3D Vision
Geometry and Learning

杨佳琪
计算机学院
jqyang@nwpu.edu.cn

Lists of Contents

Lists of Contents

What is 3DV?

3D Sensing	Pose estimation	3D reconstruction	3D understanding
Estimate the depth of 3D object/scene, or sample data on 3D object/scene	Estimate the 6-DoF of camera, object pose. In addition, track pose in sequences.	Reconstruct the 3D shape of 3D object/scene object, estimate the motion information of the reconstructed object.	3D object detection, recognition, retrieval, tracking, etc. 3D scene labeling, segmentation, etc.

3DV demos

3DV demos

3DV demos

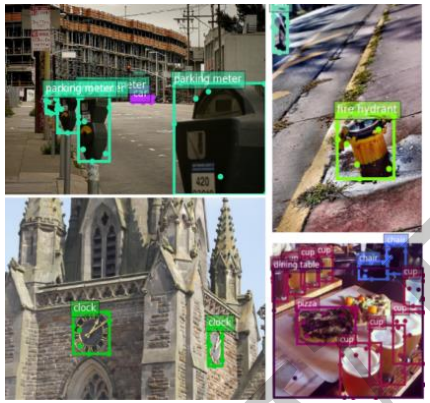
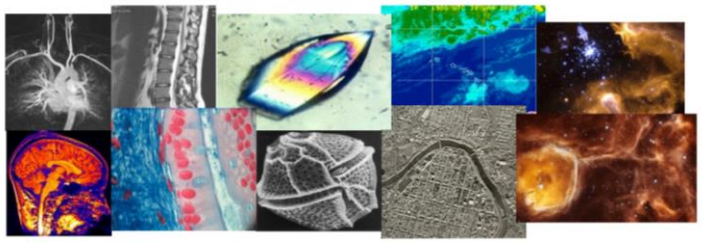
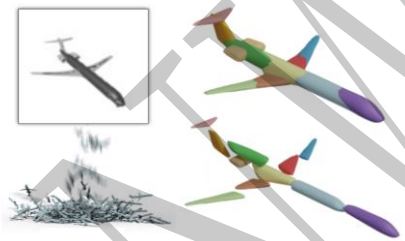
Review on 3D vision history
(ref. 清华大学 刘烨斌教授)

主要内容

- 计算机视觉与三维视觉
- 三维视觉的应用需求
- 三维视觉的发展历史
- 未来的三维视觉与计算机视觉

计算机视觉

□计算机视觉的产业价值：产业应用的技术基础

HUAWEI NVIDIA ByteDance

Alibaba Google Research Microsoft

计算机视觉的需求无处不在

机构		2017年	增长率	2020年	增长率
国内机构	CAICT	80.2亿元	166%	600亿元	96%
	腾讯研究院	>70亿元	/	660亿元	~110%
	艾瑞咨询	40亿元	264%	725亿元	162.7%
	艾媒iiMedia	68亿元	124%	780亿元	125.5%
国外机构	Ganter	10.7亿美元	110%	110亿美元	117%
	CB Insights	13.5亿美元	超过150%	160亿美元	128%

视觉市场规模飞速增长

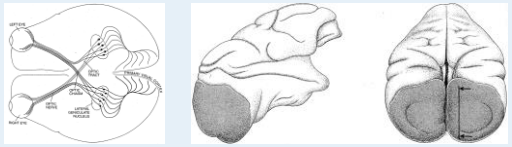
计算机视觉是人工智能最重要的组成部分

计算机视觉

□计算机视觉的起源与目标

理论基础

生物

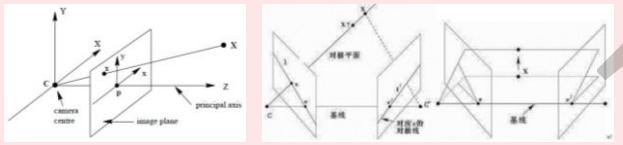


Hubel, D. H., & Wiesel, T. N. (1979). Brain mechanisms of vision. *Scientific American*

视觉分层理论的出现为计算机视觉奠定理论基础

视觉原理→语义理解

数学



Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

多视图几何的出现为计算机视觉提供数学工具

测量几何→三维感知

用计算机模拟人对物理世界的视觉感知，启发对人脑与智能的理解

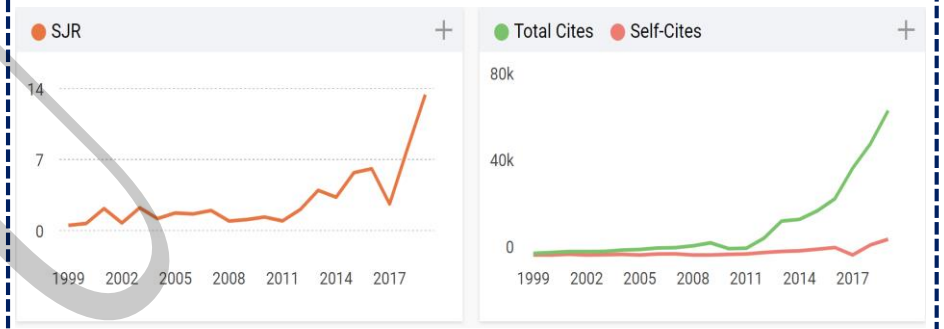
计算机视觉

计算机视觉的学术影响力

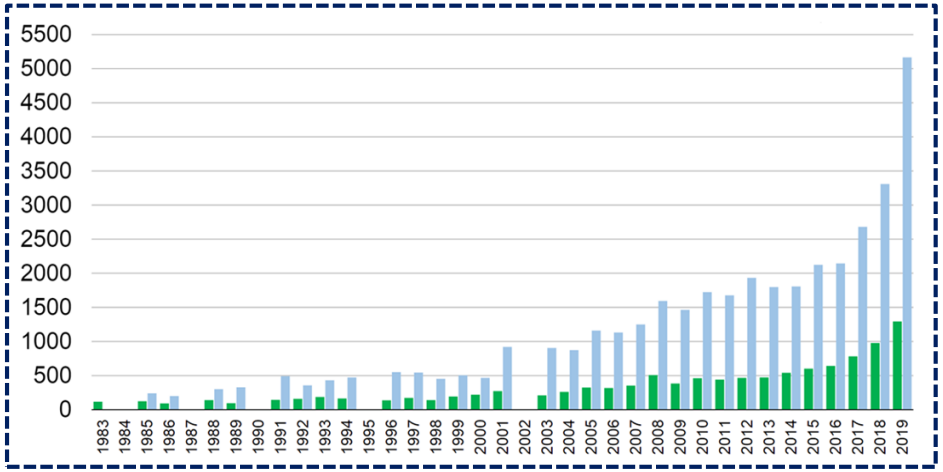
排名	期刊或会议	h5-指数	h5-中位数
1	Nature	376	552
2	The New England Journal of Medicine	365	639
3	Science	356	526
4	The Lancet	301	493
5	**IEEE CVPR**	**299**	**509**
6	Advanced Materials	273	369
7	Nature Communications	273	366
8	Cell	269	417

* h5 指数是指在过去整整 5 年中所发表文章的 h 指数。h 指在 2015-2019 年间发表的 h 篇文章每篇至少都被引用过 h 次的最大值。

谷歌学术发表2020年最新的学术期刊和会议影响力排名



CVPR会议各项指标逐年上升



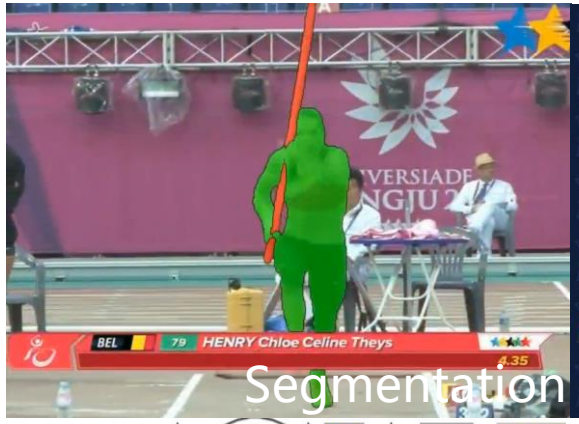
CVPR会议投稿量逐年上升

计算机视觉是国际学术热点，学术影响力名列前茅

计算机视觉

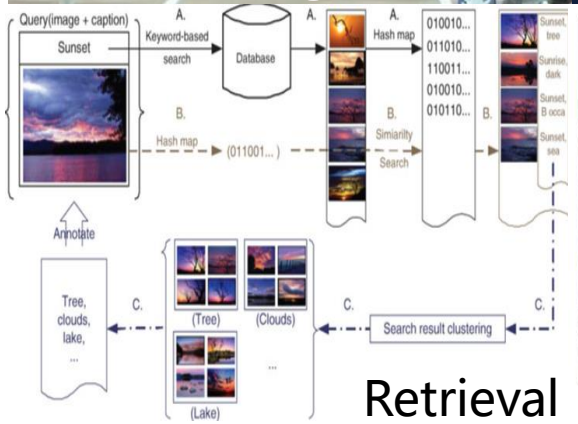
计算机视觉的研究内容

语义理解



Segmentation

Recognition

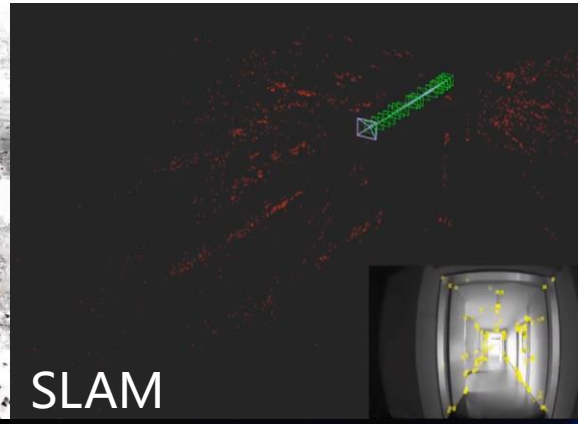


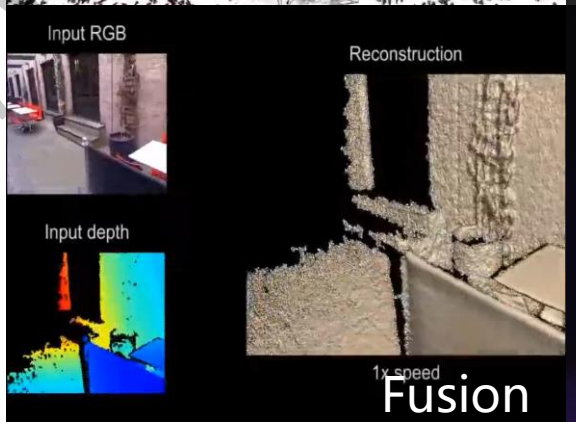
Retrieval

Caption

三维感知


Stereo


SLAM



Fusion

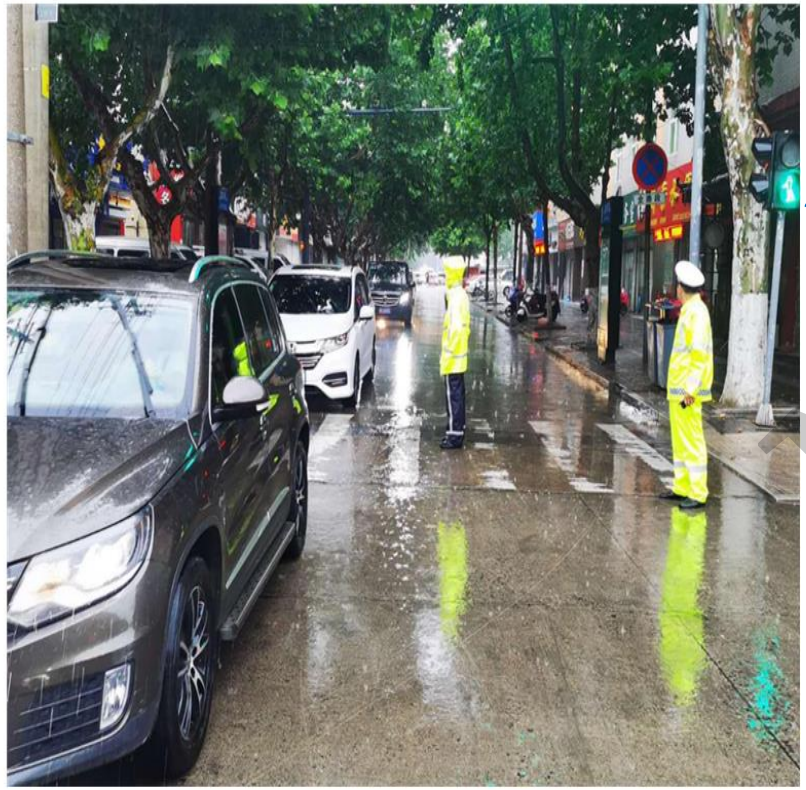

Light Field

语义理解和三维感知是计算机视觉的两大板块

三维视觉

三维视觉：从三维感知到三维理解

- 基于图片、视频以及各类深度传感器信息，采用几何、统计以及优化等数学工具对现实世界进行**三维测量、定位、建模及理解。**



- 前面是否有车？有多少车？
- 前面是否有交警或行人？
- 前面是否有障碍物？
- 交警离我有多远？
- 两车之间的距离多少？
- 交警看到什么？想什么？
- 下一秒会发生什么？
- ……

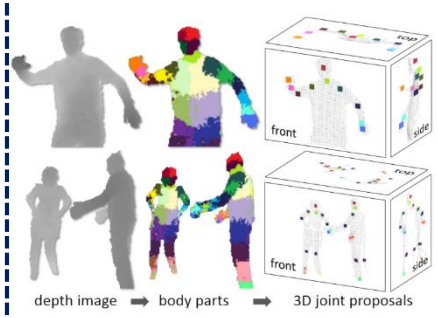
二维语义理解可以完成

需要三维视觉！

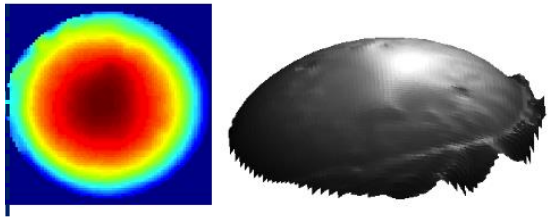
三维视觉是实现视觉智能的不可或缺的手段

三维视觉

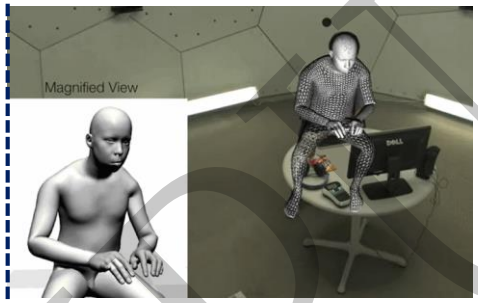
三维视觉受到极大关注



Depth Pose
2011 最佳论文



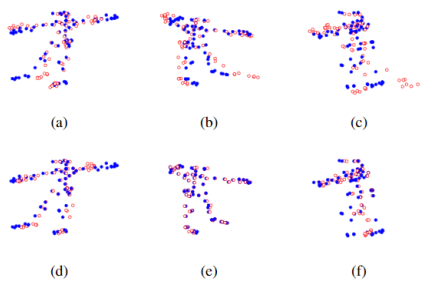
Shape from BRDF
2014 最佳论文



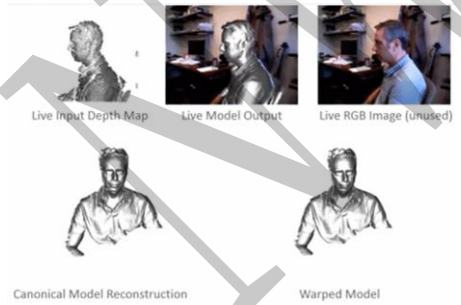
TotalCapture
2018 最佳学生论文

Fermat Paths
2019 最佳论文

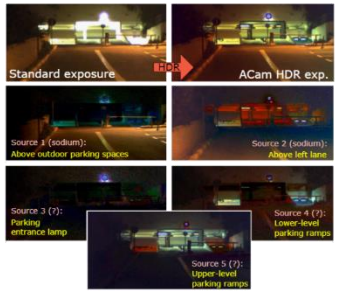
BSP-Net
2020 最佳学生论文



Non-Rigid SfM
2012 最佳论文



DynamicFusion
2015 最佳论文



Electric Grid
2017 最佳学生论文

Unsup3D
2020 最佳论文

近10年来CVPR最佳论文中，三维视觉占据半壁江山

主要内容

- 计算机视觉与三维视觉
- 三维视觉的应用需求
- 三维视觉的发展历史
- 未来的三维视觉与计算机视觉

三维视觉的应用需求

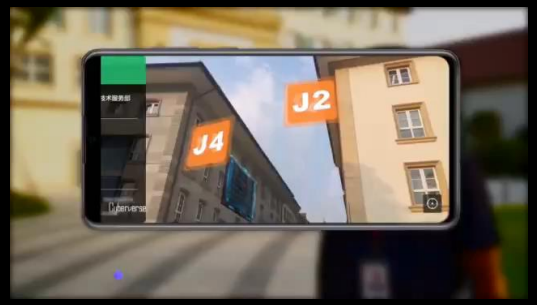
地球级数字新世界

*华为、微软、谷歌、苹果等企业纷纷投入地球级数字世界建设

人物三维重建

虚实结合城市街景

全息广告牌及点评



三维智能助手

AR多人互动游戏

AR步行导航

厘米级3D精准空间定位，真实世界与物理世界的无缝融合，万物互联的智能世界

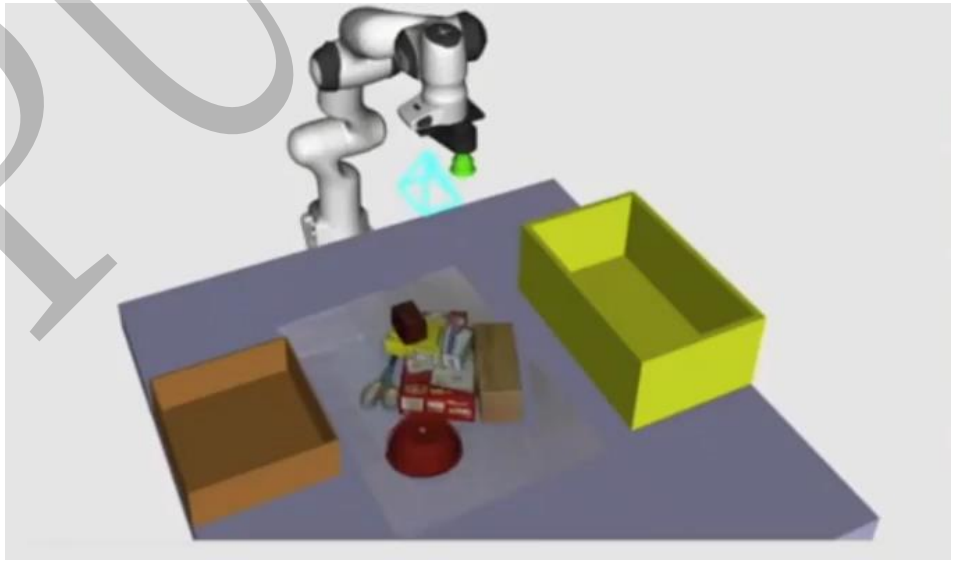
三维视觉的应用需求

机器人抓取系统

构建与现实同步的三维数字环境是机器人理解环境、自主作出判断的必要前提

3. 抓取

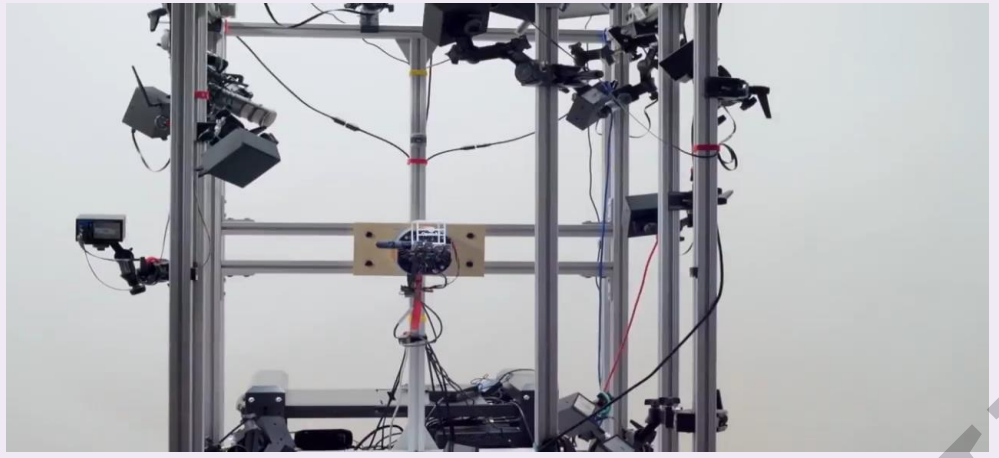
1. 三维重建



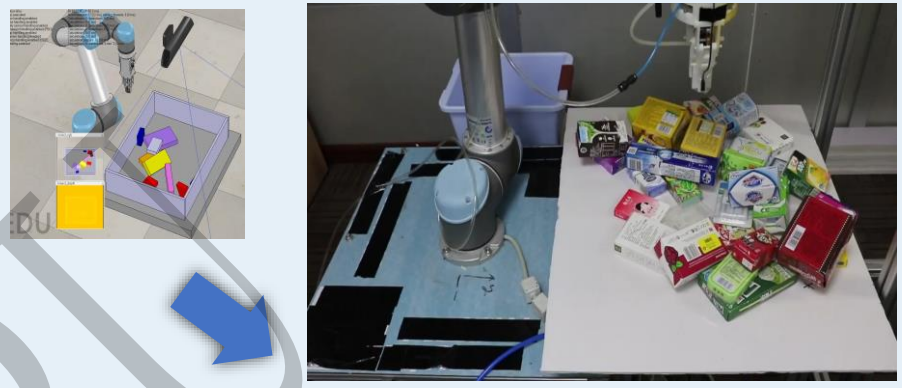
2. 预演、仿真、学习

三维重建是机器人观察世界、理解世界、与世界互动的必要手段

三维视觉的应用需求

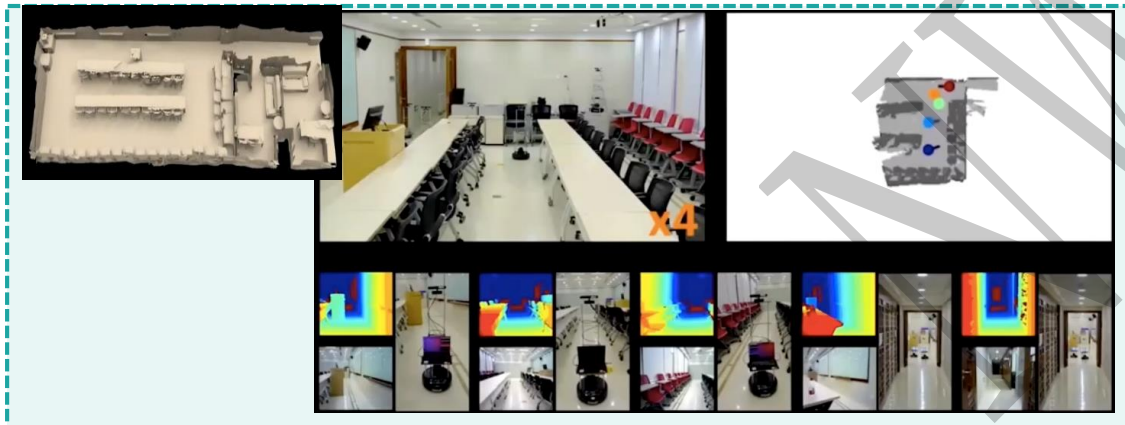


基于三维视觉的仿生人手



Huaping Liu et al. II2020

基于三维重建的复杂环境交互



Siyan Dong et al. SIGGRAPH 2019

多机器人协同室内三维重建

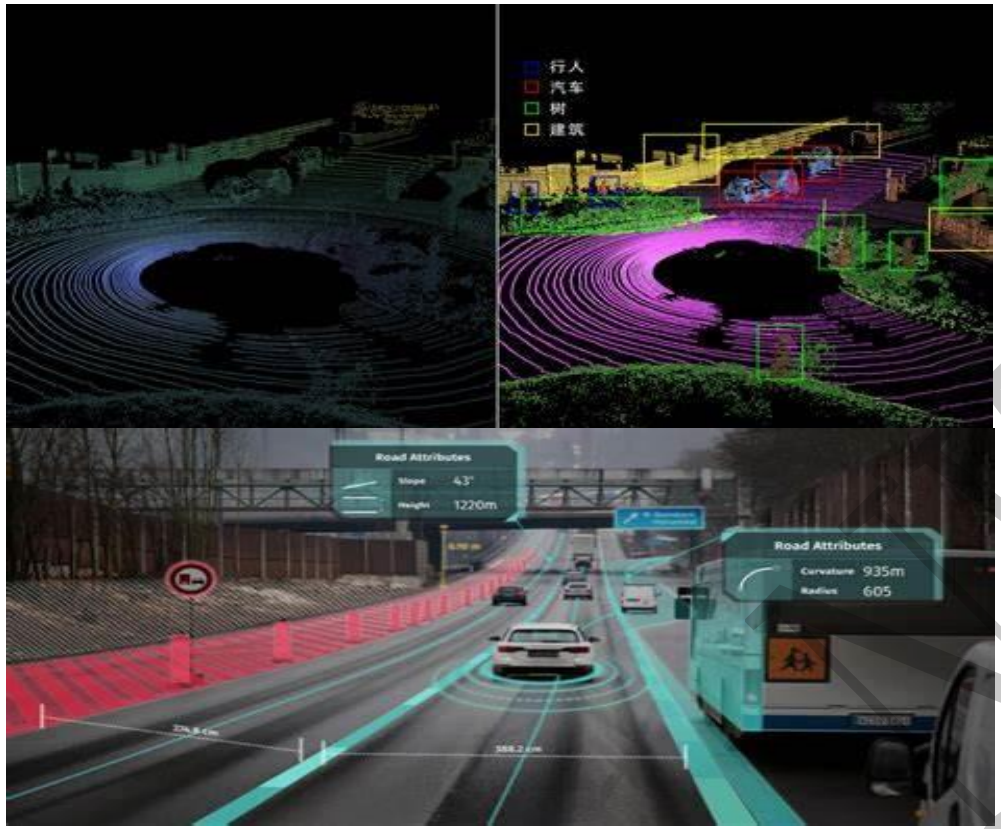
Dharmadhikari et al. ICRA20

基于三维重建的自主洞穴探索无人机

三维重建在各种机器人应用中扮演着不可或缺角色

三维视觉的应用需求

自动驾驶感知系统



高精地图构建与实时三维感知

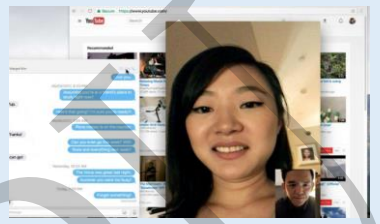
行驶轨迹可视化与预测

精准三维地图构建与实时三维环境感知是自动驾驶智能决策的关键

三维视觉的应用需求

全息交互与通信



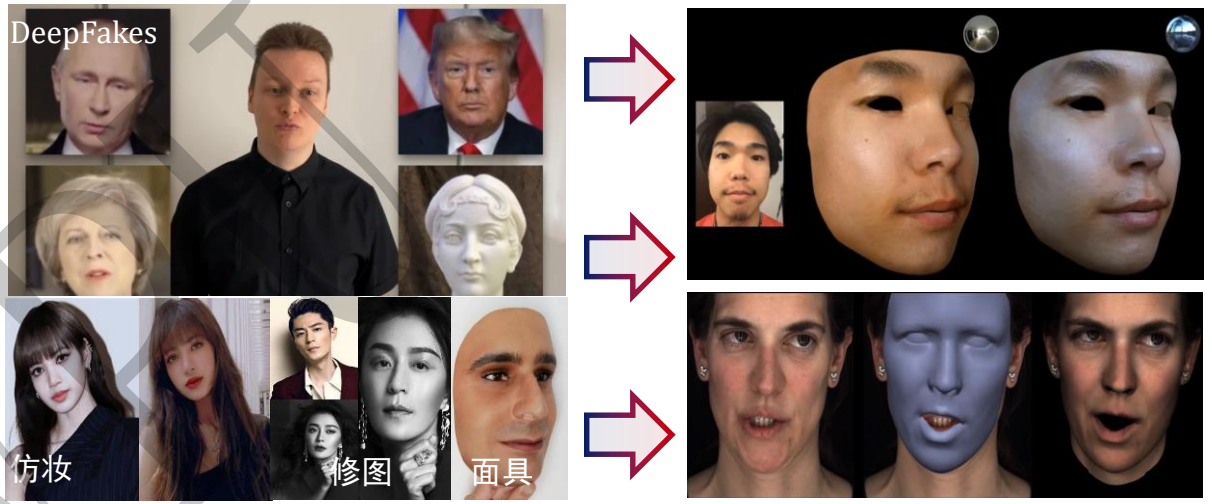

邮件 公元前1500 → 电报 1830年 → 语音电话 1860年 → 视频会议 2000年 → 全息通讯 当前到未来

动态三维重建用于未来全息交互，是媒体通信技术的革新

三维视觉的应用需求

三维视觉驱动支付级人脸识别

基于iPhone X前置深度镜头的人脸重建



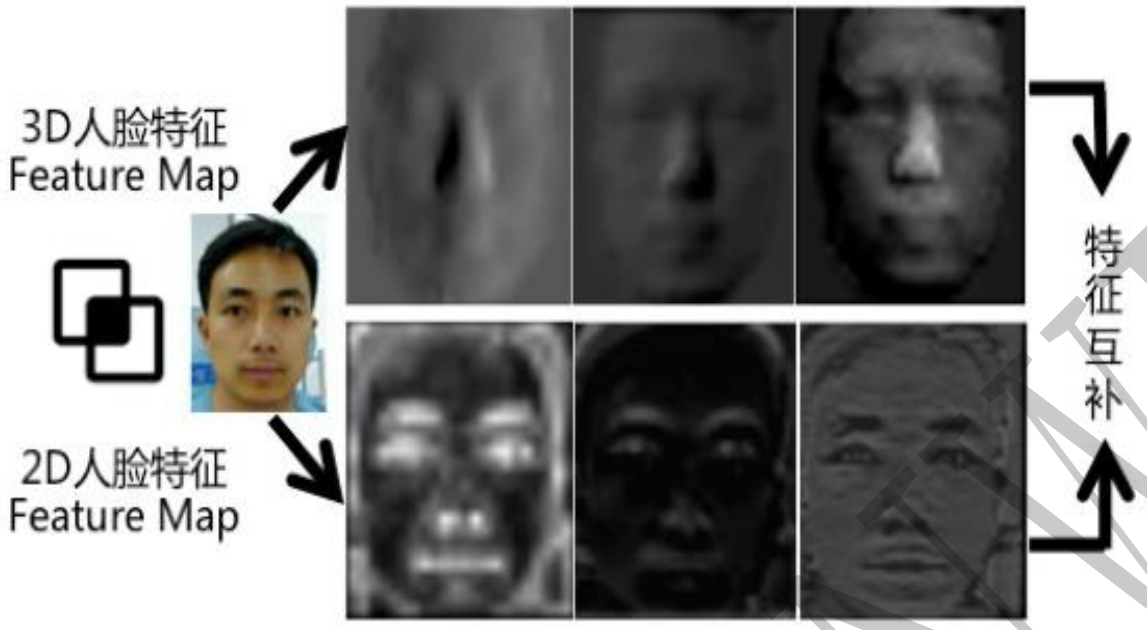
二维人脸识别困境

三维人脸识别

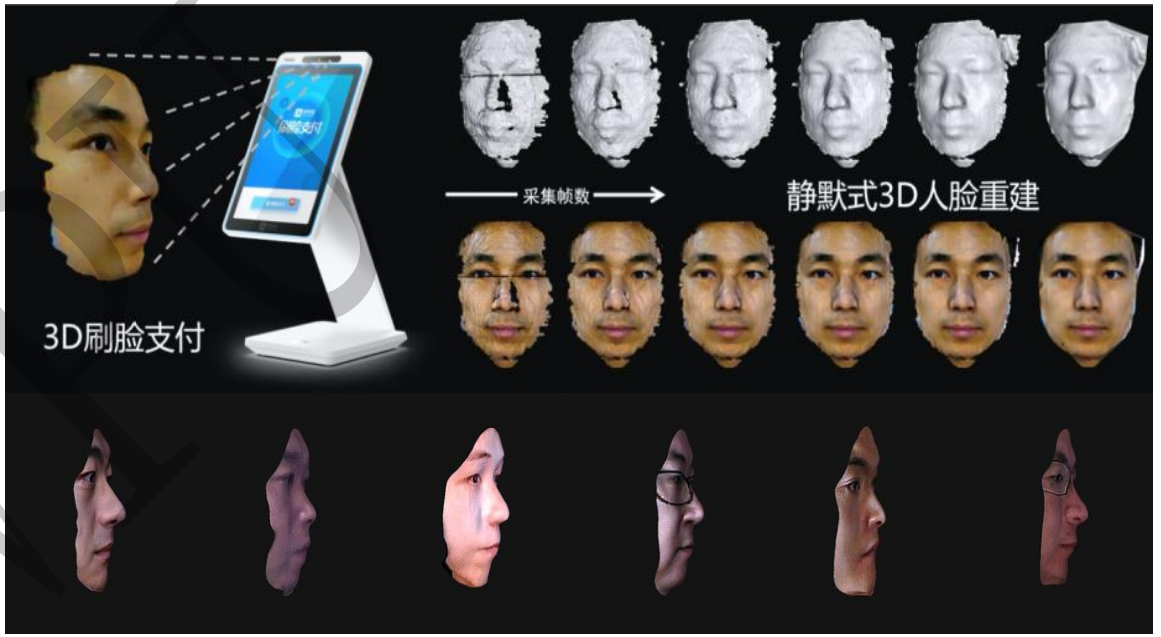
- 三维采集硬件在移动端快速发展
- 三维识别克服二维困境，将刷脸这一高隐私、高风险、高价值的技术正式带向产业应用

三维视觉的应用需求

三维视觉应用于人脸识别



二维与三维结合的多模态人脸识别



三维人脸重建应用于人脸识别

与蚂蚁金服合作，实现了**三维**人脸识别，相比二维算法，精度显著提升：

- 1对1的人脸识别**错误率下降两个数量级**；1比N识别可支持的**数据规模扩大约30倍**
- 三维活体识别和高精度三维重建将进一步提升识别精度，保障**支付级别的安全性**

三维视觉的应用需求

□动物模型行为计算

动物模型是生命科学的媒介


MONKEYS AND MICE ENLISTED TO FIGHT CORONAVIRUS

Animal models can reveal how infections develop, and aid drug and vaccine efforts.

By Ewen Callaway

With no sign that the coronavirus is going away, researchers are looking to animals to understand COVID-19. They are testing monkeys, mice and even ferrets to answer key questions about the disease and to fast-track potential drugs and vaccines for clinical trials.

Teams in China have reported initial findings from studies in which they infected monkeys and mice engineered to be susceptible to infection by the coronavirus, called SARS-CoV-2. And a team at the Australian Animal Health Laboratory in Geelong is studying the

Why does the coronavirus spread so easily?

A microscopic feature could make the virus more infectious than the SARS virus.

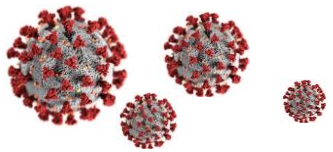
As the number of coronavirus infections passes 100,000 worldwide, researchers are racing to understand what makes it spread so easily.

A handful of genetic and structural analyses have identified a key feature of the virus — a protein on its surface — that might explain why it infects human cells much more readily than does the coronavirus that causes severe acute respiratory syndrome, or SARS.

Other groups are investigating the doorway through which the new coronavirus enters human tissues — a receptor on cell membranes. The cell

Nature, 2020.03

国际顶级学术期刊**nature**在3月份发文指出，建立猴子和小鼠等**动物模型**来对抗**新冠肺炎**



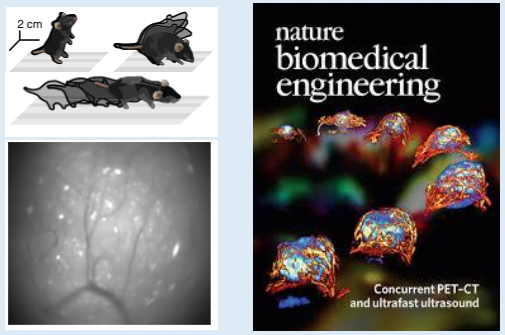
三维行为是动物固有属性

- 脑科学
- 疫苗与药物生产
- 疾病成因研究

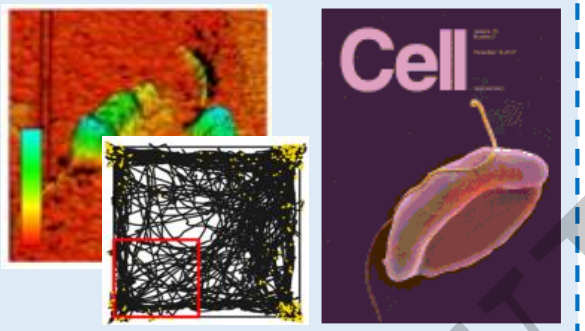
动物的三维行为计算是生命科学研究新工具

三维视觉的应用需求

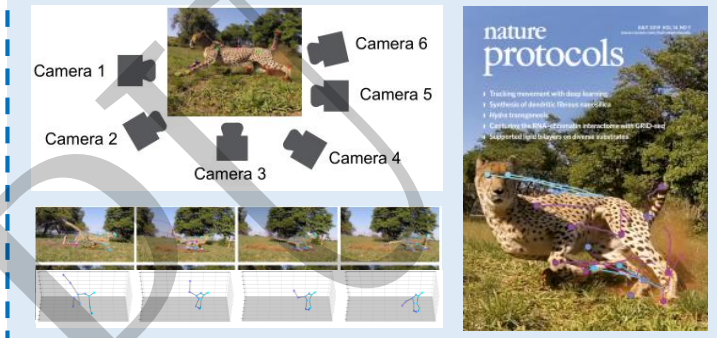
近年来，动物三维行为计算受到广泛关注



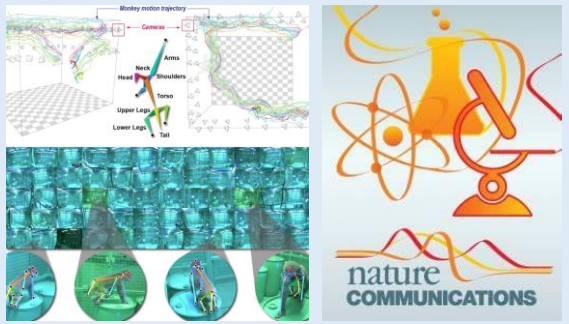
2018年：小鼠三维行为与大脑纹状体映射



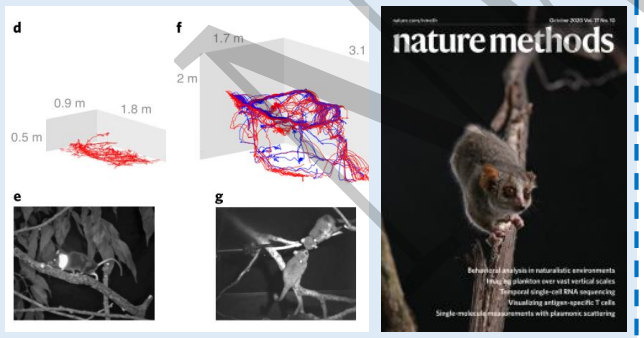
2019年：小鼠三维姿态与自闭症行为表现



2019年（封面文章）：猎豹三维姿态与物种保护



2020年：恒河猴三维姿态与三维聚类



2020年（封面文章）：尖嘴猴三维跟踪与大脑地图

2020年：动物表情行为与神经理解

未来

更高维 形态建模

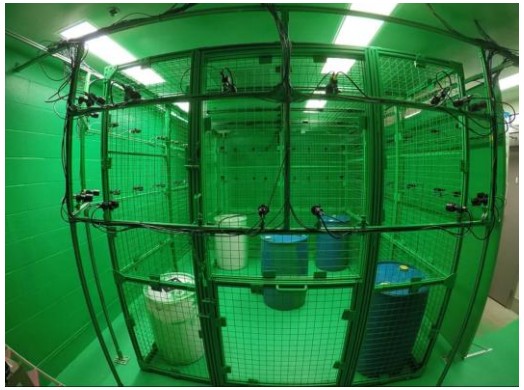
更丰富 交互运动

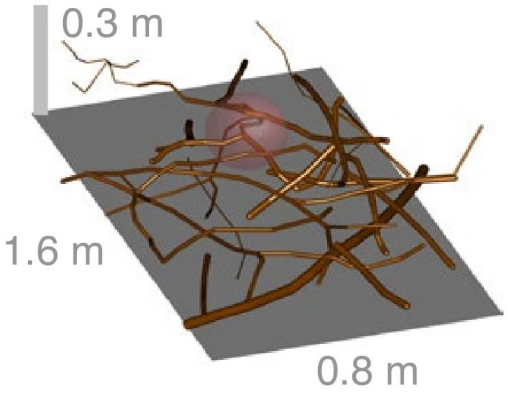
更深层 科学发现

三维视觉的应用需求

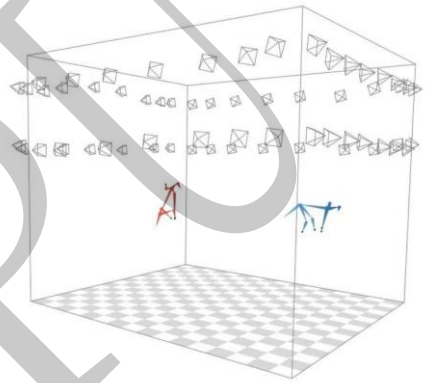
动物三维行为计算的价值

三维实验场景





三维行为计算



生命科学发现

明尼苏达大学

动物：恒河猴

结论：揭示灵长类动物的三维刻板行为的存在，以及三维交互行为的量化

瑞士日内瓦大学

动物：尖嘴狨猴/小鼠

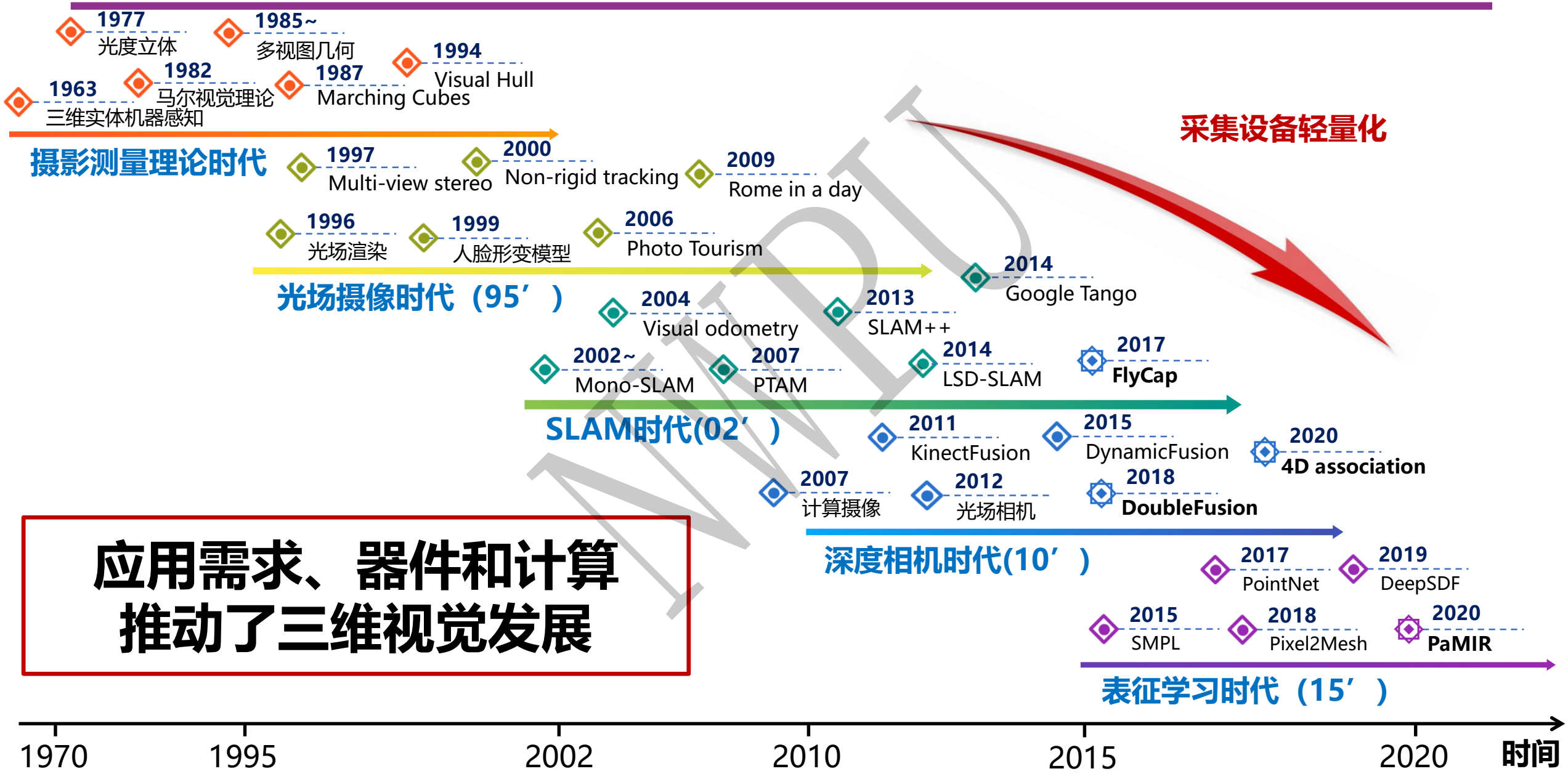
结论：在类自然环境下揭示了灵长类动物海马区有类似位置细胞的活动

三维行为计算新方法为生命科学研究带来新发现

主要内容

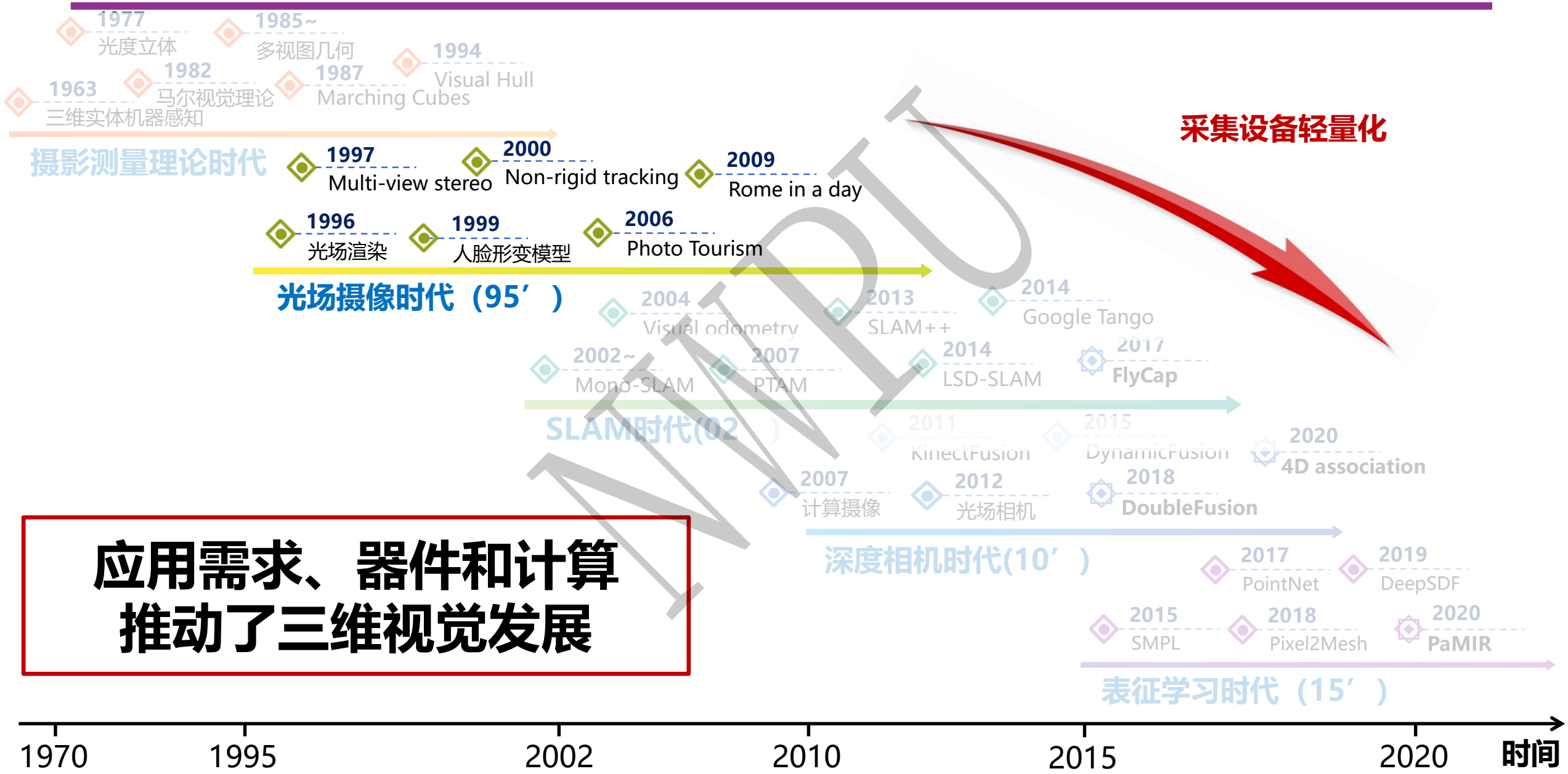
- 计算机视觉与三维视觉
- 三维视觉的应用需求
- 三维视觉的发展历史
- 未来的三维视觉与计算机视觉

三维视觉的发展历史



应用需求、器件和计算推动了三维视觉发展

三维视觉的发展历史

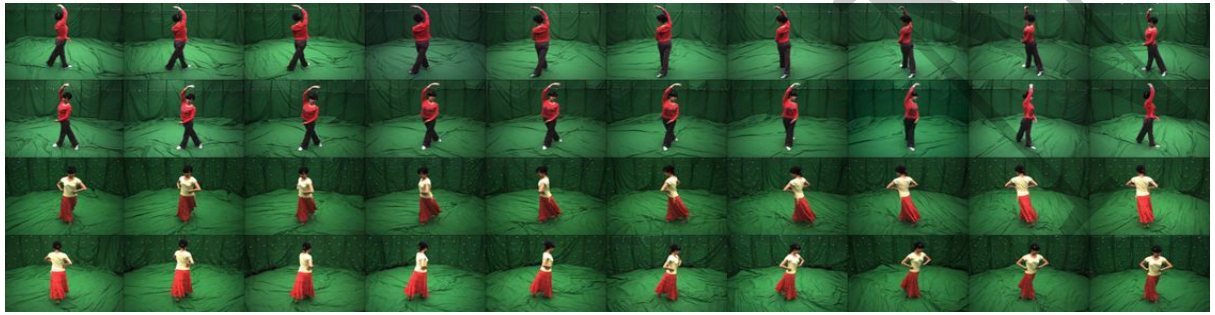


应用需求、器件和计算推动了三维视觉发展

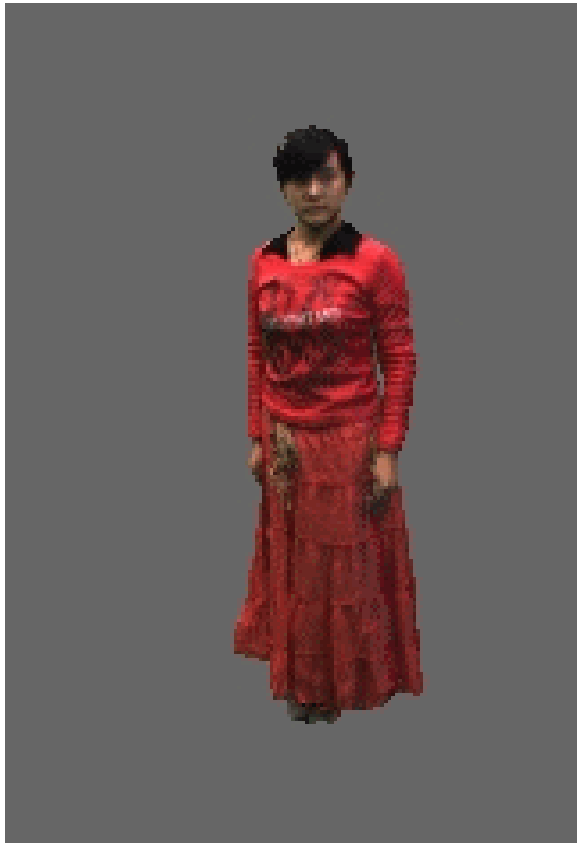
三维视觉：光场摄像时代

MVML Dome：密集光场三维数据获取的时代

2005年光场相机　2008年视角-光照协同采集装置



动态三维重建结果

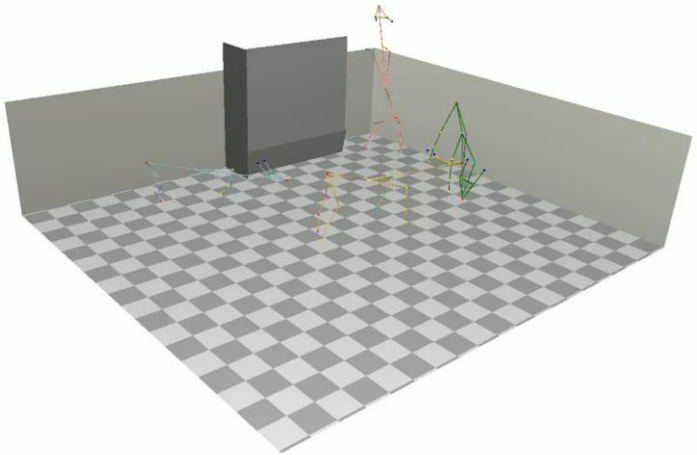


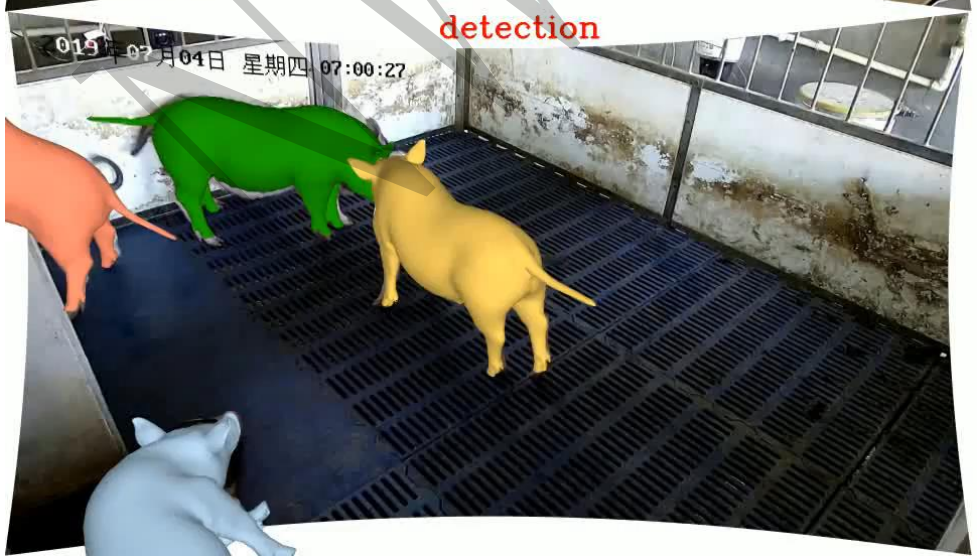
构建40相机、680个光源组成的国际首个密集视点密集光照的采集装置

三维视觉：光场摄影时代

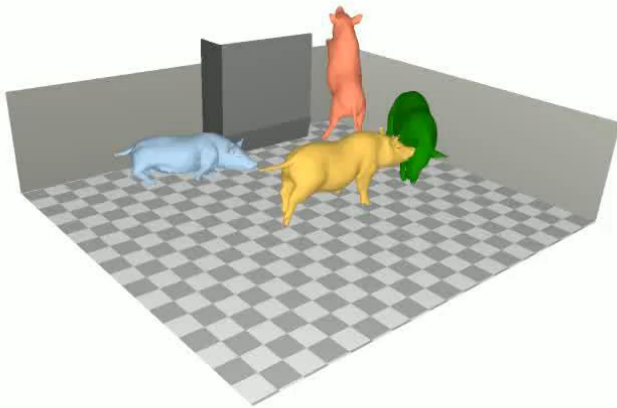
三维行为计算探索渐冻症早期症状

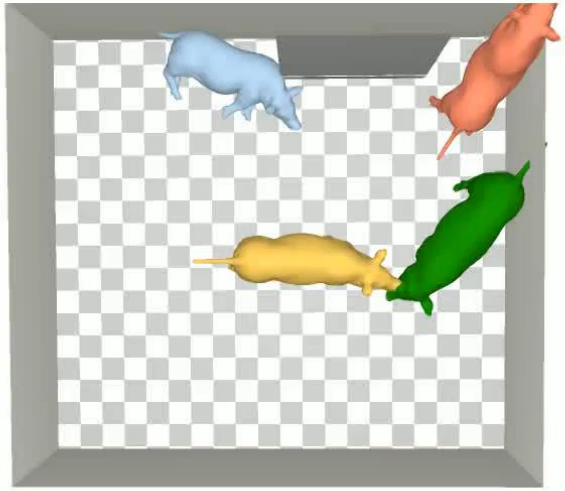
三维场景与轨迹





三维表面运动

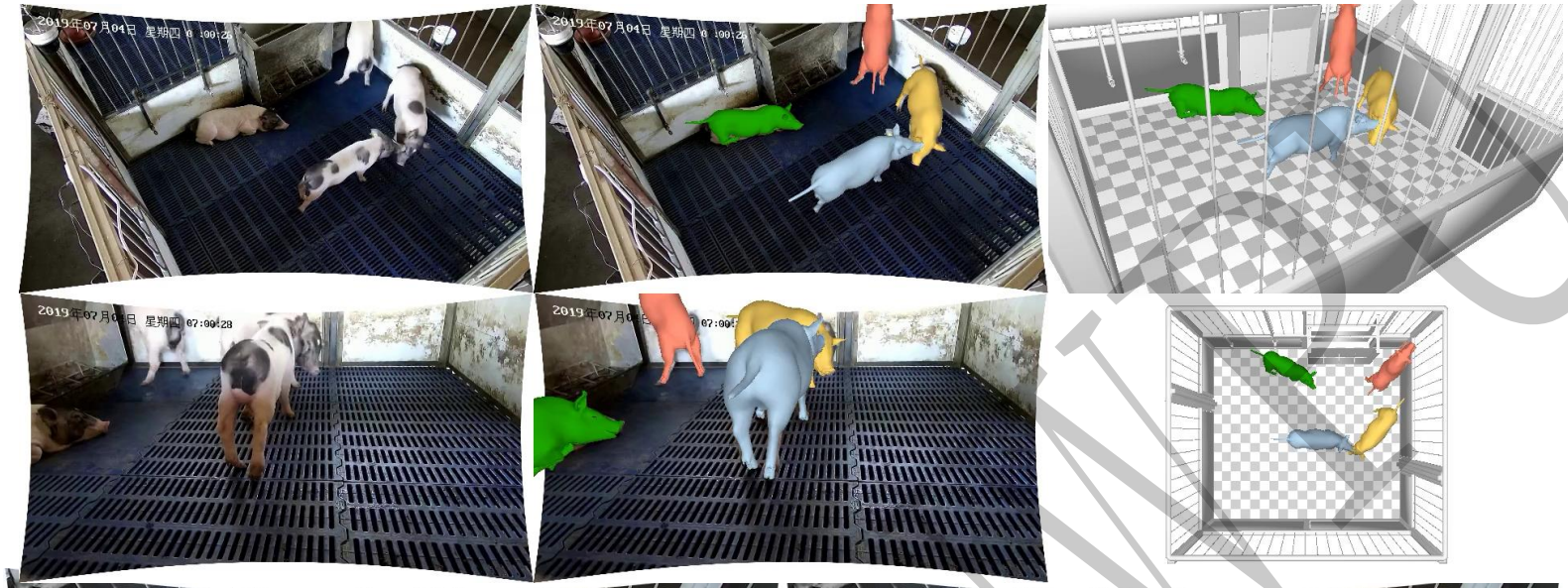




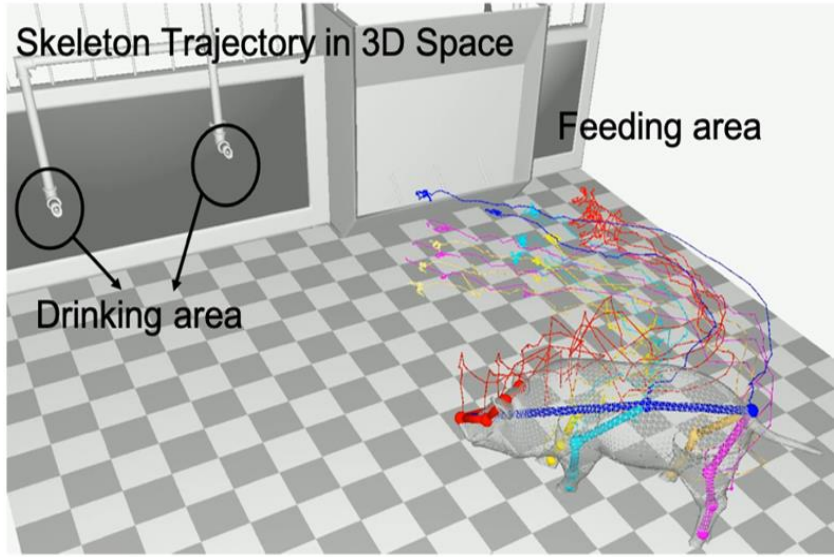
三维复杂行为计算助力神经退行性疾病的诊断

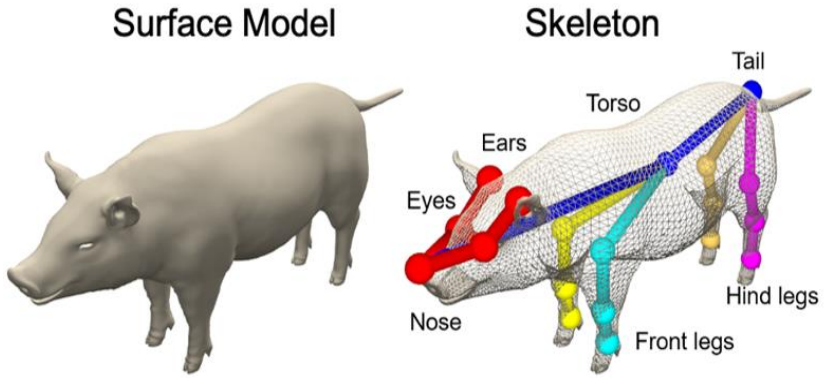
三维视觉：光场摄影时代

三维行为计算探索渐冻症早期症状



三维表面运动

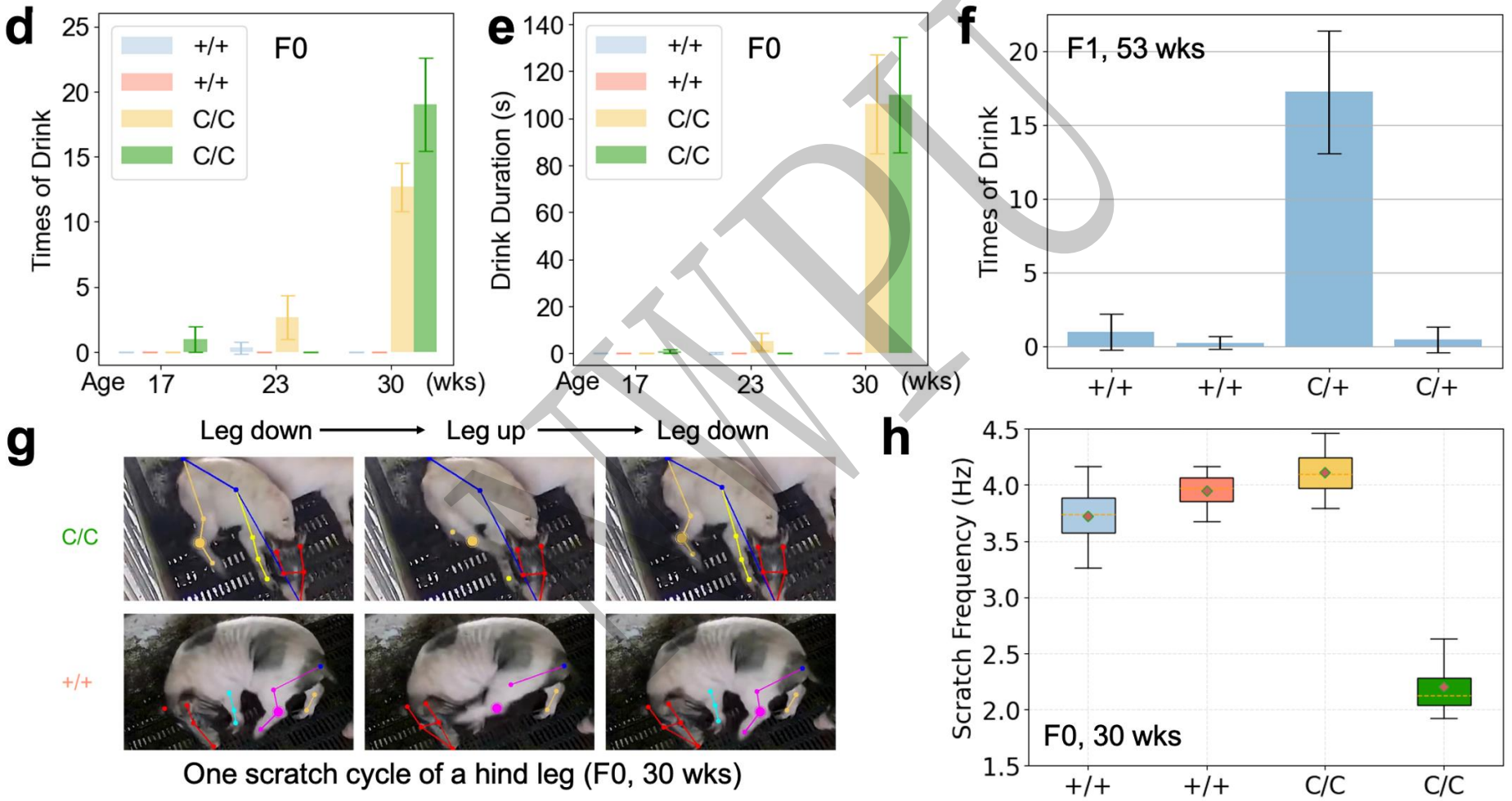




4只广西巴马小猪，2只健康，2只经基因突变，25个监控相机，时长1000小时

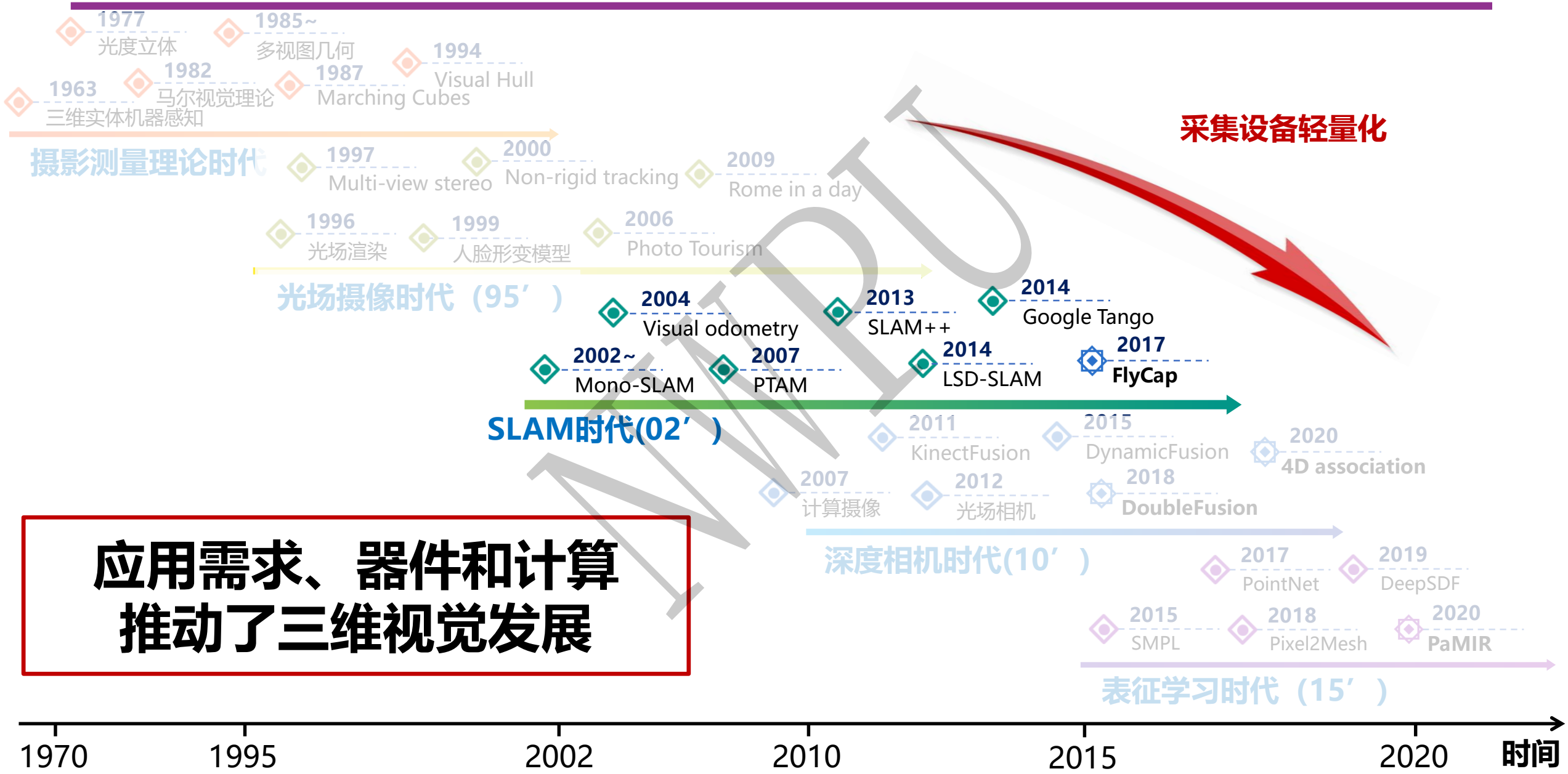
三维视觉：光场摄影时代

三维行为模式挖掘与分析



已分析的行为：喝水频率、喝水时长、侧卧挠腿频率

三维视觉的发展历史

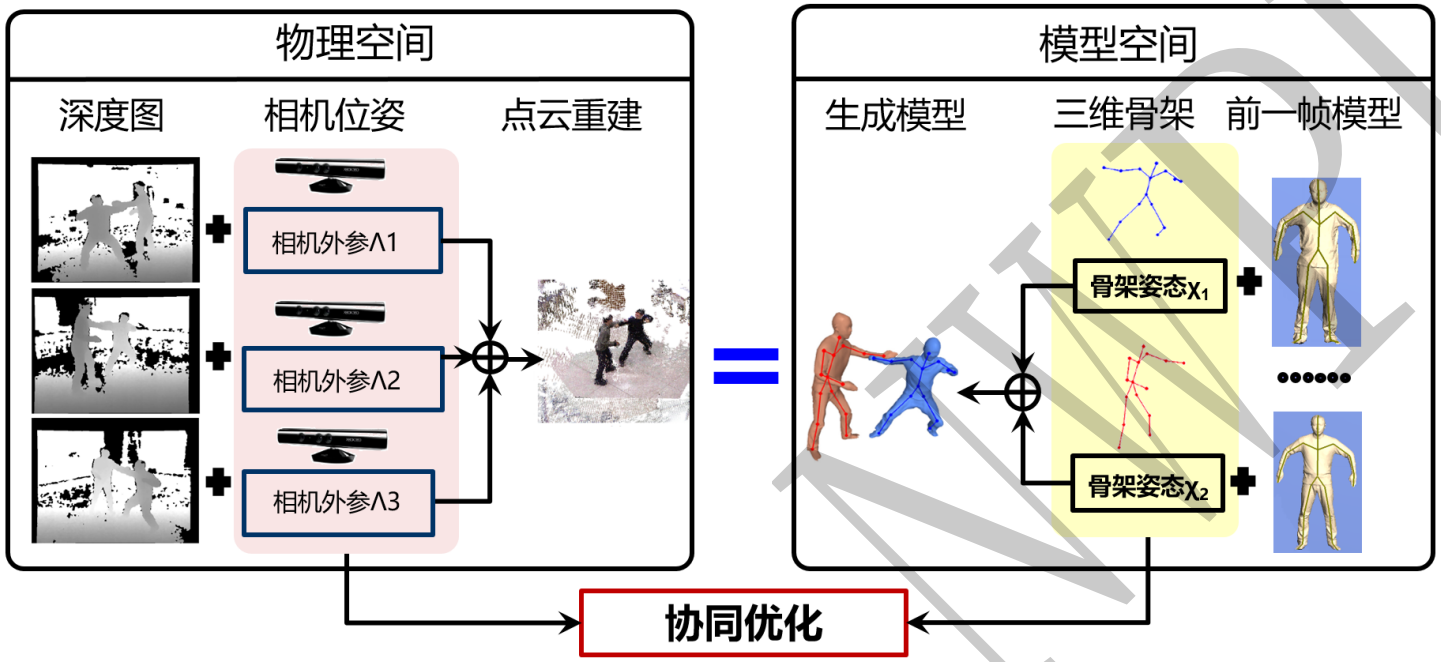


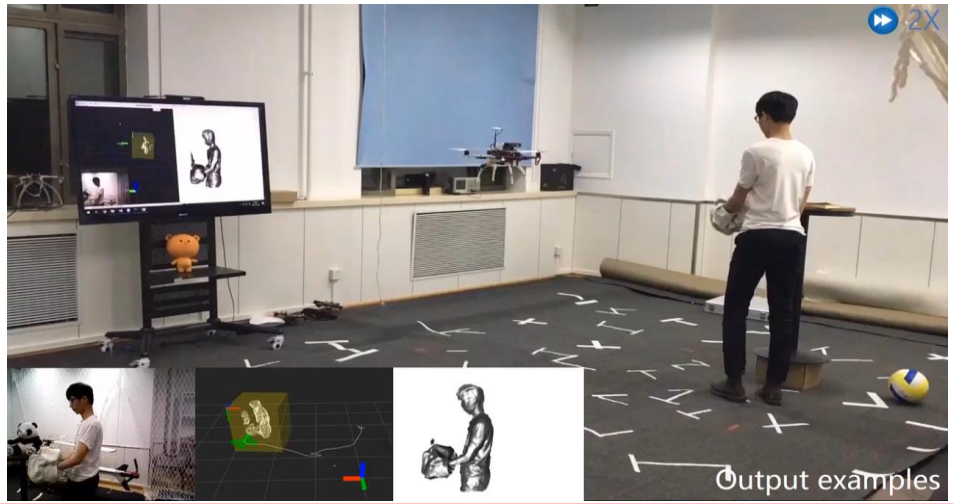
应用需求、器件和计算推动了三维视觉发展

三维视觉：相机定位时代

多手持或飞行相机动态重建方法

$$\arg\min_{\mathrm{X},\Lambda}\left\{\sum_{(p,v)\in Z_{pv}}\frac{\|p(\Lambda)-v(\mathrm{X})\|^2}{\|Z_{pv}\|}+\sum_{(p,v)\in Z_{pg}}\frac{\|p(\Lambda)-v\|^2}{\|Z_{pg}\|}+\sum_{(p,p')\in Z_s}\frac{\|p(\Lambda)-p'\|^2}{\|Z_s\|}\right\}$$





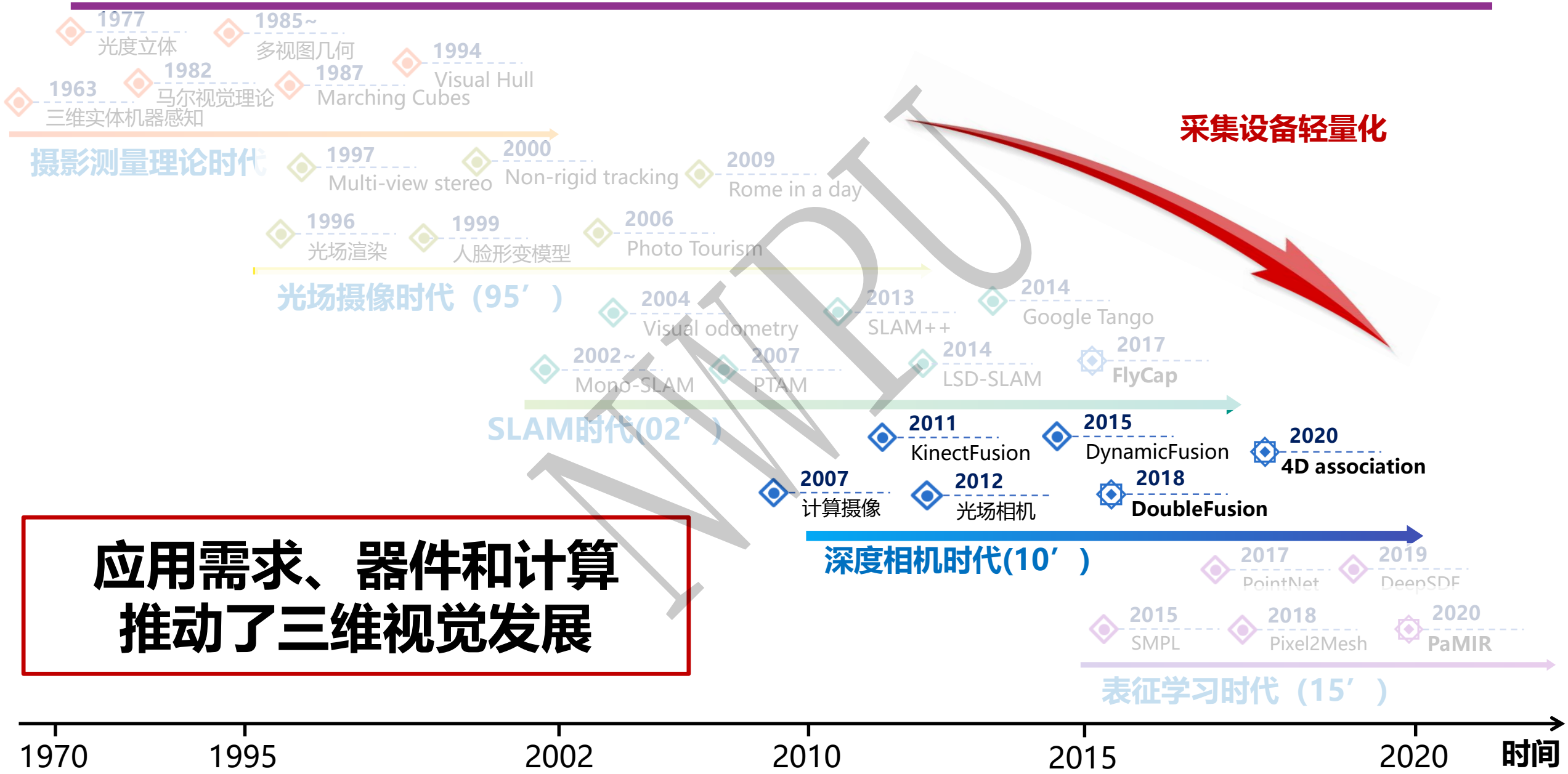
Xu et al. **UnstructuredFusion**: Realtime 4D Geometry and Texture Reconstruction using Commercial RGBD Cameras, IEEE **TPAMI** 2020

Xu et al. **FlyFusion**: Realtime Dynamic Scene Reconstruction Using a Flying Depth Camera, IEEE **TVCG** 2021

Xu et al. **FlyCap**: Markerless Motion Capture Using Multiple Autonomous Flying Cameras, IEEE **TVCG** 2018

首次实现面向动态场景的三维重建、相机协同定位和自适应视点优化

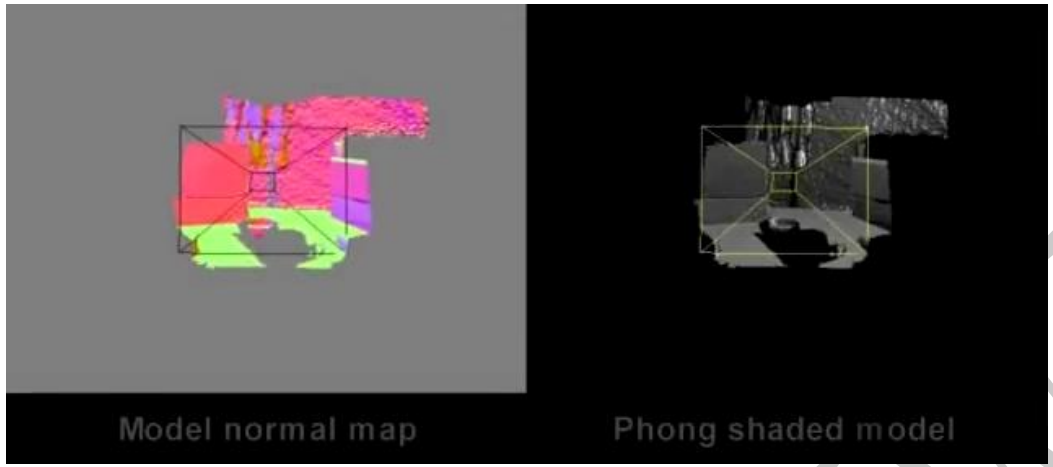
三维视觉的发展历史



应用需求、器件和计算推动了三维视觉发展

三维视觉：深度相机时代

□开启轻量级采集-实时三维数据获取时代



单视点

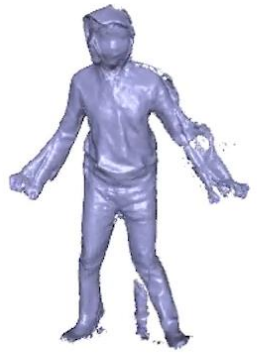
实时需求

仅能实现
时空局部映射

错误融合

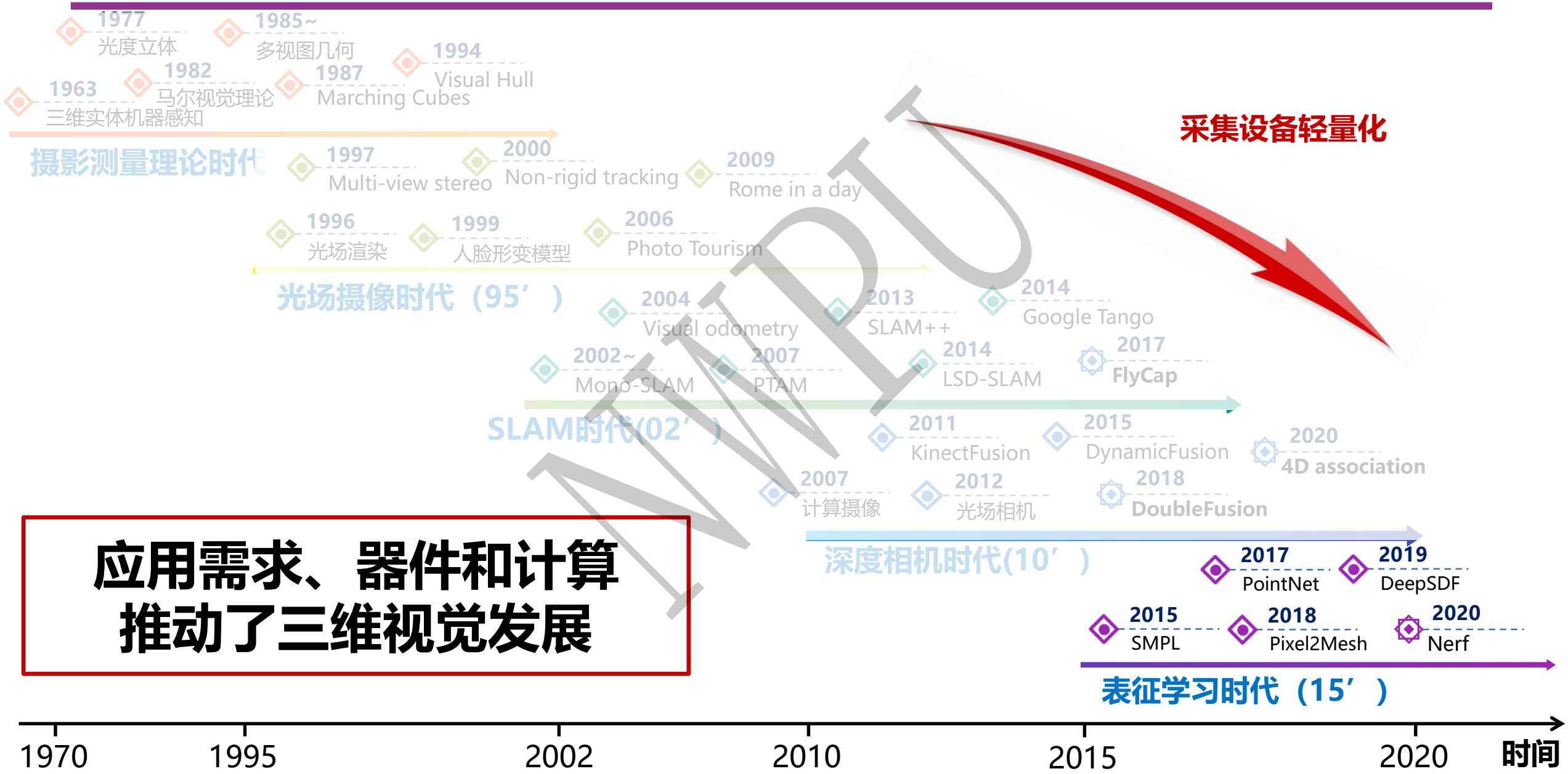
互传递

跟踪失效



已有重建方法仅实现时空局部映射，难以重建复杂运动

三维视觉的发展历史



应用需求、器件和计算推动了三维视觉发展

三维视觉：表征学习时代

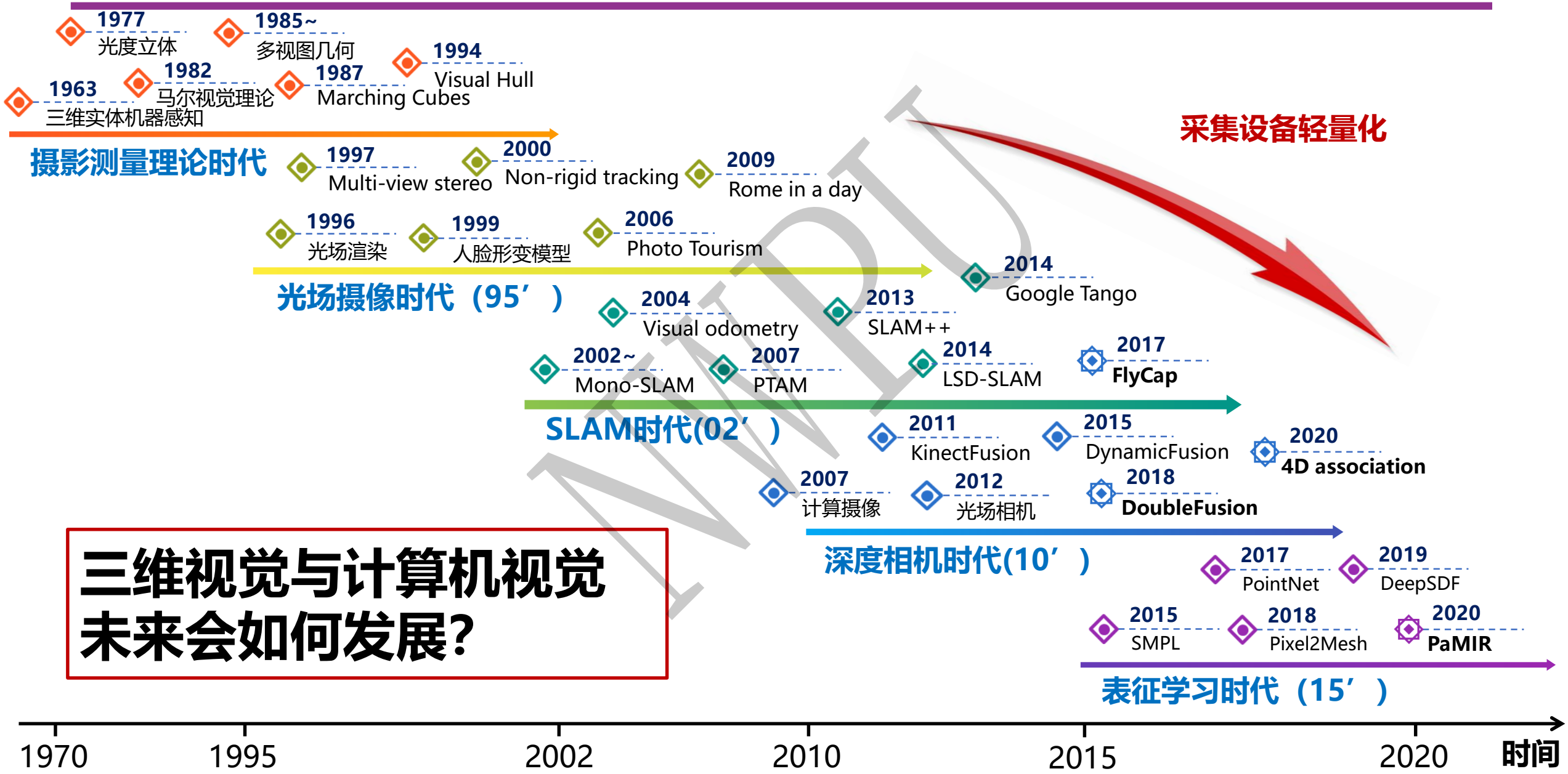
- 二维表征学习被推广到三维，单图像三维重建时代正式到来

PointNet (2016) MarrNet (2017) AtlasNet (2018) DeepSDF (2019)

单一几何表示
无法兼顾
结构、语义和细节

参数化模板	体素	网格	符号距离场	点云	光场
无法处理拓扑变化	数据量大 丢失表面细节	结构不定 难以统一	定义复杂，丢失 语义对应关系	丢失结构信息	数据量大 丢失几何信息

三维视觉的发展历史



三维视觉与计算机视觉未来会如何发展?

主要内容

- 计算机视觉与三维视觉
- 三维视觉的应用需求
- 三维视觉的发展历史
- 未来的三维视觉与计算机视觉

计算机视觉的未来

现状：语义理解和三维感知的并行独立发展

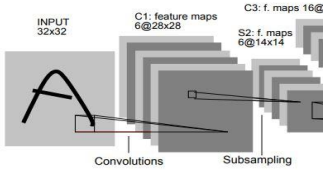
语义理解

图像光流计算

1981

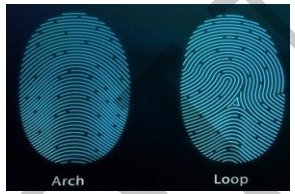
Eigenface
图像人脸识别

1990



LeNet

1998



自动指纹
识别系统

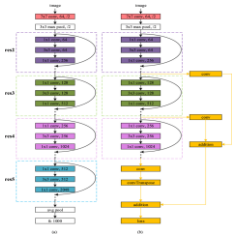
1999

Person
Re-Id

2008

ImageNet
大规模图像数据集

2010

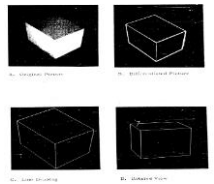


ResNet

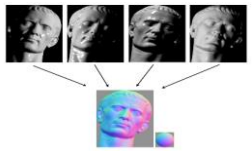
2016

三维感知

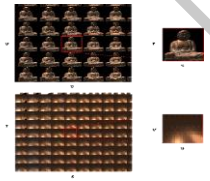
1963
机器感知
三维物体



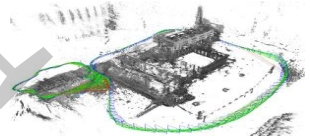
1989
Shape from
Shading



1996
Light Field
Rendering



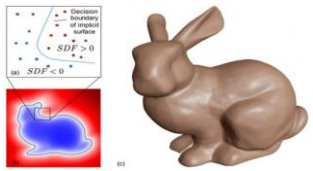
2005
视觉SLAM算法



2006
Multi-view
Stereo

2015
DynamicFusion

2019
DeepSDF



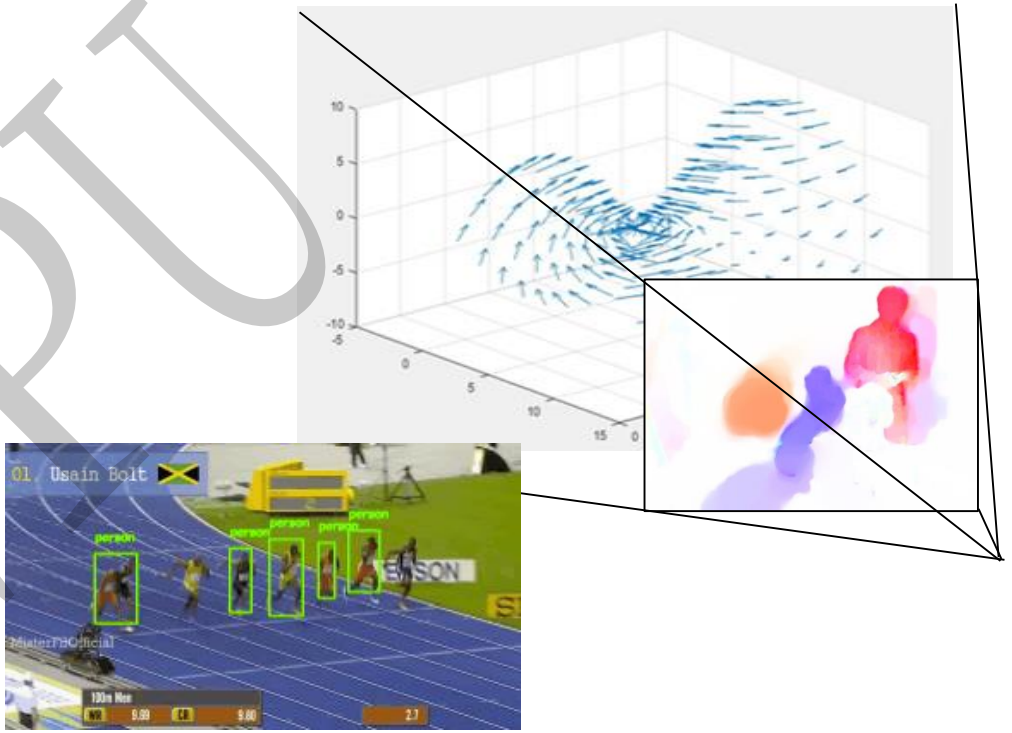
语义理解仅依靠二维图像取得，而三维感知的点云、网格和纹理等无语义信息

计算机视觉的未来

主流的场景视觉理解

二维目标识别

只是**离散的像素分类**问题



二维运动跟踪

投影导致**信息降维**

二维场景理解不足以获得连续的世界的本质

计算机视觉的未来

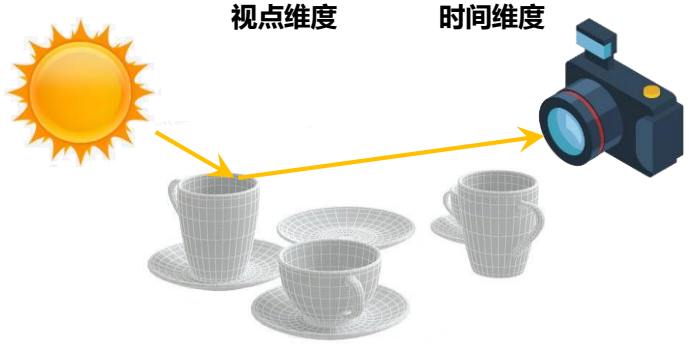
思考：三维视觉与二维视觉的关系

[Adelson 91]

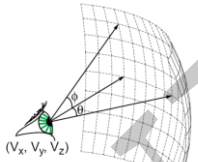
空间维度 光谱维度

$L = P(x, y, z, \theta, \varphi, t, \lambda)$

视点维度 时间维度



7维全光函数,现实世界的视觉信息
(结构、材质、运动、光照、相机)



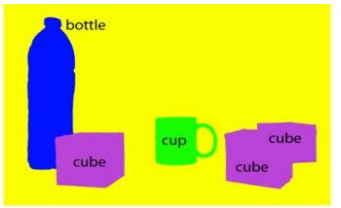
$$I(x,y) = \int \cdots \int \underbrace{L(x',y',\theta_x,\theta_y,\lambda,\ldots,t)}_{\text{Most of this gets lost}} \, d\mathbf{A} \, d\boldsymbol{\omega} \, d\lambda \cdots dt$$

成像过程是全光函数
离散下采样，滤波等

二维视觉

二维图像

模式识别



场景理解

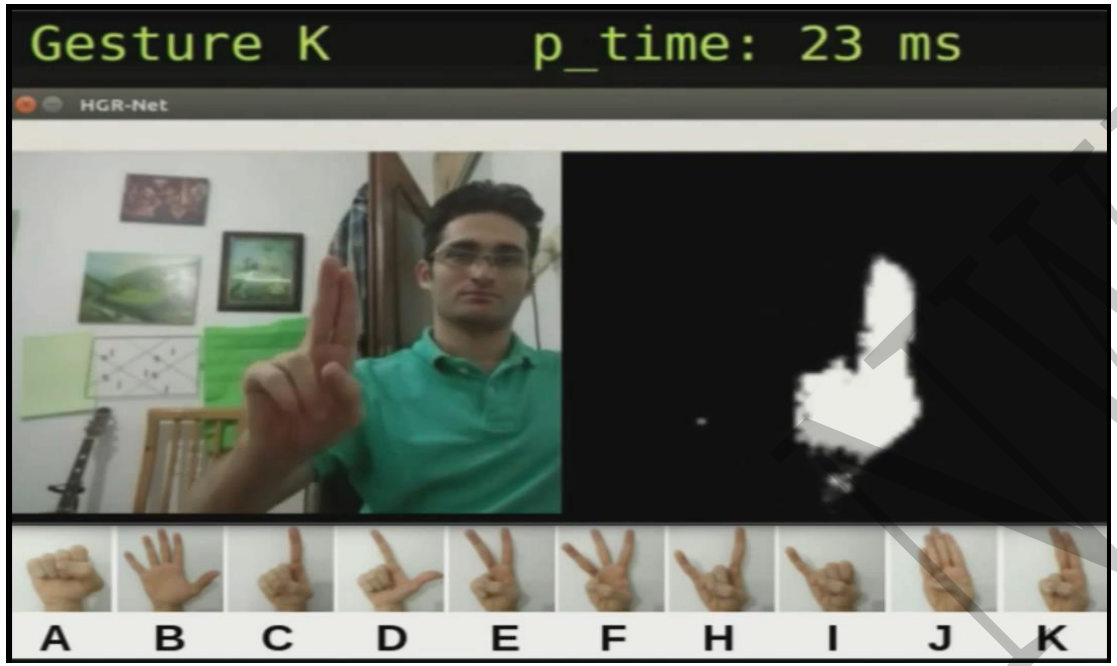
三维视觉

相比于二维视觉，三维视觉建模了视觉成像机理，更接近视觉世界本质

计算机视觉的未来

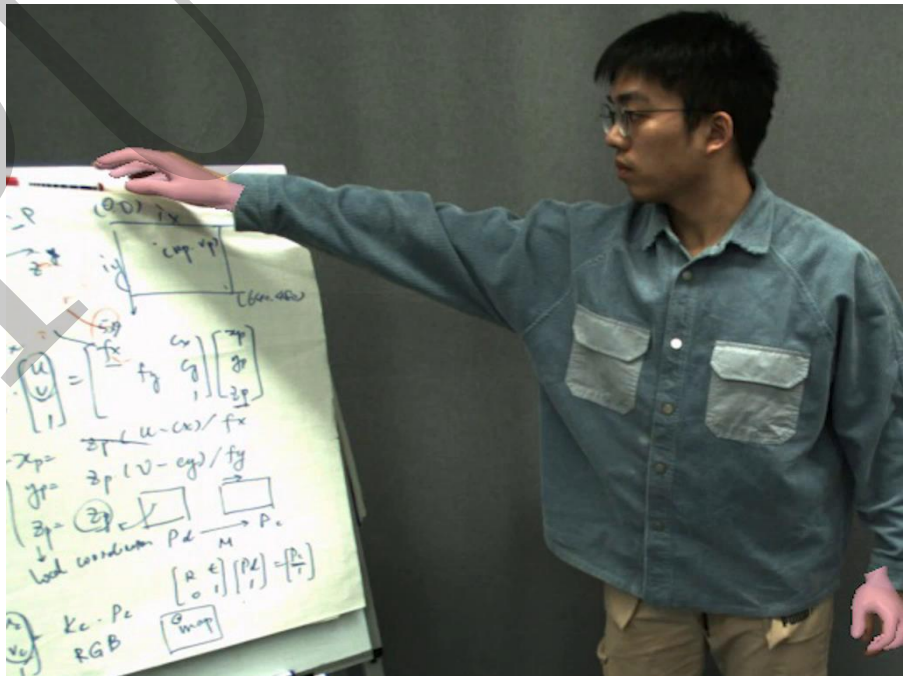
二维到三维的发展：以人体/人手姿态检测为例

提取少量特征进行离散的分类



二维视觉（分类）

获取完整的信息并呈现连续的姿势

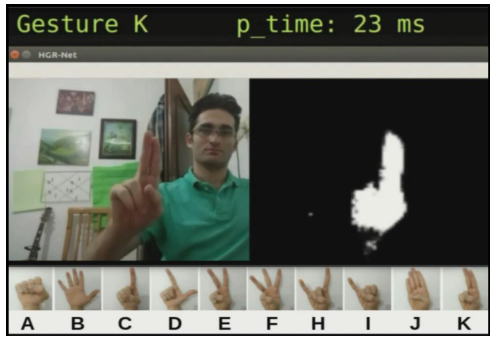


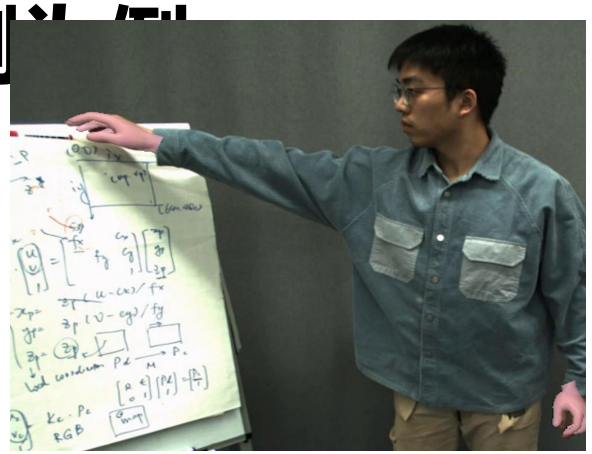
三维视觉（单图像重建）

从二维到三维，提高了手势识别细粒度

计算机视觉的未来

- 二维到三维的发展：以人体/人手姿态检测为例



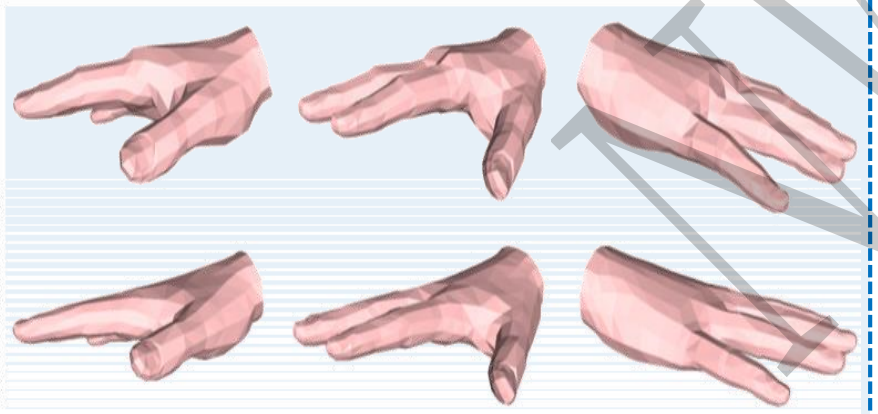


(三维知识先验)

(三维数据先验)

三维表征技术

人手几何模板



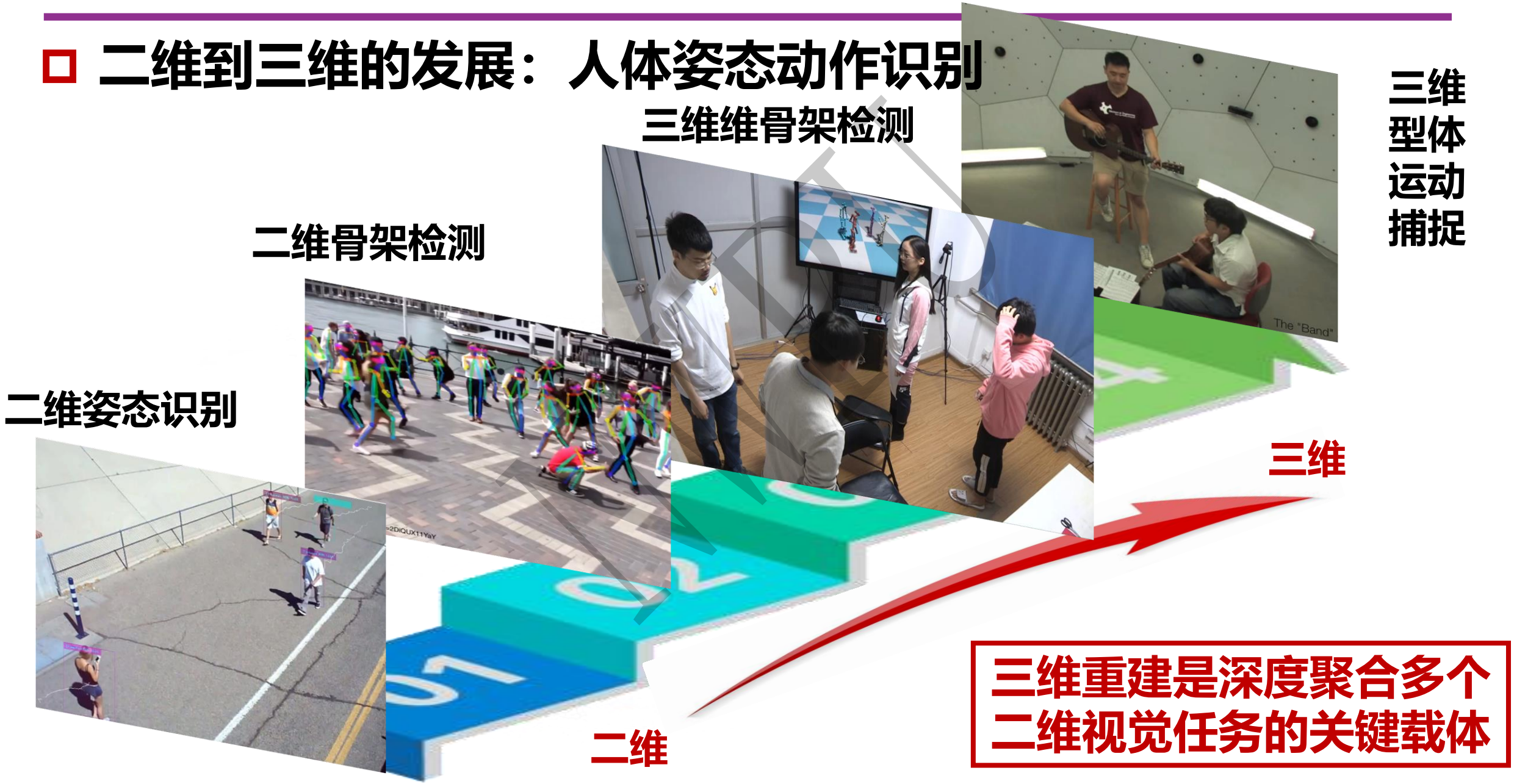
人手多视角数据集

三维采集技术

三维技术提供知识和数据，实现完整、紧致、语义化、连续的智能视觉感知

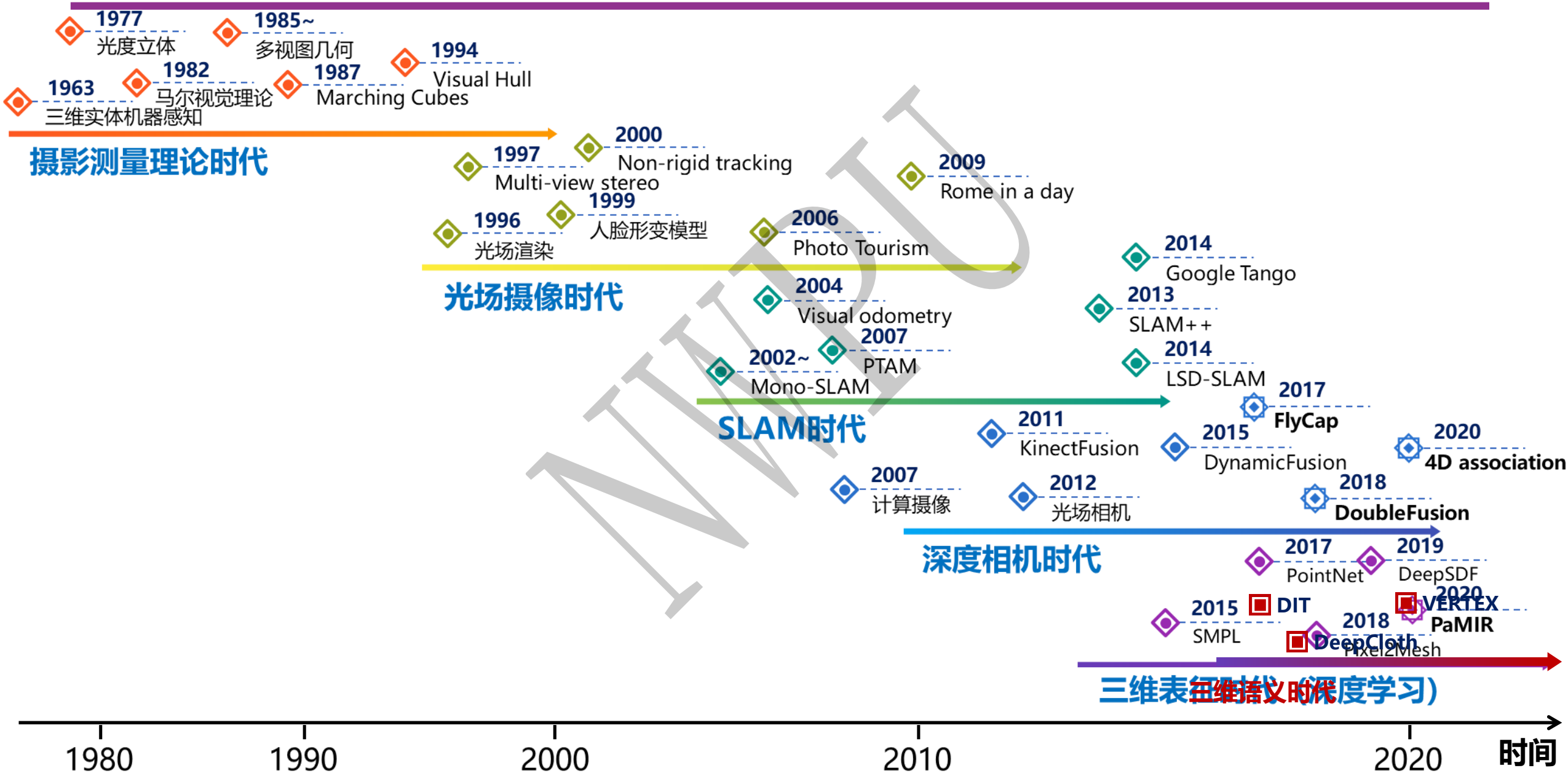
计算机视觉的未来

二维到三维的发展：人体姿态动作识别



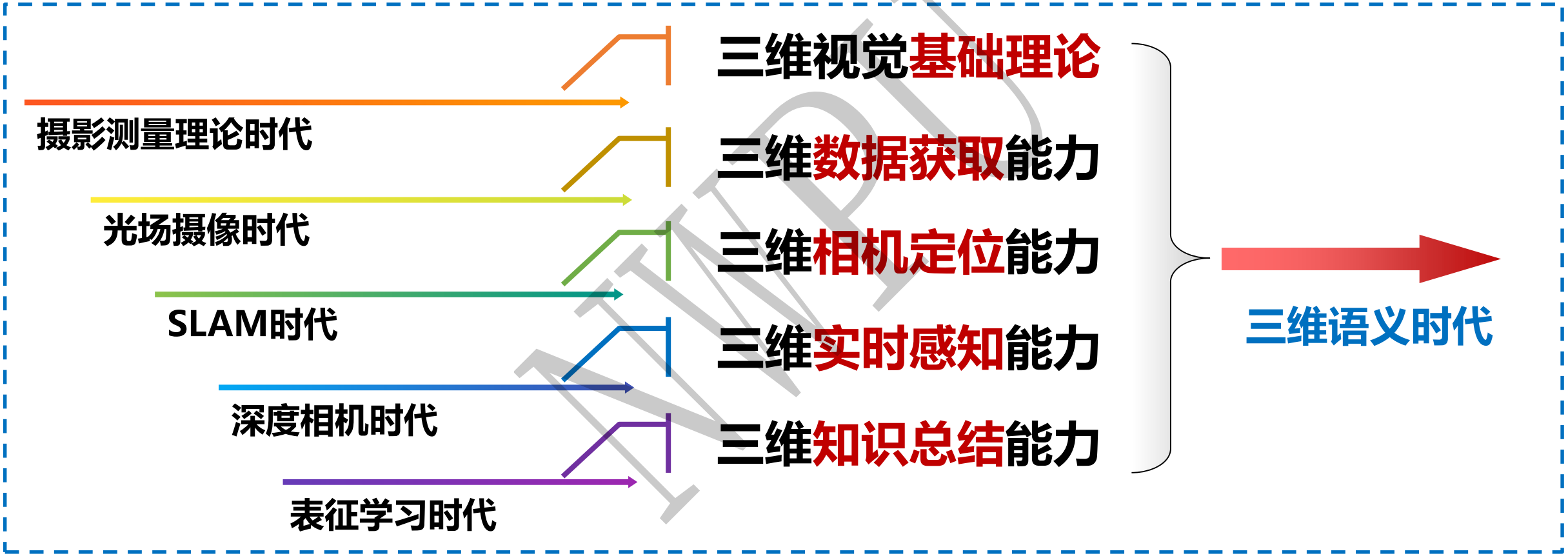
三维重建是深度聚合多个二维视觉任务的关键载体

计算机视觉的未来



三维语义时代

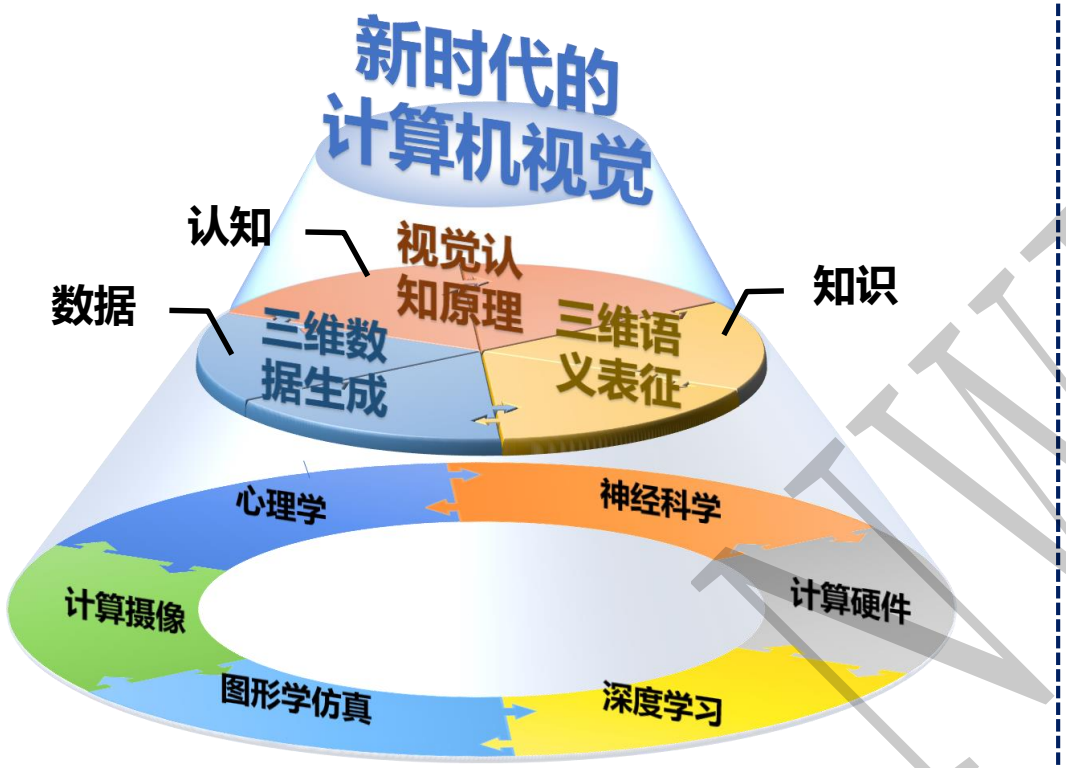
已有时代与三维语义时代的关系



已有三维视觉的理论和技术赋能三维语义时代，实现新一代智能视觉感知

三维语义时代

三维感知技术能够为场景理解提供更多先验



三维语义时代

定义：

带有语义的三维，带有三维的语义

特点：

- 蕴含连续高维知识的三维表征
- 可编辑可交互
- 未来目标：单一图像全景三维语义重建

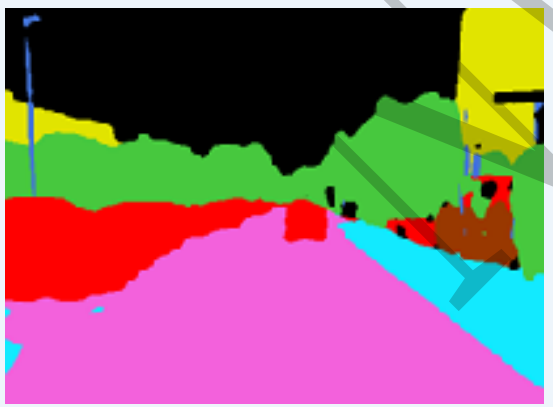
数据+知识+认知=未来视觉智能（三维语义智能）

必须将目标识别与三维感知结合，才能触及计算机视觉最终目标

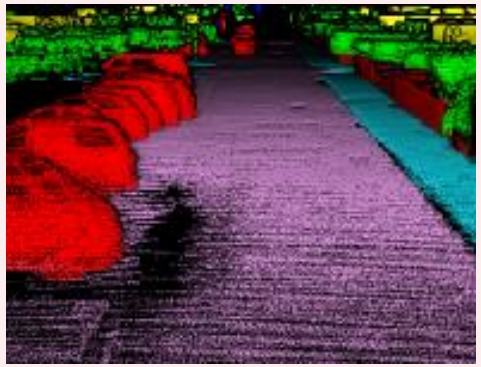
三维语义时代

二维语义：离散化、单一属性 → **三维语义：连续化、多维度属性**

二维

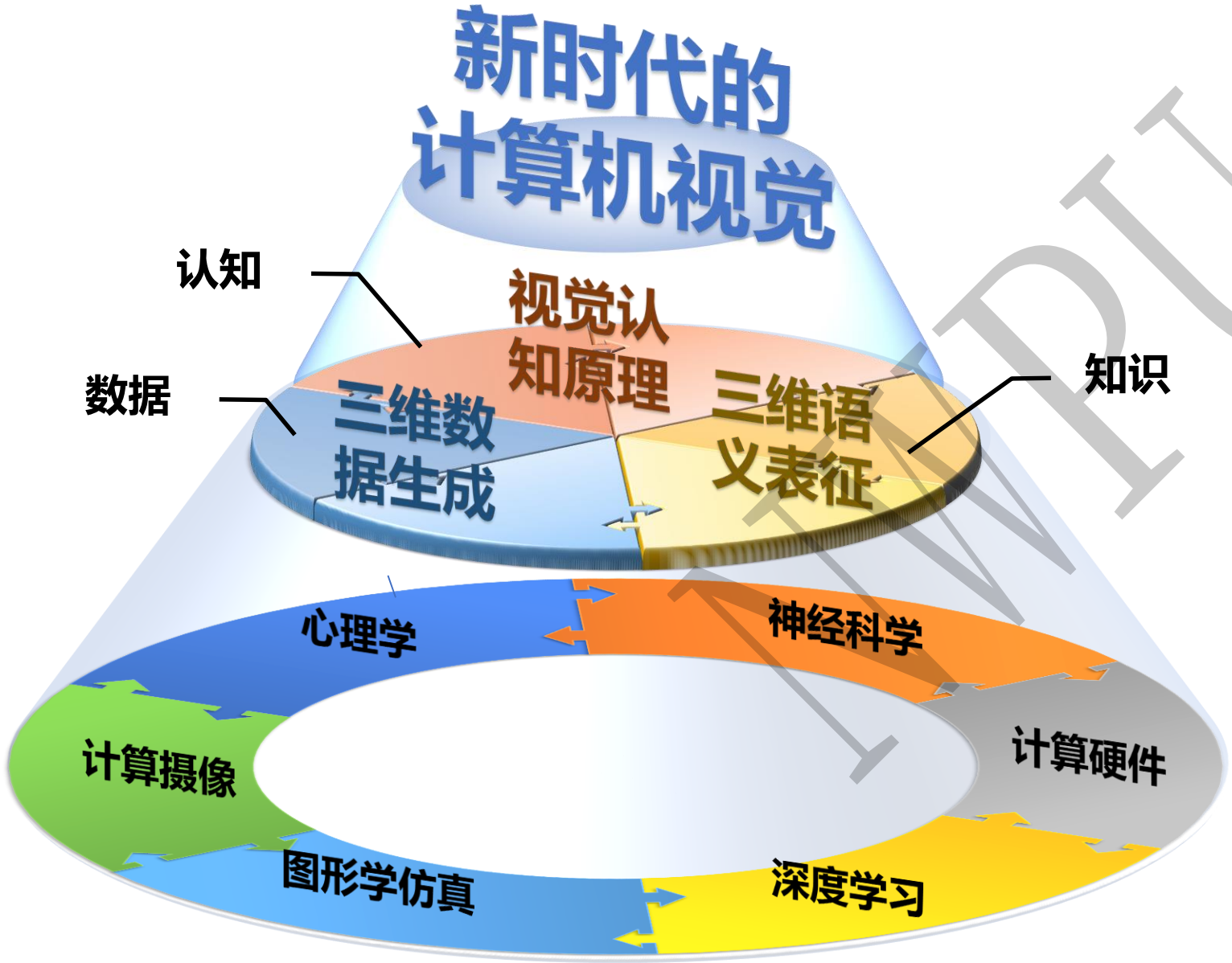


三维



三维语义具有天然的连续性特征，提供了更准确、更高密度的语义信息

总 结



- 三维视觉在人工智能各产业具有重要地位
- 三维视觉是计算机视觉的必然趋势
- 知识+数据+认知 驱动计算机视觉发展
- 三维语义重建是三维视觉未来十年的热点

Sensors

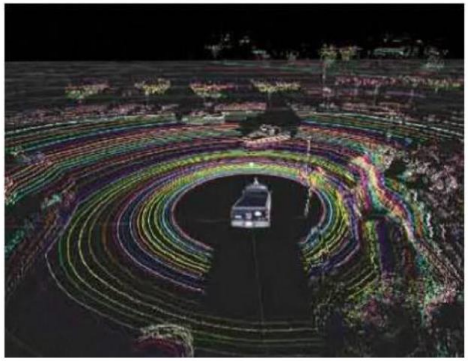
- Goal: create a **point cloud** of (samples from) the surface of an object/scene
  - Collection of distance measurements from the sensor to the surface
  - Distances are them transformed into 3D coordinates (x,y,z) by means of calibration information
  - Usually, 3D sensors acquire only a view of the object (**2.5D data**)
  - Some sensors also acquire information concerning color or light intensity (**RGB-D data**)
- Contact sensors
- Active sensors
  - LIDAR, rangefinders
  - Time-of-Flight cameras
  - Laser Triangulation
  - Structured light
  - Medical imaging (CT, MRI)
- Passive sensors
  - Stereo
  - Structure-from-motion
  - Shape-from-shading, shape-from-silhouette, shape-from-defocus, ..

Sensors: LiDAR

- **LIDAR**: Light Detection And Ranging
- A light pulse is emitted from the sensor and the round-trip time is computed as $d = \frac{ct}{2}$
  - The higher the time, the further away the point from the sensor
- Usually visible or near-infrared light is used
- Arrays of emitters are employed together to yield a set of simultaneous range measureaments (3D slice)
- Slices can be swept using a **motor** to yield a slice array
- Pros:
  - High range (hundred meters / kms)
  - Works indoor/outdoor
  - Real-time (eg. 100 Hz slices)
- Cons
  - Accuracy is an issue due to speed of light (3 10^8 m/s -> 1 mm every 3.3 ps)
  - Cost
  - Color/intensity information is usually not provided

Velodyne

SICK LMS500



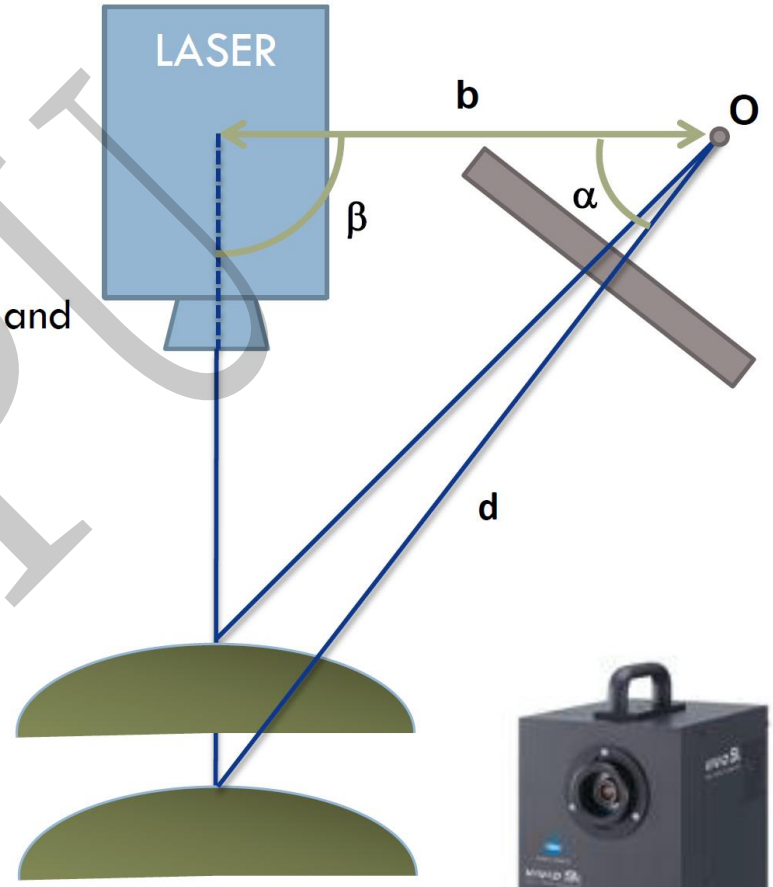
Sensors: Time of Flight camera

- A particular LIDAR device yielding a full 2D array of range measurements
- A light pulse is emitted through an infrared illuminator; then:
  - Phase-shift measurament of the returning light pulse on each pixel (Photonic Mixed devies, Canesta Vision (now Microsoft), Swiss Ranger)
  - «range-gated imager»: each pixel has a shutter that starts closing when the light pulse is emitted; the less light received, the further away the point from the sensor (Zcam by 3DV Systems, now Microsoft)
- Pros
  - No motor needs to be employed
  - Real-time (30-100 fps)
  - Cost effective
- Cons
  - Low resolution
  - Dark, non-reflective objects are hard to be acquired
  - Hardly works under sunlight (no outdoors)
  - Multiple reflections can yield false measurements
  - Interference between different sensors

MESA SwissRanger 4000

Sensors: Laser Triangulation

- Laser + camera system
  - A laser dot or stripe is emitted on the scene
  - The camera locates the dot/stripe.
  - The distance is determined via triangulation of the position of the dot (emitting angle, receiving angle and baseline need to be known)



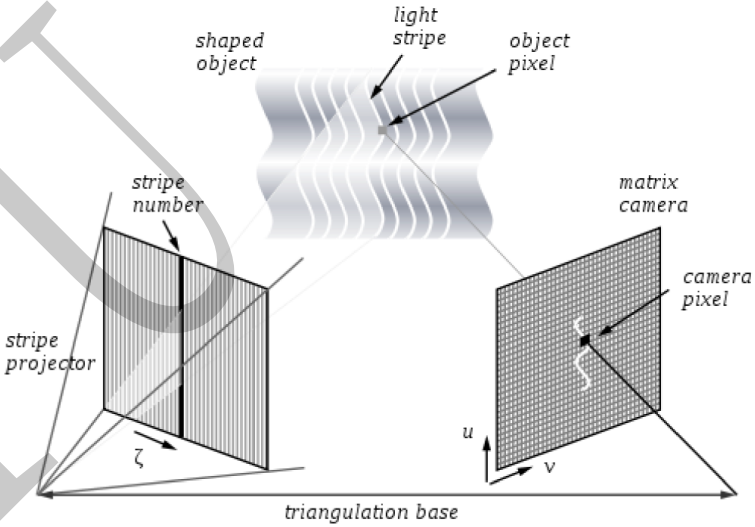
- Pros
  - Accuracy (tens of micrometers)

- Cons
  - Limited range
  - Slow scanning time (often requires static scene)
  - $$
  - No color information, needs to be paired with a color camera

Minolta Vivid 9I

Sensors: Structured Light

- Camera + projector system
- Similar to laser triangulation, where the laser stripe is replaced by a stripe projected by a light projector
- Projecting a set of stripes (2D pattern) allows for multiple sampling, hence a full 2D range image can be acquired at once (but problem of confusing different fringes)
- Using infrared projection and two cameras (one in infrared, one in the visible band) yields accurate RGB-D data
- Pros:
  - Relatively cheap
  - Sub-millimiter accuracy (down to tens of micrometers)
  - Real-time
- Cons:
  - Limited range
  - Hardly usable outdoor or in presence of other light sources
  - Highly dependent from the object surface characteristics (eg. reflective, translucent, ..)
  - Interference between different sensors

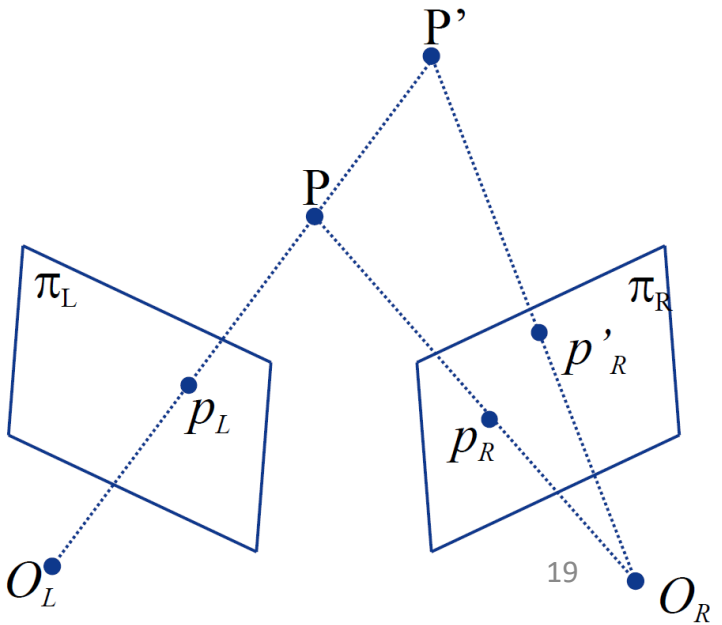


Microsoft Kinect

Sensors: Stereo Vision

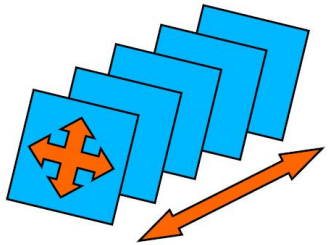
- Two (or more) cameras
- Cameras have to be sync-ed, especially in presence of non-static scenes
- Depth is retrieved via triangulation of the point projections on the two views
- **Correspondence** problem!
- Pros:
  - Cheap
  - Passive
  - Real-time
  - Color/intensity can be directly associated to range data (RGB-D)
- Cons
  - Low accuracy (tend to fail on low-textured regions, repetitive patterns and depth borders)
- A projector can help adding texture to the scene to improve accuracy on low-textured regions

Videre Design

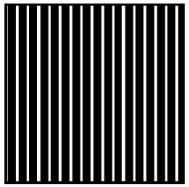


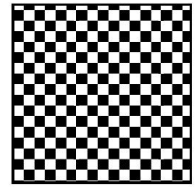
Sensors: Space Time Stereo

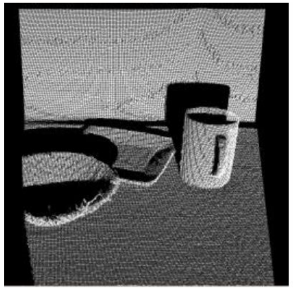
- Stereo using **spatial** and **temporal** information[Davis 05][Zhang 03]
- To gather information, the appearance must change over time (but not the geometry!)
- A random pattern is projected to augment each frame with a different texture (no structured light, no interference)
- More accurate than standard stereo, but depth must be constant in time (static objects)



Joint spatio-temporal window

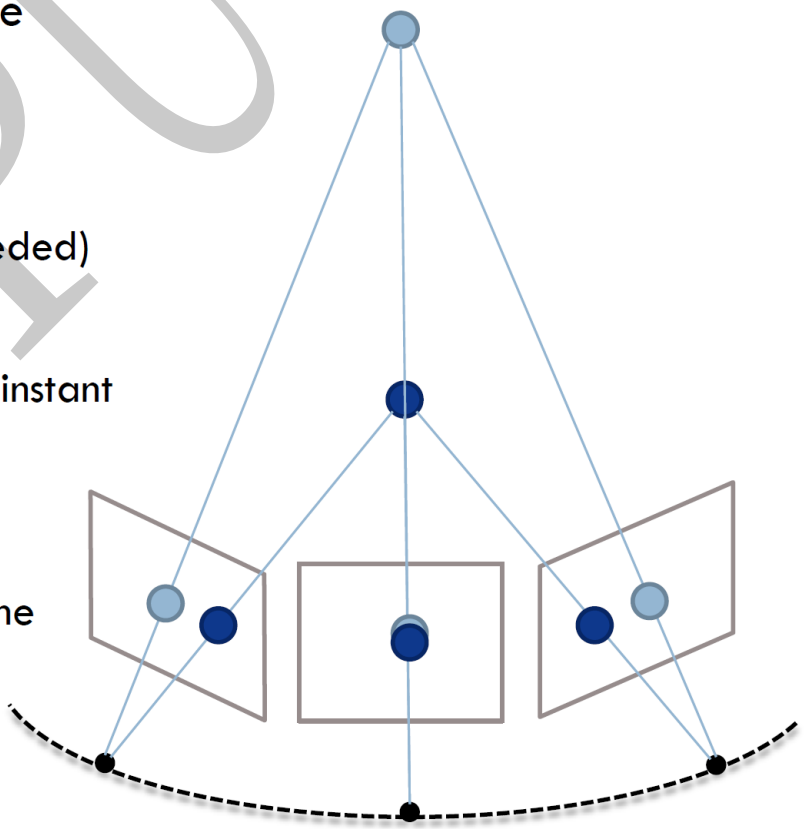






Sensors: Structure from Motion

- Monocular system
- Instead of spatially extending multiple views, they are temporally extended
- Requires either the surface or the camera to move
- Tracking and matching features
- Pros
  - Cheap and simple hardware (only one camera needed)
  - RGB-D data
  - Solving SfM also yields camera pose at each time instant
- Cons
  - Highly dependent from the available motion that the object/camera can undergo
  - Sparse depth information

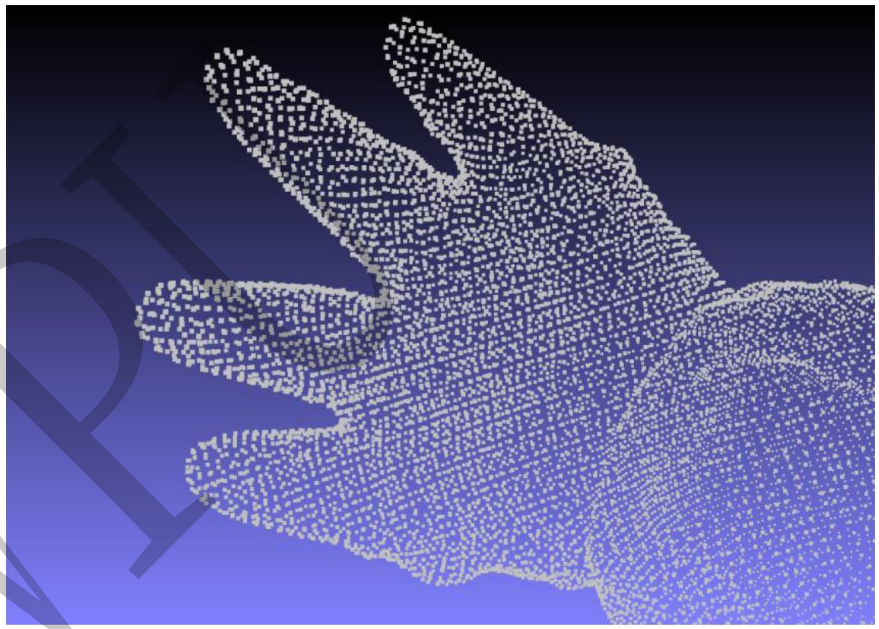


Data

- 3D data may be represented in different formats
  - Unorganized
    - **Point cloud:** a set of vertices.
    - **Polygon mesh:** a set of vertices and their connections (topology).
  - Organized
    - **(Binary) voxelized cloud:** a 3D regular grid of (binary) density values.
    - **Range image:** a 2D regular grid (an image) of 3D coordinates.
- 3D data may be
  - **Pure 3D:** represent only the geometry of the scene
  - **RGB-D:** store the geometry of the scene as well as the intensity of the 3 color channels
- 3D data may represent
  - **Full 360° (3D) view** of an object /scene
  - **2.5D view** of an object /scene

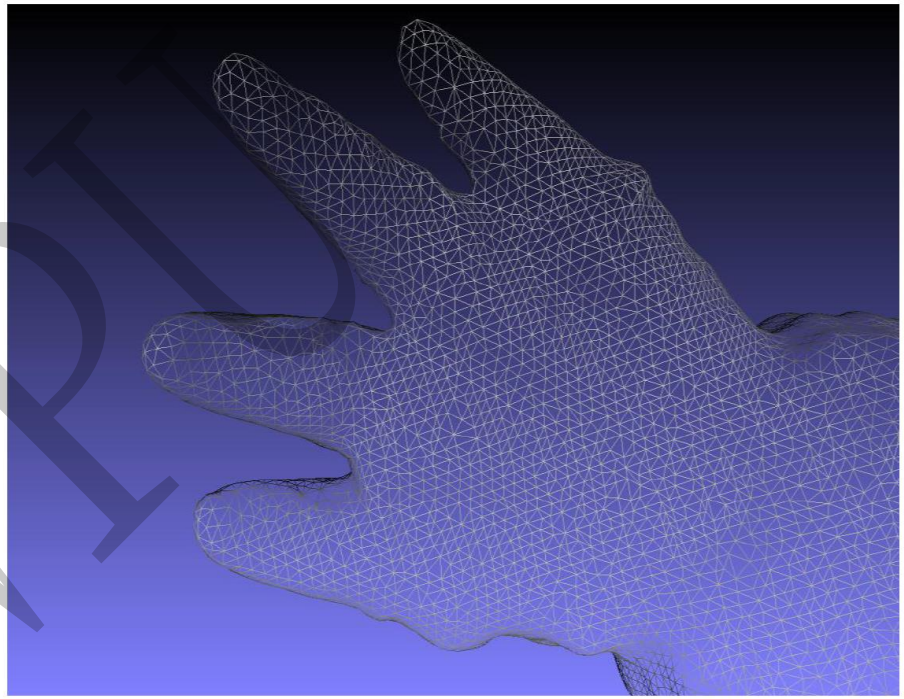
Data: Point Cloud

- Point cloud is the most common 3D data representation in computer vision.
- It is just a collection of 3D coordinates:
  - Unorganized: hence, nearest neighbor searches are costly
    - Must build an index structure (e.g. kD-tree or Oc-tree)
  - no topology
    - Inner and outer parts of the surface are hard to discriminate

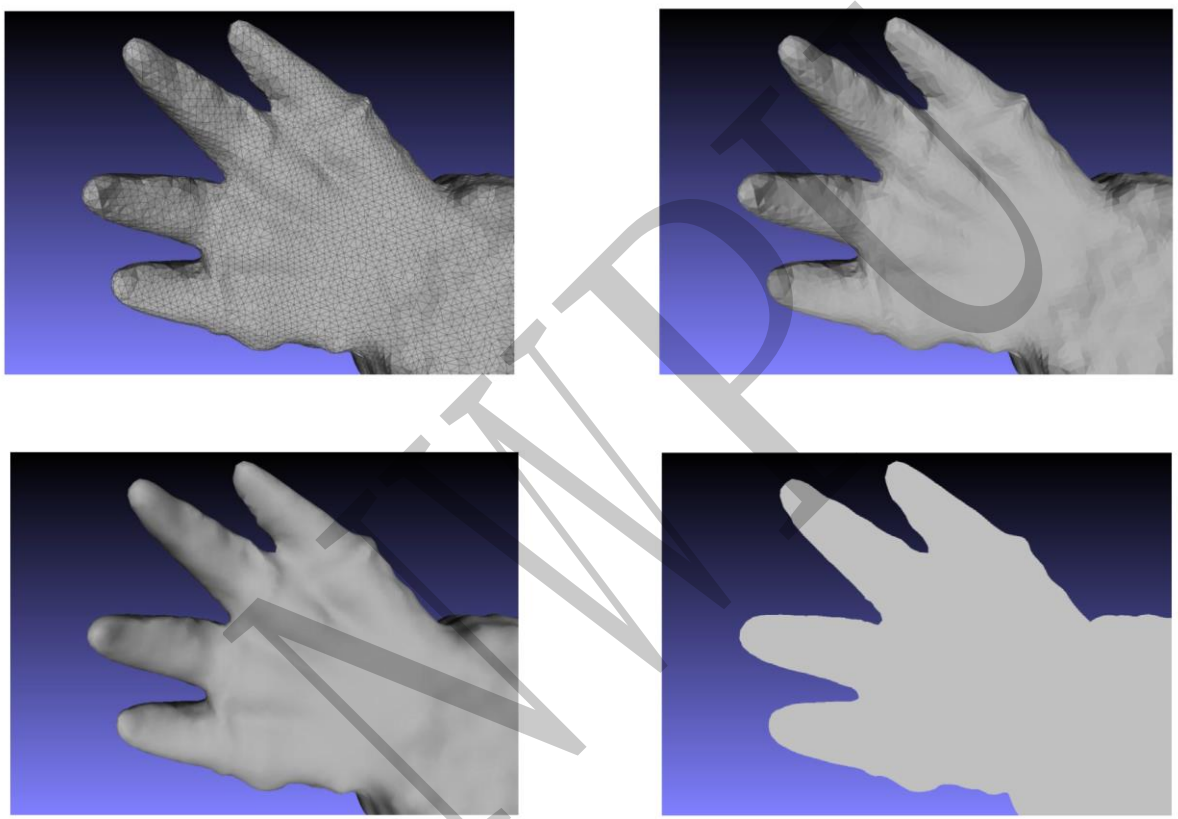


Data: Polygon Mesh

- Polygon mesh is the most common 3D data representation in computer graphics.
- It is a collection of 3D vertices and of faces connecting them:
  - Unorganized: hence, nearest neighbor searches are costly
    - Must build an index structure (e.g. kD-tree or Octree)
  - Has topology
    - Essential for rendering
    - Provides surface orientation



Data: Polygon Mesh



Rendering in **MeshLab**

Data: Range Image

- Range image is a useful representation for efficient 3D data processing
- The term is a bit ambiguous as it is used to denote
  - a single channel image whose pixels encode the distance of the scene point from the sensor
  - **a three channels image whose pixels encode the coordinates of the scene points**
- It is an organized representation
  - Nearest neighbor searches can be carried out efficiently by exploiting the image lattice
- No topology
  - But it is easy to create it from the lattice
- **Only 2.5D views**: it provides also the sensor position (point (0,0,0)).

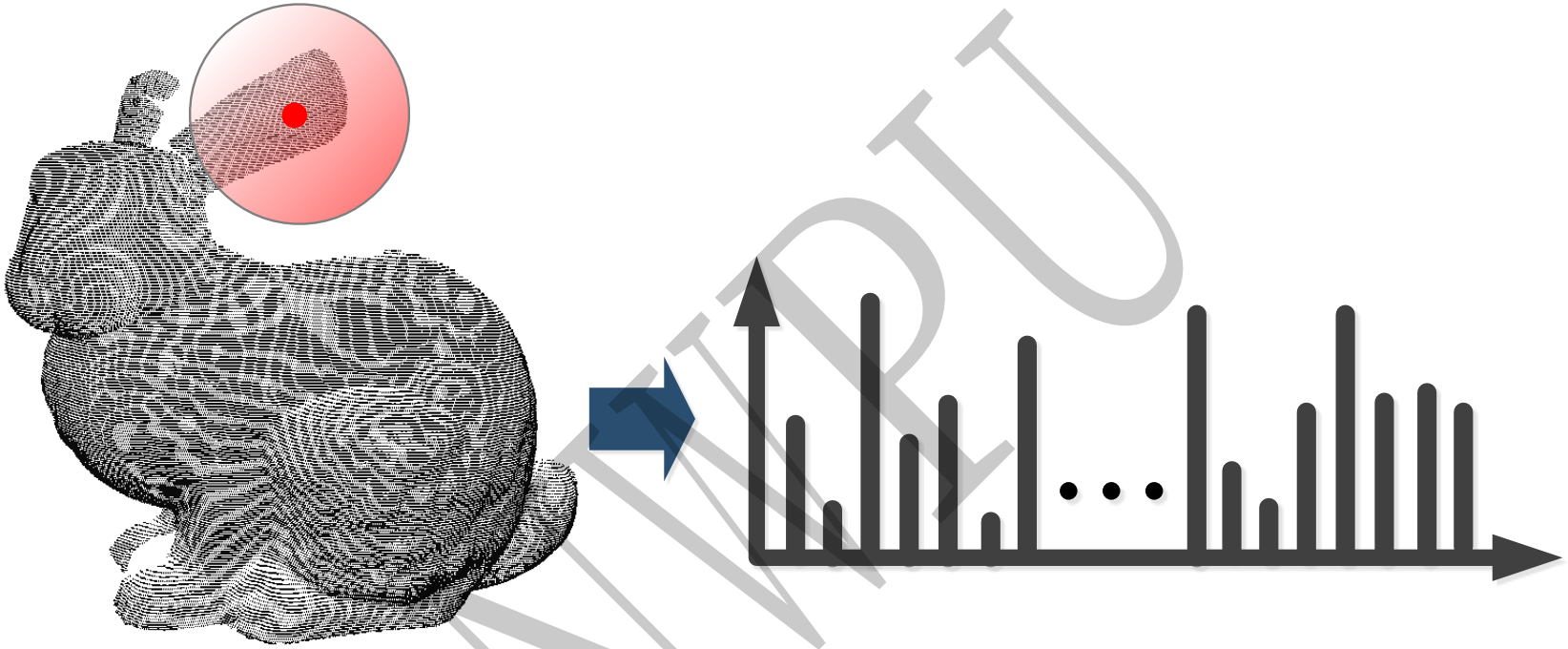
Image from Stuttgart Range Image Database (http://range.informatik.uni-stuttgart.de/)

Data: Voxelized data

- Voxelization is a useful representation for efficient 3D data processing
  - It is also the output of some sensors (e.g. metal detectors, body scanners)
- It represents surfaces with a regular grid of voxels (volumetric pixels)
  - 3D coordinates are implicitly defined by the index in the grid.
- It is an organized representation
  - Nearest neighbor searches can be carried out efficiently by exploiting the regular structure
- No topology
  - But it is easy to create it from the grid
- Suitable also for full 3D views.

Lists of Contents

Concept



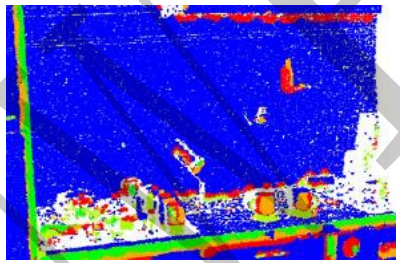
- Design a rotation-invariant vector to encode the underlying geometry of a local surface

What is a good feature?

Descriptiveness | Robustness | Efficiency | Compactness

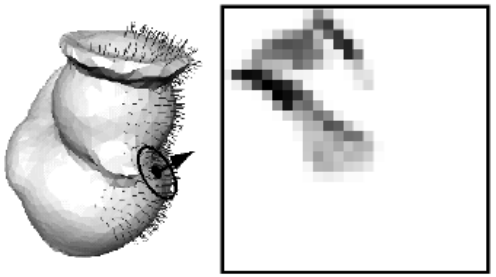
Normals

[ICRA 2007] FPFH
[ECCV 2010] SHOT



Curvature

[PRL 2007] LSP
[CVIU 2009] LPDM



Point densities

[PAMI 1999] Spin image
[IJCV 2013] RoPS

Basic cue1: the neighborhood concept

From an usage point of view, there are two specific application examples for the determination of $\mathcal{P}^k$ for a query point $\boldsymbol{p}_q$, namely:

- determine the closest k neighbors of the query point (k search);
- determine all the k neighbors of the query point up to a radius r from it (radius search).

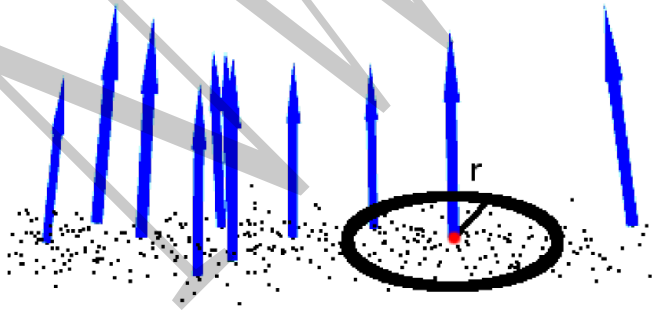


Figure 4.2 Example of a radius r search to determine the set of neighbors $\mathcal{P}^k$ for the query point (marked with red). $\mathcal{P}^k$ is used to estimate an approximative surface normal at the query point (blue arrow).

Basic cue2: Normals

- An important problem in describing the geometry of the surface is to first infer its orientation in a coordinate system, that is, estimate its normal

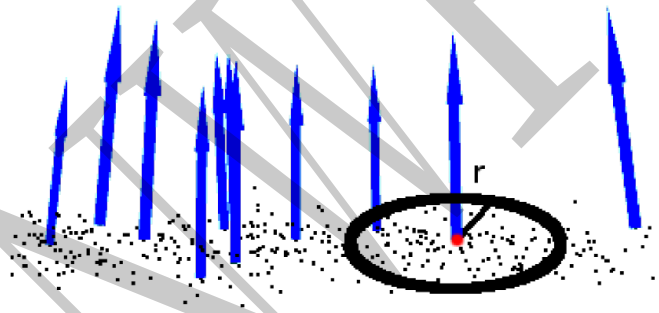


Figure 4.2 Example of a radius r search to determine the set of neighbors $\mathcal{P}^k$ for the query point (marked with red). $\mathcal{P}^k$ is used to estimate an approximative surface normal at the query point (blue arrow).

Basic cue2: Normals

- The way of computing normals-> covariance analysis

The plane is represented as a point $\boldsymbol{x}$ and a normal vector $\vec{\boldsymbol{n}}$, and the distance from a point $\boldsymbol{p}_i \in \mathcal{P}^k$ to the plane is defined as $d_i = (\boldsymbol{p}_i - \boldsymbol{x}) \cdot \vec{\boldsymbol{n}}$. The values of $\boldsymbol{x}$ and $\vec{\boldsymbol{n}}$ are computed in a least-square sense so that $d_i = 0$. By taking:

$$\boldsymbol{x} = \overline{\boldsymbol{p}} = \frac{1}{k} \cdot \sum_{i=1}^{k} \boldsymbol{p}_i \tag{4.6}$$

as the centroid of $\mathcal{P}^k$, the solution for $\vec{\boldsymbol{n}}$ is given by analyzing the eigenvalues and eigenvectors

Basic cue2: Normals

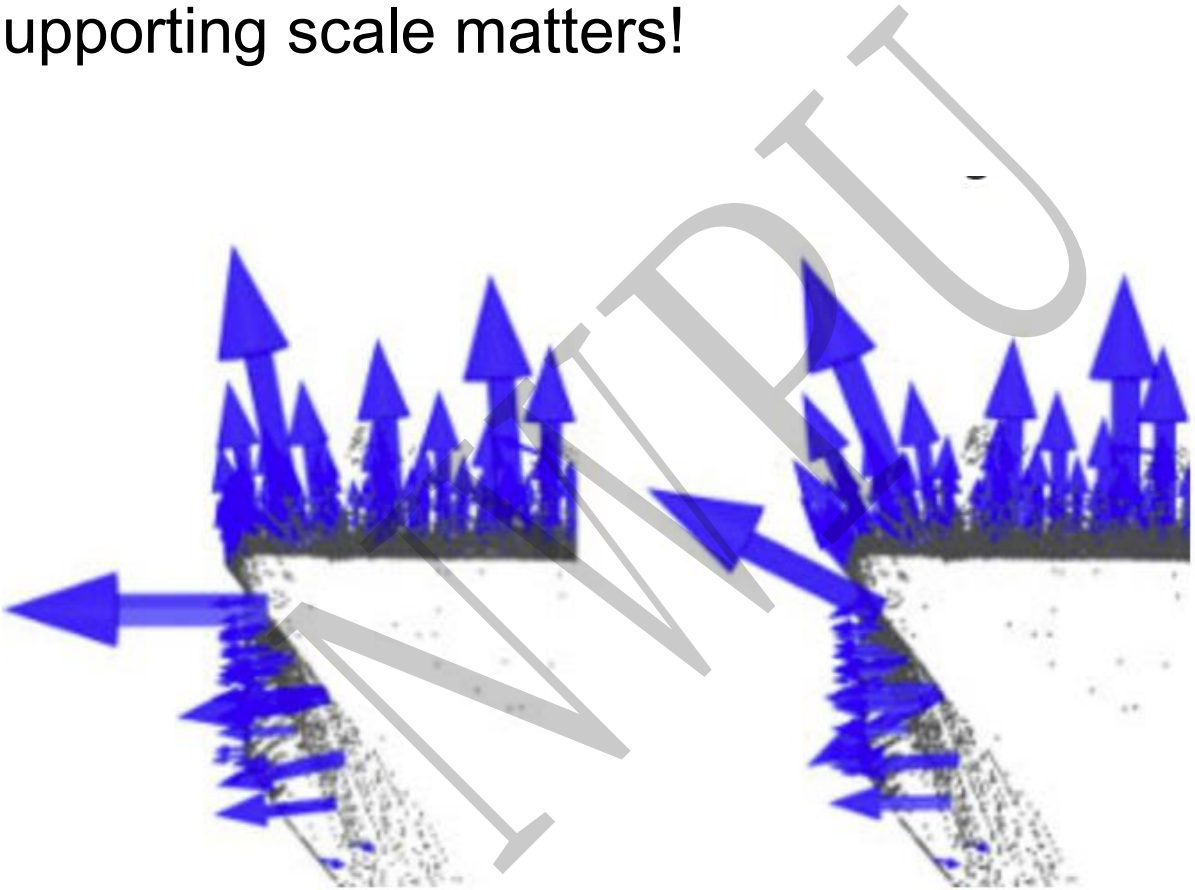
- The way of computing normals-> covariance analysis

of the covariance matrix $\mathcal{C} \in \mathbb{R}^{3\times3}$ of $\mathcal{P}^k$, expressed as:

$$\mathcal{C} = \frac{1}{k}\sum_{i=1}^{k} \xi_i \cdot (\boldsymbol{p}_i - \overline{\boldsymbol{p}}) \cdot (\boldsymbol{p}_i - \overline{\boldsymbol{p}})^T,\ \mathcal{C} \cdot \vec{\mathrm{v}}_j = \lambda_j \cdot \vec{\mathrm{v}}_j,\ j \in \{0,1,2\} \tag{4.7}$$

Basic cue2: Normals

- The supporting scale matters!



Basic cue2: Normals

- The sign ambiguity of normals

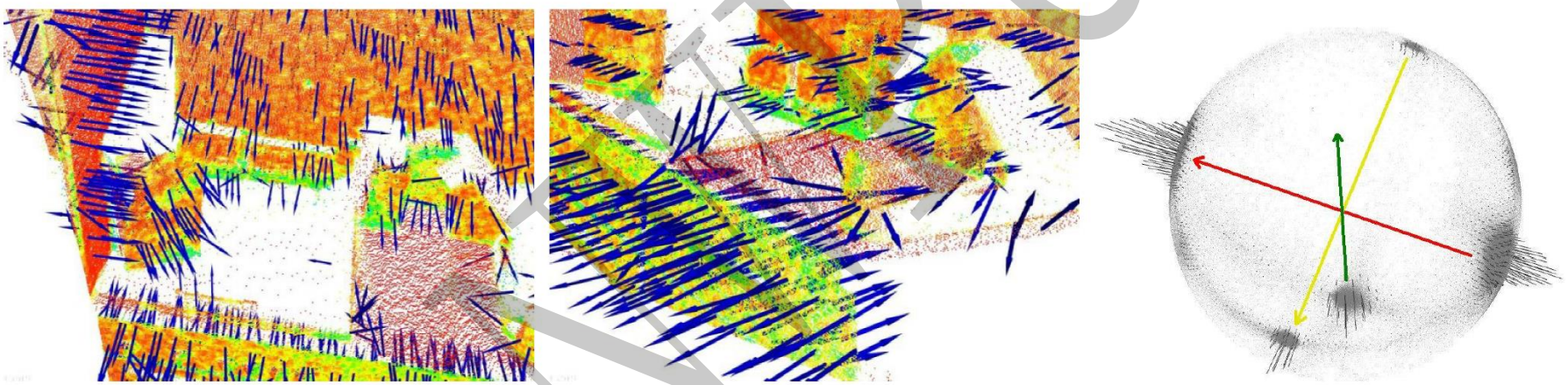


Figure 4.8 Left and middle: inconsistently oriented surface normals $\vec{n}_i$ acquired using PCA in a 2.5D dataset $\mathcal{P}$. Right: the resultant EGI (normal sphere) for $\mathcal{P}$.

Basic cue2: Normals

- Remove sign ambiguity with a given viewpoint

$$\vec{n}_i \cdot (\mathrm{v}_p - \boldsymbol{p}_i) > 0 \tag{4.9}$$

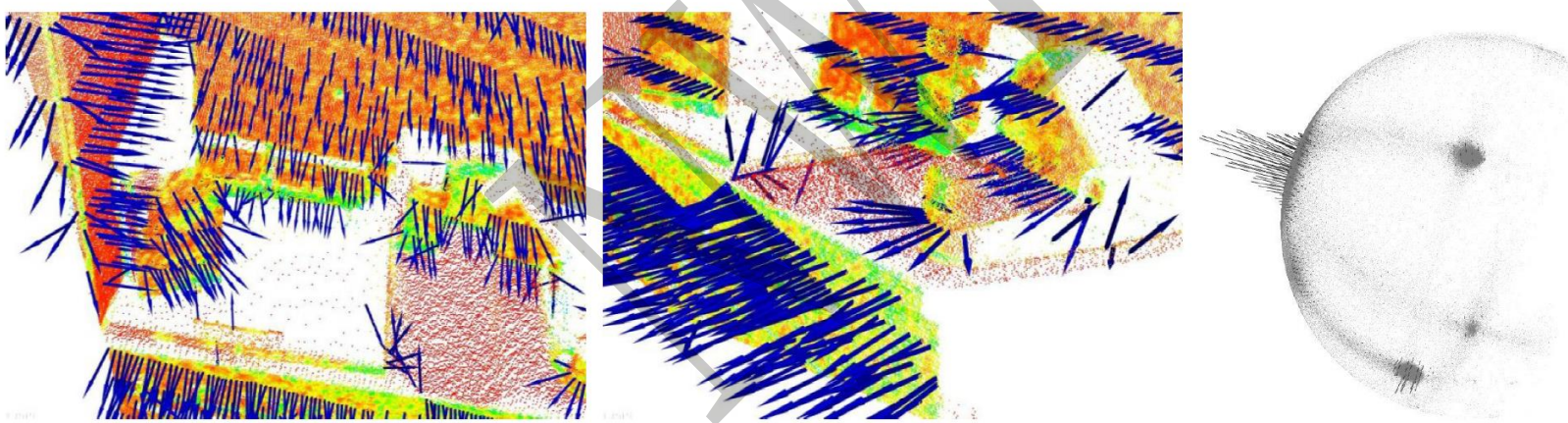
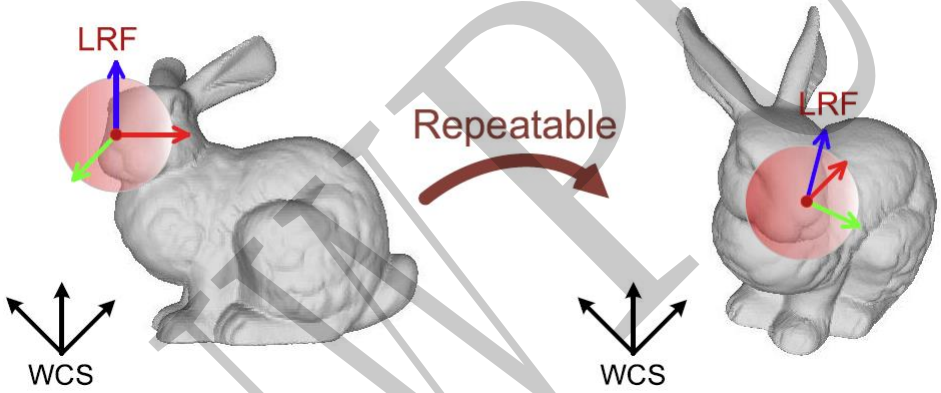


Figure 4.9 Left and middle: consistently re-oriented surface normals $\vec{n}_i$ satisfying equation 4.9. Right: the resultant EGI (normal sphere) after re-orientation for $\mathcal{P}$.

Basic cue3: Local reference frames

- An independent local reference frame (LRF) at a local scale



$$\mathcal{L} = \left[\mathcal{L}.\mathbf{x} \quad \mathcal{L}.\mathbf{y} \quad \mathcal{L}.\mathbf{z} \right], \tag{1}$$

where $\mathcal{L}.\mathbf{x}$, $\mathcal{L}.\mathbf{y}$ and $\mathcal{L}.\mathbf{z}$ are 3×1 vectors that store the coordinates of the unit vectors.

Next, some traditional yet good features for point clouds

1. Spin image

- Background

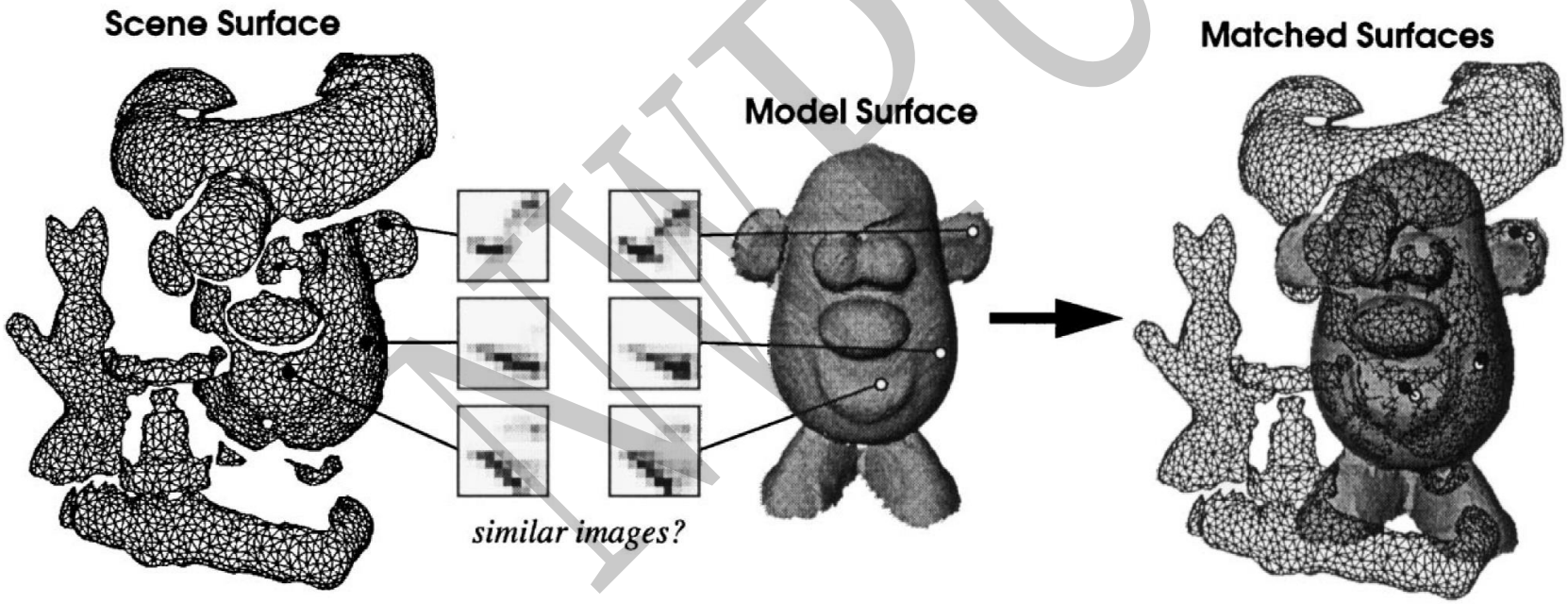
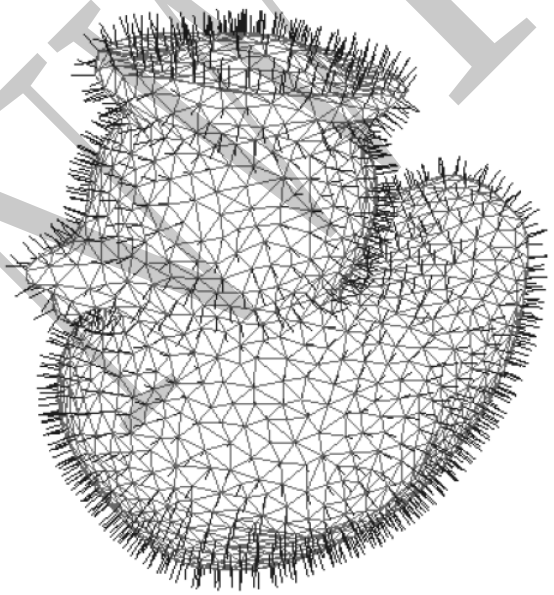


Fig. 1. Recognition algorithm overview. Images generated at oriented points on a model surface mesh are matched to images generated from a scene surface mesh. Matched images determine point correspondences from which a rigid transformation mapping the model into the scene can be computed.

1. Spin image

1) Normal estimation

- The fundamental shape element we use to perform object recognition is an oriented point, a three-dimensional point with an associated direction



1. Spin image

2) 3D-to-2D invariant projection

$$S_O : R^3 \rightarrow R^2$$

$$S_O(x) \rightarrow (\alpha, \beta) = (\sqrt{||x-p||^2 - (n\cdot(x-p))^2}, n\cdot(x-p))$$

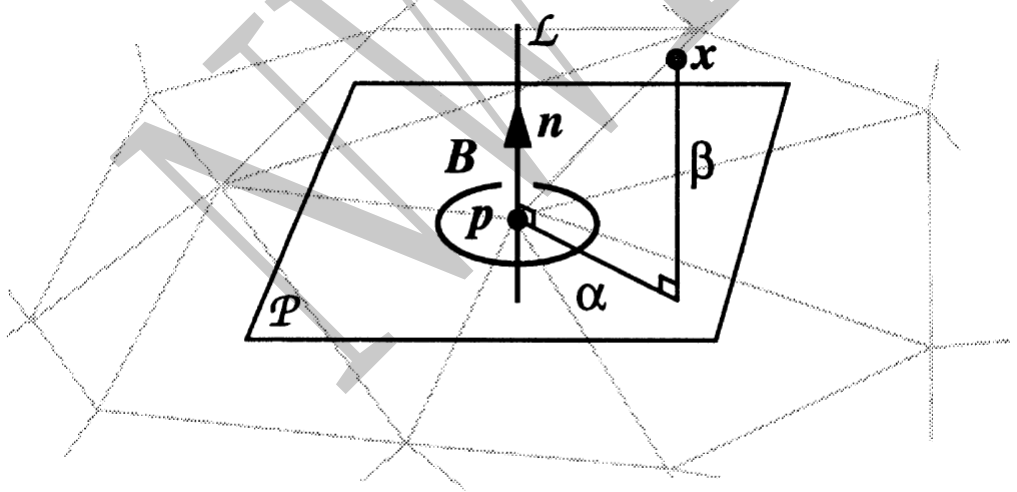


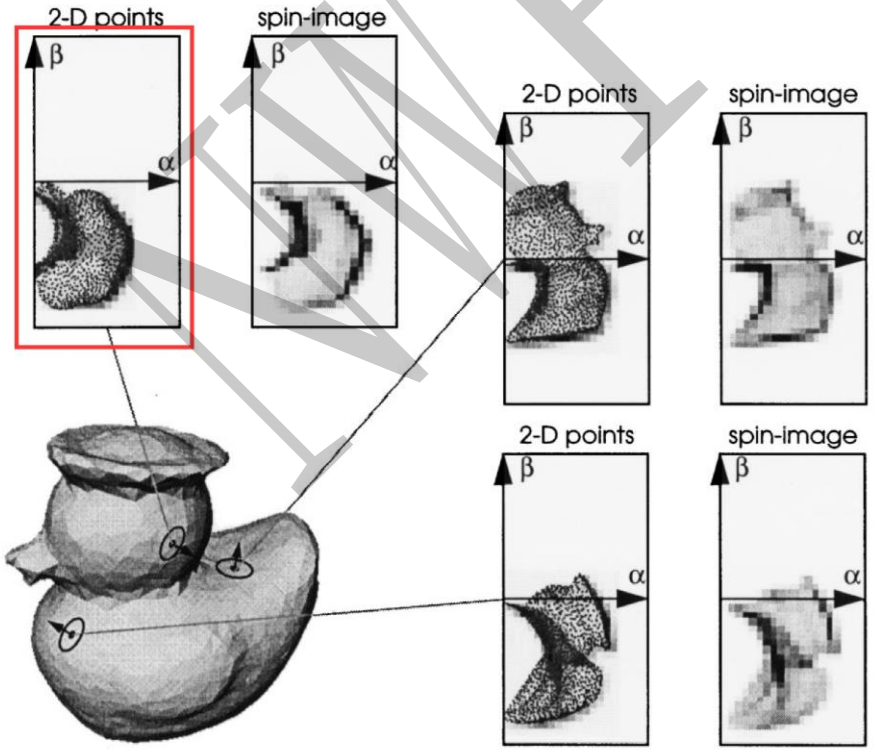
Fig. 2. Oriented point basis.

1. Spin image

2) 3D-to-2D invariant projection

$$S_O : R^3 \rightarrow R^2$$

$$S_O(x) \rightarrow (\alpha, \beta) = (\sqrt{||x-p||^2 - (n\cdot(x-p))^2}, n\cdot(x-p))$$

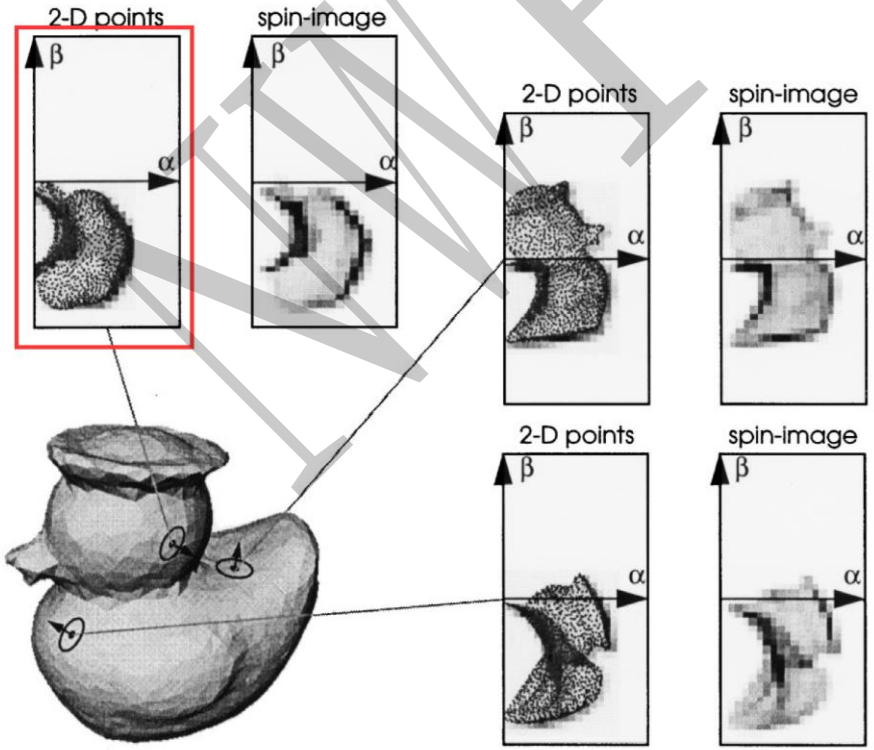


1. Spin image

2) 3D-to-2D invariant projection

$$S_O : R^3 \rightarrow R^2$$

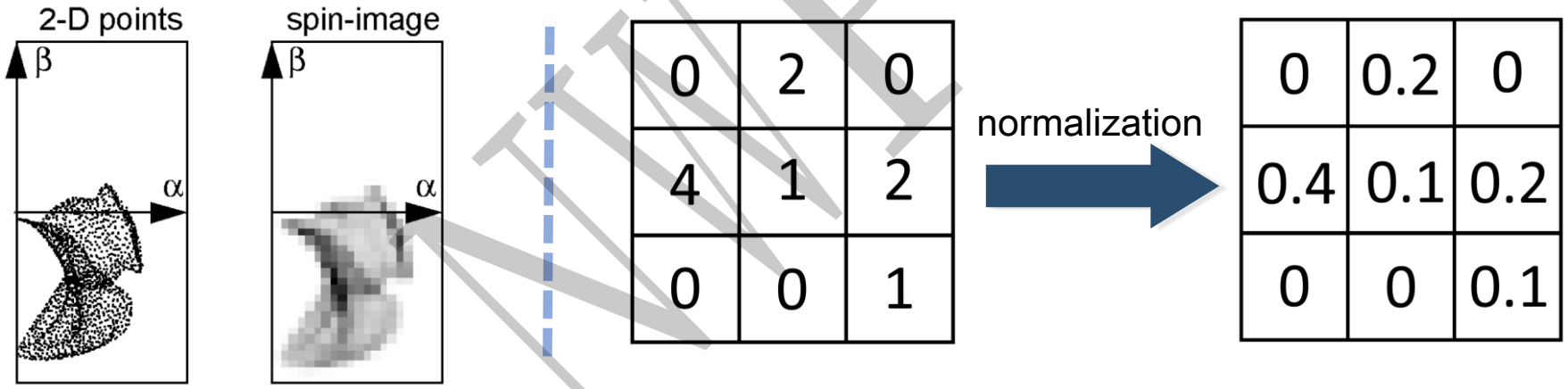
$$S_O(x) \rightarrow (\alpha, \beta) = (\sqrt{||x-p||^2 - (n\cdot(x-p))^2}, n\cdot(x-p))$$



1. Spin image

3) Bin value calculation & feature generation

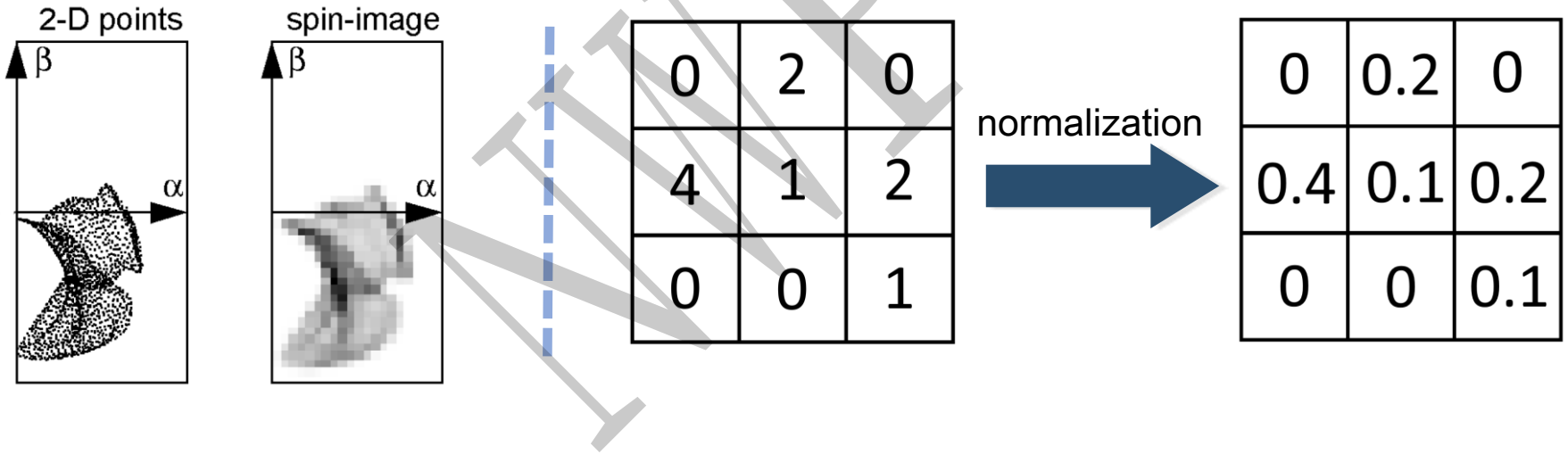
- The density of each bin is the bin value of the pixel
- The concatenation of all pixel values is the spin image feature



1. Spin image

3) Bin value calculation & feature generation

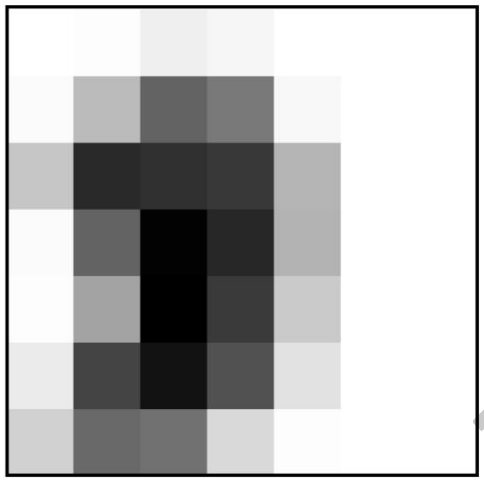
- The density of each bin is the bin value of the pixel
- The concatenation of all pixel values is the spin image feature



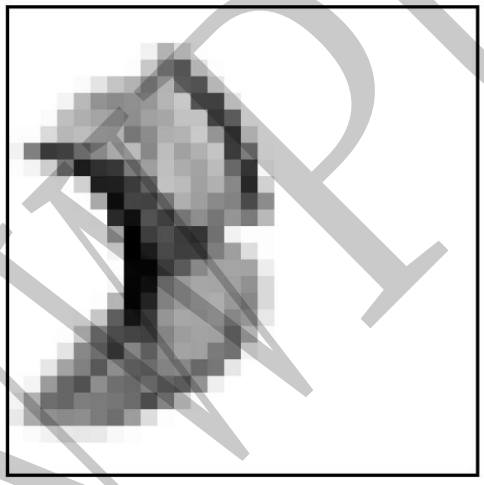
Q: Why normalization?

1. Spin image

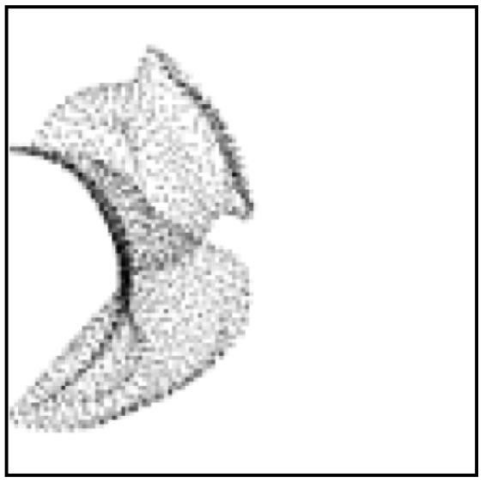
The bin number of spin image matters



(a)



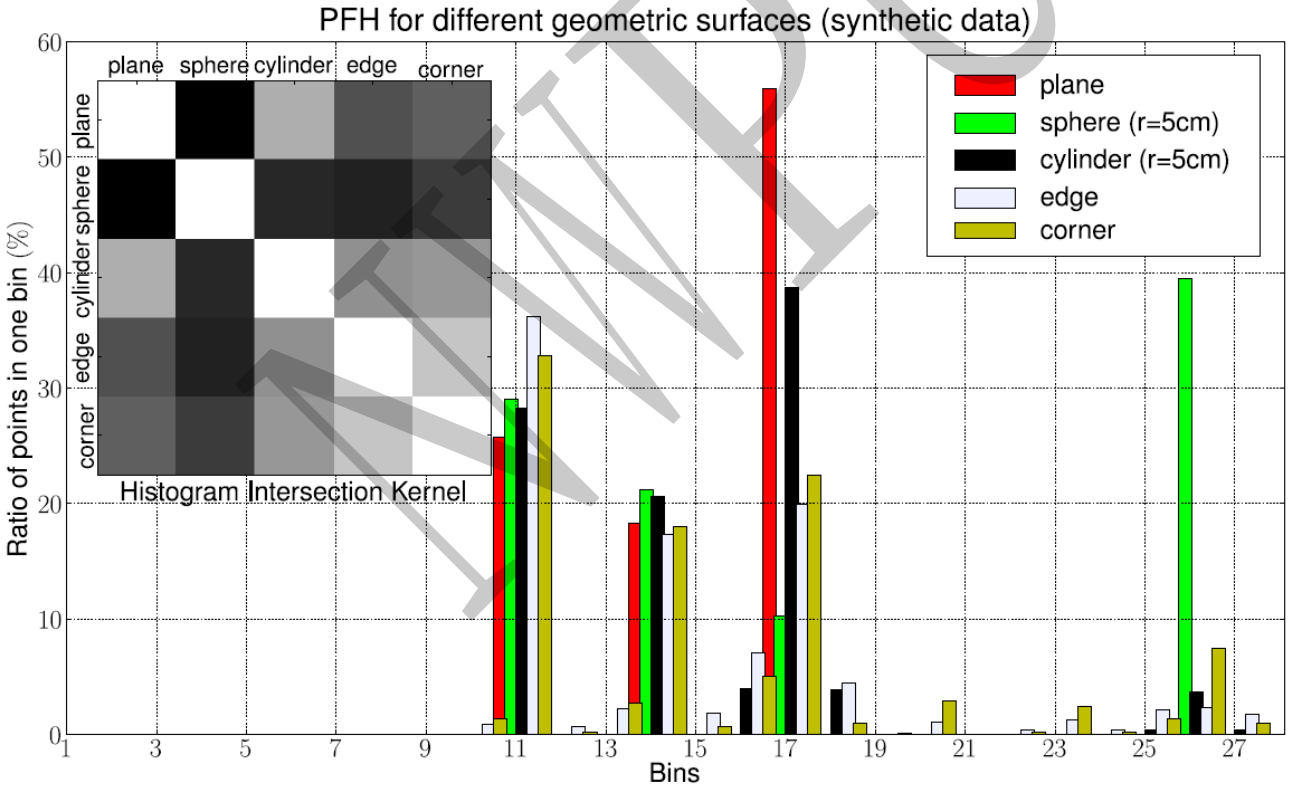
(b)



(c)

2. FPFH

- Fast Point Feature Histogram
- A speed-up version of Point Feature Histogram



2. FPFH

1) Point pair attribute calculation

- The Darboux frame
- The point pair feature

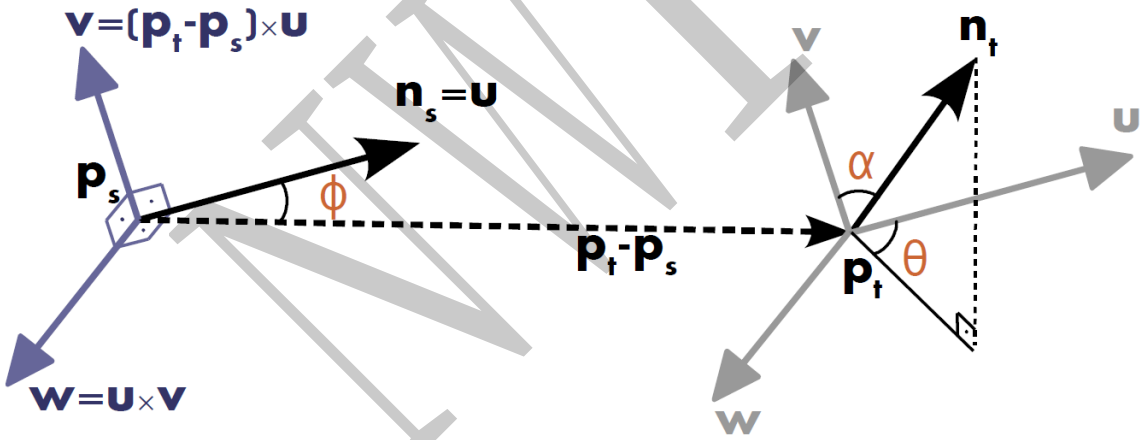


Figure 4.12 A graphical representation of the Darboux frame and the angular PFH features for a pair of points p_s and p_t with their associated normals n_s and n_t.

2. FPFH

1) Point pair attribute calculation

$$\begin{cases} u = n_s \\ v = u \times \dfrac{(p_t - p_s)}{\|p_t - p_s\|_2} \\ w = u \times v \end{cases}$$

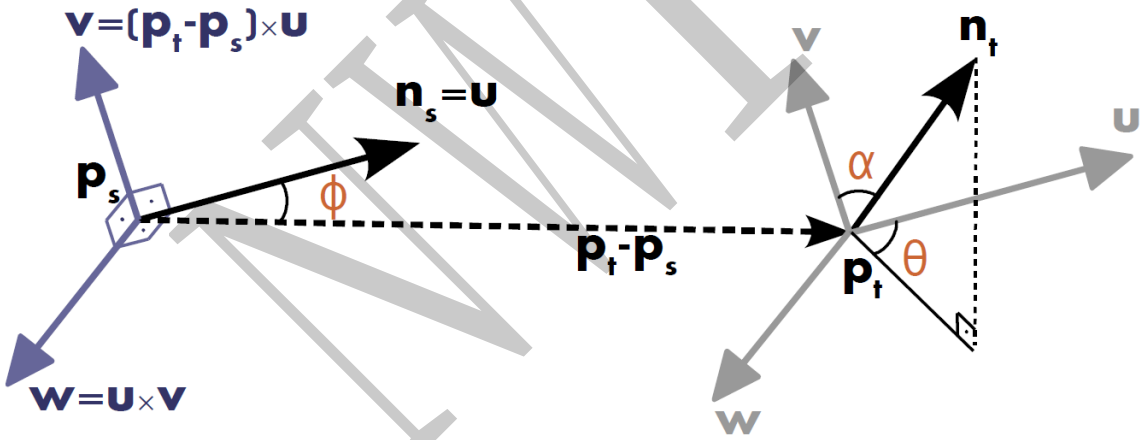


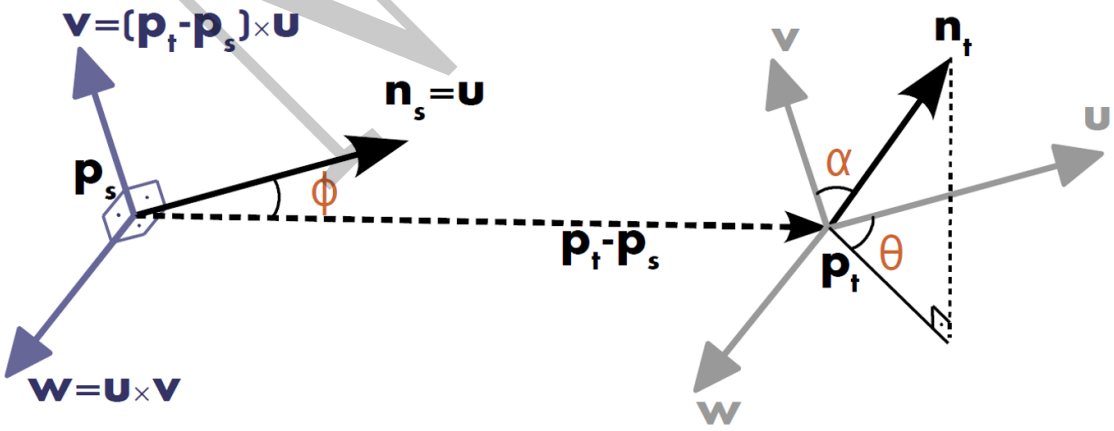
Figure 4.12 A graphical representation of the Darboux frame and the angular PFH features for a pair of points p_s and p_t with their associated normals n_s and n_t.

2. FPFH

1) Point pair attribute calculation

$$\alpha = \mathrm{v} \cdot \boldsymbol{n}_t$$
$$\phi = \mathrm{u} \cdot \frac{(\boldsymbol{p}_t - \boldsymbol{p}_s)}{d} \tag{4.20}$$
$$\theta = \arctan(\mathrm{w} \cdot \boldsymbol{n}_t, \mathrm{u} \cdot \boldsymbol{n}_t)$$

where d is the Euclidean distance between the two points $\boldsymbol{p}_s$ and $\boldsymbol{p}_t$, $d = \|\boldsymbol{p}_t - \boldsymbol{p}_s\|_2$. The quadruplet $\langle \alpha, \phi, \theta, d \rangle$ is computed for each pair of two points in the $\mathcal{P}^{k_2}$ neighborhood, therefore reducing the 12 values (x, y, z, n_x, n_y, n_z for each point) of the two points and their normals to 4.



2. FPFH

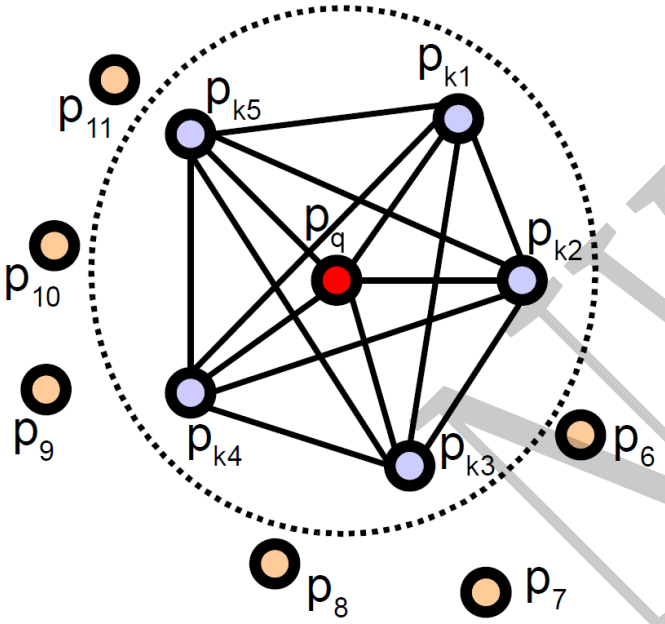
2) Statistical histogram generation

- The distance attribute d is less distinctive via experiments
- Final: 3 angle attributes
- Way1: N^3 bin in a 3-dimensional attribute space
- Way2: Incremental binning (ref: the following)

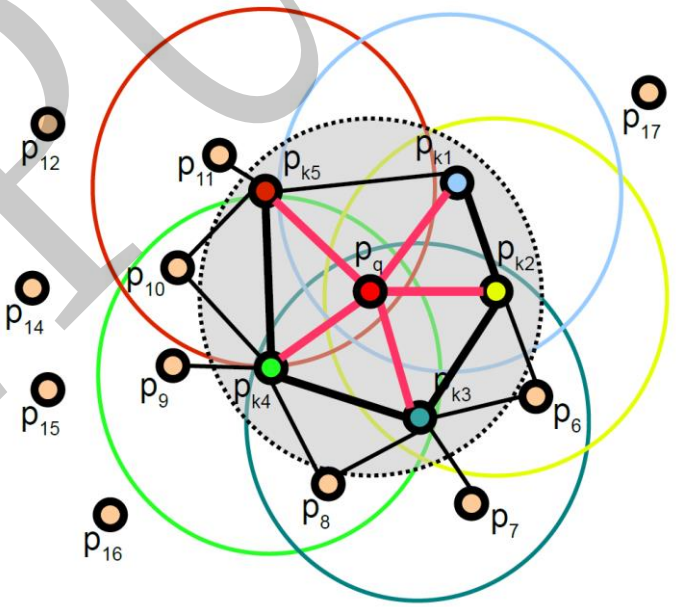
[RMBB08b] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, December 17-20 2008.

2. FPFH

Neighbor scale definition for PFH and FPFH



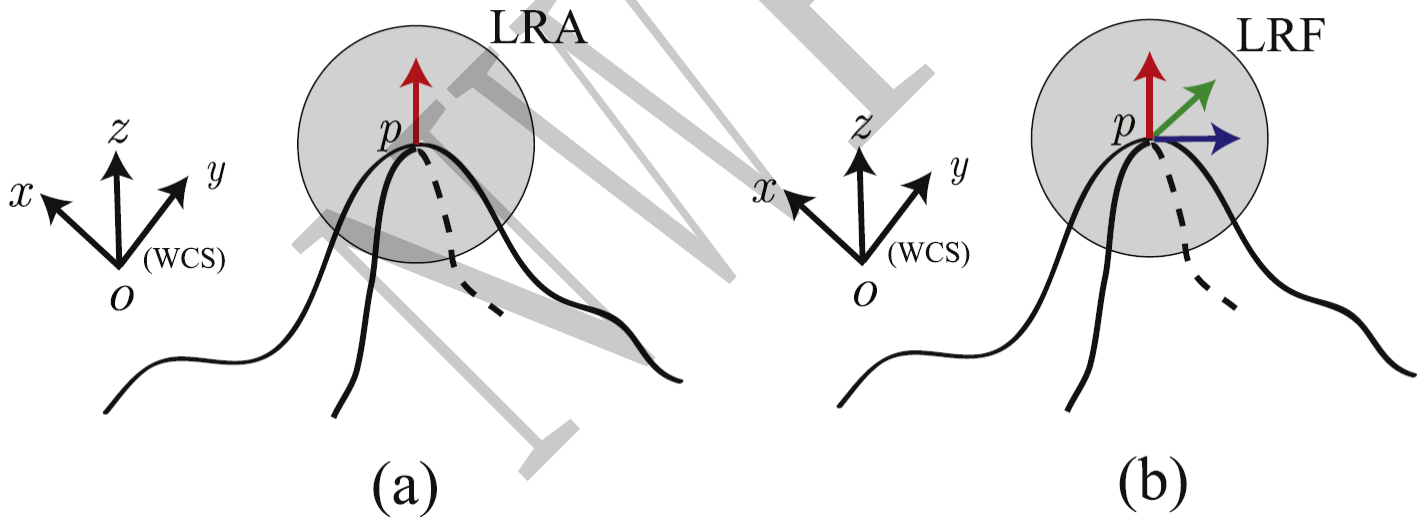
PFH: O(n^2)



FPFH: O(kn)

3. TOLDI

- Triple Orthogonal Local Depth Images
- An LRF-based descriptor



3. TOLDI

- LRF: more comprehensive spatial information encoding

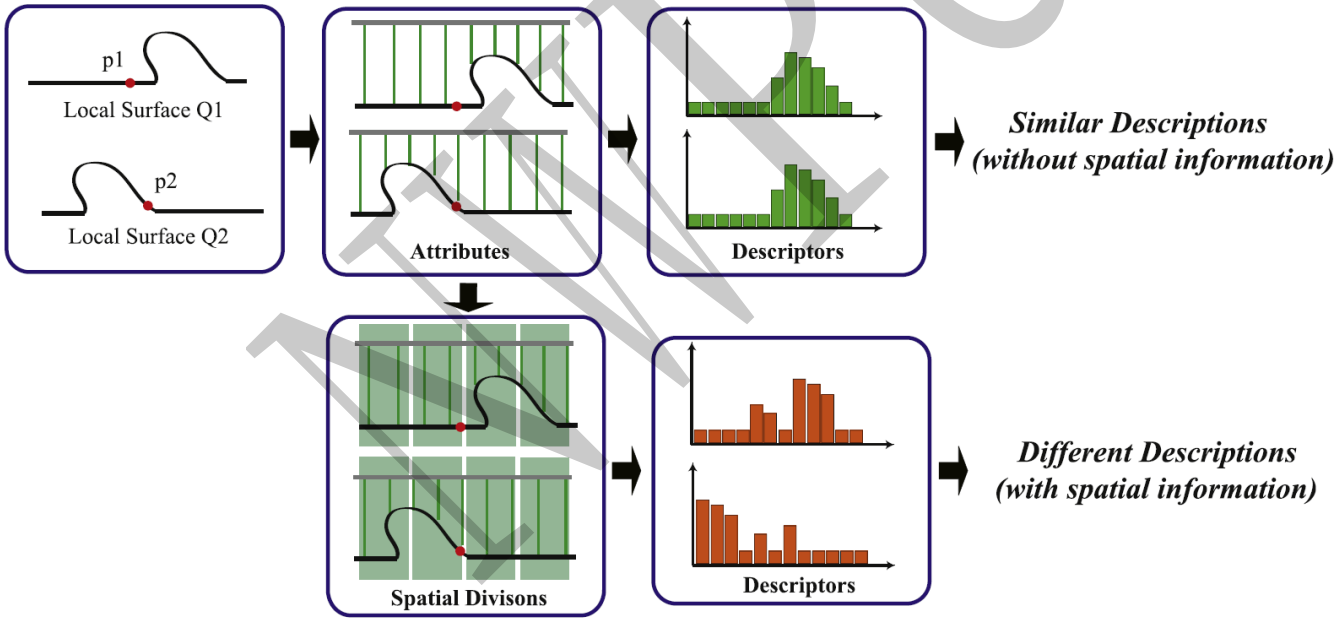


Fig. 2. Descriptions of two local surfaces (shown in 2D) with and without spatial information. The red points p_1 and p_2 denote two keypoints, which are not corresponded.

3. TOLDI

1) LRF construction

- Z-axis

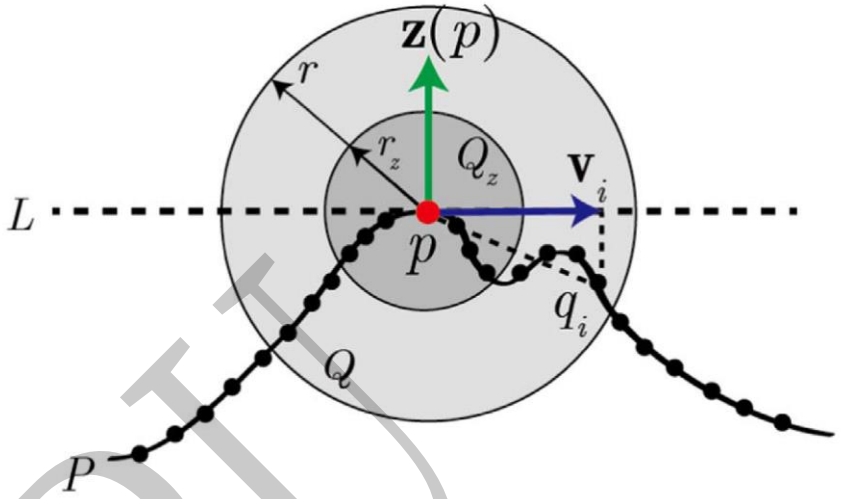


Fig. 1. Illustration of the proposed LRF in 2D. The red point, green arrow and blue arrow denote the keypoint in the local surface, the *z*-axis and the projection vector of an exemplar radius neighbor of the keypoint, respectively. See text for the representations of variables. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$Cov(Q_z) = \begin{bmatrix} q_1^z - \overline{q^z} \\ \cdots \\ q_s^z - \overline{q^z} \end{bmatrix}^T \cdot \begin{bmatrix} q_1^z - \overline{q^z} \\ \cdots \\ q_s^z - \overline{q^z} \end{bmatrix}, \tag{2}$$

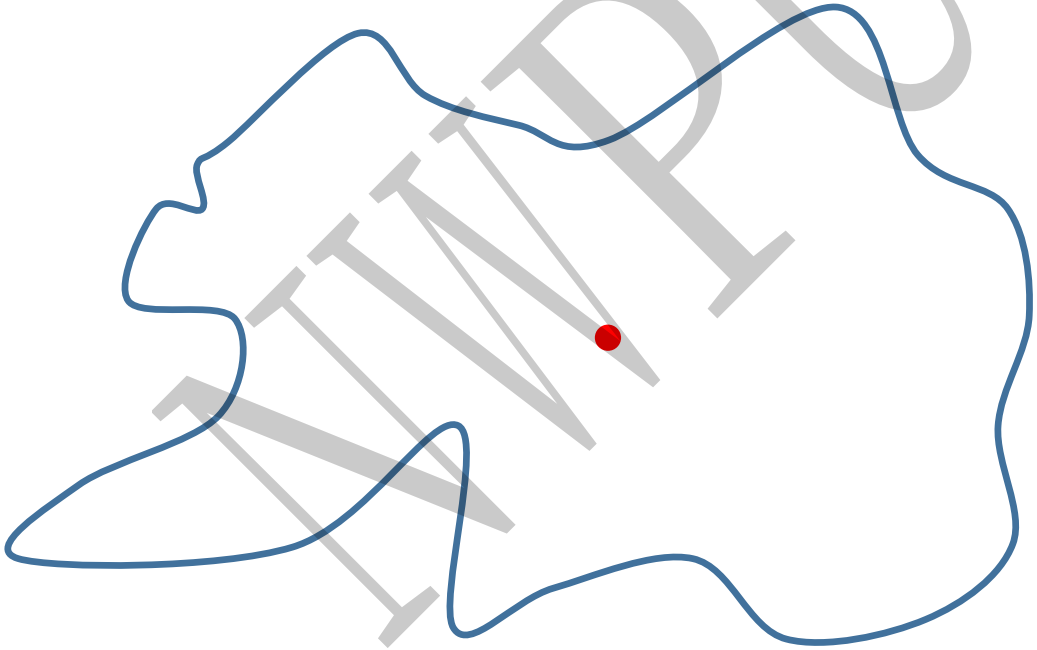
where s is the size of Q_z, and $\overline{q^z}$ is the centroid of Q_z. The eigenvector $\mathbf{n}(p)$ corresponding to the minimum eigenvalue of $Cov(Q_z)$ is computed

$$\mathbf{z}(p) = \begin{cases} \mathbf{n}(p), & \text{if } \mathbf{n}(p) \cdot \sum_{i=1}^{k} \mathbf{q}_i \mathbf{p} \geq 0 \\ -\mathbf{n}(p), & \text{otherwise} \end{cases},$$

3. TOLDI

1) LRF construction

- X-axis: a more challenging issue

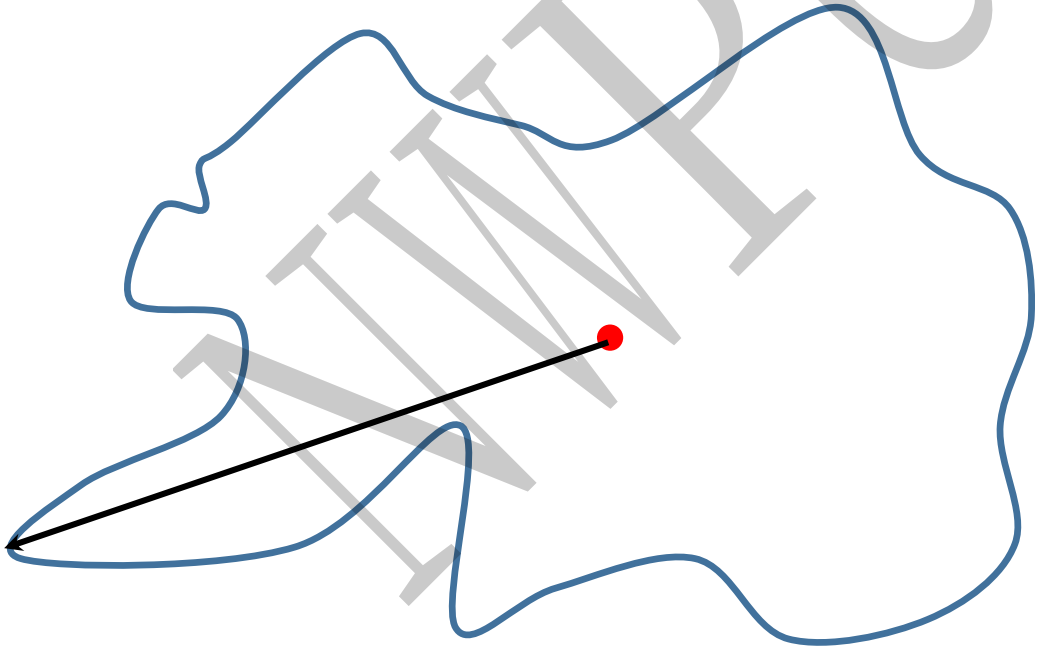


Help me find a stable direction

3. TOLDI

1) LRF construction

- X-axis: a more challenging issue

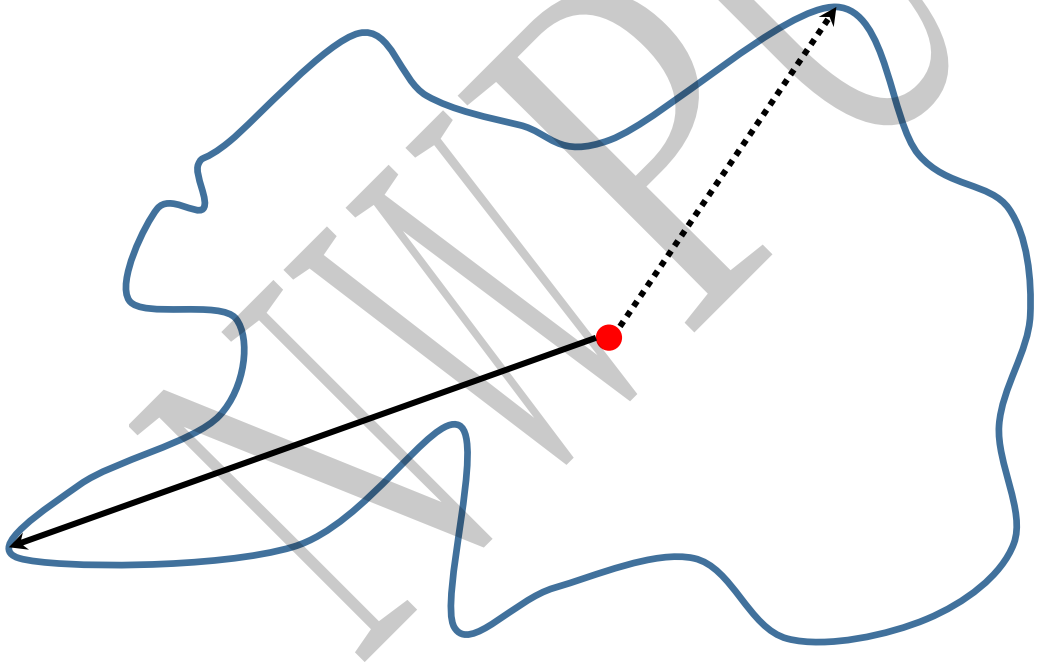


Solution 1

3. TOLDI

1) LRF construction

- X-axis: a more challenging issue

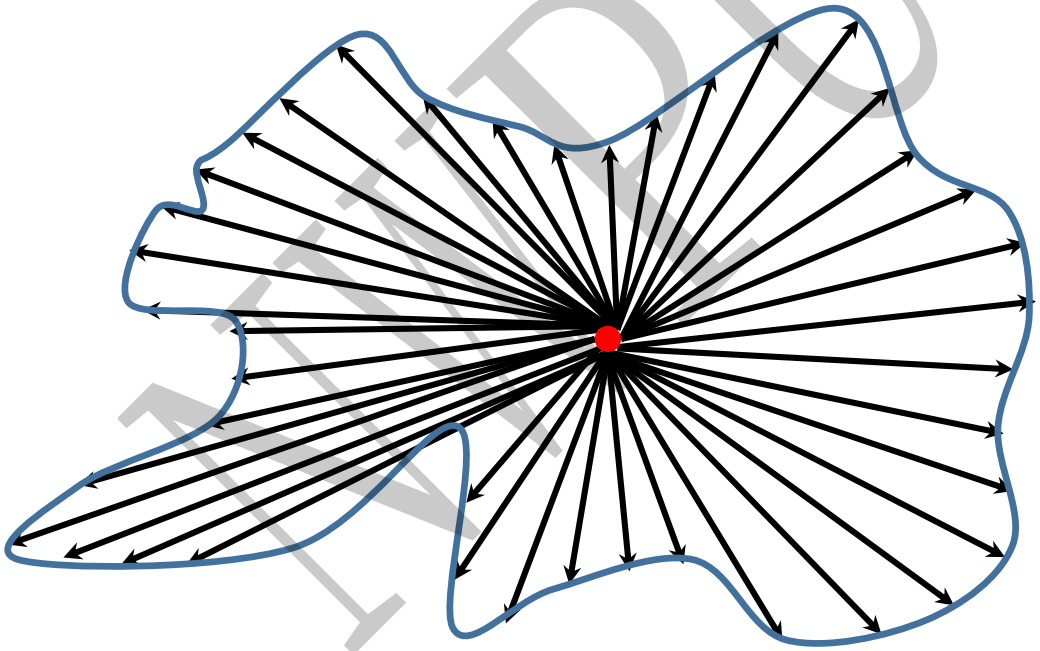


Solution 2

3. TOLDI

1) LRF construction

- X-axis: a more challenging issue

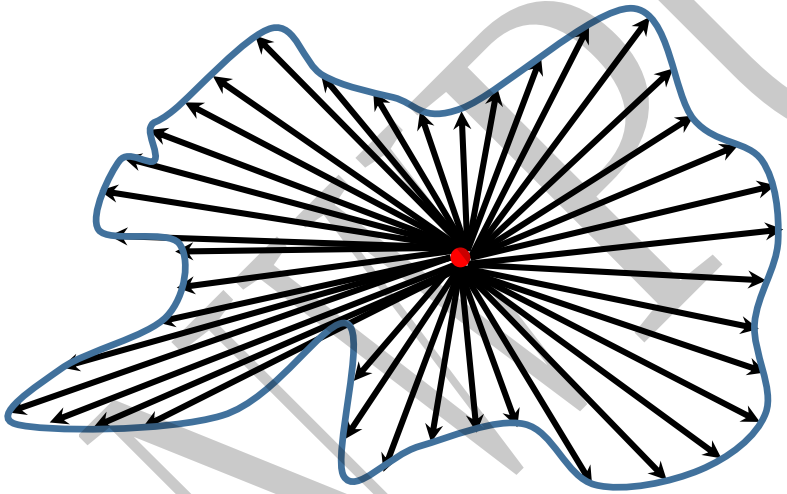


Final solution

3. TOLDI

1) LRF construction

- X-axis: a more challenging issue



1. Why use all points?
   To enhance robustness
2. How to integrate all these vectors into one vector?
   Vector sum (efficient and a powerful representation)

3. TOLDI

1) LRF construction

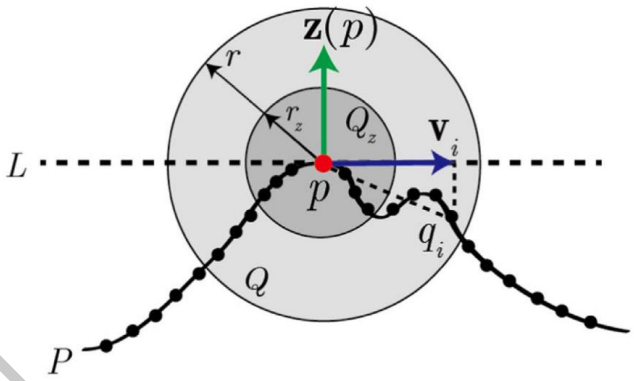


Fig. 1. Illustration of the proposed LRF in 2D. The red point, green arrow and blue arrow denote the keypoint in the local surface, the *z*-axis and the projection vector of an exemplar radius neighbor of the keypoint, respectively. See text for the representations of variables. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- X-axis: a more challenging issue

$$\mathbf{v}_i = \mathbf{pq}_i - (\mathbf{pq}_i \cdot \mathbf{z}(p)) \cdot \mathbf{z}(p). \tag{4}$$

$$\mathbf{x}(p) = \sum_{i=1}^{k} w_{i1} w_{i2} \mathbf{v}_i \Big/ \left\| \sum_{i=1}^{k} w_{i1} w_{i2} \mathbf{v}_i \right\|, \tag{5}$$

$$w_{i1} = (r - \|p - q_i\|)^2, \tag{6}$$

w_{i2} is a weight that related to the projection distance from q_i to L, that is:

$$w_{i2} = (\mathbf{pq}_i \cdot \mathbf{z}(p))^2, \tag{7}$$

3. TOLDI

2) Feature generation

- Overall pipeline

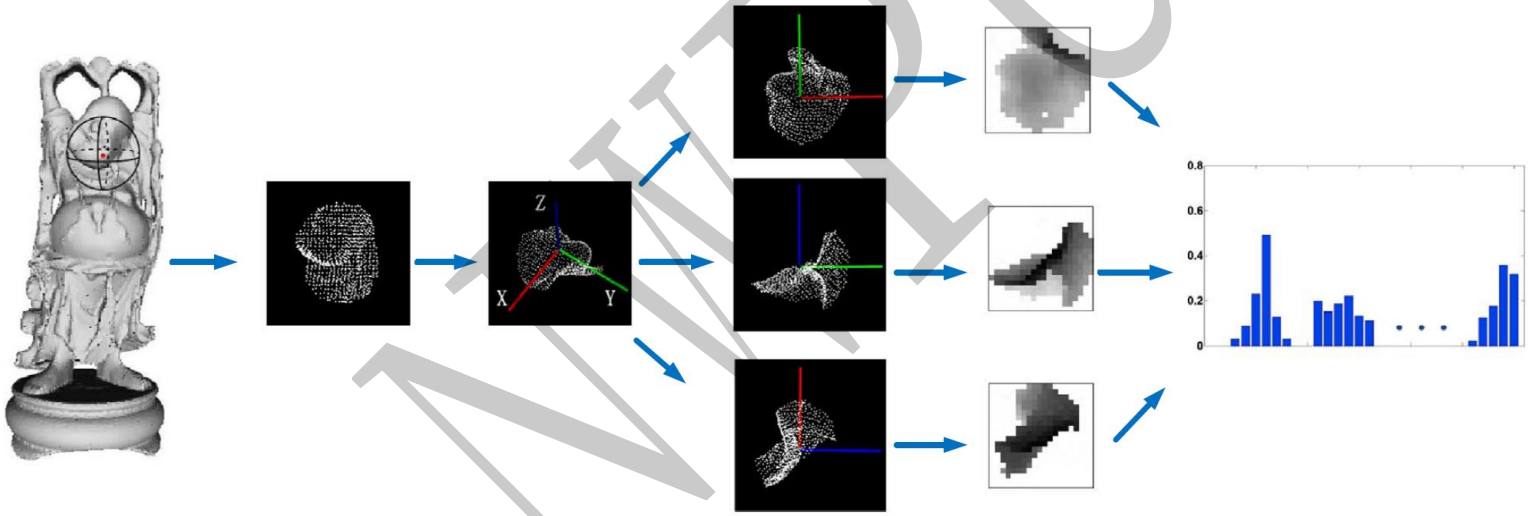


Fig. 2. An illustration of the TOLDI feature descriptor. The red point in (a) represents a keypoint in the 3D model and the points inside the surrounding sphere of the keypoint constitute a local surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. TOLDI

2) Feature generation

- Multi-view is necessary

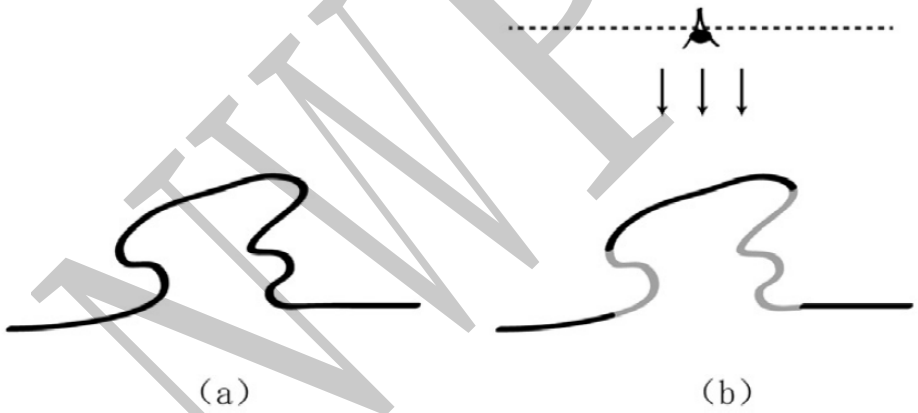
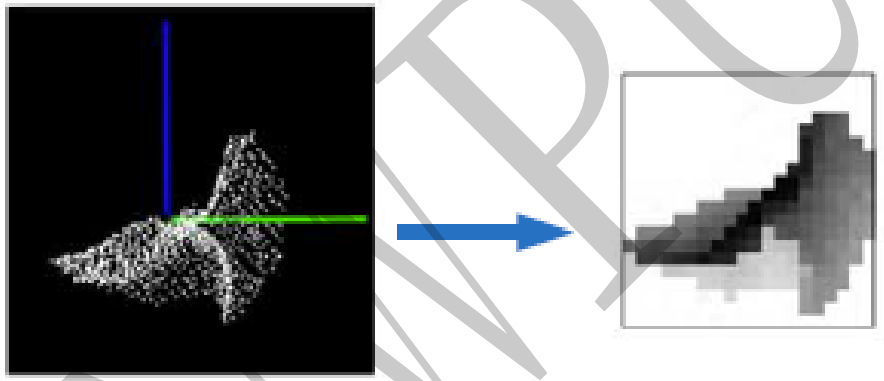


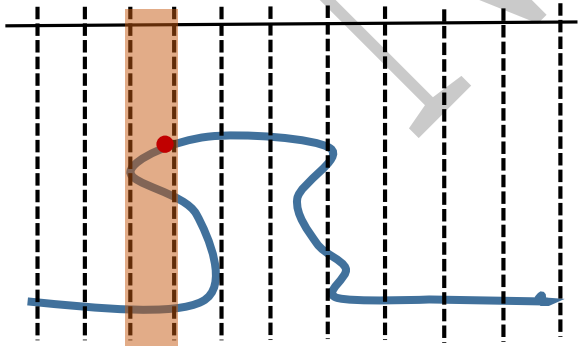
Fig. 3. Illustration of loss of information caused by occlusion from a single view. (a) A shape in 2D. (b) The captured (shown in dark) and hidden (shown in gray) information from one view.

3. TOLDI

2) Feature generation

- The pixel value of a local depth image



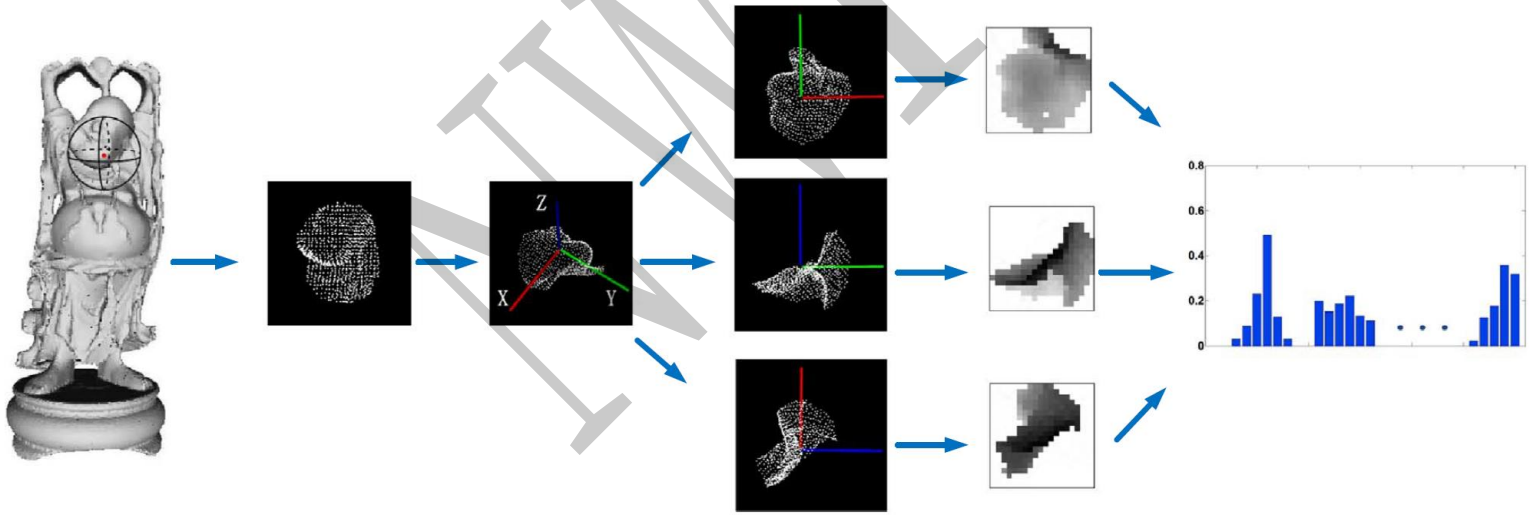


$$\begin{cases} f_i^{xy} = r - q' \; i.\, z \\ f_i^{yz} = r - q' \; i.\, x, \\ f_i^{xz} = r - q' \; i.\, y \end{cases}$$

3. TOLDI

2) Feature generation

- The concatenation of all pixel values of 3 LDI is the TOLDI feature
- 1200 dim.



(a) 3D model (b) Local surface (c) LRF (d) Three orthogonal views (e) LDI sub-features (f) TOLDI feature

Fig. 2. An illustration of the TOLDI feature descriptor. The red point in (a) represents a keypoint in the 3D model and the points inside the surrounding sphere of the keypoint constitute a local surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. RCS

- Rotational Contour Signatures
- An LRF-based descriptor as well

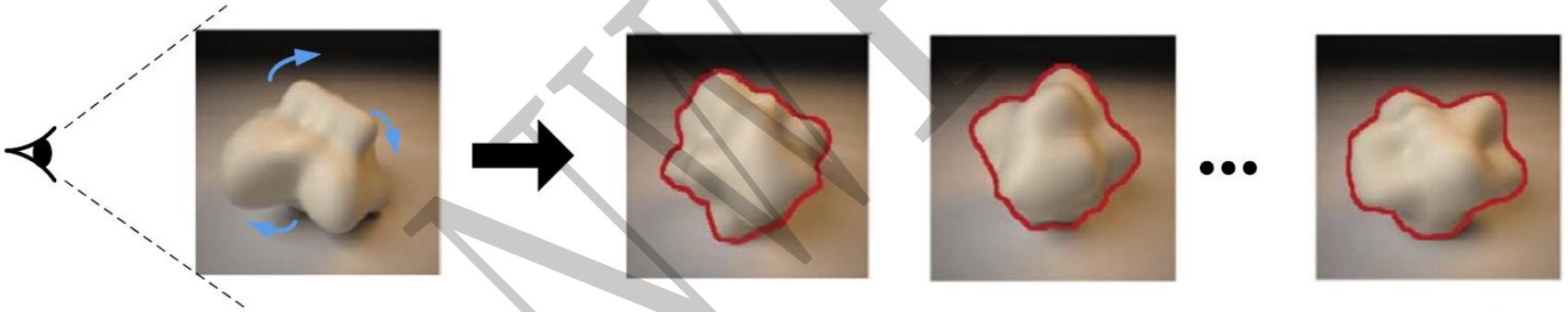
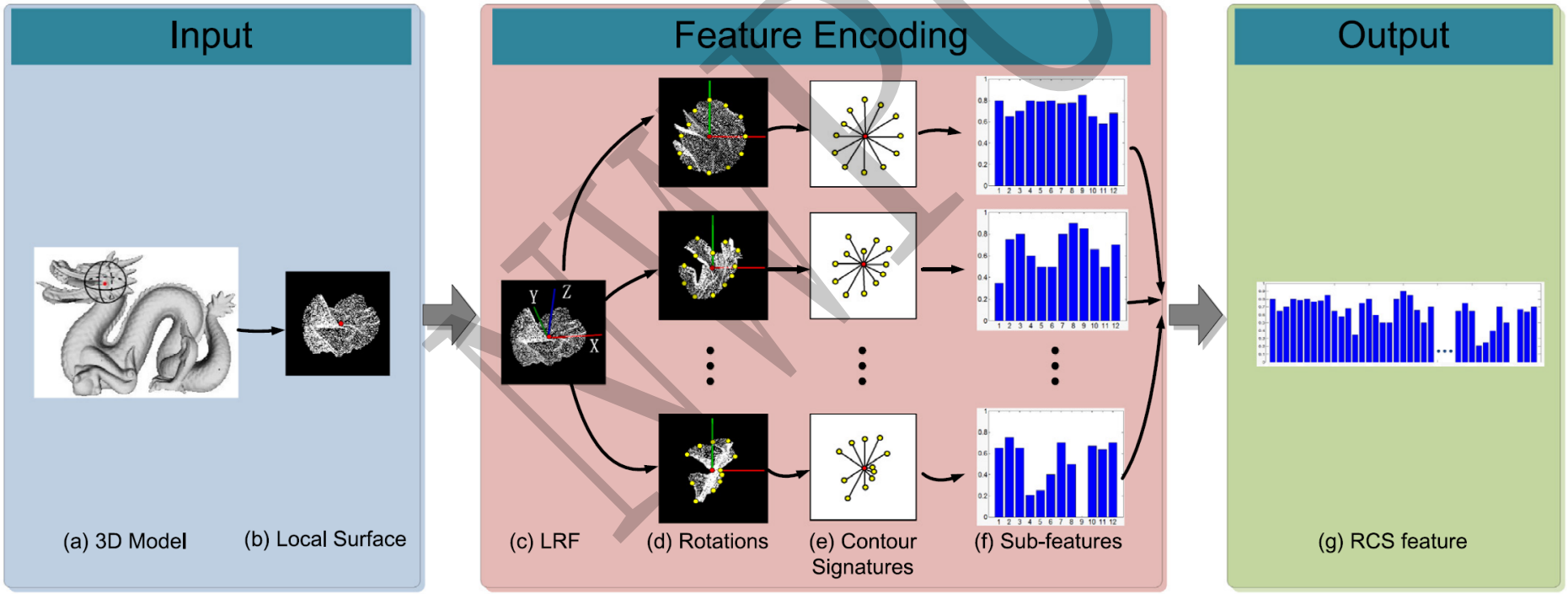


Fig. 1. Illustration of the basic inspiration of the proposed RCS method.

4. RCS

- Overall pipeline: LRF-> Surface rotation -> contour signature calculation -> contour signature concatenation

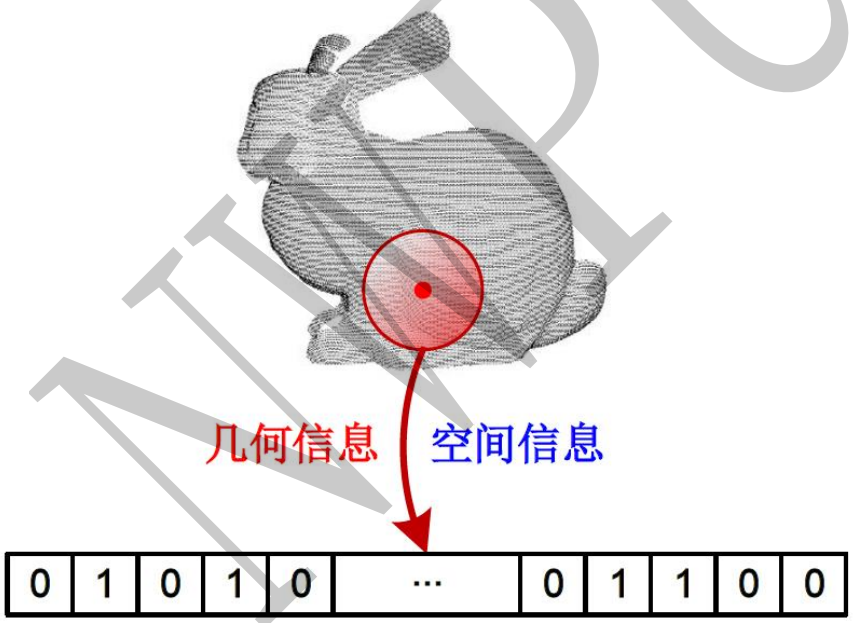


4. RCS

- Binary descriptor

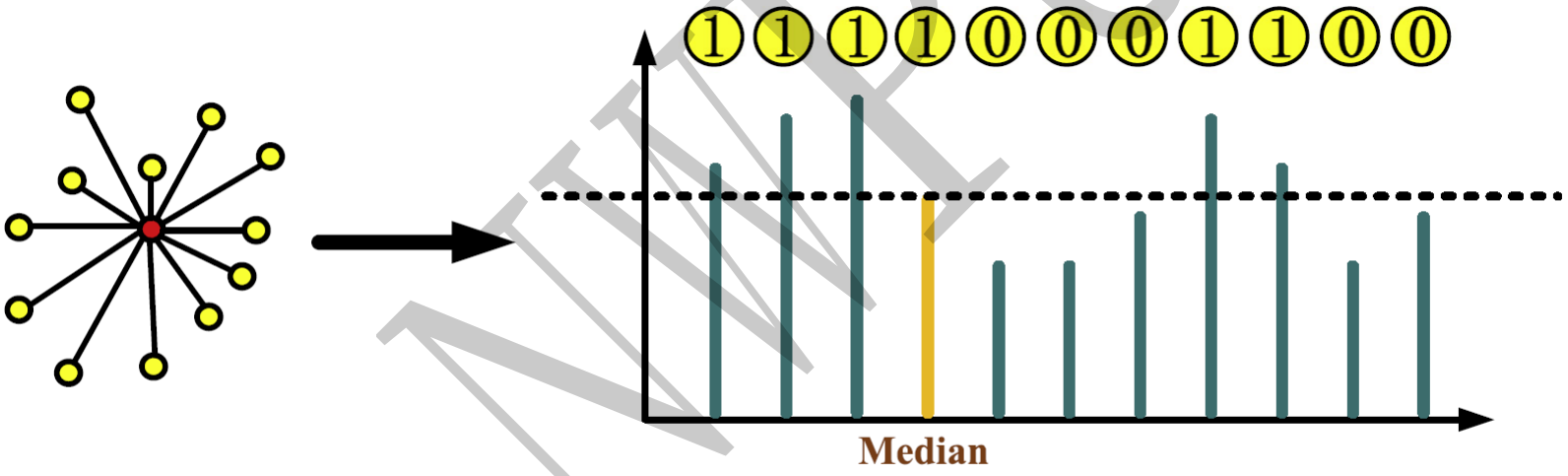
4. RCS

- Binary descriptor



4. RCS

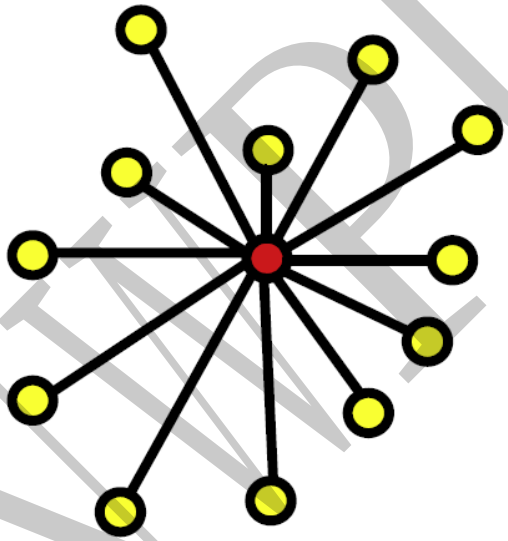
- Binary descriptor



(a) Thresholding based on Maximum Entropy

4. RCS

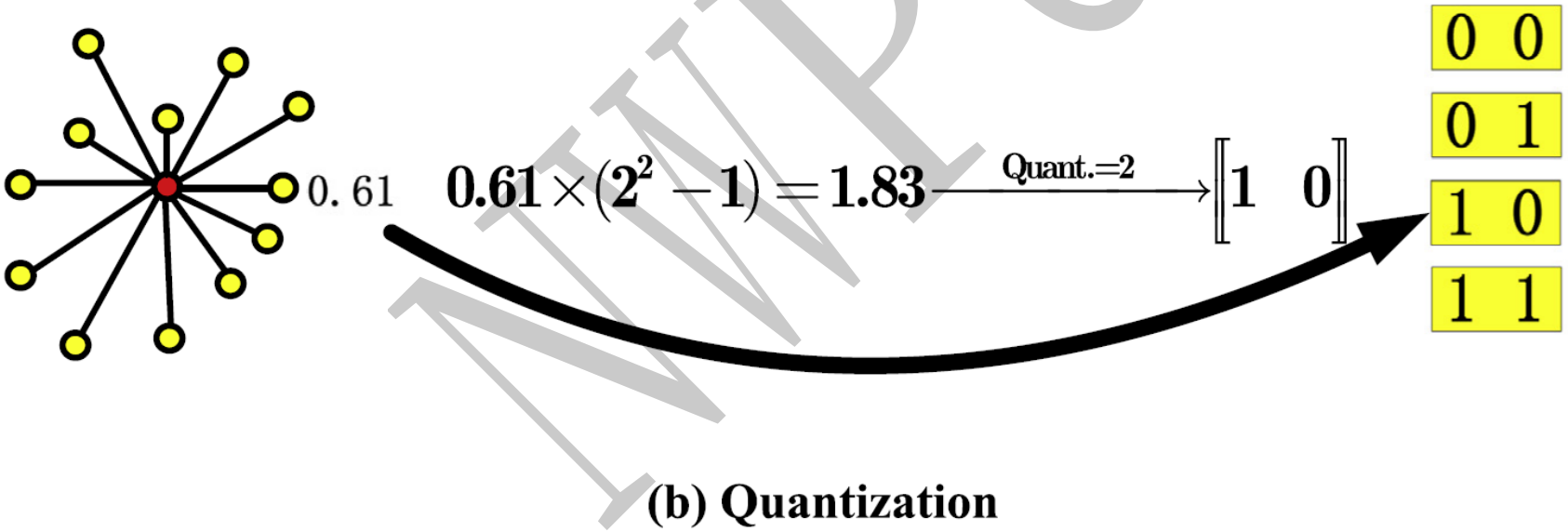
- Binary descriptor



How to transform contour signature to binary feature?

4. RCS

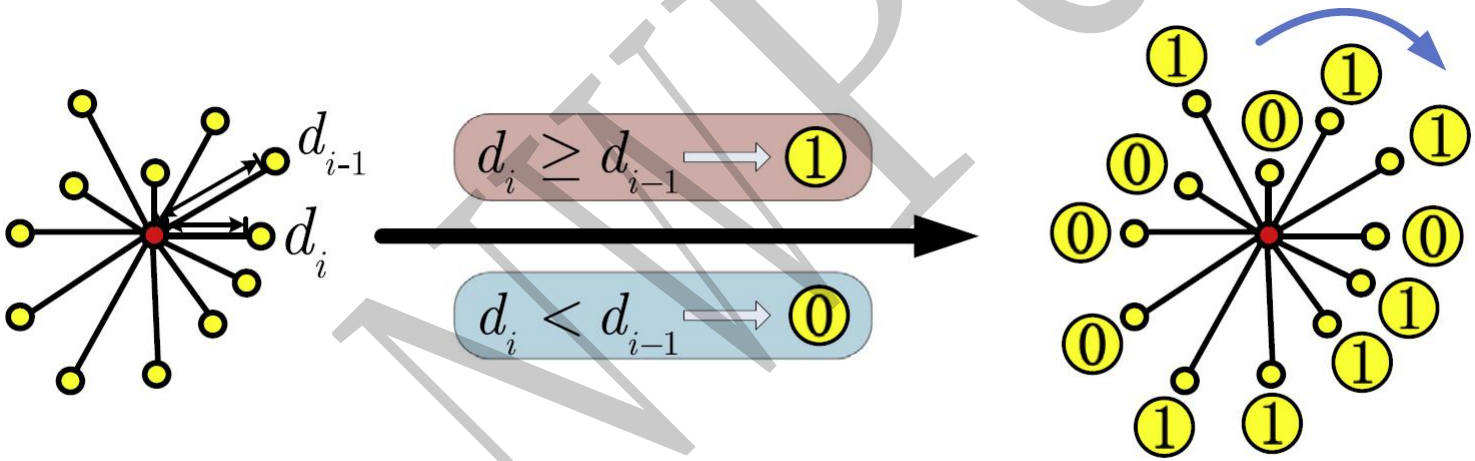
- Binary descriptor



(b) Quantization

4. RCS

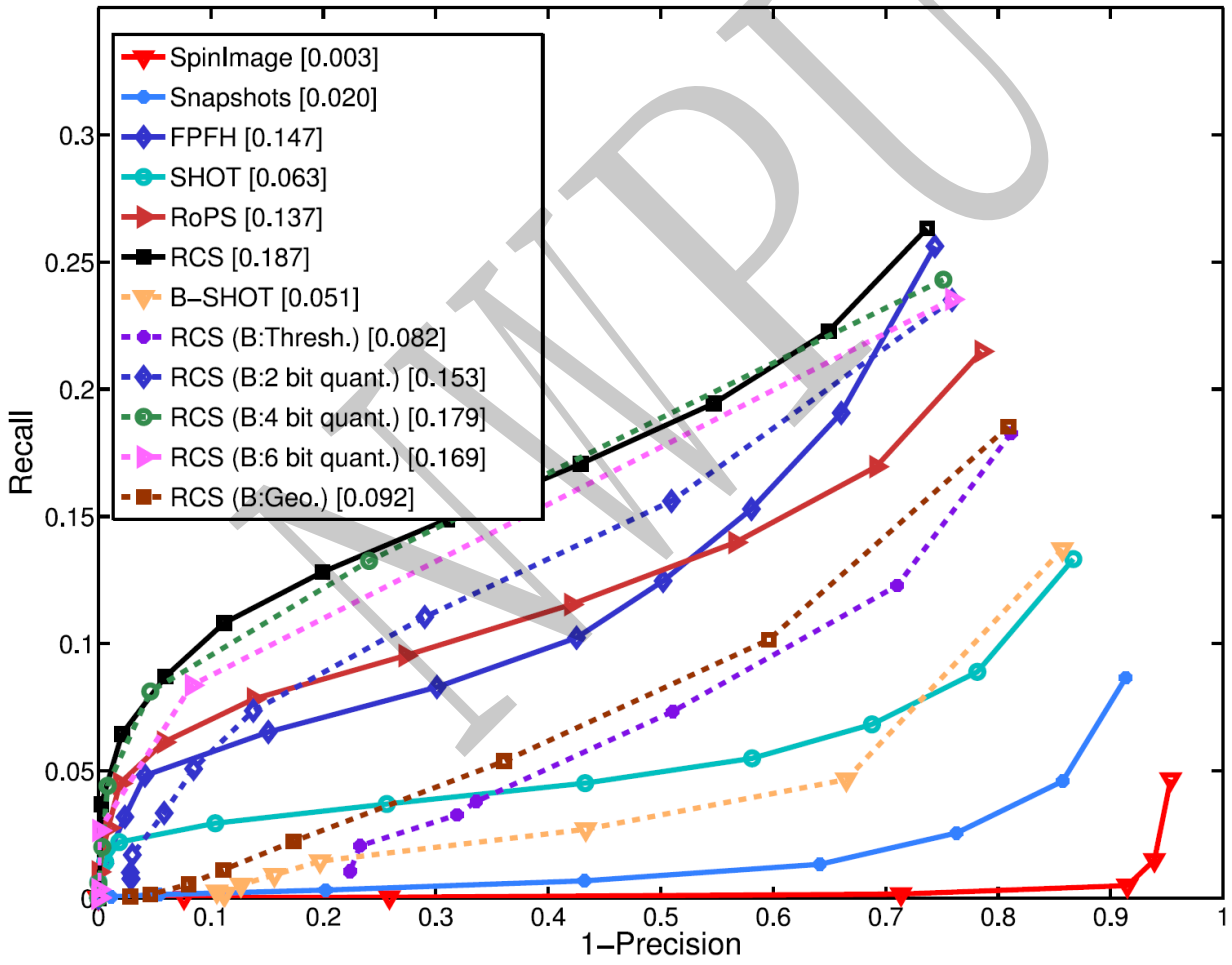
- Binary descriptor



(c) Geometric binary encoding

4. RCS

- Binary descriptor: efficient yet effective

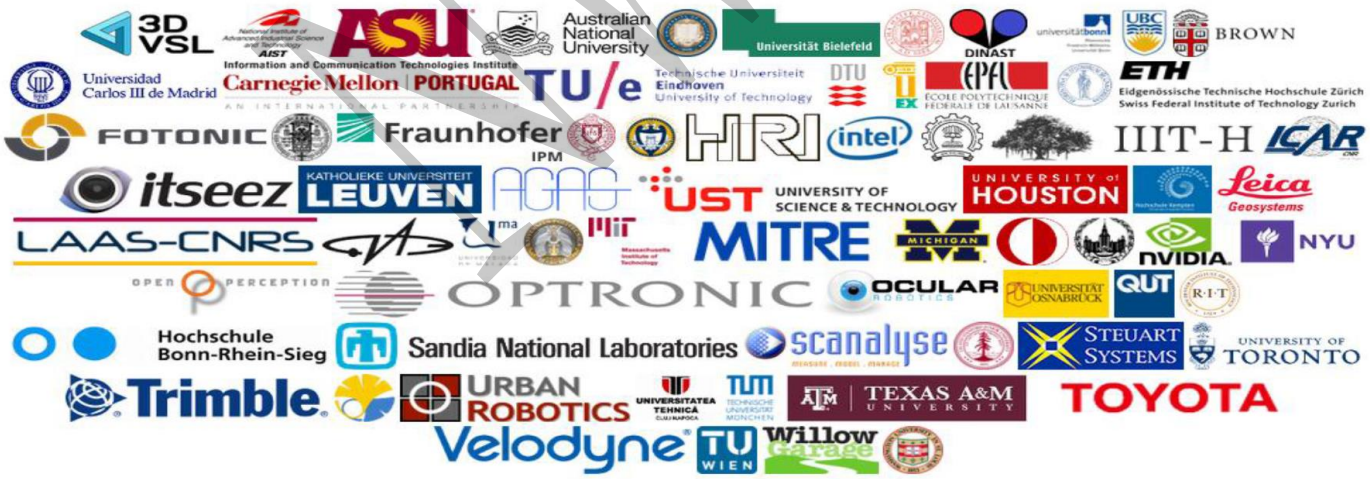


Practice: A C++ library---PCL

- Reference open source community for 3D computer vision and robotic perception
- Includes modules for
  - Keypoint extraction (pcl_keypoint)
  - Global/local descriptors (pcl_features)
  - Object Recognition in clutter (pcl_recognition)
  - Surface registration (pcl_registration)
  - Cloud segmentation (pcl_segmentation)
- And many more .. (machine learning, stereo, I/O, ...)

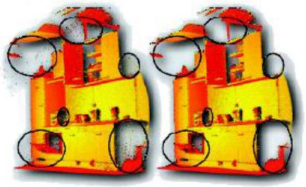
pcl

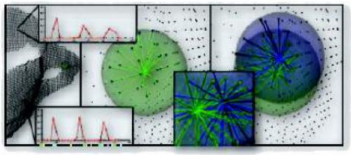
www.pointclouds.org

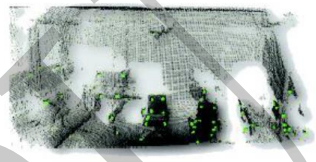


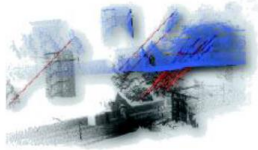
Practice: A C++ library---PCL

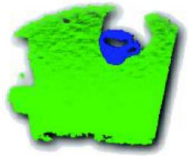
- 22 Code libraries + separate CUDA/GPU/Mobile modules


Features

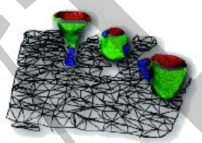

Filters

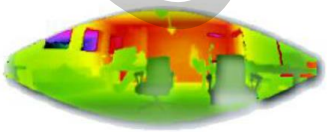

Keypoints


Registration


Segmentation


Sample Consensus


Surface

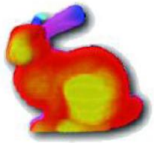

Range Image

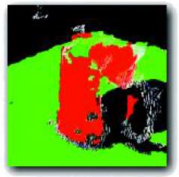

I/O

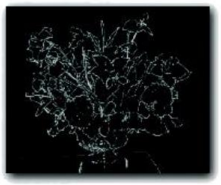

People


Simulation


Out-of-core

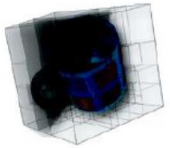

Visualization


Segmentation


2D


ML

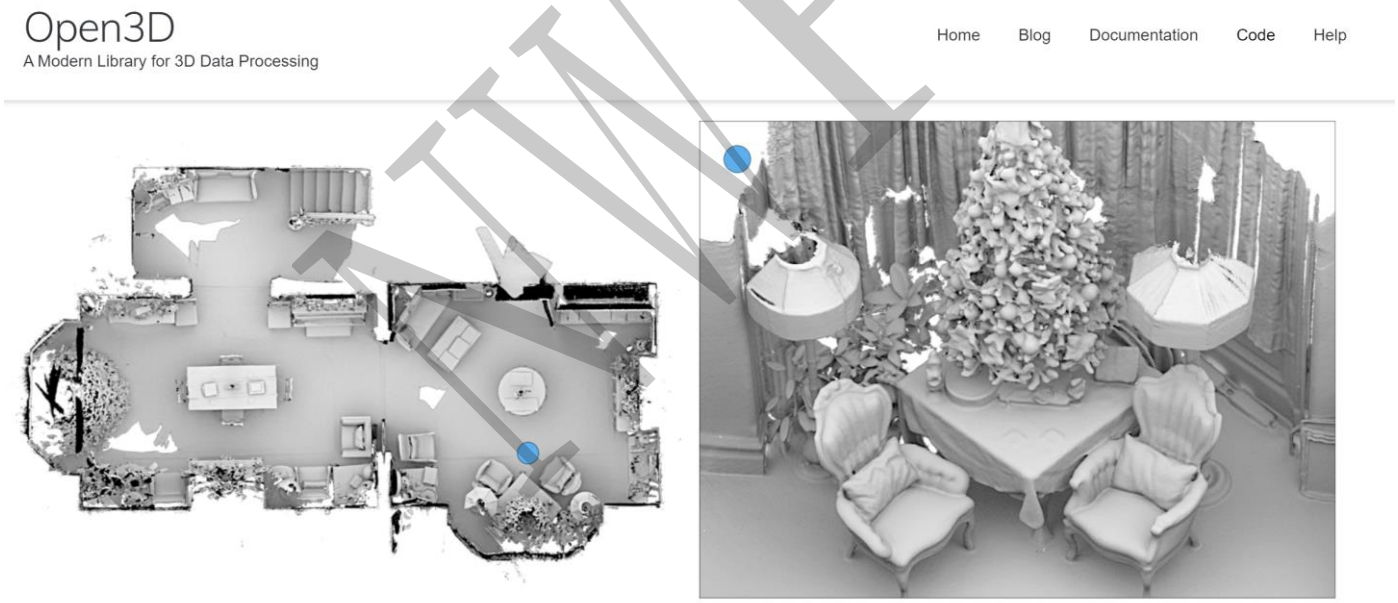

Recognition


Kdtree


Octree

Practice: A C++/Python library---Open3D

http://www.open3d.org/

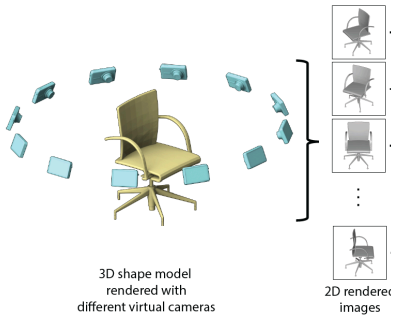


Course Reading Time

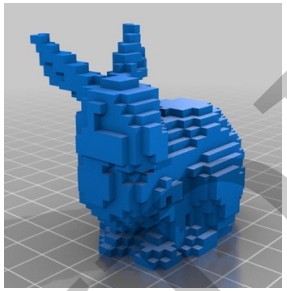
1. Johnson A E, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes[J]. IEEE Transactions on pattern analysis and machine intelligence, 1999, 21(5): 433-449.
2. Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//2009 IEEE international conference on robotics and automation. IEEE, 2009: 3212-3217.
3. Yang J, Zhang Q, Xiao Y, et al. TOLDI: An effective and robust approach for 3D local shape description[J]. Pattern Recognition, 2017, 65: 175-187.
4. Yang J, Xiao Y, Cao Z. Toward the repeatability and robustness of the local reference frame for 3D shape matching: An evaluation[J]. IEEE Transactions on Image Processing, 2018, 27(8): 3766-3781.

Lists of Contents

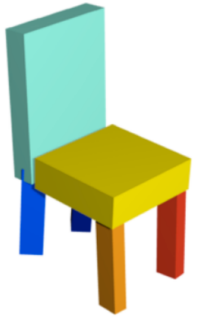
The Representation Challenge of 3D Deep Learning



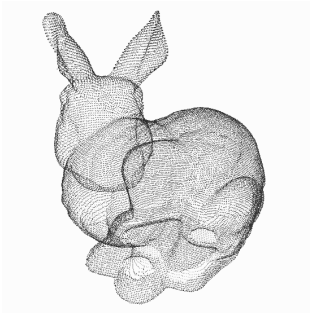
Multi-view



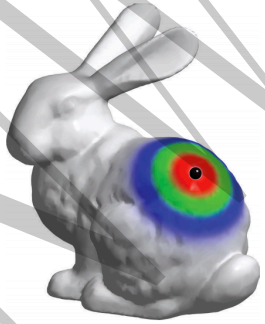
Volumetric



Part Assembly



Point Cloud



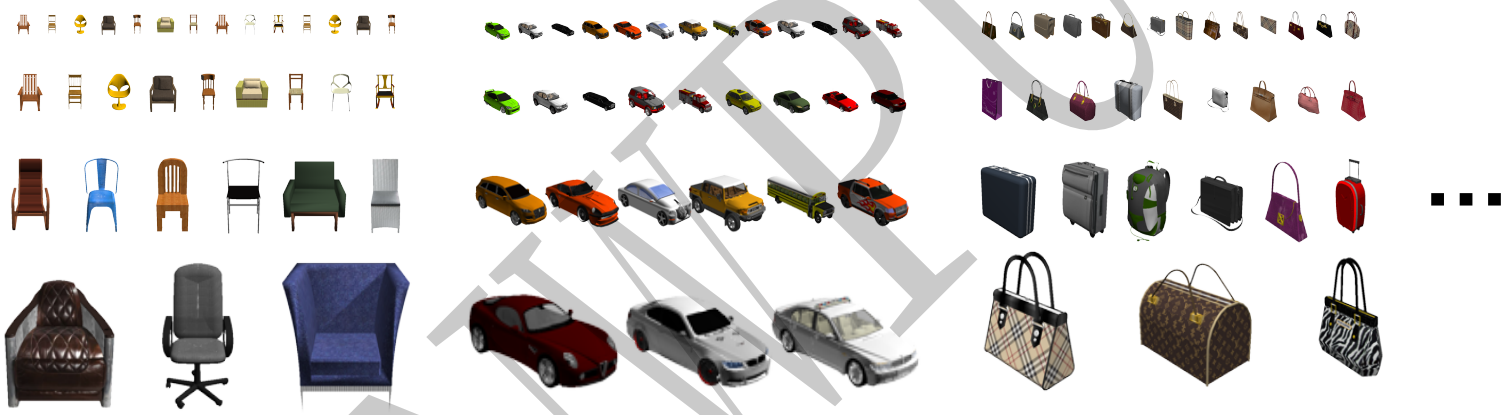
Mesh (Graph CNN)

$$F(x) = 0$$

Implicit Shape

Datasets for 3D Objects

Large-scale Synthetic Objects: ShapeNet



3DScan: Consumer-grade 3D scanning (click to open)

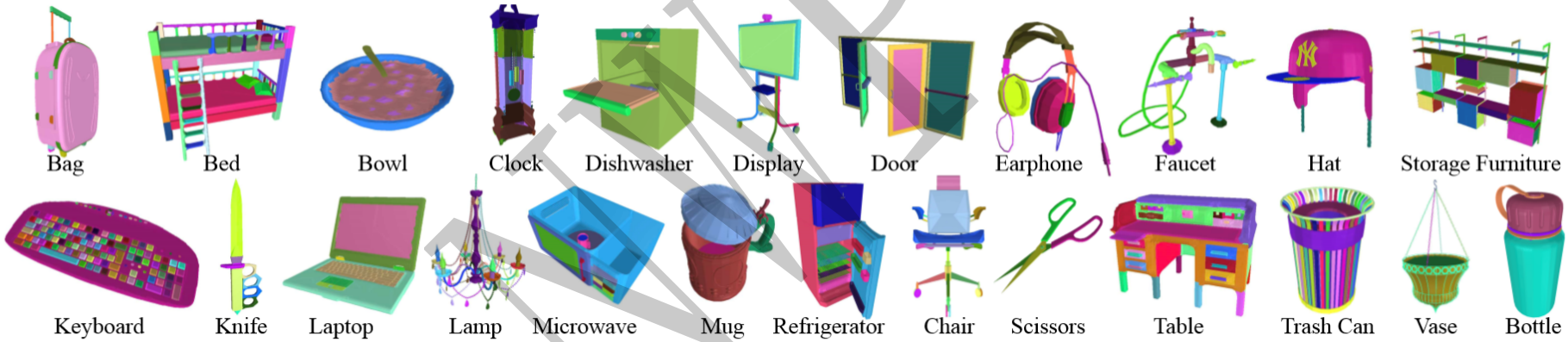
ModelNet: absorbed by ShapeNet

Chang et al., "**ShapeNet: An Information-Rich 3D Model Repository**" , *arXiv*
Wu et al., "**3D ShapeNets: A deep representation for volumetric shapes**", *CVPR 2015*
Choi et al., "**A Large Dataset of Object Scans**", *arXiv*

Datasets for 3D Object Parts

Fine-grained Part: PartNet (ShapeNetPart2019)

- Fine-grained (towards mobility)
- Instance-level
- Hierarchical



Mo et al., "**PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding** ", *CVPR 2019*

Datasets for Indoor 3D Scenes

Large-scale Synthetic Scenes: SceneNet

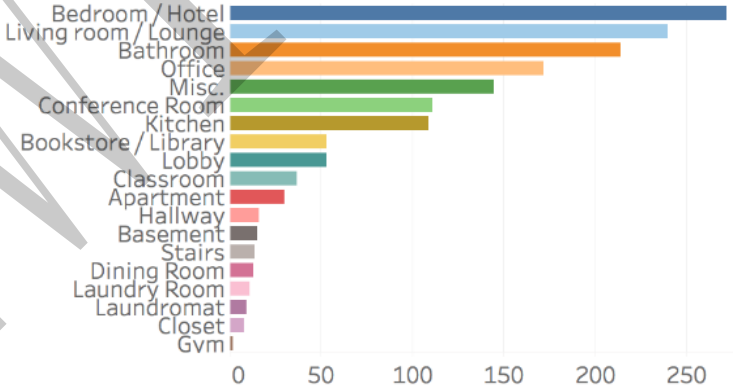
- 3D meshes
- 5M Photorealistic Images

Ankur et al., "**Understanding RealWorld Indoor Scenes with Synthetic Data**", *CVPR 2016*
McCormac et al., "**SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?**", *ICCV 2017*

Datasets for Indoor 3D Scenes

Large-scale Scanned Real Scenes: ScanNet

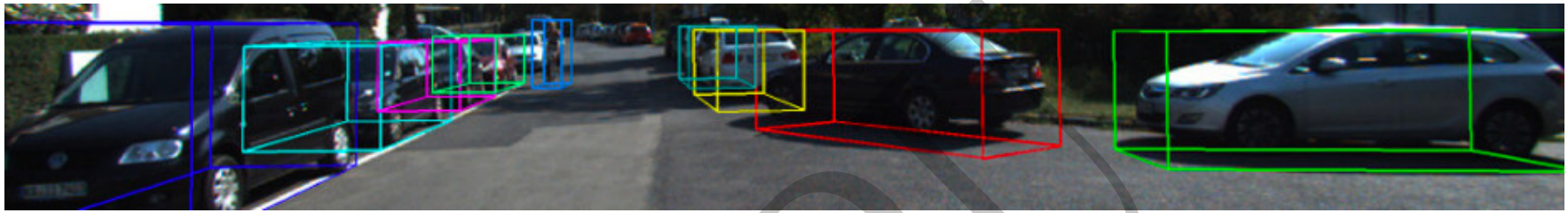
- 2.5 M Views in 1500 RGBD scans
- 3D camera poses
- surface reconstructions
- Instance-level semantic segmentations



Dai et al., "**ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes**", *CVPR 2017*

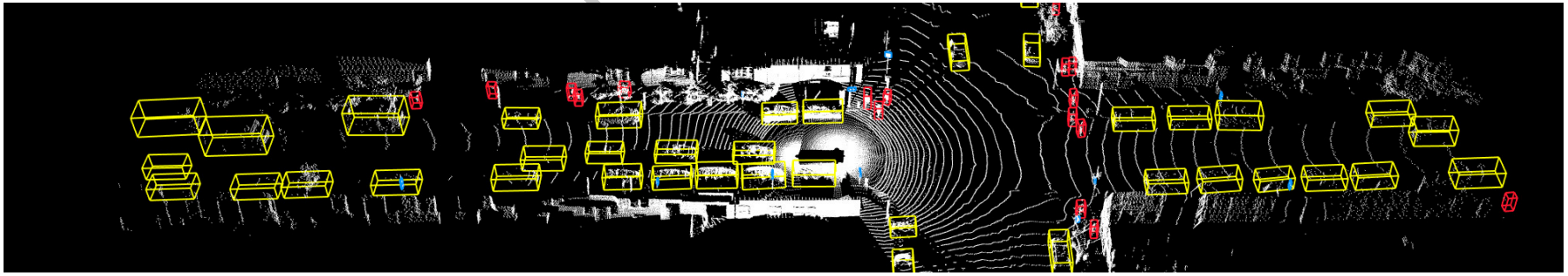
Datasets for Outdoor 3D Scenes

KITTI: LiDAR data, labeled by 3D b.boxes



Semantic KITTI: LiDAR data, labeled per point

Waymo Open Dataset: LiDAR data, labeled by 3D b.boxes



Topics

- 3D Data
- **Classification**
- Segmentation and Detection
- Reconstruction

Task: 3D Classification

This is a chair!

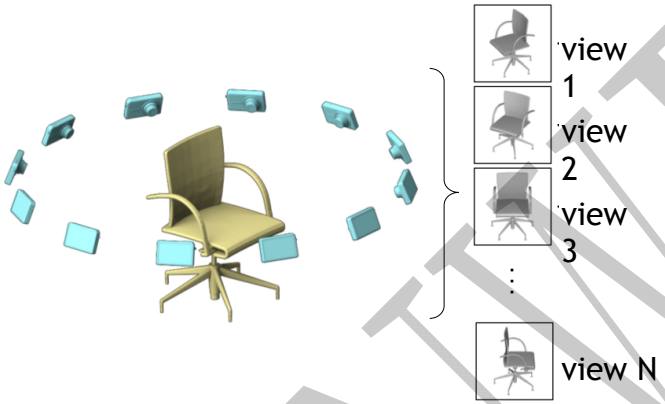
Covered methods: Volumetric CNN, OctNet, O-CNN, SparseConvNet, PointNet, PointNet++, RS CNN, DGCNN, Point ConvNet, KPConv, Monte Carlo Point Convolution, PConv, Multi-View CNN, Spectral CNN, Synchronized Spectral CNN, Spherical CNN

Multi-View CNN

Given an Input Shape

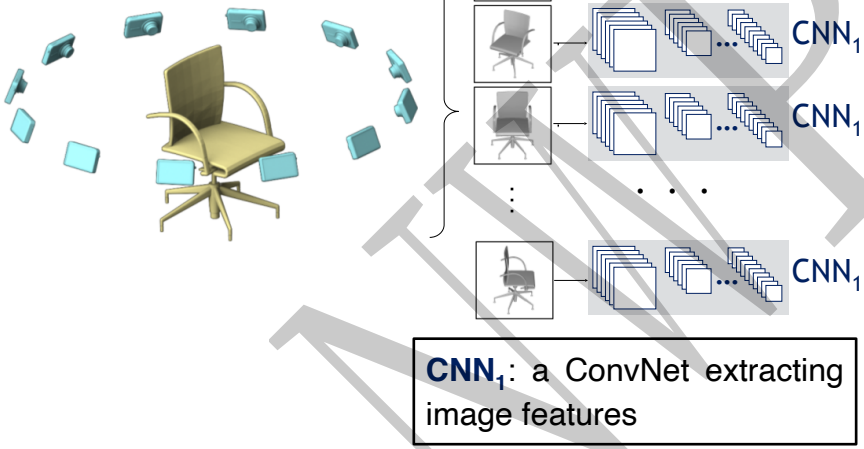
Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

Render with Multiple Virtual Cameras



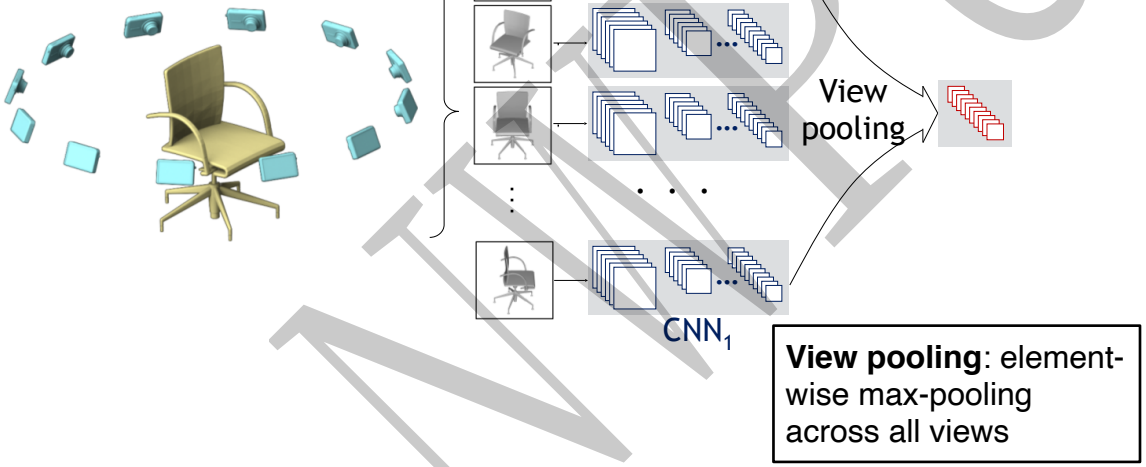
Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

The Rendered Images are Passed through CNN_1 for Image Features



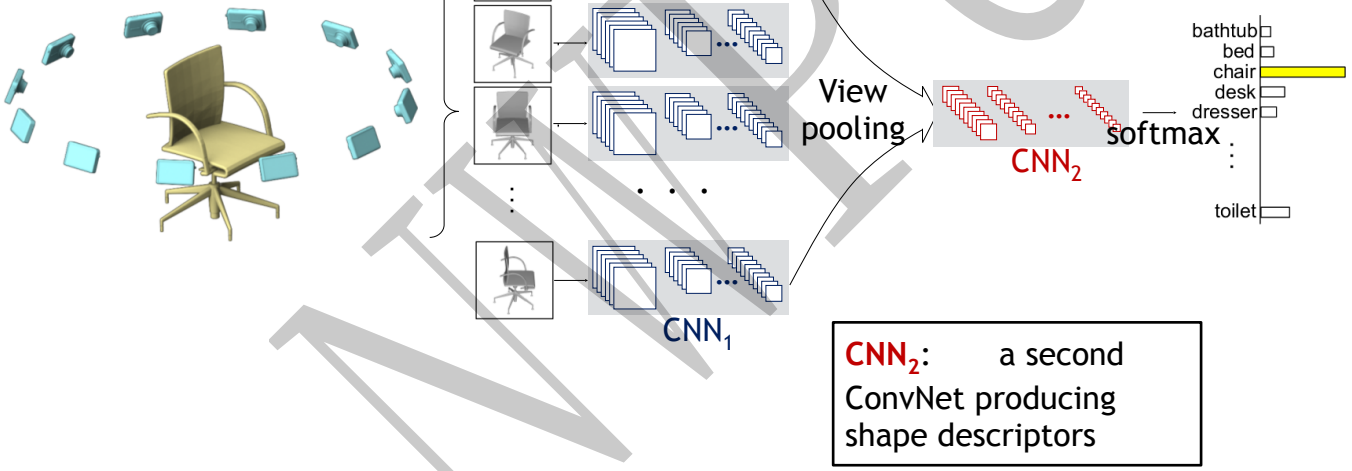
Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

All Image Features are Combined by View Pooling



Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

… and then Passed through CNN_2 and to Generate Final Predictions



Su et al., "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *ICCV 2015*

Experiments – Classification & Retrieval

	Method	Classification (Accuracy)	Retrieval (mAP)
Non-deep	SPH [16]	68.2%	33.3%
	LFD [5]	75.5%	40.9%
	3D ShapeNets [37]	77.3%	49.2%
	FV, 12 views	84.8%	43.9%
	CNN, 12 views	88.6%	62.8%
	MVCNN, 12 views	**89.9%**	70.1%
	MVCNN+metric, 12 views	89.5%	**80.2%**
	MVCNN, 80 views	90.1%	70.4%
	MVCNN+metric, 80 views	**90.1%**	**79.5%**

On ModelNet40

[credit: Hang Su]

- Indeed gives good performance
- Can leverage vast literature of image classification
- Can use pertained features
- Need projection
- What if the input is noisy and/or incomplete? e.g., point cloud

Volumetric CNN

Can we use CNNs without 2D-3D projection?

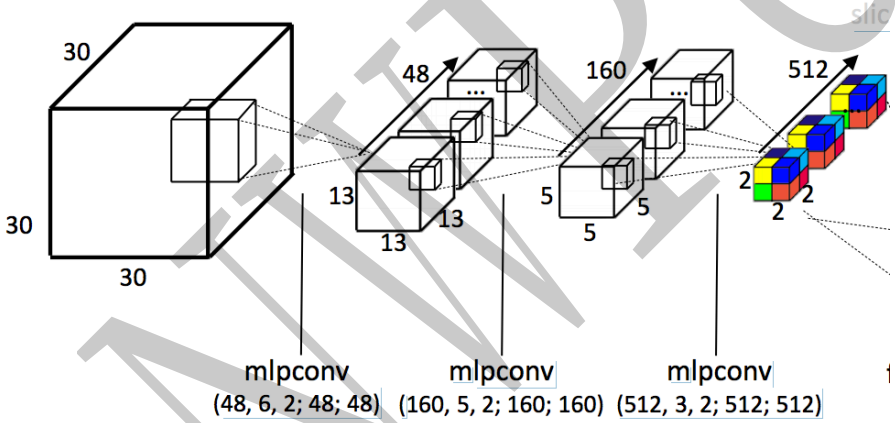
Straight-forward idea: 3D native convolution

Voxelization

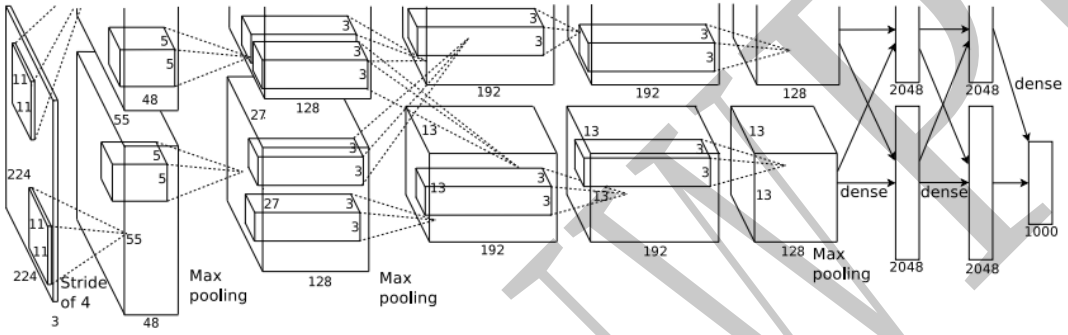
Represent the occupancy of regular 3D grids

3D CNN on Volumetric Data

3D convolution uses 4D kernels



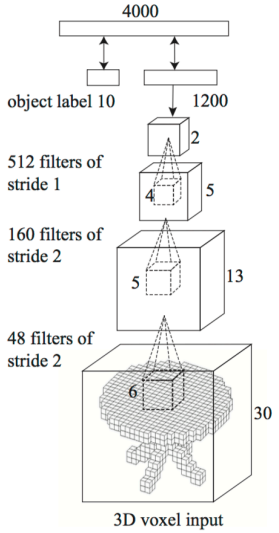
Complexity Issue



AlexNet, 2012

Input resolution: 224x224

224x224=50176

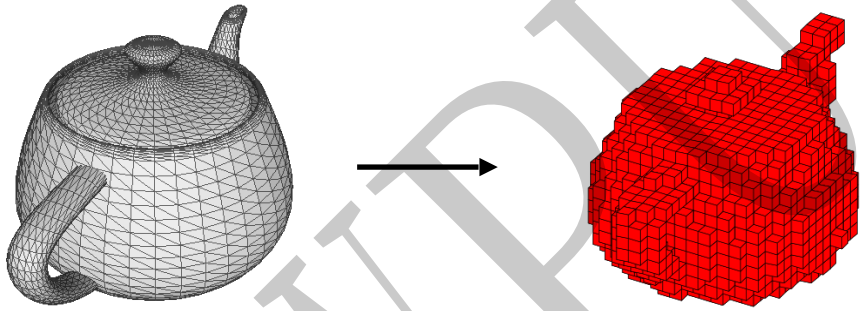


3DShapeNets, 2015

Input resolution: 30x30x30

224x224=27000

Complexity Issue



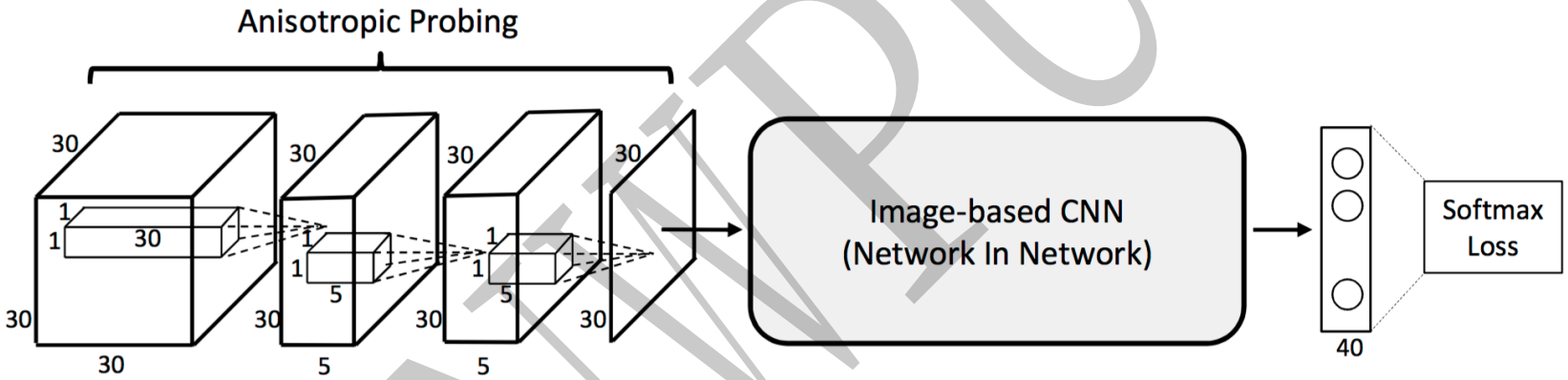
Polygon Mesh

Occupancy Grid
30×30×30

Information loss in voxelization

Idea 1: Learn to Project

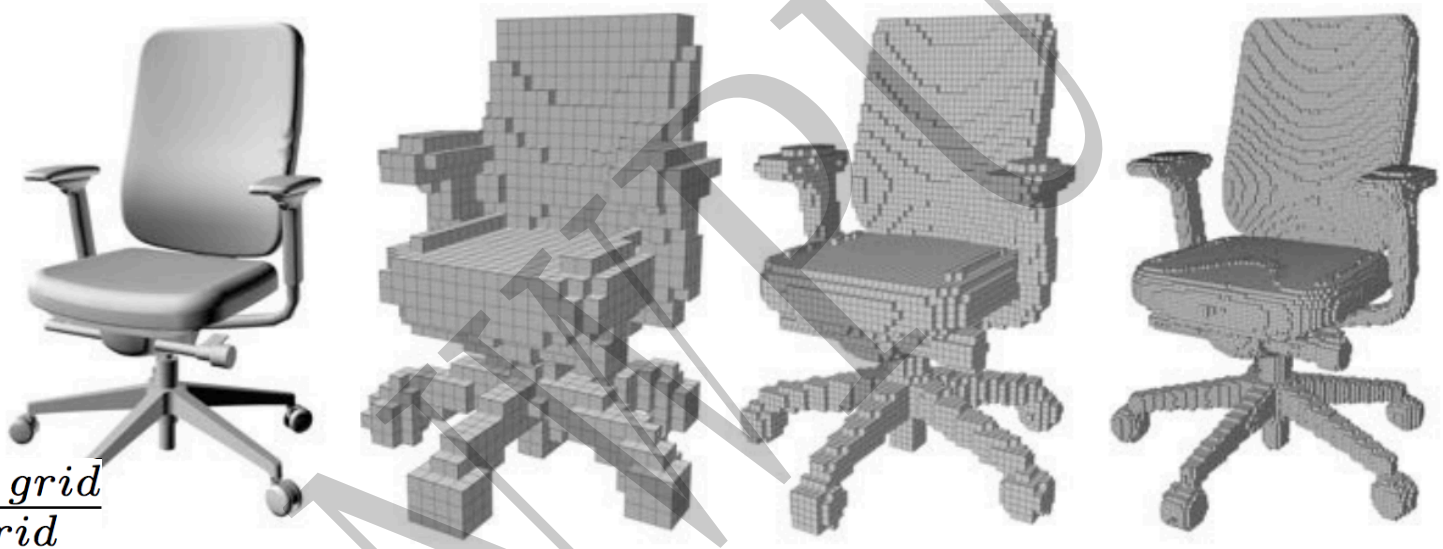
Idea: "X-ray" rendering + Image (2D) CNNs
very low #param, very low computation



Su et al., "**Volumetric and Multi-View CNNs for Object Classification on 3D Data**", *CVPR 2016*

Many other works in autonomous driving that uses bird's eye view for object detection

More Principled: Leverage Sparsity of 3D Surface Data

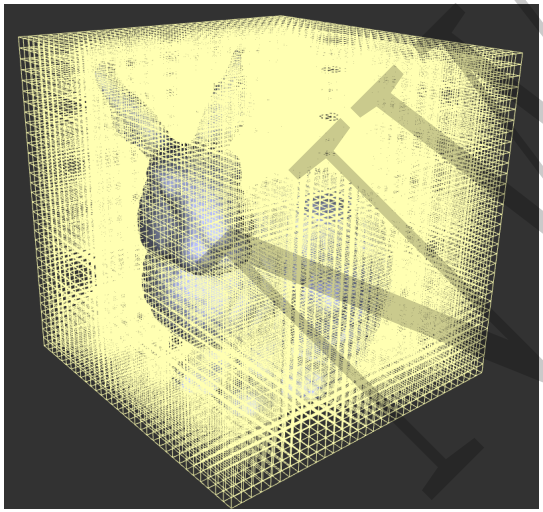


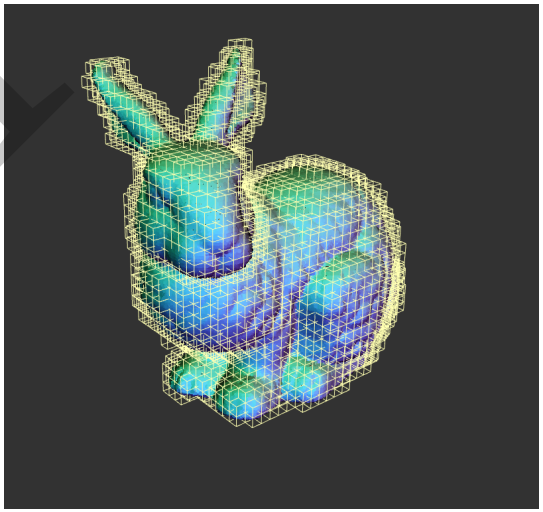
$$\frac{\#occupied\ grid}{\#total\ grid}$$

Occupancy:	10.41%	5.09%	2.41%
Resolution:	32	64	128

Store only the Occupied Grids

- Store the sparse surface signals
- Constrain the computation near the surface

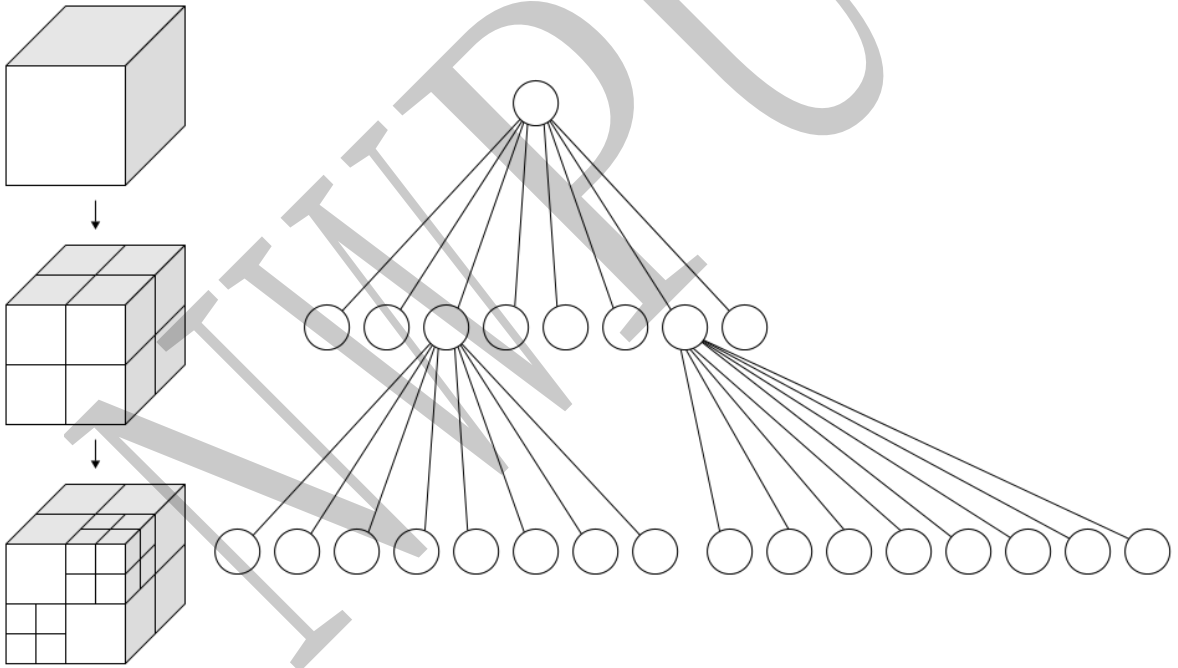




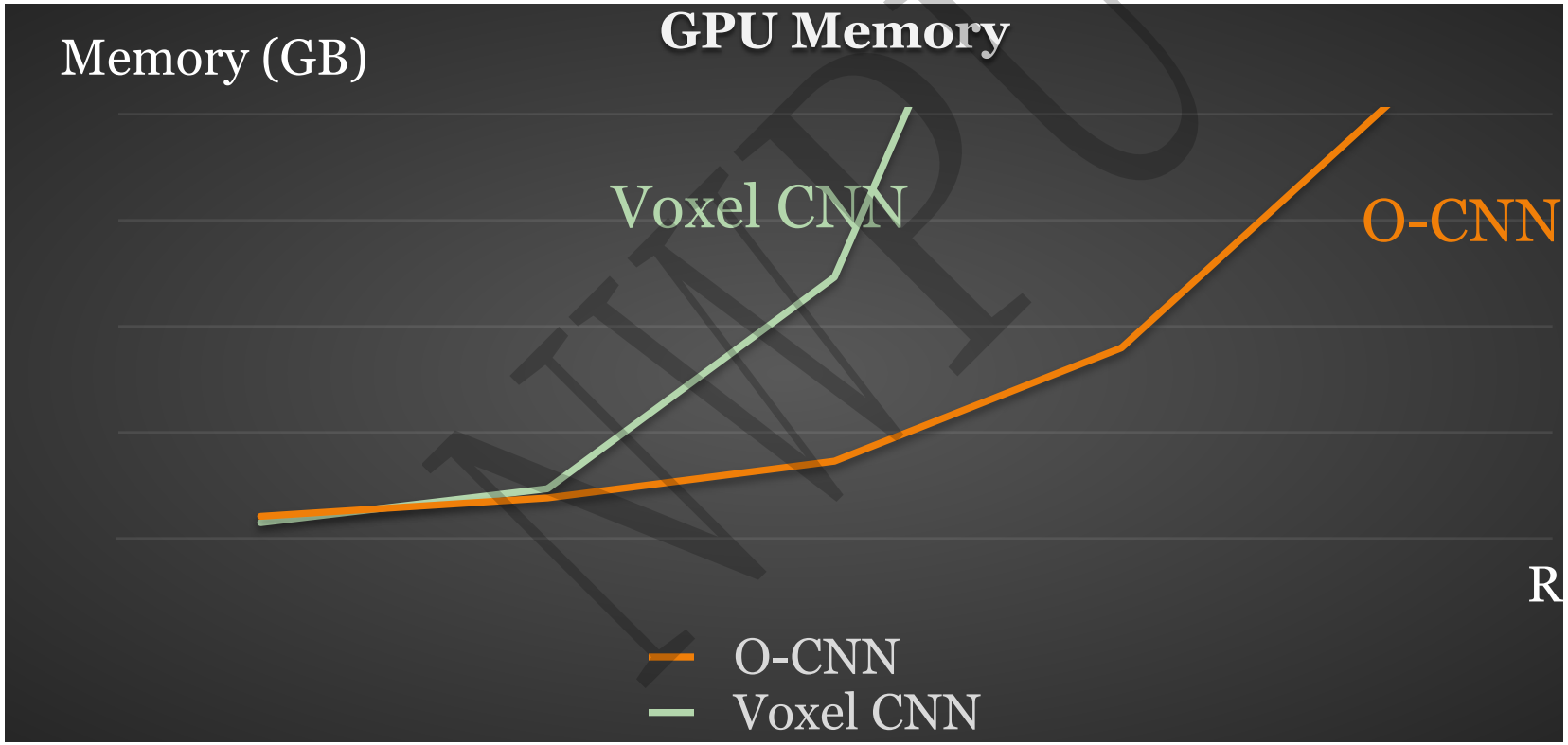
Octree: Recursively Partition the Space

Each internal node has exactly eight children

Neighborhood searching: Hash table



Memory Efficiency

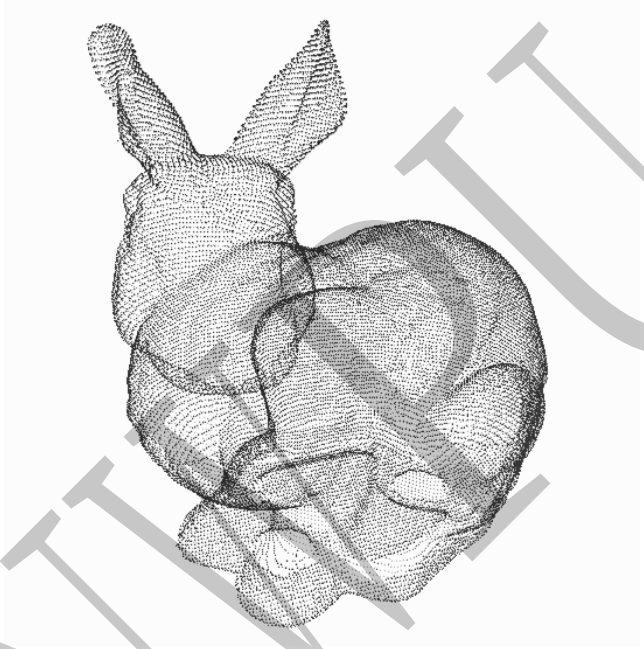


Implementation

- SparseConvNet
  - https://github.com/facebookresearch/SparseConvNet
  - Uses ResNet architecture
  - State-of-the-art for 3D analysis
  - Takes time to train

Graham et al., "**Submanifold Sparse Convolutional Networks**", *arxiv*

Point Networks



Point cloud

(The most common 3D sensor data)

List of Content

1. Supervised general 3D learned features

- ***PointNet***: Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.
- ***PointNet++:*** Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. arXiv preprint arXiv:1706.02413, 2017.

2. Unsupervised general 3D learned features

- ***FoldingNet:*** Yang Y, Feng C, Shen Y, et al. Foldingnet: Point cloud auto-encoder via deep grid deformation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 206-215.

List of Content

3. 3D learned features for 3D registration/reconstruction

- *3DMatch*: Zeng A, Song S, Nießner M, et al. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1802-1811.
- *PPFNet*: Deng H, Birdal T, Ilic S. Ppfnet: Global context aware local features for robust 3d point matching[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 195-205.
- *CGF*: Khoury M, Zhou Q Y, Koltun V. Learning compact geometric features[C]//Proceedings of the IEEE international conference on computer vision. 2017: 153-161.
- *PPF-FoldNet*: Deng H, Birdal T, Ilic S. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 602-618.

Learn and Miss

Learn

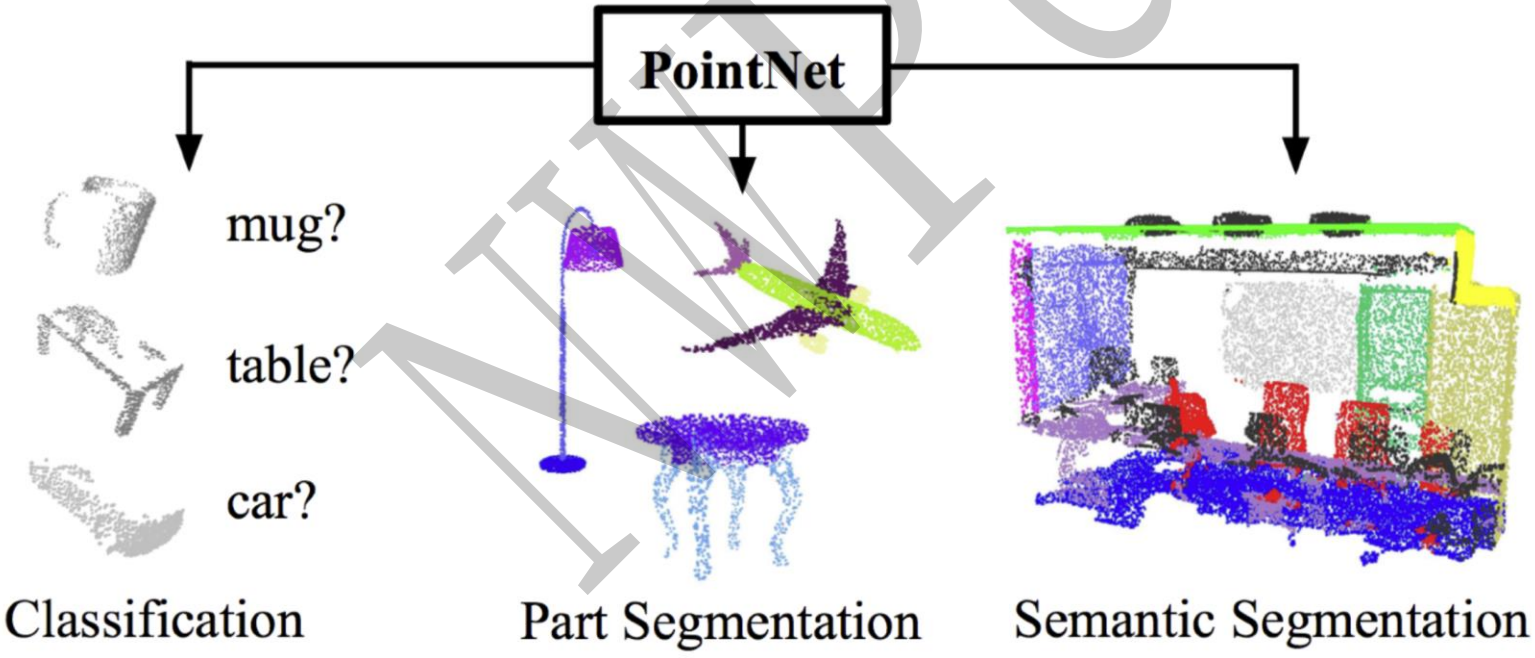
- 3 typical **general** learned point cloud features, i.e., PointNet, PointNet++, and FoldingNet
- 4 learned local features for 3D registration/reconstruction

Miss

- Learned features for other 3D vision tasks will be introduced by your guys.
- 3D Spectral learning for non-rigid 3D data

PointNet

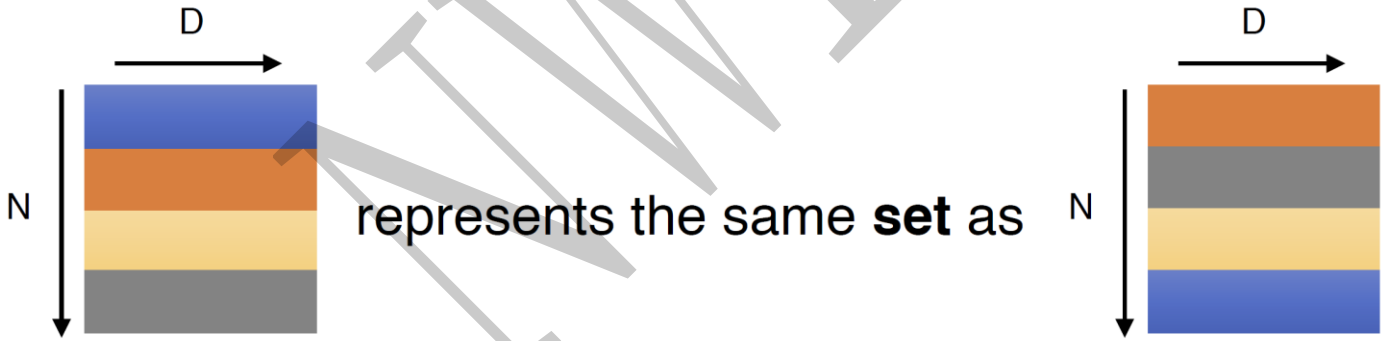
- End-to-end learning for scattered, unordered point data
- Unified framework for various tasks



PointNet

- The network should be invariant to N! permutations

Point cloud: N **orderless** points, each represented by a D dim vector



PointNet

- The network should be invariant to N! permutations

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

…

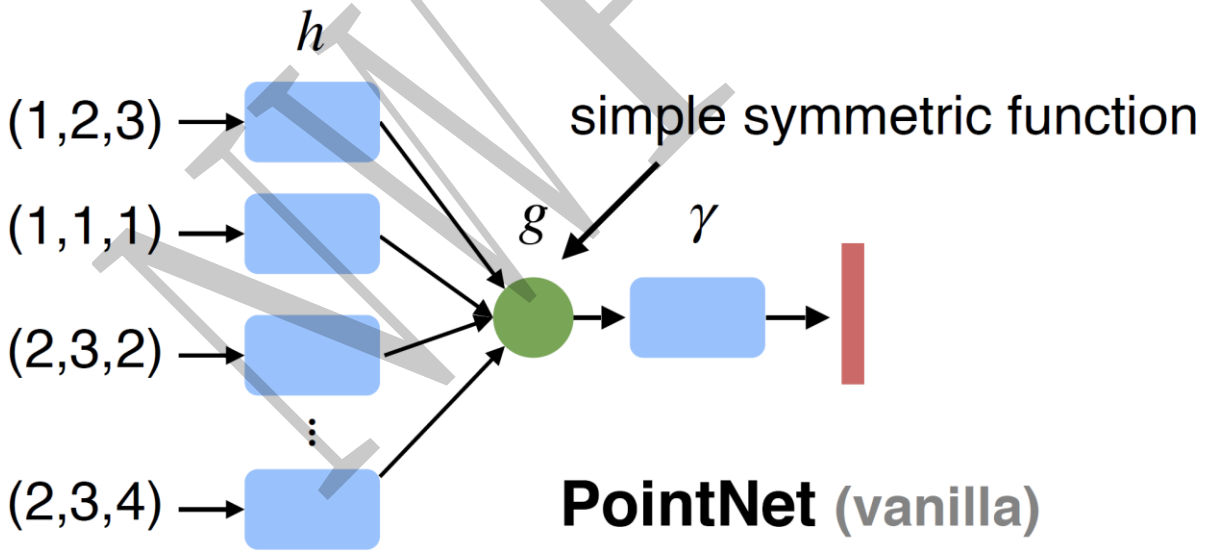
How can we construct a family of symmetric functions by neural networks?

PointNet

- The network should be invariant to N! permutations

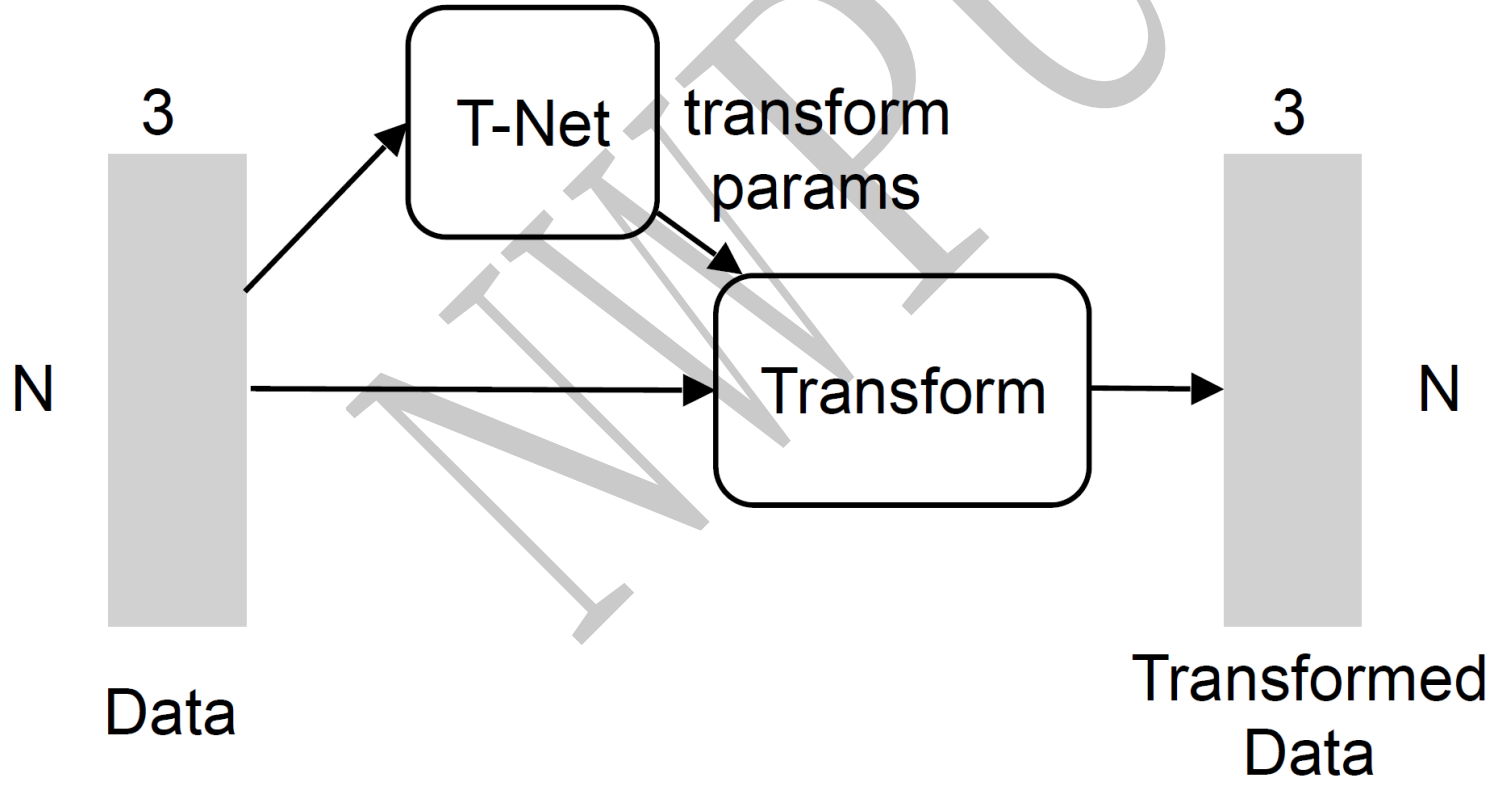
Observe:

$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet

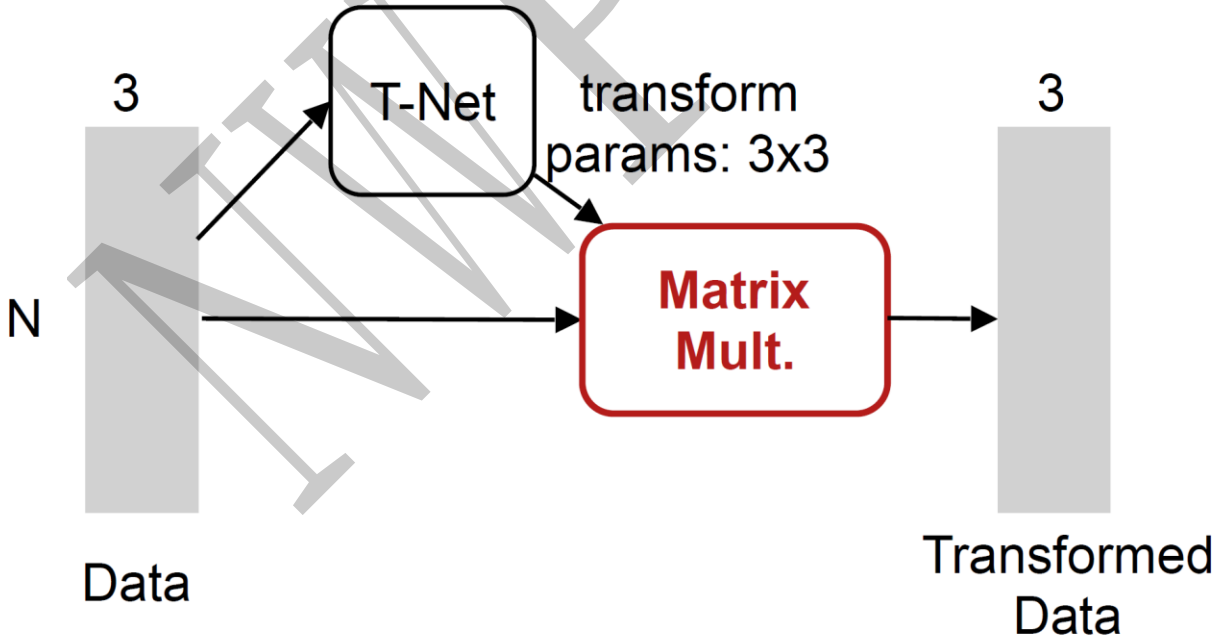
- The network should be invariant to geometric transformations



PointNet

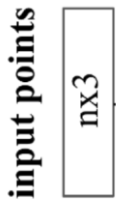
- The network should be invariant to geometric transformations

The transformation is just matrix multiplication!



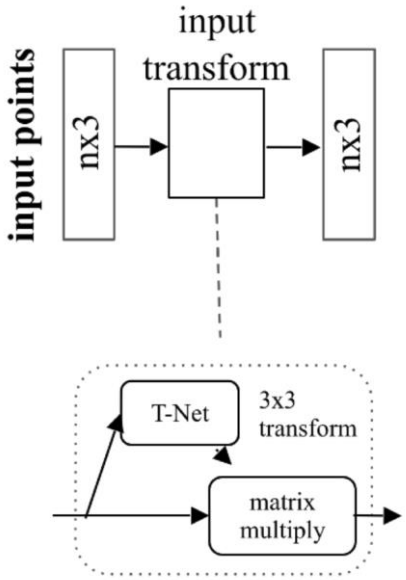
PointNet

- Overall architecture--classification



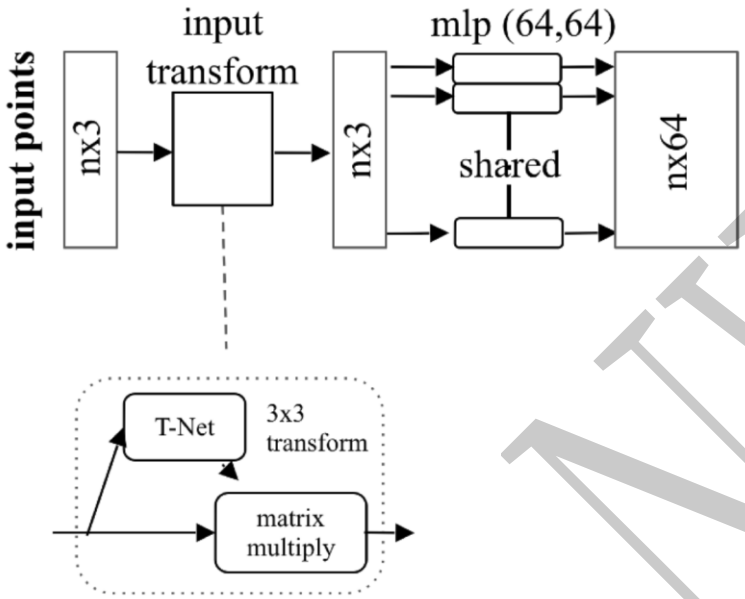
PointNet

- Overall architecture--classification



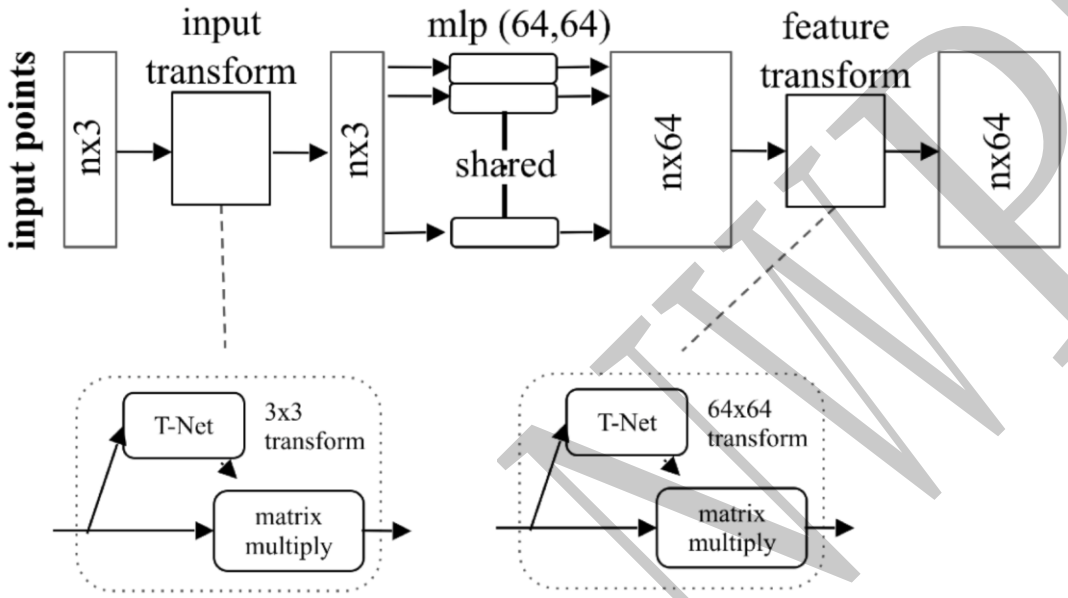
PointNet

- Overall architecture--classification



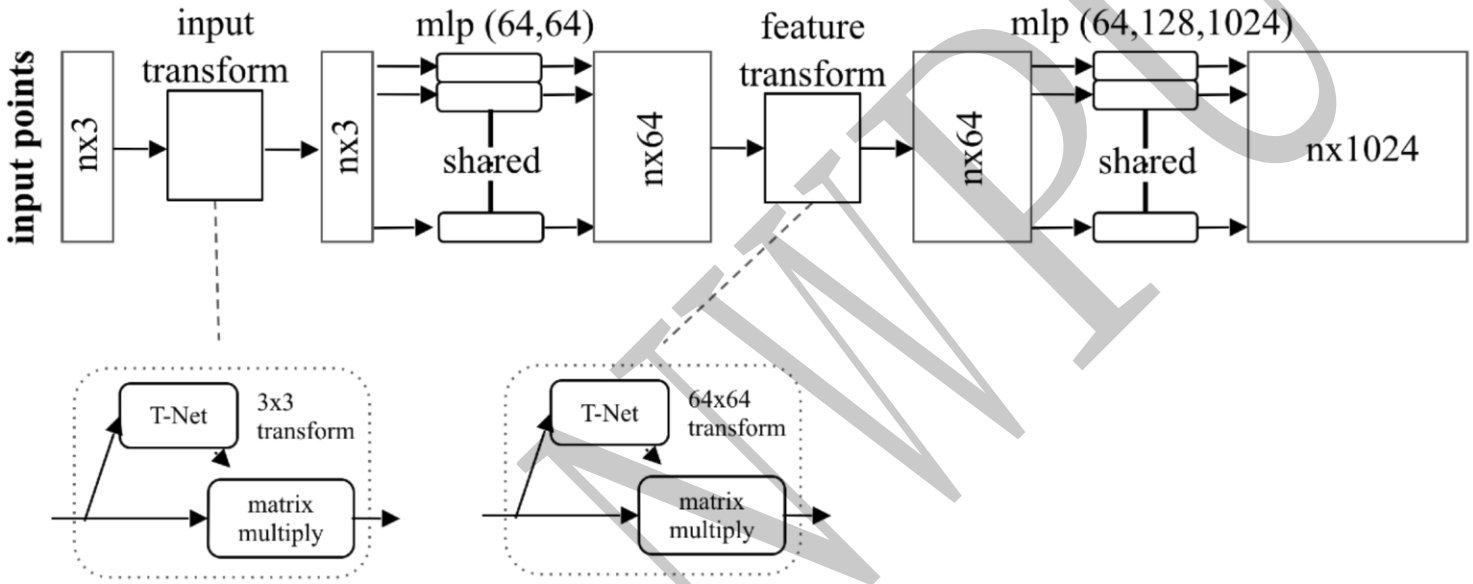
PointNet

- Overall architecture--classification



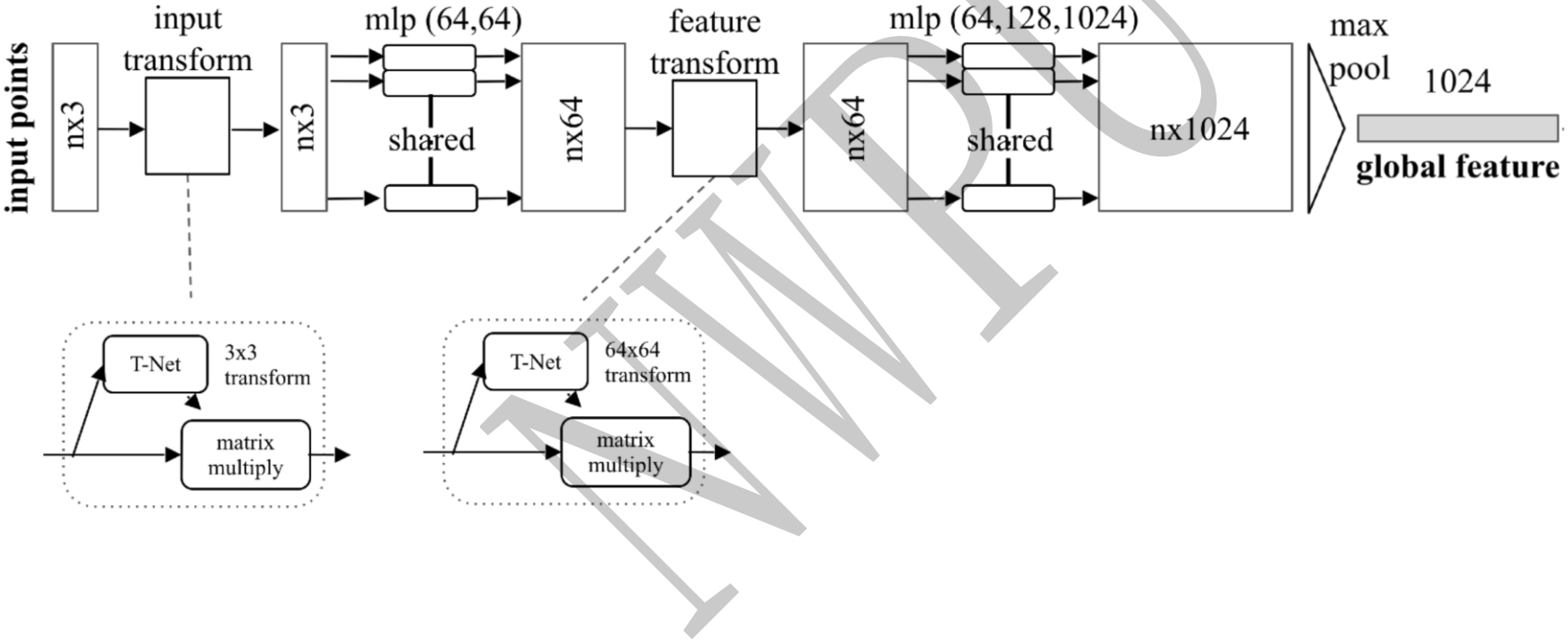
PointNet

- Overall architecture--classification



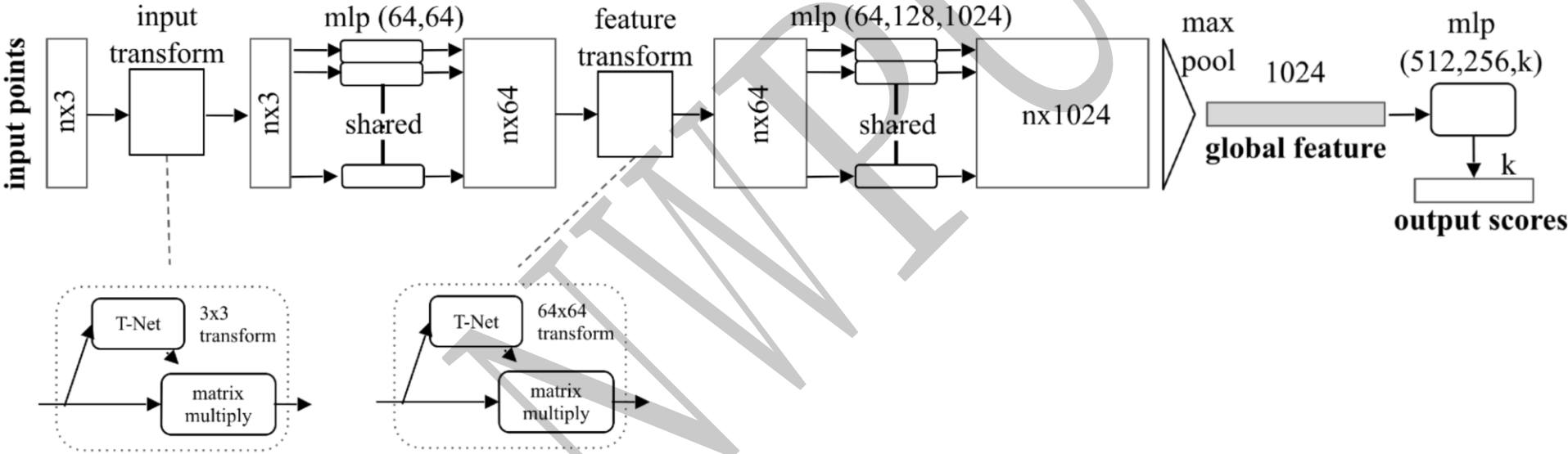
PointNet

- Overall architecture--classification



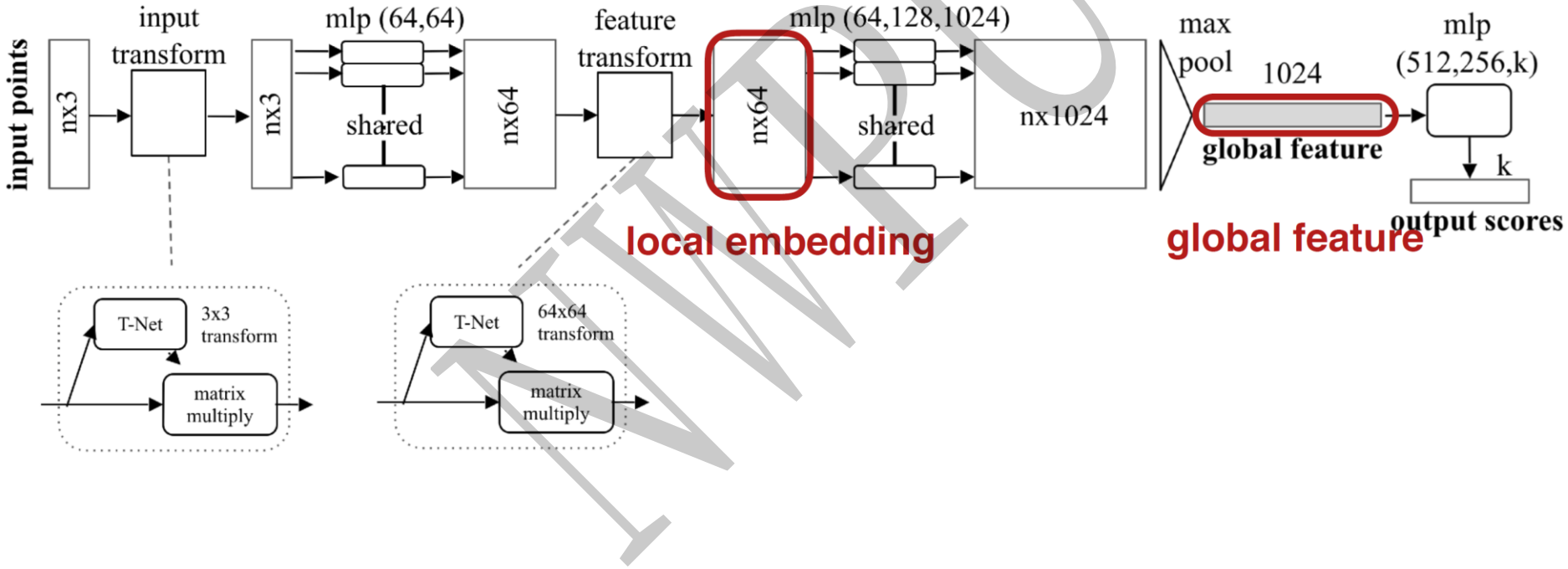
PointNet

- Overall architecture--classification



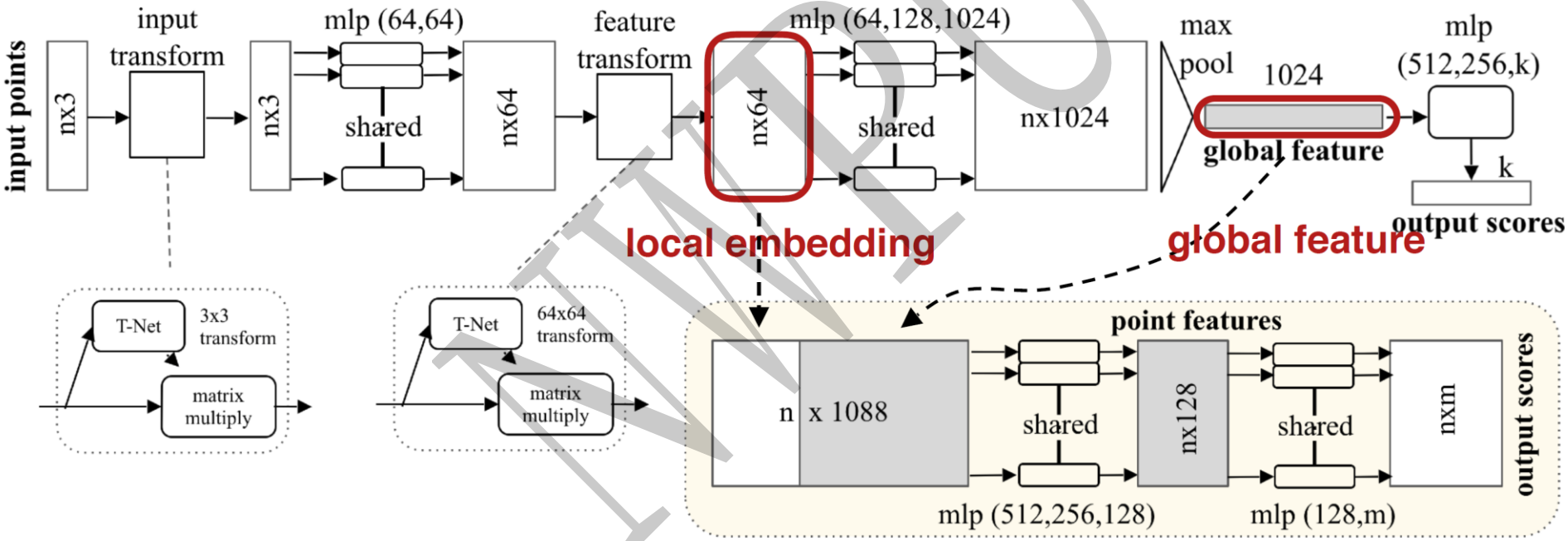
PointNet

- Overall architecture--classification



PointNet

- Overall architecture--segmentation



PointNet++

- Try to capture fine-grained local structure of point clouds

Abstract

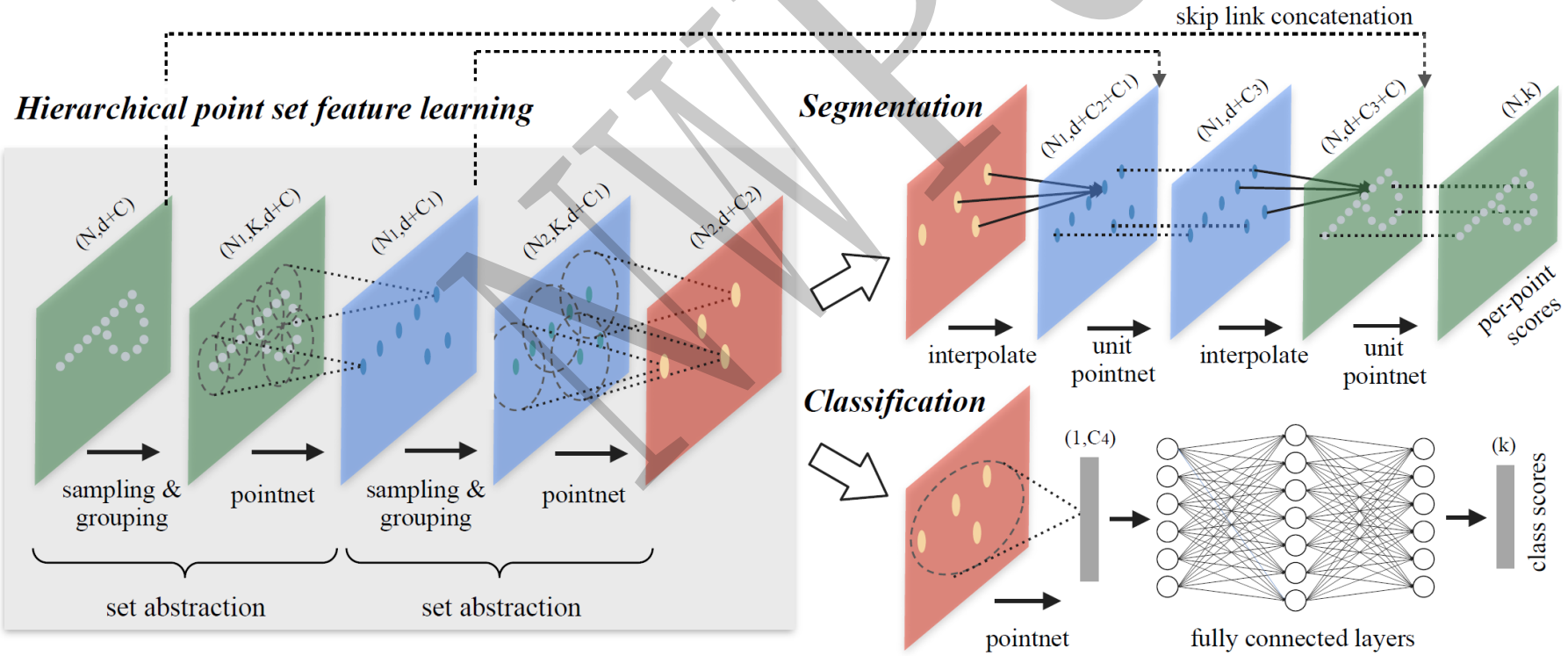
Few prior works study deep learning on point sets. PointNet [20] is a pioneer in this direction. However, by design PointNet does not capture local structures induced by the metric space points live in, limiting its ability to recognize fine-grained patterns and generalizability to complex scenes. In this work, we introduce a hierarchical

PointNet++

- The hierarchical structure is composed by a number of set abstraction levels
- At each level, a set of points is processed and abstracted to produce a new set with fewer elements
- The set abstraction level is made of 3 key layers: ***Sampling layer, Grouping layer and PointNet layer***.
  - **The Sampling layer** selects a set of points from input points, which defines the centroids of local regions.
  - ***Grouping layer*** then constructs local region sets by finding "neighboring" points around the centroids.
  - PointNet layer uses a mini-PointNet to encode local region patterns into feature vectors

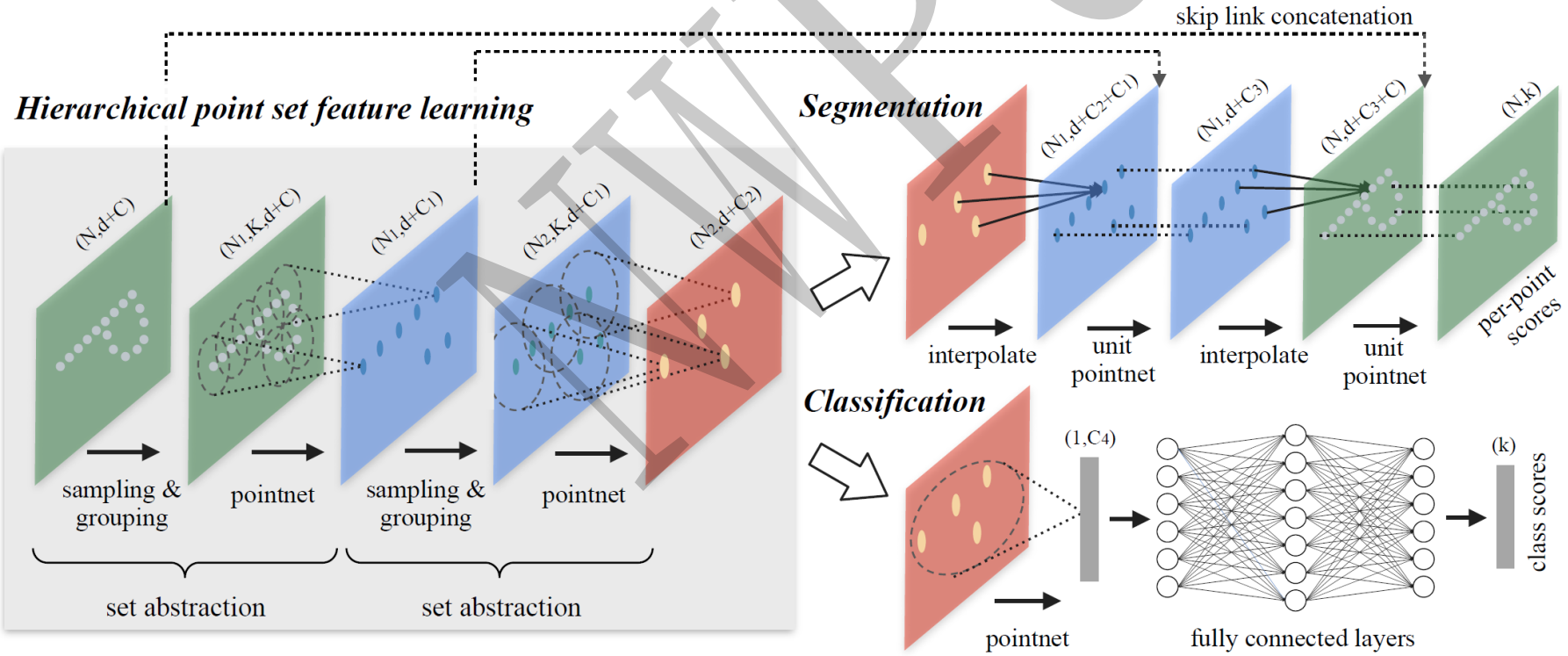
PointNet++

Sampling layer. Given input points $\{x_1, x_2, ..., x_n\}$, we use iterative farthest point sampling (FPS) to choose a subset of points $\{x_{i_1}, x_{i_2}, ..., x_{i_m}\}$, such that x_{i_j} is the most distant point (in metric distance) from the set $\{x_{i_1}, x_{i_2}, ..., x_{i_{j-1}}\}$ with regard to the rest points. Compared with random sampling, it has better coverage of the entire point set given the same number of centroids. In contrast to CNNs that scan the vector space agnostic of data distribution, our sampling strategy generates receptive fields in a data dependent manner.



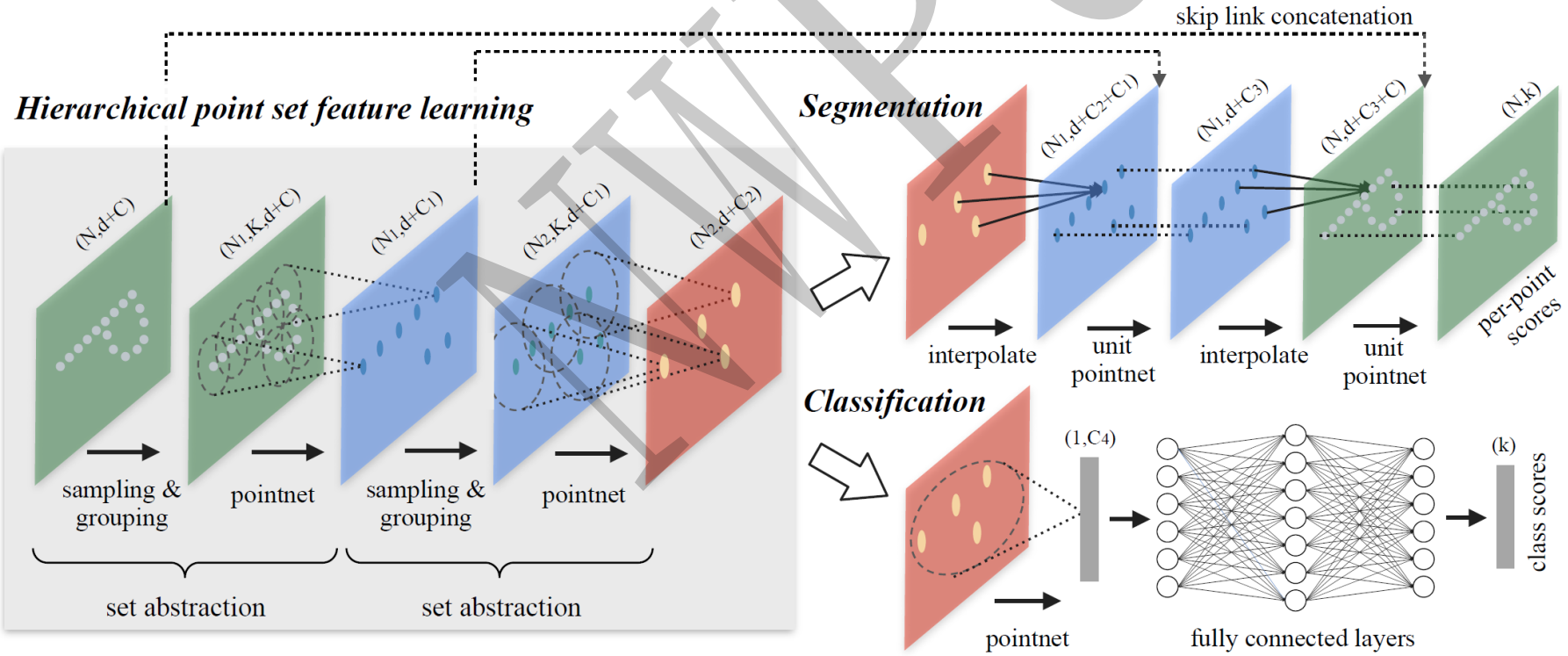
PointNet++

Grouping layer. The input to this layer is a point set of size $N \times (d + C)$ and the coordinates of a set of centroids of size $N' \times d$. The output are groups of point sets of size $N' \times K \times (d + C)$, where each group corresponds to a local region and K is the number of points in the neighborhood of centroid points. Note that K varies across groups but the succeeding *PointNet layer* is able to convert flexible number of points into a fixed length local region feature vector.



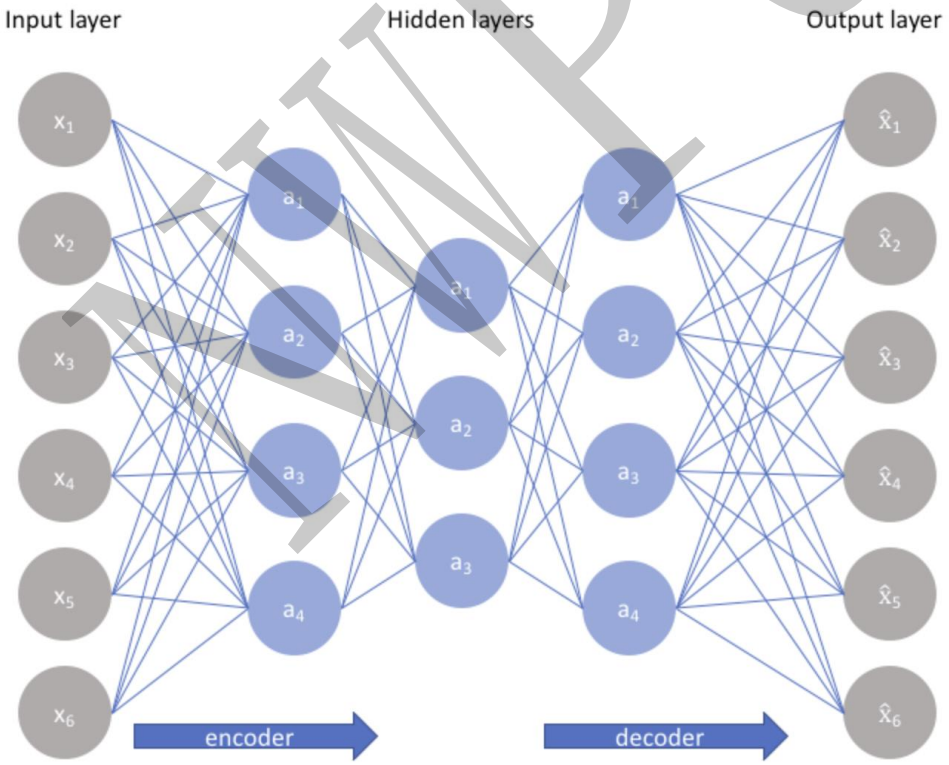
PointNet++

PointNet layer. In this layer, the input are N' local regions of points with data size $N' \times K \times (d+C)$. Each local region in the output is abstracted by its centroid and local feature that encodes the centroid's neighborhood. Output data size is $N' \times (d + C')$.



FoldingNet

- A 3D Auto-encoder
- What is Auto-encoder? A powerful unsupervised feature learning mechanism



FoldingNet

- A novel end-to-end deep auto-encoder is proposed to address unsupervised learning challenges on point clouds

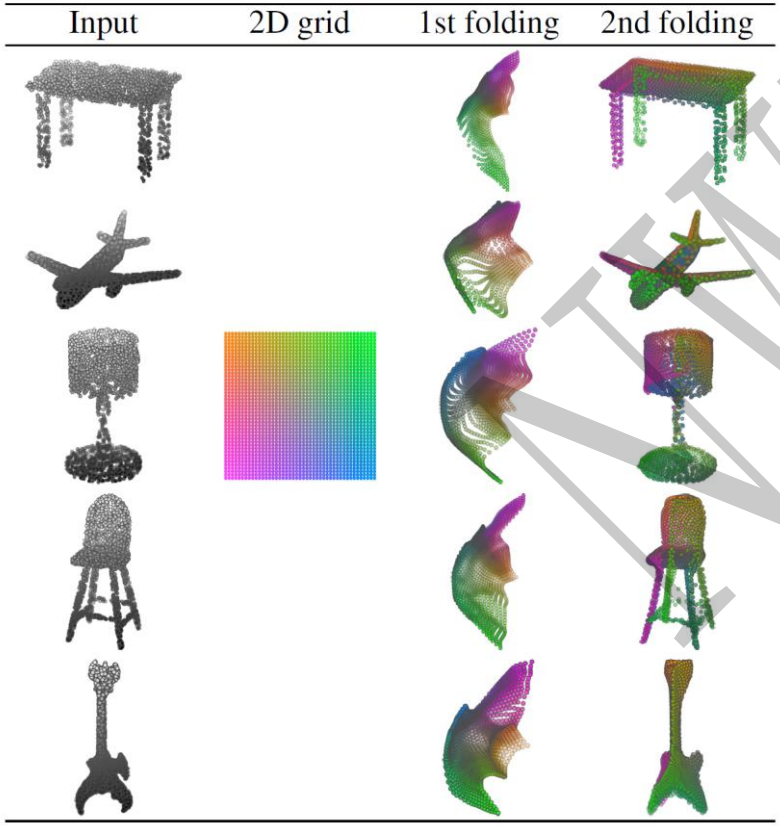


Table 1. **Illustration of the two-step-folding decoding**. Column one contains the original point cloud samples from the ShapeNet dataset [57]. Column two illustrates the 2D grid points to be folded during decoding. Column three contains the output after one folding operation. Column four contains the output after two folding operations. This output is also the reconstructed point cloud. We use a color gradient to illustrate the correspondence between the 2D grid in column two and the reconstructed point clouds after folding operations in the last two columns. Best viewed in color.

FoldingNet

- Overall architecture

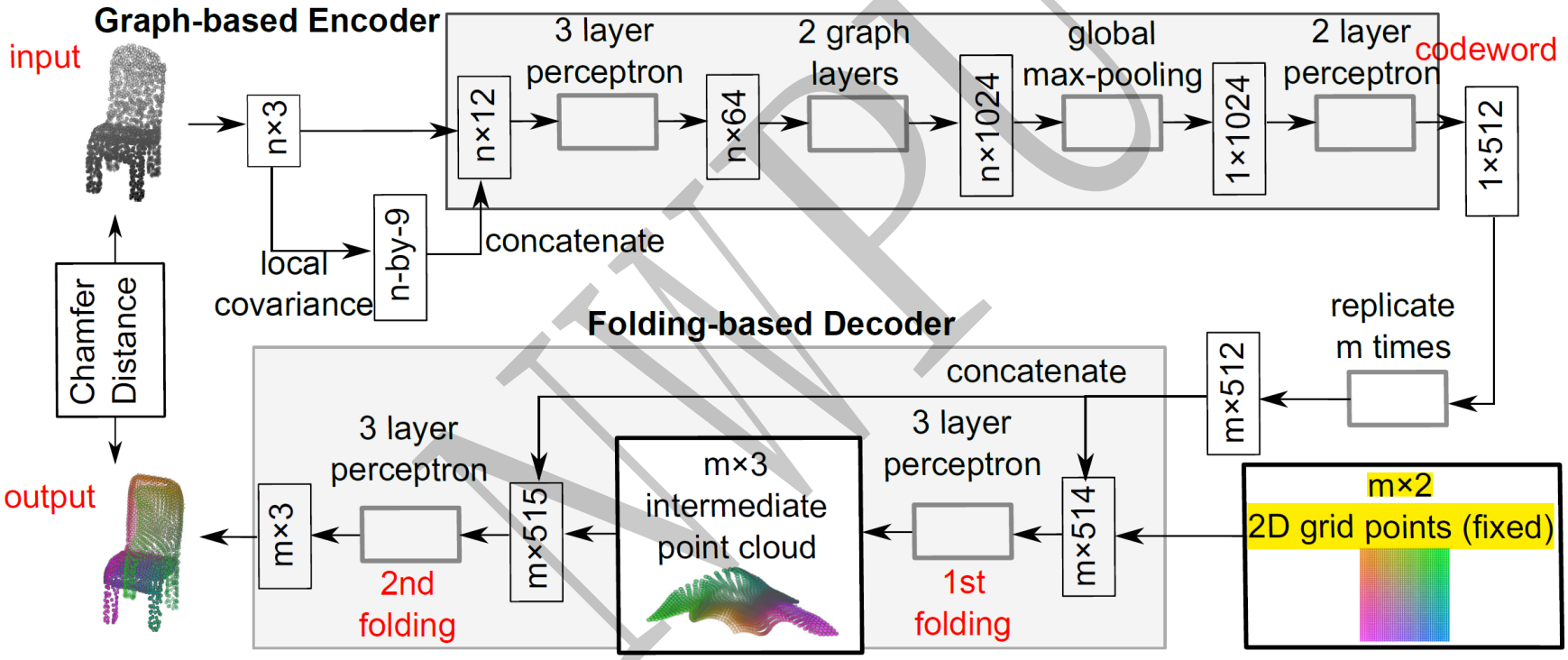


Figure 1. **FoldingNet Architecture**. The graph-layers are the graph-based max-pooling layers mentioned in (2) in Section 2.1. The 1st and the 2nd folding are both implemented by concatenating the codeword to the feature vectors followed by a 3-layer perceptron. Each perceptron independently applies to the feature vector of a single point as in [41], i.e., applies to the rows of the m-by-k matrix.

FoldingNet

- Overall architecture

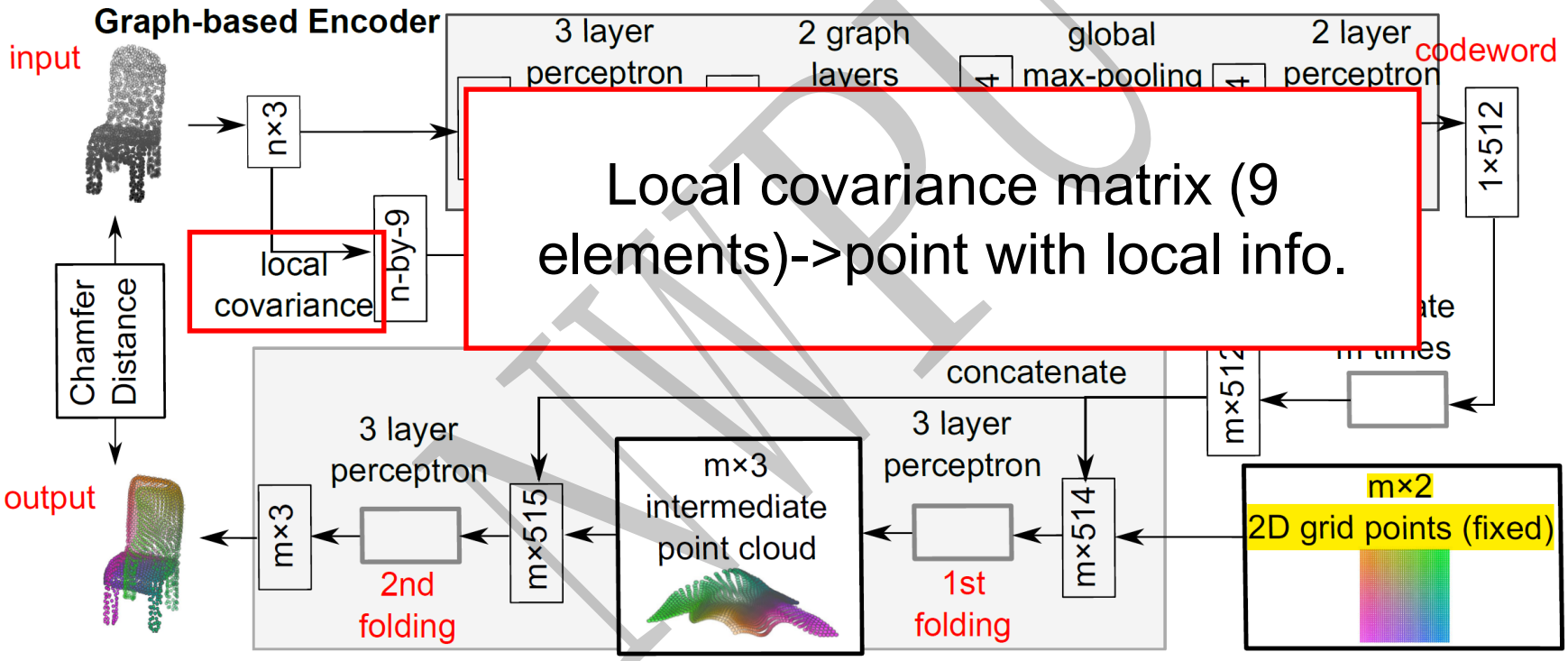


Figure 1. **FoldingNet Architecture**. The graph-layers are the graph-based max-pooling layers mentioned in (2) in Section 2.1. The 1st and the 2nd folding are both implemented by concatenating the codeword to the feature vectors followed by a 3-layer perceptron. Each perceptron independently applies to the feature vector of a single point as in [41], i.e., applies to the rows of the m-by-k matrix.

FoldingNet

- Overall ar

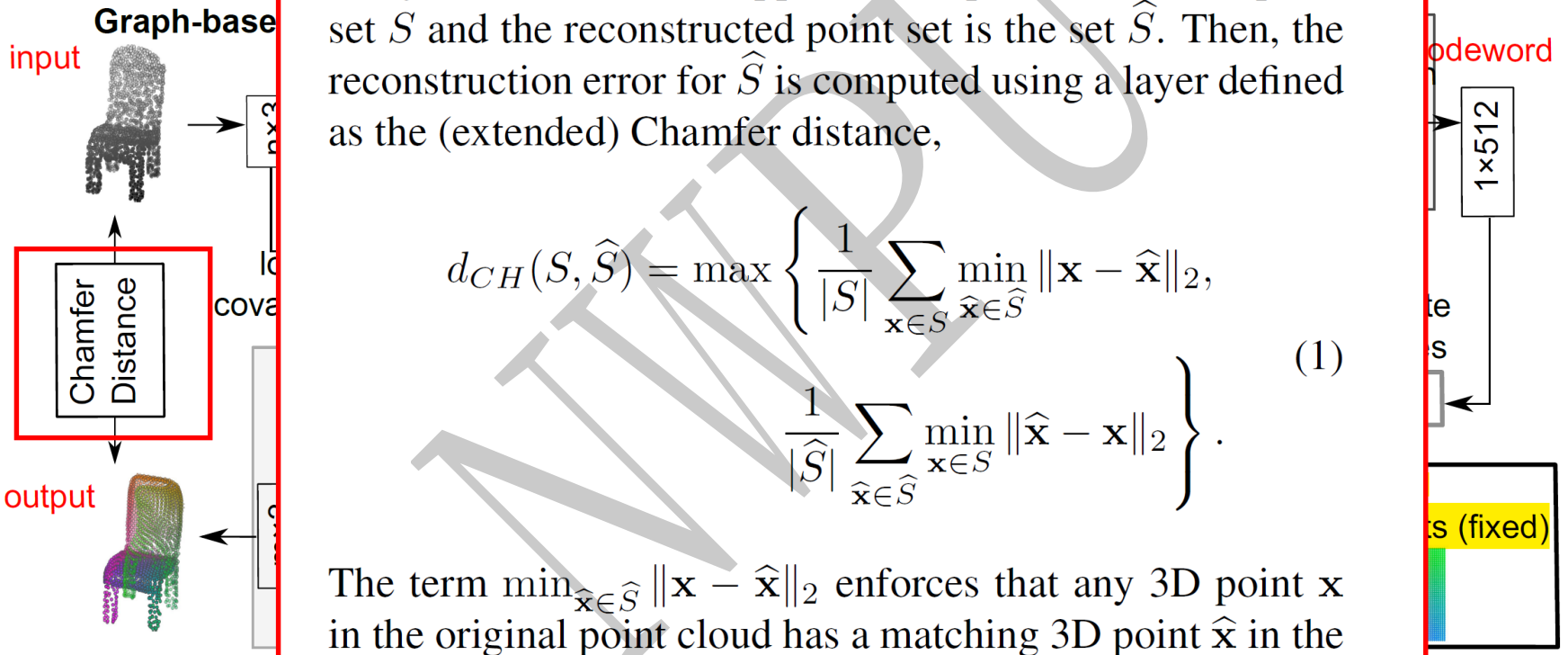


Figure 1. **FoldingNet Arch** and the 2nd folding are bo perceptron independently a

sarily the same as n. Suppose the input contains the point set S and the reconstructed point set is the set $\widehat{S}$. Then, the reconstruction error for $\widehat{S}$ is computed using a layer defined as the (extended) Chamfer distance,

$$d_{CH}(S,\widehat{S}) = \max\left\{\frac{1}{|S|}\sum_{\mathbf{x}\in S}\min_{\widehat{\mathbf{x}}\in\widehat{S}}\|\mathbf{x}-\widehat{\mathbf{x}}\|_2,\ \frac{1}{|\widehat{S}|}\sum_{\widehat{\mathbf{x}}\in\widehat{S}}\min_{\mathbf{x}\in S}\|\widehat{\mathbf{x}}-\mathbf{x}\|_2\right\}. \quad (1)$$

The term $\min_{\widehat{\mathbf{x}}\in\widehat{S}}\|\mathbf{x}-\widehat{\mathbf{x}}\|_2$ enforces that any 3D point $\mathbf{x}$ in the original point cloud has a matching 3D point $\widehat{\mathbf{x}}$ in the reconstructed point cloud, and the term $\min_{\mathbf{x}\in S}\|\widehat{\mathbf{x}}-\mathbf{x}\|_2$

on 2.1. The 1st rceptron. Each trix.

FoldingNet

- Training visualization

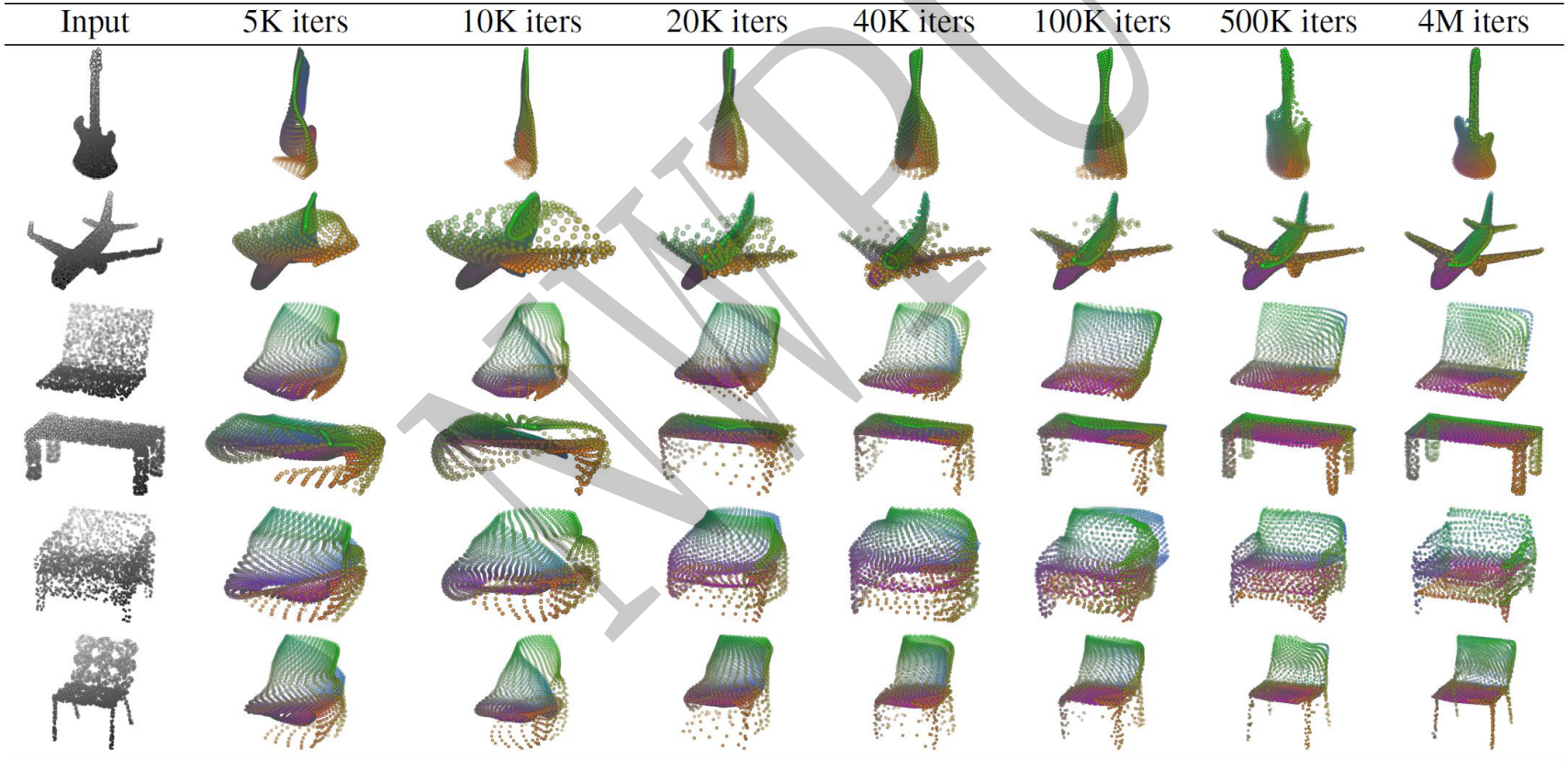
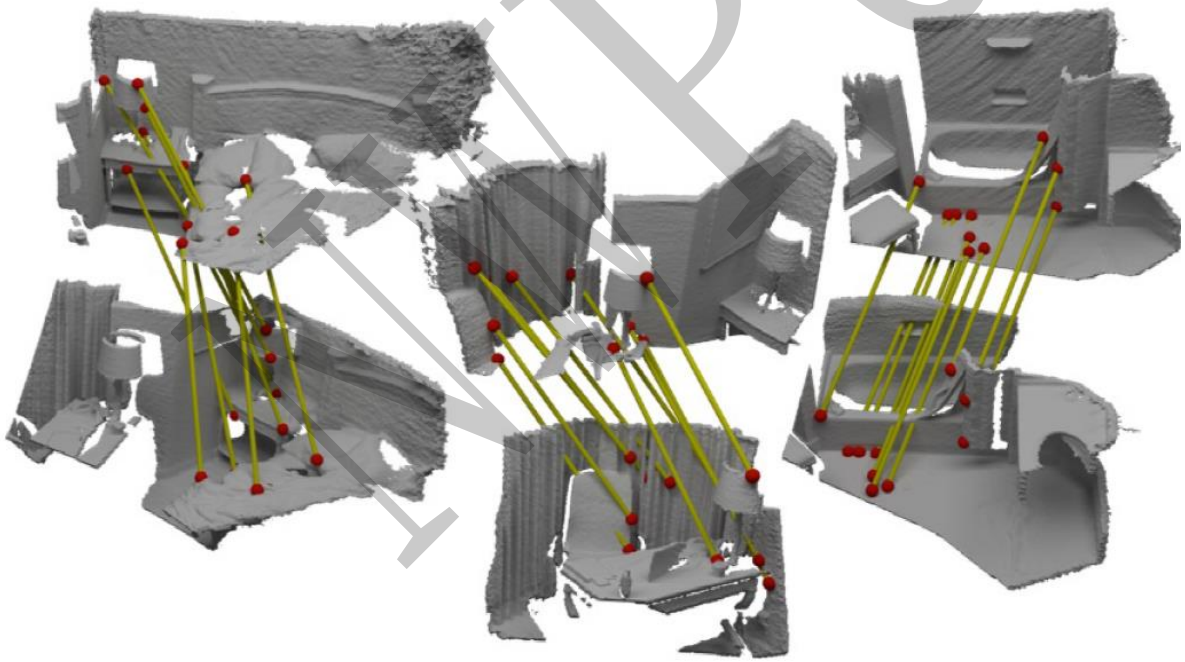


Table 2. Illustration of the training process. Random 2D manifolds gradually transform into the surfaces of point clouds.

3DMatch

- The first learned local descriptor for 3D registration



3DMatch

- Use Harris 3D effective keypoints for feature extraction and network training

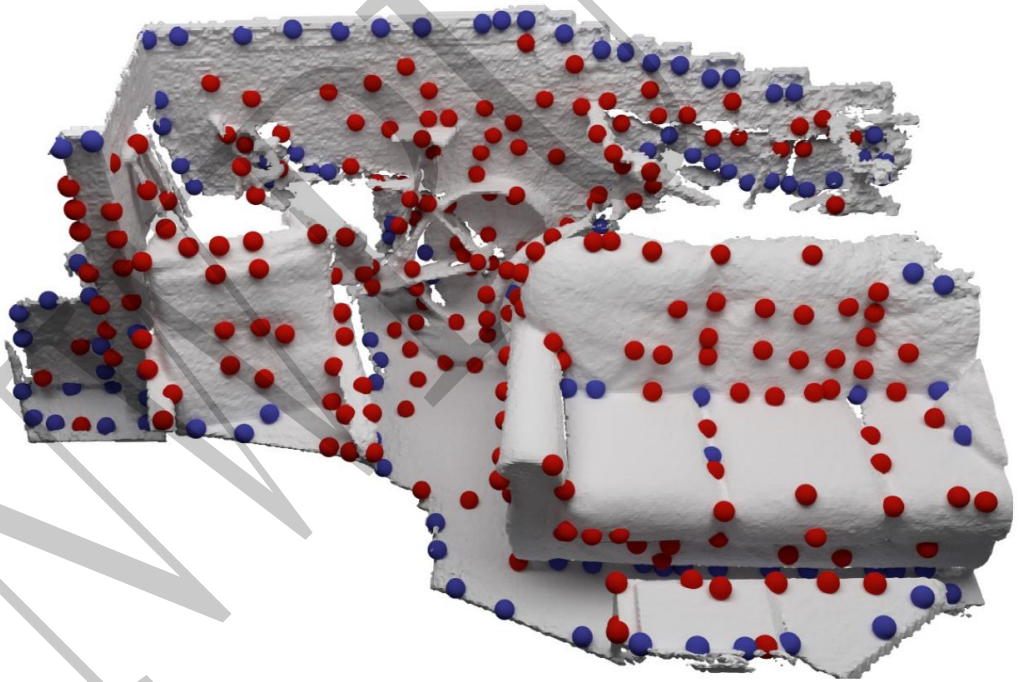
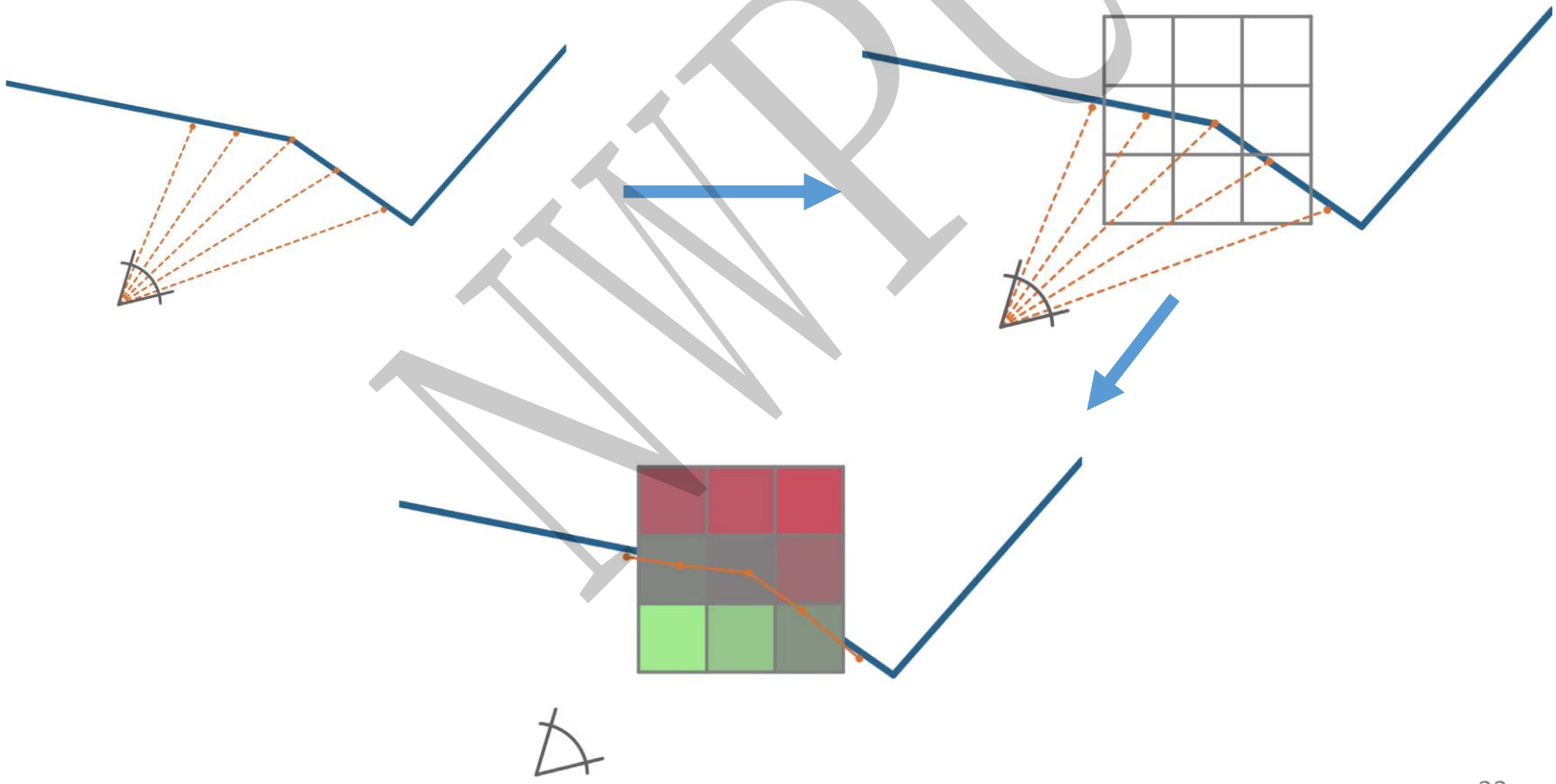


Figure 2: We use Harris corner responses to detect keypoints. We discard keypoints whose local regions are not observed by enough frames (blue). Only well-observed keypoints (red) and their local regions are used to train 3DMatch.

3DMatch

- How to represent the data for learning?
- TSDF: Truncated Signed Distance Function



3DMatch

- Voxel (TSDF) representation
- Feature learning (3D CNN)
- Metric learning (MLP)

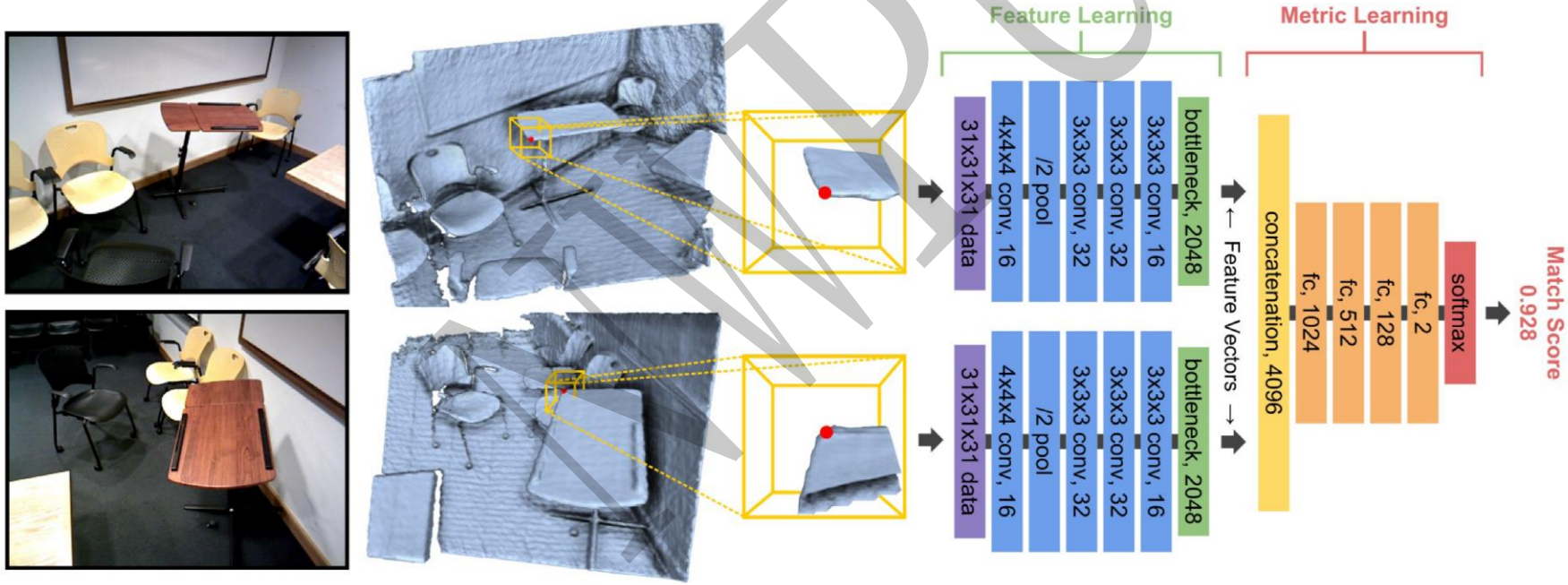


Figure 3: Matching two keypoints from different geometric fragments using 3DMatch. For each keypoint, its local 3D region is extracted from the TDF volume of its fused fragment. This 3D region is then passed through the the feature network to return a feature vector (green). To compare two keypoints to each other, their corresponding feature vectors are concatenated and fed to the metric network which returns a similarity score.

3DMatch

- Learned feature visualization with t-SNE

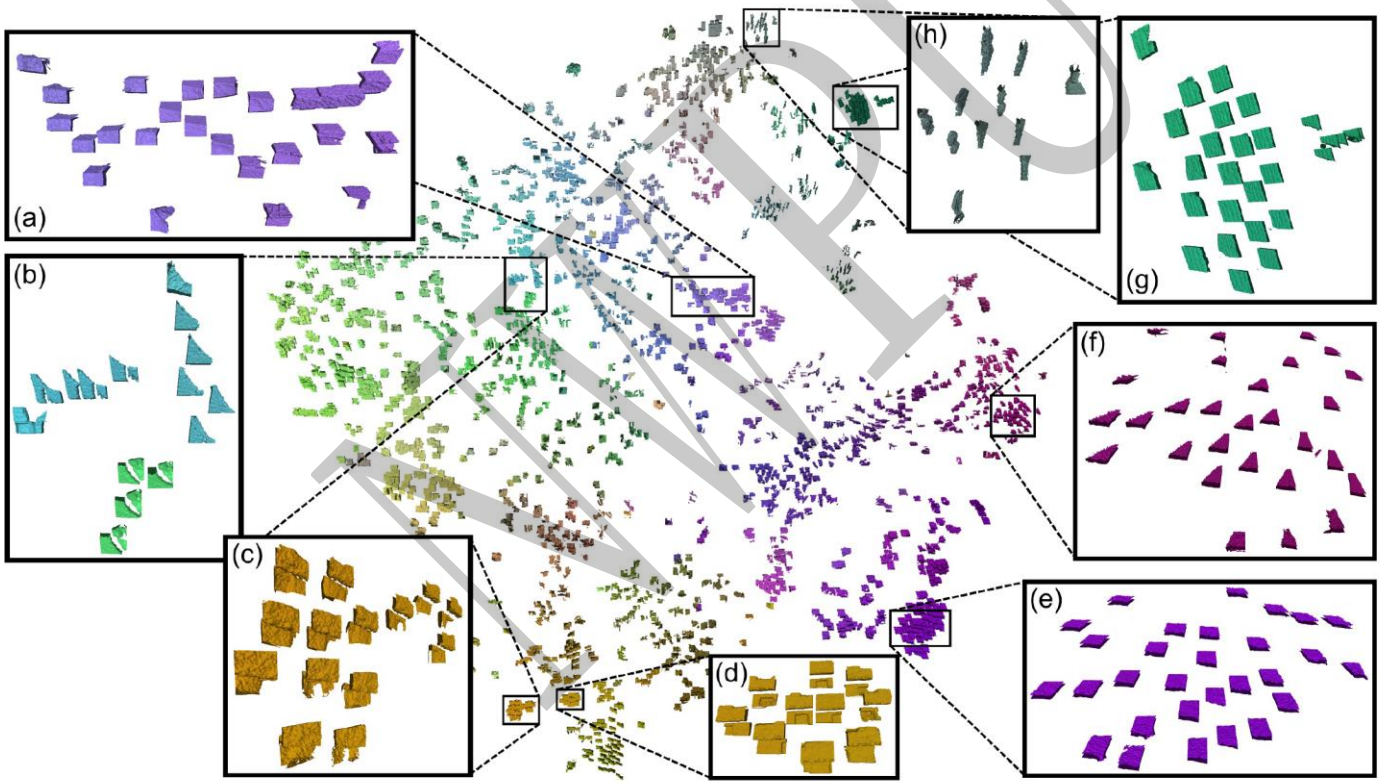


Figure 4: A feature embedding of local keypoint regions in a test scene, visualized using t-SNE. The learned features are highly predictive of geometric structure as well as local context. This embedding suggests that our 3DMatch network is able to coherently group 3D local keypoint volumes based on properties such as edges (a,f), planes (e), corners (c,d), and other local geometric structures (g, b, h) in the face of noisy and partial data.

3DMatch

- Open issue: what is the limitation of 3DMatch?

PPFNet

- Directly learn on point clouds

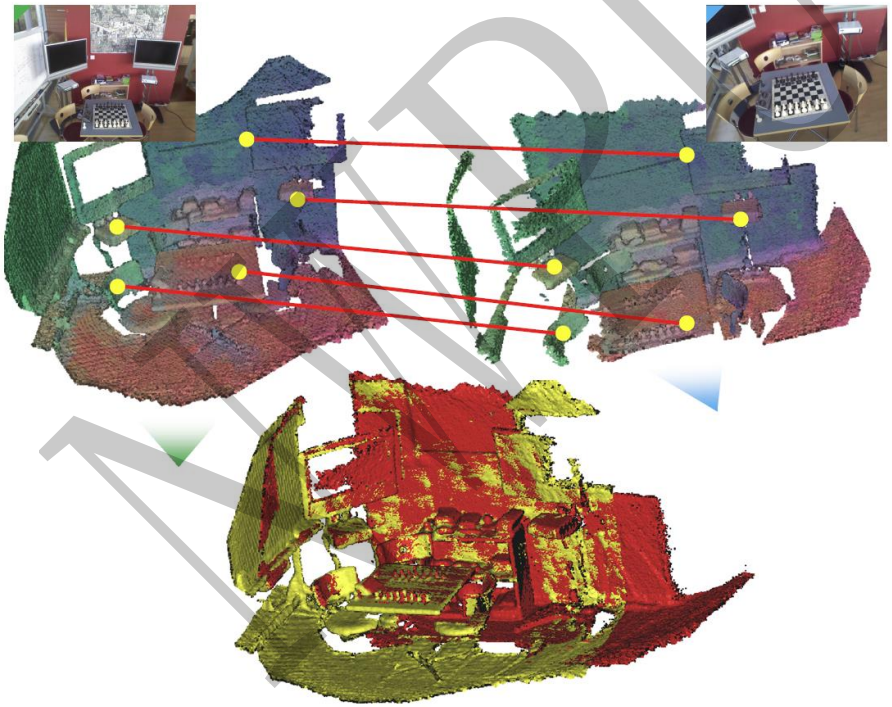


Figure 1. PPFNet generates repeatable, discriminative descriptors and can discover the correspondences simultaneously given a pair of fragments. Point sets are colored by a low dimensional embedding of the local feature for visualization. 3D data and the illustrative image are taken from 7-scenes dataset [39].

PPFNet

- Encoding of the input local patch

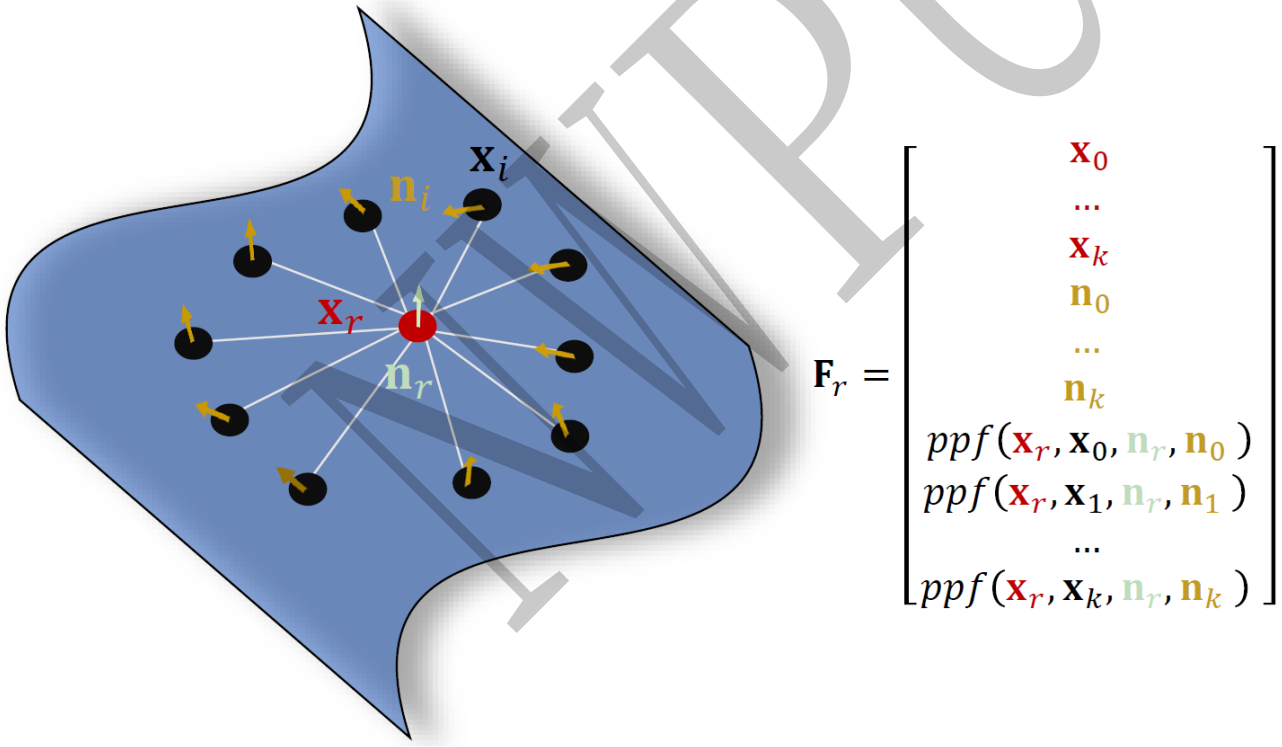
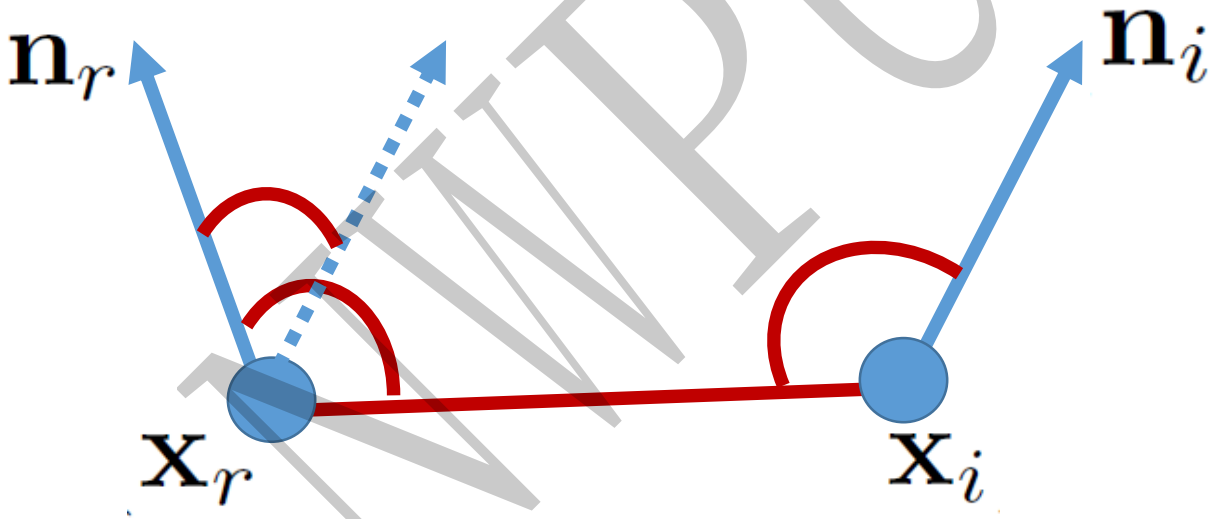


Figure 3. Simplistic encoding of a local patch.

PPFNet

- Definition of point pair feature (PPF)
- The distinction with respect to FPFH?



$$\mathbf{f} : (\mathbf{x}_r^T, \mathbf{x}_i^T)^T \rightarrow (\angle(\mathbf{n}_r, \mathbf{d}), \angle(\mathbf{n}_i, \mathbf{d}), \angle(\mathbf{n}_r, \mathbf{n}_i), \|\mathbf{d}\|_2)^T$$

PPFNet

- The architecture

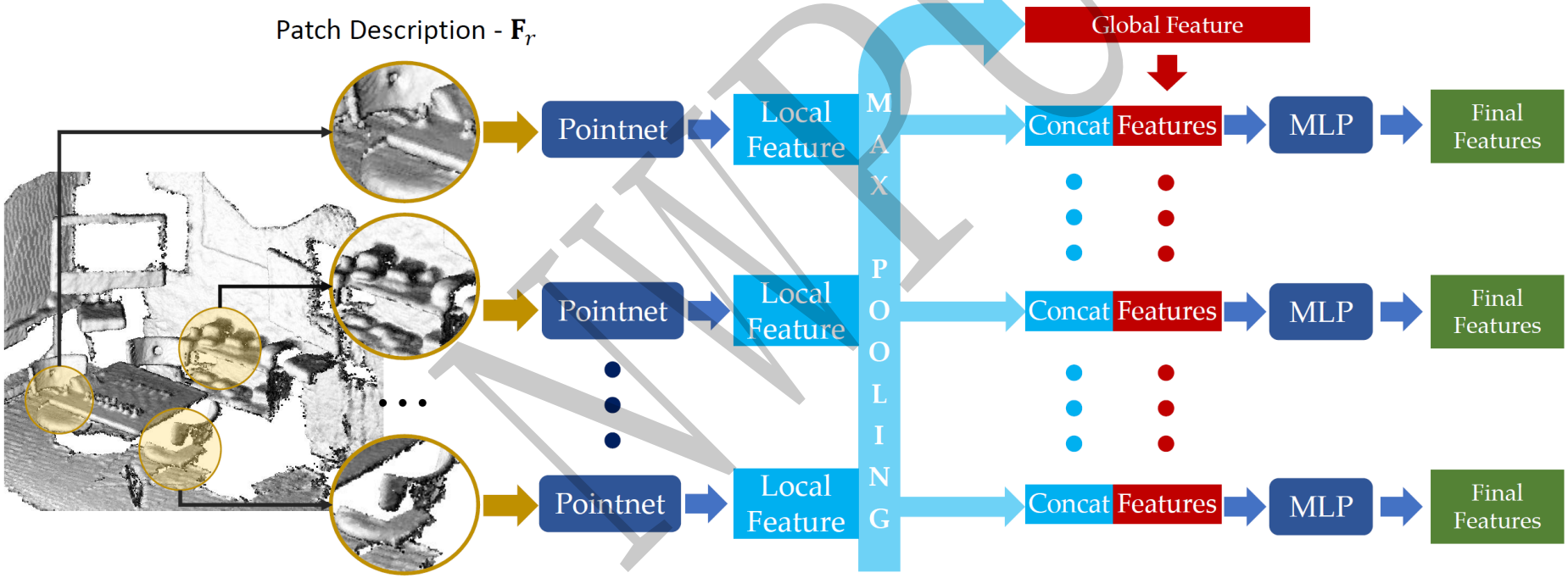


Figure 2. PPFNET, our inference network, consists of multiple PointNets, each responsible for a local patch. To capture the global context across all local patches, we use a max-pooling aggregation and fusing the output back into the local description. This way we are able to produce stronger and more discriminative local representations.

PPFNet

- What do learned feature get and miss?

Table 1. Our evaluations on the 3DMatch benchmark before RANSAC. *Kitchen* is from 7-scenes [39] and the rest from SUN3D [47].

	Spin Images [22]	SHOT [38]	FPFH [35]	USC [42]	PointNet [31]	CGF [24]	3DMatch [49]	3DMatch-2K [49]	PPFNet (ours)
Kitchen	0.1937	0.1779	0.3063	0.5573	0.7115	0.4605	0.5751	0.5296	**0.8972**
Home 1	0.3974	0.3718	0.5833	0.3205	0.5513	0.6154	**0.7372**	0.6923	0.5577
Home 2	0.3654	0.3365	0.4663	0.3077	0.5385	0.5625	**0.7067**	0.6202	0.5913
Hotel 1	0.1814	0.2080	0.2611	0.5354	0.4071	0.4469	0.5708	0.4779	**0.5796**
Hotel 2	0.2019	0.2212	0.3269	0.1923	0.2885	0.3846	0.4423	0.4231	**0.5769**
Hotel 3	0.3148	0.3889	0.5000	0.3148	0.3333	0.5926	**0.6296**	0.5185	0.6111
Study	0.0548	0.0719	0.1541	0.5068	0.4315	0.4075	**0.5616**	0.3904	0.5342
MIT Lab	0.1039	0.1299	0.2727	0.4675	0.5065	0.3506	0.5455	0.4156	**0.6364**
Average	0.2267	0.2382	0.3589	0.4003	0.4710	0.4776	0.5961	0.5085	**0.6231**

CGF

- Learning compact geometric features (CGF)

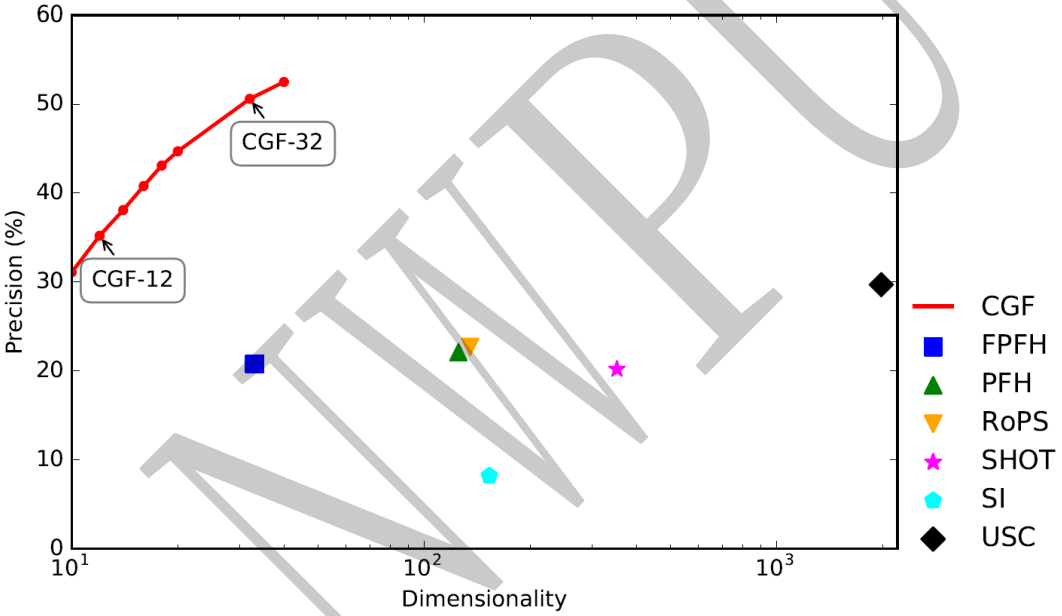
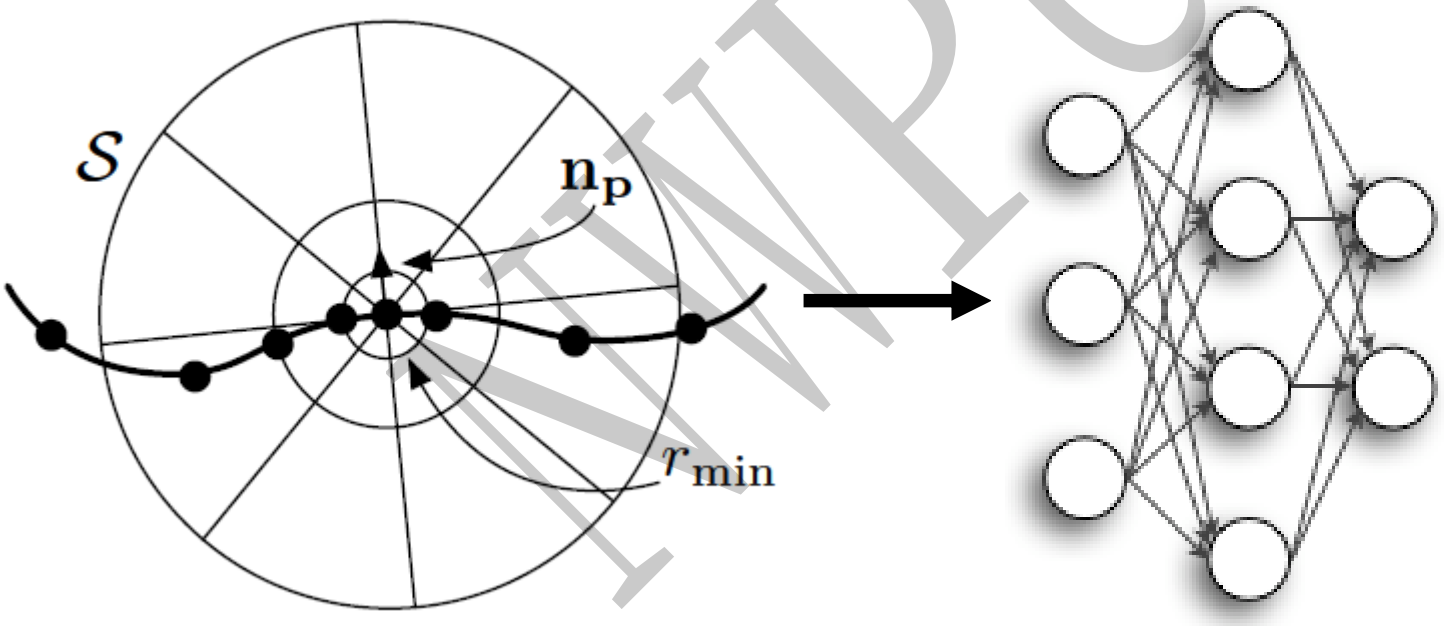


Figure 1. Our approach yields a family of Compact Geometric Features (CGF), parameterized by dimension. This figure illustrates the performance of CGF on the SceneNN test set. Our features are both more compact and more precise than the baselines. The horizontal axis (dimensionality) is on a logarithmic scale.

CGF

- Handcrafted + MLP Learning



CGF

- Training

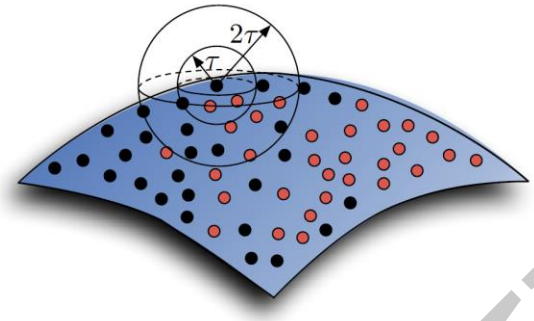


Figure 3. Two overlapping points clouds, shown here in red and black, are sampled from an underlying surface. We consider two concentric spheres of radius τ and 2τ around a black point $\mathbf{p}$. The red points in the innermost sphere, which form the set $\mathcal{N}_{\mathbf{p},\tau}$, are good correspondences for $\mathbf{p}$. The red points in the outermost sphere form the set $\mathcal{N}_{\mathbf{p},2\tau}$. We generate triplets for $\mathbf{p}$ by sampling $\mathbf{x}^p \in \mathcal{N}_{\mathbf{p},\tau}$ and $\mathbf{x}^n \in \mathcal{N}_{\mathbf{p},2\tau} \setminus \mathcal{N}_{\mathbf{p},\tau}$.

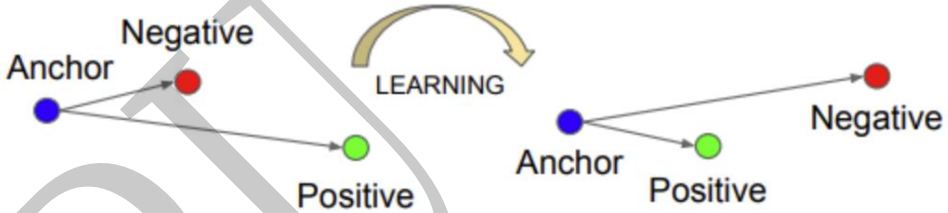


Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

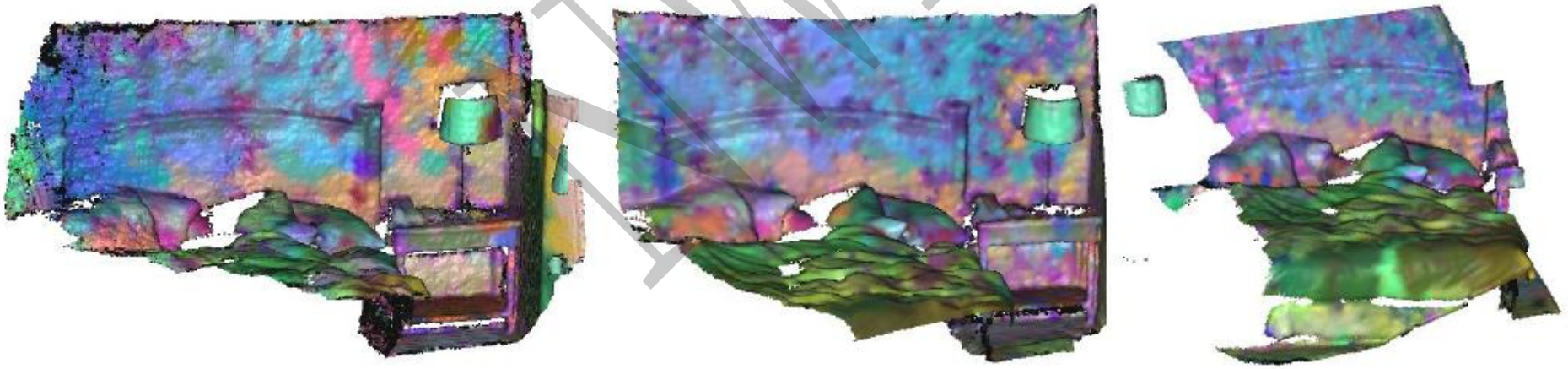
$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \Big[\|f(\mathbf{x}_i^a;\boldsymbol{\theta}) - f(\mathbf{x}_i^p;\boldsymbol{\theta})\|^2 - \|f(\mathbf{x}_i^a;\boldsymbol{\theta}) - f(\mathbf{x}_i^n;\boldsymbol{\theta})\|^2 + 1 \Big]_+, \quad (2)$$

where $\boldsymbol{\theta}$ are the parameters of the mapping f and $[\cdot]_+$ denotes $\max(\cdot, 0)$. Intuitively, f is optimized such that $\mathbf{x}_i^a$ is embedded closer to $\mathbf{x}_i^p$ than to $\mathbf{x}_i^n$, with a margin separating the distances.

- Difficult negative samples
- Negative samples

PPF-FoldNet

- A variant of PPFNet
- Unsupervised
- Invariant to rotation



PPF-FoldNet

- Encoding xyz positions with invariant PPF attributes as the input of the network

A point cloud/set: $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^6\}$

Each point is represented by: $\mathbf{x} = \{\mathbf{p}, \mathbf{n}\} \in \mathbb{R}^6$

A local patch around a reference point $\mathbf{x}_r$ **is:**

$$\Omega_{\mathbf{x}_r} \subset \mathbf{X}$$

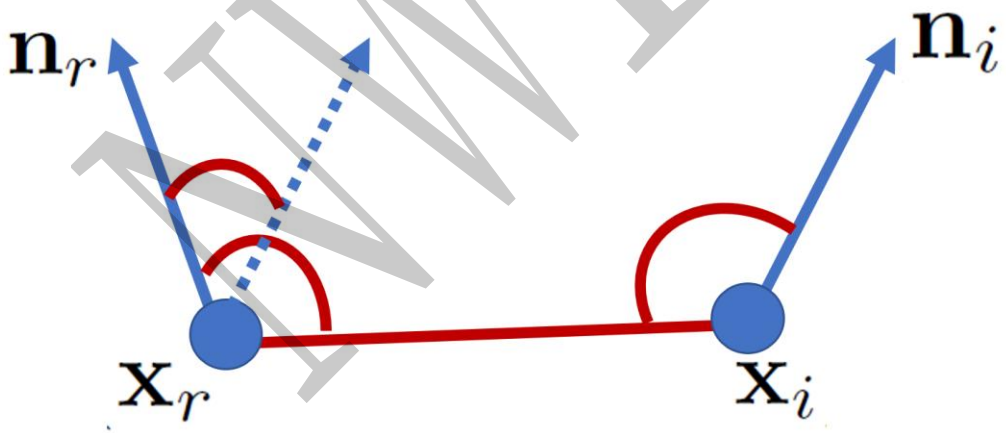
Use a feature $\mathbf{F}_\Omega$ **to represent** $\Omega_{\mathbf{x}_r}$

PPF-FoldNet

- Encoding xyz positions with invariant PPF attributes as the input of the network

Use a feature $\mathbf{F}_{\Omega}$ to represent $\Omega_{\mathbf{x}_r}$

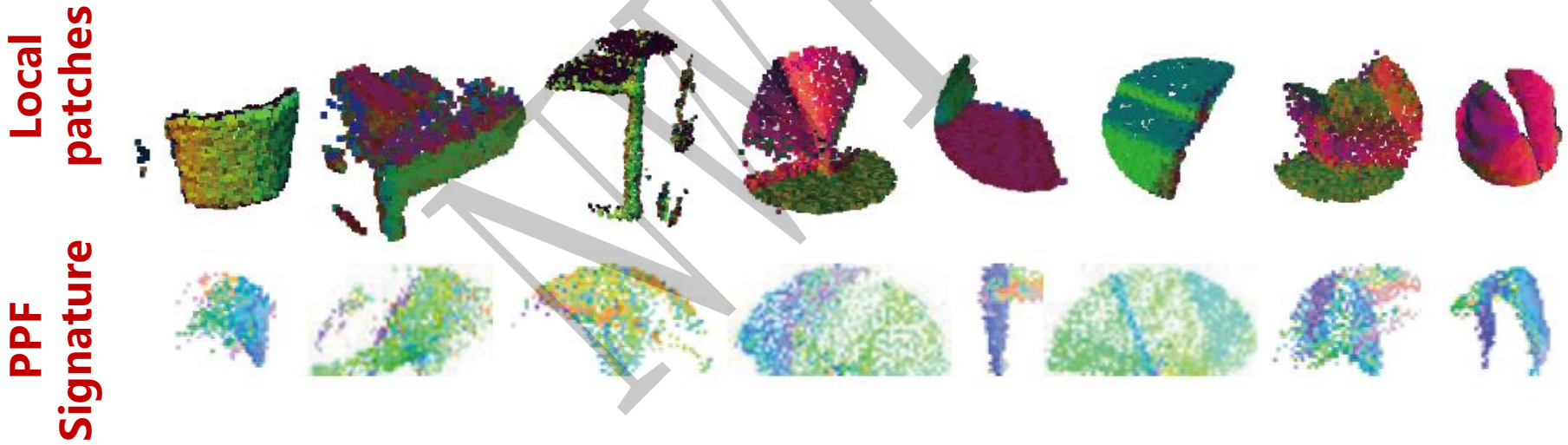
$$\mathbf{F}_{\Omega} = \{\, \mathbf{f}(\mathbf{x}_r, \mathbf{x}_1) \cdots \mathbf{f}(\mathbf{x}_r, \mathbf{x}_i) \cdots \mathbf{f}(\mathbf{x}_r, \mathbf{x}_N) \,\} \in \mathbb{R}^{4 \times N-1},\ i \neq r$$



$$\mathbf{f} : (\mathbf{x}_r^T, \mathbf{x}_i^T)^T \rightarrow (\angle(\mathbf{n}_r, \mathbf{d}), \angle(\mathbf{n}_i, \mathbf{d}), \angle(\mathbf{n}_r, \mathbf{n}_i), \|\mathbf{d}\|_2)^T$$

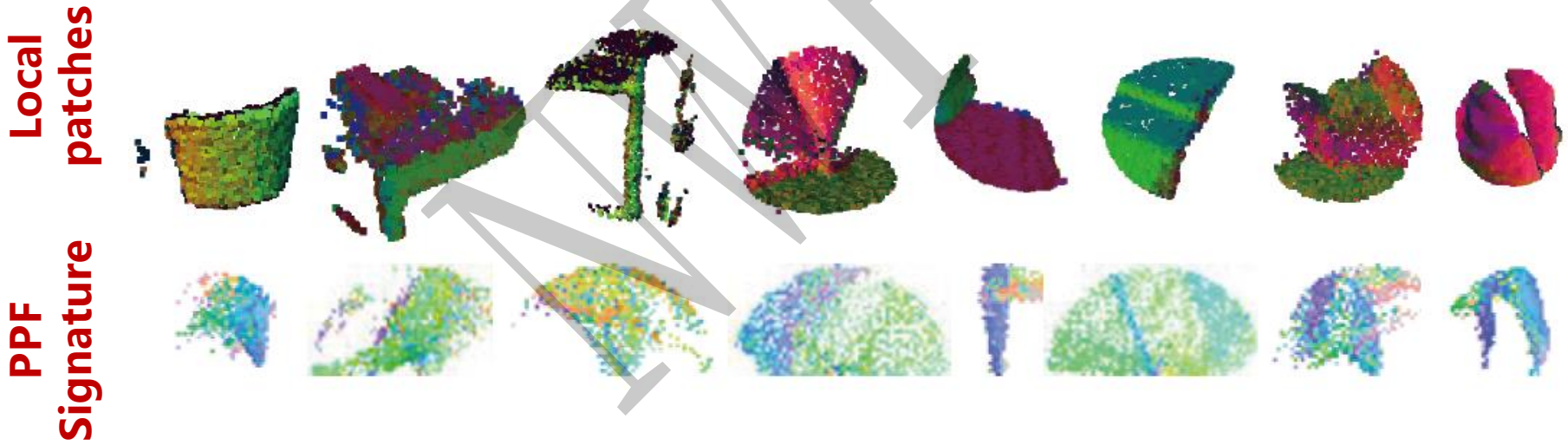
PPF-FoldNet

- Encoding xyz positions with invariant PPF attributes as the input of the network (*Visualization*)



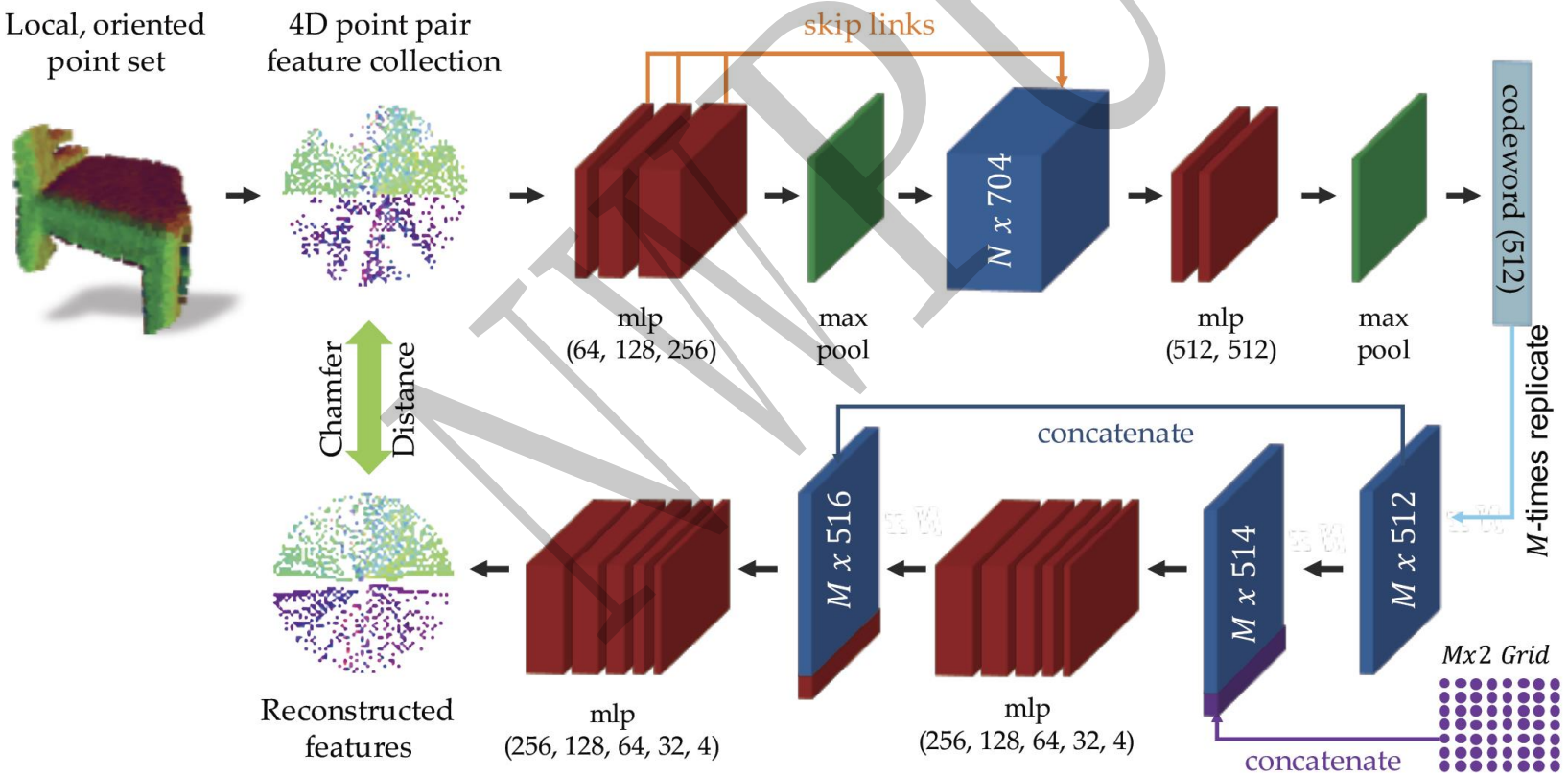
PPF-FoldNet

- Encoding xyz positions with invariant PPF attributes as the input of the network (*Visualization*)



PPF-FoldNet

- Overall architecture
- Technical part->Pointnet & FoldingNet



PPF-FoldNet

- This work (2008 ECCV) finally highlights the importance of rotation invariance for end2end learning

Table 2. Our results on the rotated 3DMatch benchmark. *Red Kitchen* data is from 7-scenes [37] and the rest imported from SUN3D [47].

	Spin Image [16]	SHOT [35]	FPFH [34]	3DMatch [51]	CGF [18]	PPFNet [8]	FoldNet [48]	Ours	Ours-5K
Kitchen	0.1779	0.1779	0.2905	0.0040	0.4466	0.0020	0.0178	0.7352	**0.7885**
Home 1	0.4487	0.3526	0.5897	0.0128	0.6667	0.0000	0.0321	0.7692	**0.7821**
Home 2	0.3413	0.3365	0.4712	0.0337	0.5288	0.0144	0.0337	0.6202	**0.6442**
Hotel 1	0.1814	0.2168	0.3009	0.0044	0.4425	0.0044	0.0133	0.6637	**0.6770**
Hotel 2	0.1731	0.2404	0.2981	0.0000	0.4423	0.0000	0.0096	0.6058	**0.6923**
Hotel 3	0.3148	0.3333	0.5185	0.0096	0.6296	0.0000	0.0370	0.9259	**0.9630**
Study	0.0582	0.0822	0.1575	0.0000	0.4178	0.0000	0.0171	0.5616	**0.6267**
MIT Lab	0.1169	0.1299	0.2857	0.0260	0.4156	0.0000	0.0260	0.6104	**0.6753**
Average	0.2265	0.2337	0.3640	0.0113	0.4987	0.0026	0.0233	0.6865	**0.7311**

Other features

1. Spezialetti R, Salti S, Stefano L D. ***Learning an effective equivariant 3d descriptor without supervision***[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6401-6410.
2. Gojcic Z, Zhou C, Wegner J D, et al. ***The perfect match: 3d point cloud matching with smoothed densities***[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 5545-5554.
3. Ao S, Hu Q, Yang B, et al. ***SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration***[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 11753-11762.
4. Li L, Fu H, Ovsjanikov M. ***UPDesc: Unsupervised Point Descriptor Learning for Robust Registration***[J]. arXiv preprint arXiv:2108.02740, 2021.

Summary

	input	Rotation invariance	Unsupervised
PointNet	xyz	×	×
PointNet++	xyz	×	×
FoldingNet	xyz	×	✓
3DMatch	voxel	×	×
PPFNet	xyz+normal	×	×
CGF	Descriptor	✓	×
PPF-FoldNet	xyz+normal	✓	✓

Future direction

1. Unsupervised
2. Few shot learning
3. Registration guided learning

See you

联系方式：jqyang@nwpu.edu.cn or YJQ_hust_nwpu （微信）

办公室：数字化大楼A101

个人主页：https://teacher.nwpu.edu.cn/person/2019010121