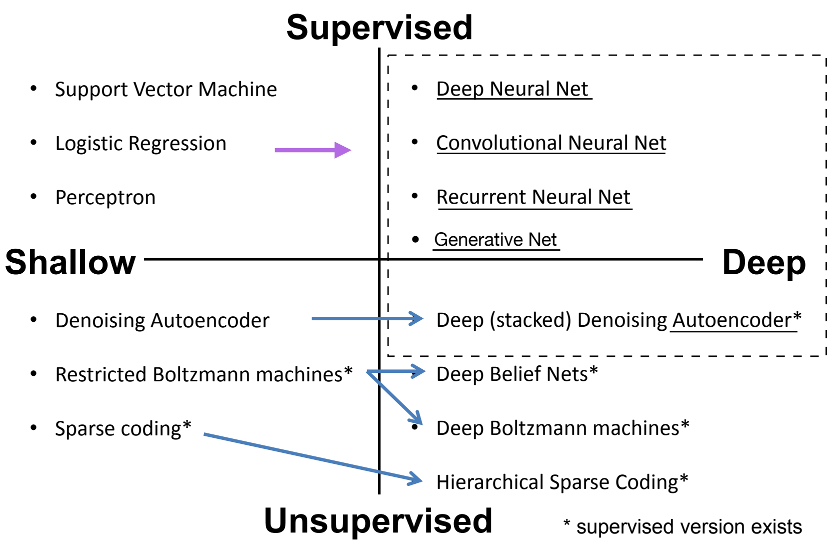
Simulation

- Advanced Methods
  - Semi-Supervised / Self-Supervised

Comparison

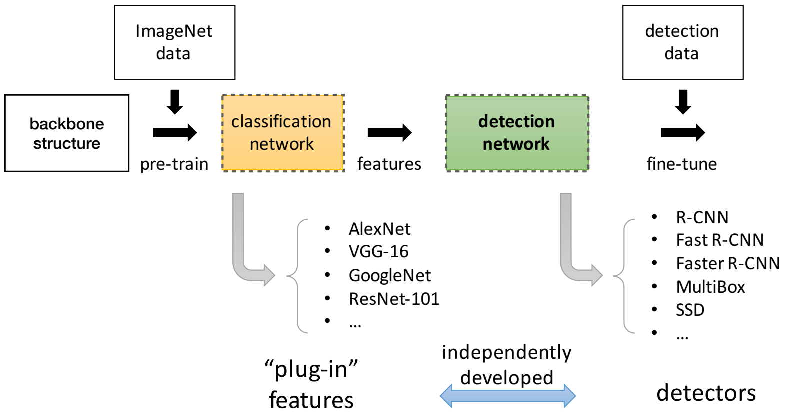
- Supervised Learning
  - End-to-end learning of deep architectures (e.g., CNNs)
  - Works well when the amounts of labels is large
  - Structure of the model is important (e.g. convolutional structure)
- Unsupervised Learning
  - Learn statistical structure or dependencies of the data from unlabeled data
  - Layer-wise training
  - Useful when the amount of labels is not large

Taxonomy



Combining Networks

- Example: Classification / Detection

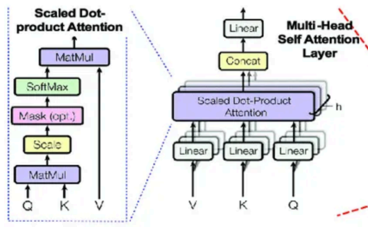


Recent Network Architectures

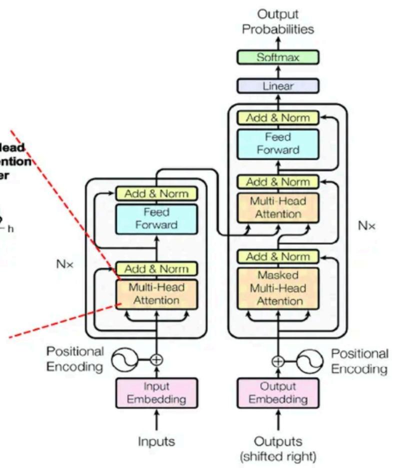
- Transformers
- Diffusion Networks
- Student / Teacher
- Mixture of Experts
- Contrastive Methods
- Recursive Language Models

Transformers

- Learn Embeddings (Structure)



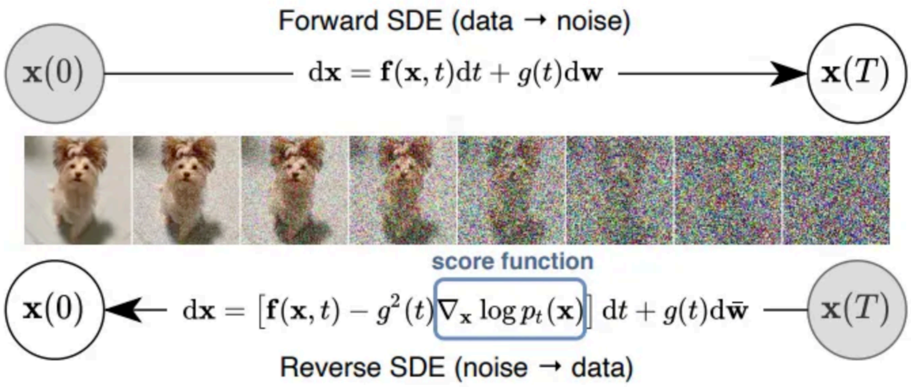
attention



embeddings

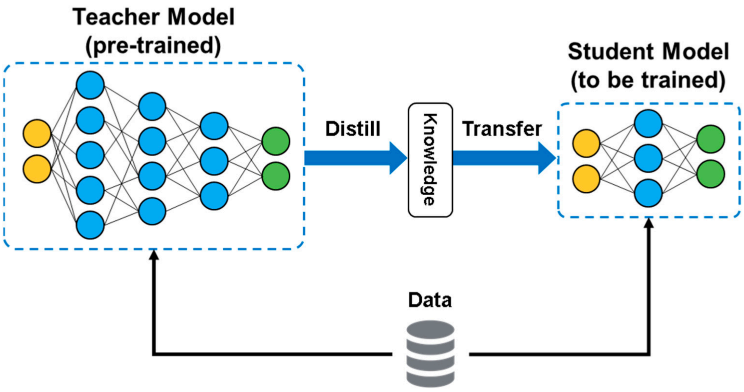
Diffusion Networks

- Learn a Distribution



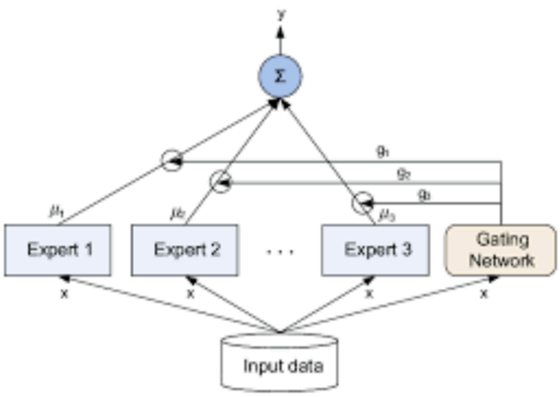
Student / Teacher

- Knowledge <u>Distillation</u>



Mixture of Experts

- Adaptative <u>Combination</u>



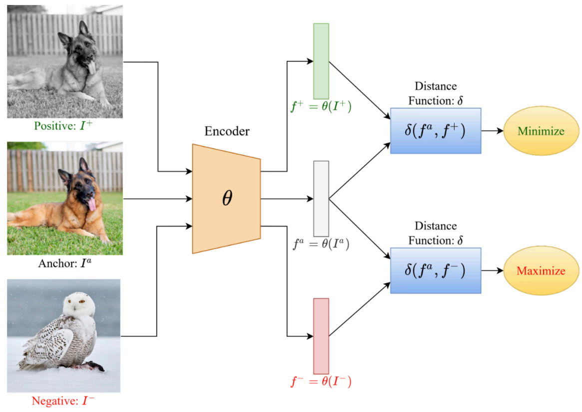
Contrastive Methods

- Self-Supervised Learning

Positive

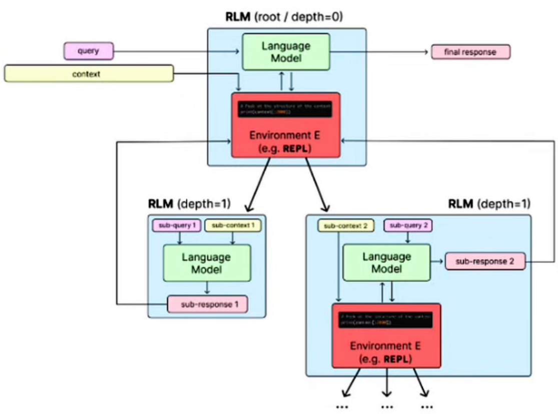
$\neq$

Negative



Recursive Language Models

- Active Inference by Recursive Evaluation



3D Graphics & Vision

Scope

Machine Learning Models for Media

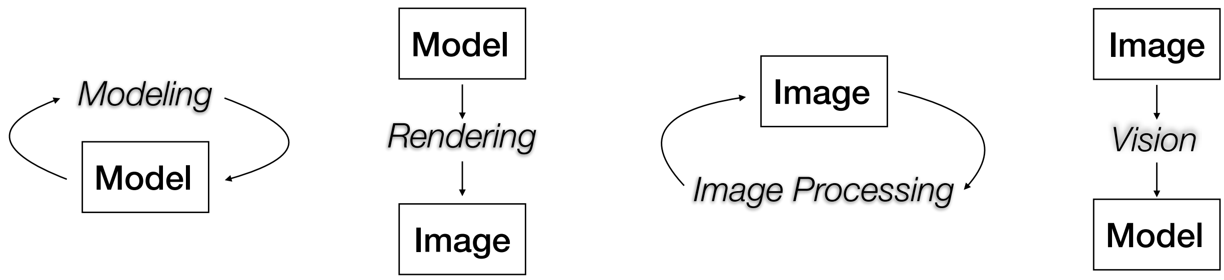
- 3D Graphics
  - Synthetic Images
- Vision
  - Scene Models
- *Applications*
  - *3D Animation / Games*
  - *Multimedia*
  - *Expanded Reality*

Evolution of the Area

Traditional

- Four Separated Areas

1960’s - 1990’s



Modern

- Integrated Areas 1990 - 2005

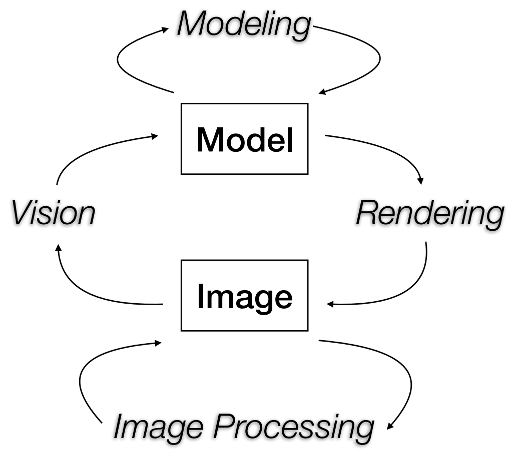
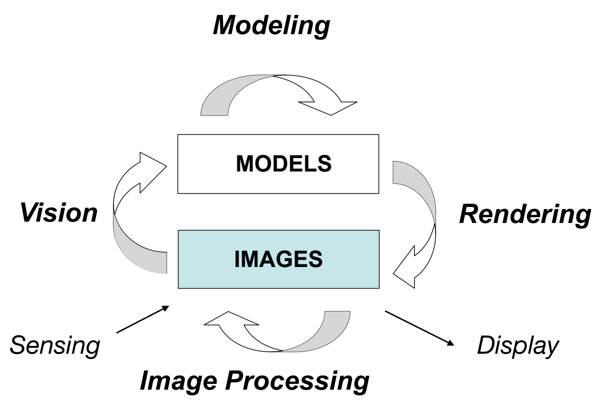


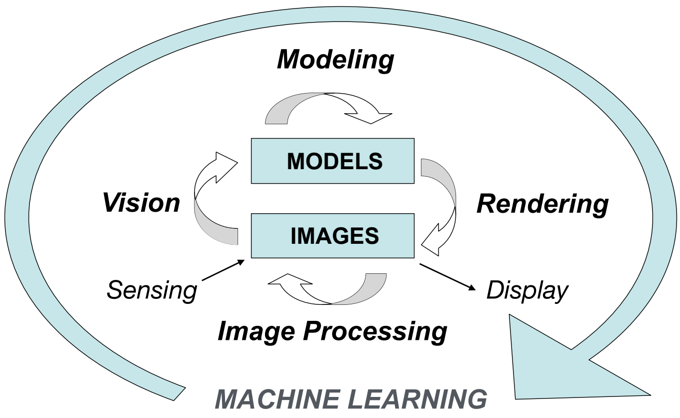
Image-Based

- From Appearance to Algorithms 2005 - 2015



AI Graphics / 3D Vision

- From Data to Networks 2015 - …



Basic Concepts
for
3D Graphics and Vision

Concepts

<u>*Computational Applied Mathematics for Media*</u>

- Paradigm of the 4 Universes
- Media Objects

4 Universes Paradigm

- Levels of Abstraction

Physical Universe

$\updownarrow$

Mathematical Universe

$\updownarrow$

Representation Universe

$\updownarrow$

Implementation Universe

(Gomes and Velho, 1995)

Example: Numbers

Scalar Quantities	length, area
Real Numbers	7.598
Floating Point Representation	$7598 * 10^{-3}$
IEEE Standard	mantissa exponent

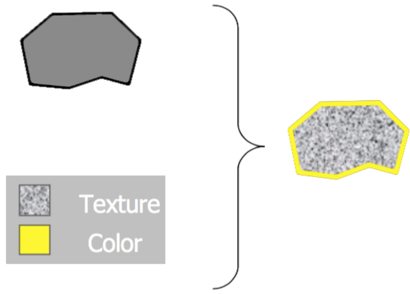
Issues

- Study each Level
  - Objects in level
- Relation between Levels
  - How to Map (e.g. Correspondences)
  - Equivalences and Losses

Media Objects

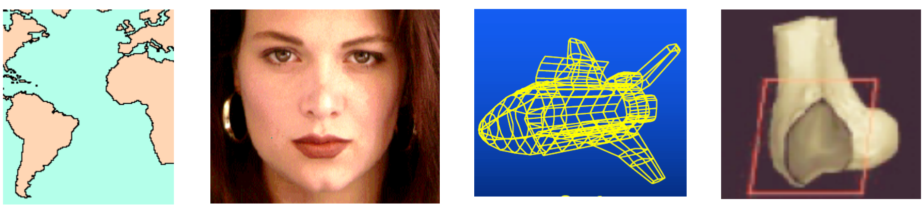
$$o = (U, f) \qquad\qquad f : U \subset \mathbb{R}^m \to \mathbb{R}^n$$

- Support $\boldsymbol{U}$
  - Shape / Geometry
- Properties $\boldsymbol{f}$
  - Attributes / Appearance



Unifying Concept

Examples



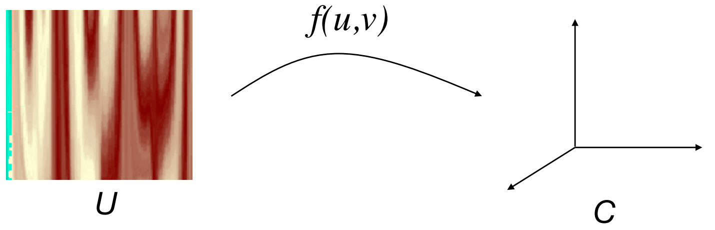
drawings **images** **surfaces** **volumes** *…time …*

Static or Dynamic

- General Encompassing Definition (Gomes and Velho, 1996)

Defining Attributes

- Ex: <u>Image</u>
  - Simple Shape: $U = [0,1] \times [0,1]$
  - Complex Attributes: (color, density, etc) $f(u,v) = [r, g, b, a, ..]$



Defining Shape

- Ex: <u>Curves</u>

  (unit circle)

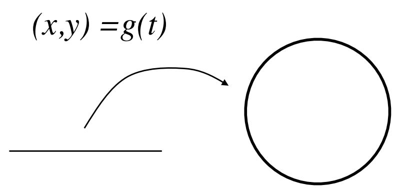
  - Parametric Description

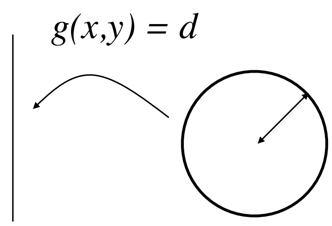
    $g(\theta) = (\cos\theta, \sin\theta)$

    $\theta \in [0,2\pi]$

  - Implicit Description

    $g^{-1}(0) = \{(x,y) \,|\, x^2 + y^2 - 1 = 0\}$





Putting it All Together

- Ex: 3D Object
  (Diana statue)
  - Shape:
    - Scanned 3D Surface
    - Solid Object
  - Properties:
    - Material Marble

(Wann Jensen et al.)

Questions

- How to Define:
  - Function (model)
  - Support (domain)
  - Attribute (range)

Mathematical Models

- Deterministic
  - Function (one object)
- Probabilistic
  - Stochastic Process (class of objects)
- Procedural
  - Generators + Operators (algebra / expression)
- Neural
  - Data + AI Models (cognitive learning)

Computation and Representation

- How to Compute
- Conversion between Models / Representations

✳ Hybrid Models

Media Objects Operators

- Spaces of Media Objects

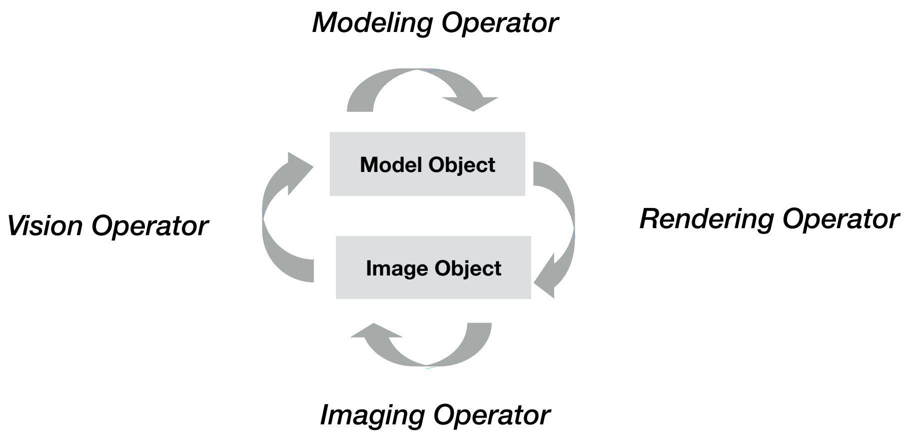
$$o \in \mathcal{O}$$

- Media Objects Operators

$$T : \mathcal{O}_1 \mapsto \mathcal{O}_2$$

- Intra $(i = j)$ / Inter Levels $(i \neq j)$
- Representation

Landscape of Operations



Direct Problems

- Given T and x, find y

$$y = Tx$$

Ex: Visualization

- x is the scene (geometry, lighting, camera)
- T is the rendering operator
- y is the rendered image

Inverse Problems

- Given T and y, find x such that $T(x) = y$

  Ex: Object Recognition

  - y is a captured image
  - T is an acquisition system
  - x is a template

- Given x and y, find T

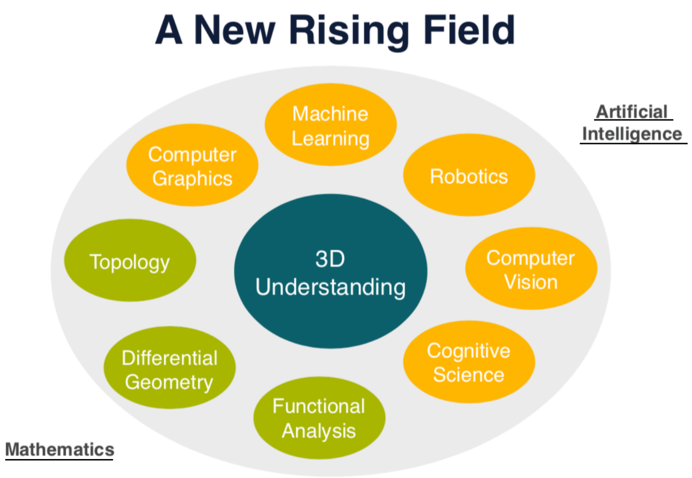
  Ex: Camera Calibration

  - x is a calibration pattern
  - y is a transformed pattern
  - T is a projective transformation

3D Deep Learning

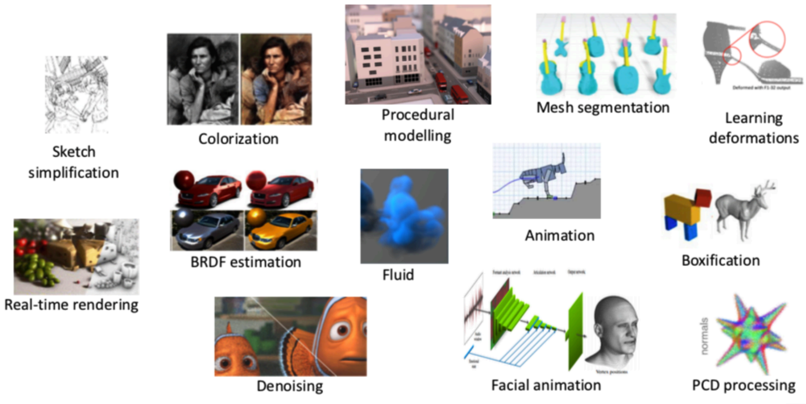
Slides from: (Mitra et. al, "CreativeAI: Deep Learning for Computer Graphics", 2019)

Deep Learning for 3D Data



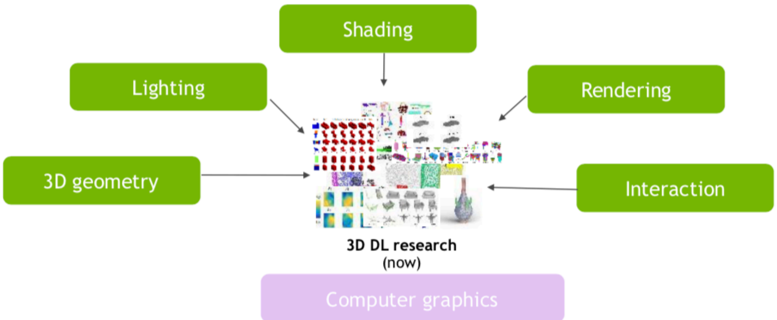
(Mitra et. al, 2019)

Applications



(Mitra et. al, 2019)

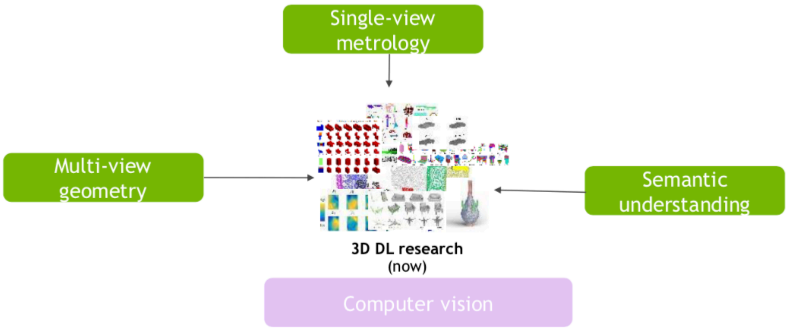
Computer Graphics



(Synthesis)

(Mitra et. al, 2019)

Computer Vision



(Analysis)

(Mitra et. al, 2019)

Synergistic Opportunities

What is Special about CG?

1. Regular data structure and easy to parallelize
   (e.g., image translation)

2. Many sources of input data — model building
   (e.g., images, scanners, motion capture)

3. Many sources of synthetic data — can serve as supervision data
   (e.g., rendering, animation)

4. Many problems in generative models and need for user-control

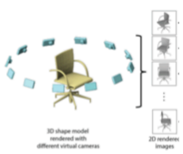
(Mitra et. al, 2019)

Representations in CG

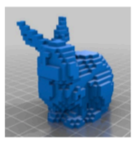
- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)
- Point clouds (e.g., collection of points)
- Animation (e.g., skeletal positions over time; cloth dynamics over time)

(Mitra et. al, 2019)

Representation Challenge



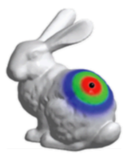
Multi-view



Volumetric

Part Assembly

Point Cloud



Mesh (Graph)

$$F(x) = 0$$

Implicit Shape

(Mitra et. al, 2019)

3D Deep Learning

Goal: Learn a Parametric Function (Distribution)

$$f_\theta : \mathbb{X} \longrightarrow \mathbb{Y}$$

θ: function parameters, these are learned

$\mathbb{X}$: source domain

$\mathbb{Y}$: target domain

Examples:

Image Classification: $f_\theta : \mathbb{R}^{w \times h \times c} \longrightarrow \{0, 1, \ldots, k-1\}$

$w \times h \times c$: image dimensions $\quad k$: class count

Image Synthesis: $f_\theta : \mathbb{R}^{n} \longrightarrow \mathbb{R}^{w \times h \times c}$

n : latent variable count $\quad w \times h \times c$: image dimensions

(Mitra et. al, 2019)

Problems in GC

analysis:

- Feature detection (image features, point features) $\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$
- Denoising, Smoothing, etc. $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Embedding, Metric learning $\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d}$

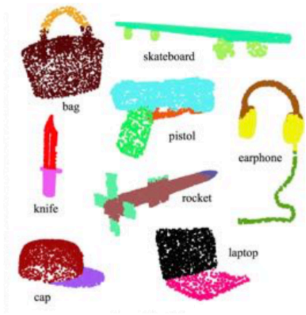
synthesis:

- Rendering $\mathbb{R}^{3n} \rightarrow \mathbb{R}^{m \times m}$
- Animation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Generative models $\mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m}$

(Mitra et. al, 2019)

3D Analysis

Classification



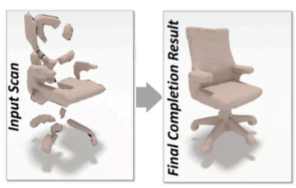
Parsing
(object/scene)

Correspondence

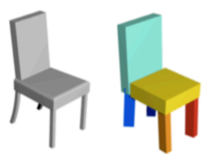
(Mitra et. al, 2019)

3D Synthesis

Monocular
3D reconstruction



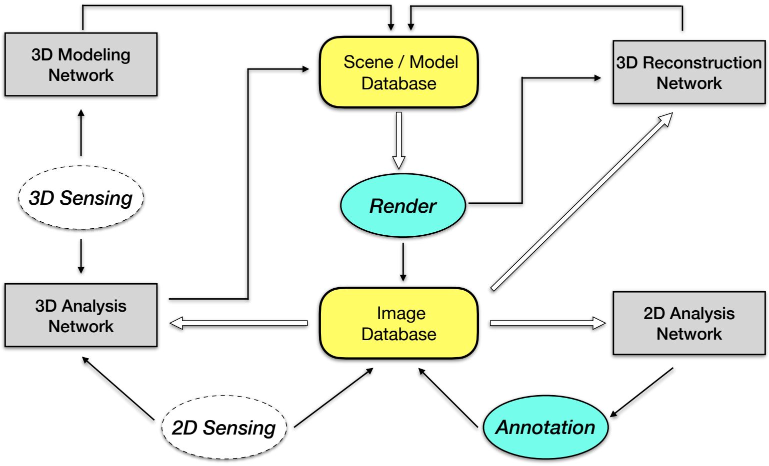
Shape completion



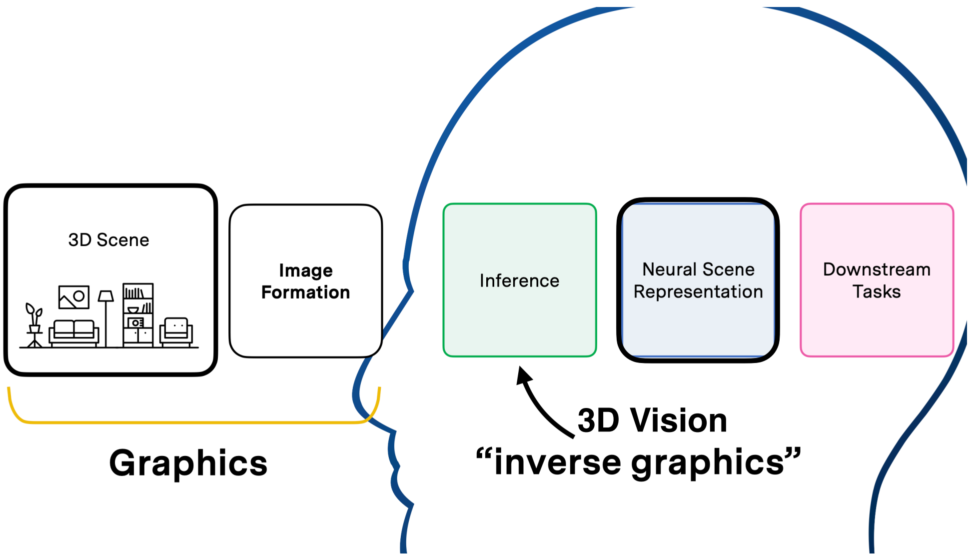
Shape modeling

(Mitra et. al, 2019)

A.I. Graphics



Neural 3D Vision



(V. Sitzmann, 2024)