目录

[2017 September 5 Tuesday 3](#_Toc495238999)

[(1) Geometric Loss Functions for Camera Pose Regression with Deep Learning (cvpr17) 3](#_Toc495239000)

[2017 September 1 Friday 3](#_Toc495239001)

[(1)–(5) Face-3D-Reconstruction\_Column 3](#_Toc495239002)

[2017 September 2 Saturday 3](#_Toc495239003)

[(1)-(7) Face-3D-Reconstruction\_Column 3](#_Toc495239004)

[2017 September 3 Sunday 4](#_Toc495239005)

[(1) PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization (iccv15) 4](#_Toc495239006)

[2017 September 4 Monday 5](#_Toc495239007)

[(1) End-to-End Learning of Geometry and Context for Deep Stereo Regression (cvpr17) 5](#_Toc495239008)

[2017 September 5 Tuesday 7](#_Toc495239009)

[(2) Geometric Loss Functions for Camera Pose Regression with Deep Learning (cvpr17) 7](#_Toc495239010)

[(3) Point Net: Deep Learning on Point Sets for 3D Classification and Segmentation (cvpr17) 8](#_Toc495239011)

[2017 September 6 Wednesday 11](#_Toc495239012)

[(1) Monocular Dense 3D Reconstruction of a Complex Dynamic Scene from Two Perspective Frames (arXiv) 11](#_Toc495239013)

[2017 September 7 Thursday 13](#_Toc495239014)

[(1) A Combinatorial Solution to Non-Rigid 3D Shape-to-Image Matching (cvpr17) 13](#_Toc495239015)

[(2) KillingFusion: Non-rigid 3D Reconstruction without Correspondences (cvpr17) 13](#_Toc495239016)

[2017 September 8 Friday 14](#_Toc495239017)

[(1) A Point Set Generation Network for 3D Object Reconstruction from a Single Image (cvpr17) 14](#_Toc495239018)

[2017 September 9 Saturday 15](#_Toc495239019)

[(1) Template-based Monocular 3D Shape Recovery(tpami16) 15](#_Toc495239020)

[2017 September 11 12 13 Monday-Wednesday 16](#_Toc495239021)

[2017 September 14 Thursday 23](#_Toc495239022)

[(1) Pixelwise View Selection for Unstructured (eccv16) 23](#_Toc495239023)

[2017 September 15 Friday 25](#_Toc495239024)

[(1) Generative and Discriminative Voxel Modeling with Convolutional Neural Networks (arXiv) 25](#_Toc495239025)

[2017 September 16 Saturday 27](#_Toc495239026)

[(1) 3D Bounding Box Estimation Using Deep Learning and Geometry Networks (cvpr17) 27](#_Toc495239027)

[2017 September 17-23 Saturday-Saturday 29](#_Toc495239028)

[2017 September 24 Sunday 30](#_Toc495239029)

[(1) Multi-View 3D Object Detection Network for Autonomous Driving (cvpr17) 30](#_Toc495239030)

[2017 September 25 Monday 32](#_Toc495239031)

[(1) 3D Bounding Box Estimation Using Deep Learning and Geometry Networks (cvpr17) - (twice read) 32](#_Toc495239032)

[2017 September 26-October 2 Tuesday-Tuesday 33](#_Toc495239033)

[2017 October 4 Wednesday 34](#_Toc495239034)

[(1) A Morphable Model For The Synthesis Of 3D Faces(sig07) 34](#_Toc495239035)

[2017 October 5-6 Thursday – Friday 34](#_Toc495239036)

[2017 October 7 Saturday 35](#_Toc495239037)

[(1) Face-3D-Reconstruction\_Column 35](#_Toc495239038)

# **2017 September 5 Tuesday**

1. Geometric Loss Functions for Camera Pose Regression with Deep Learning (cvpr17)

pipeline:

BackGround

Large scale localisation research can be divided into two categories; place recognition and metric localisation

consequence:

Dataset

Cambridge Landmarks [22], 7 Scenes [46], Cambridge Landmarks [22]

Sequence

summary:

no code

no better than SIFT

# **2017 September 1 Friday**

(1)–(5) Face-3D-Reconstruction\_Column

# **2017 September 2 Saturday**

(1)-(7) Face-3D-Reconstruction\_Column

# **2017 September 3 Sunday**

1. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization (iccv15)

pipeline:

Learning target

Quaternion: arbitrary 4-D values are easily mapped to legitimate rotations by normalizing them to unit length

Loss Function

loss(I) =

individual networks to regress position(x) and orientation(q) separately performed poorly

Network

GoogleNet

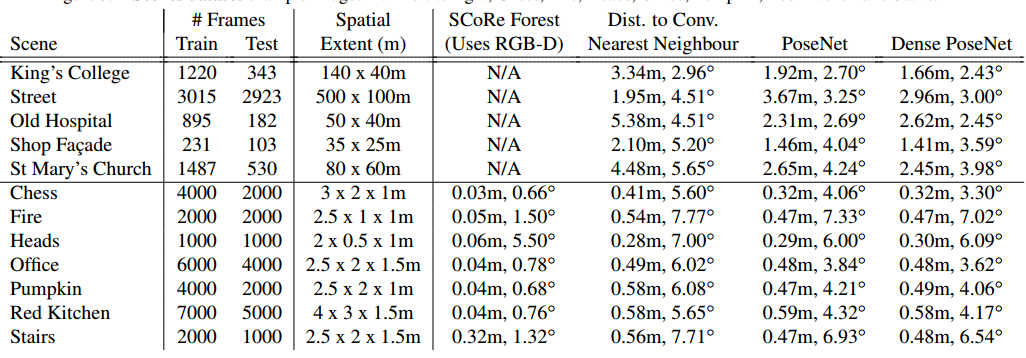
consequence:

dataset

Cambridge Landmark, 7 Scenes dataset [20],

sequence

bad at indoor things



summary:

code in caffe and tf

new dataset

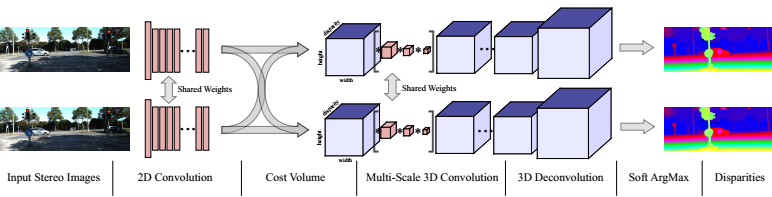
# **2017 September 4 Monday**

1. End-to-End Learning of Geometry and Context for Deep Stereo Regression (cvpr17)

pipeline:

Learning target

Quaternion: arbitrary 4-D values are easily mapped to legitimate rotations by normalizing them to unit length

Network 

2D Conv

share the parameters between the left and right towers to more effectively learn corresponding features

Cost Volume

cost volume of dimensionality *height×width×*(*max disparity + 1)×feature size*

concatenatingeach unary feature with their corresponding unary from the opposite stereo image across each disparity level, and packing these into the 4D volume

3-D convolutions

learn feature representations from the height, width and disparity dimensions

the additional dimension is a burden on the computational time

use encoder-decoder structure solve time problem

ArgMin

consequence:

dataset

Scene Flow [36] and KITTI [14, 35]

sequence

summary:

no code

constant learning rate of 1*×*10*-*3

# **2017 September 5 Tuesday**

1. Geometric Loss Functions for Camera Pose Regression with Deep Learning (cvpr17)

pipeline:

BackGround

Large scale localisation research can be divided into two categories; place recognition and metric localisation

Network

GoogleNet

Using pre-train weight

Output

We can easily learn camera position in Euclidean space

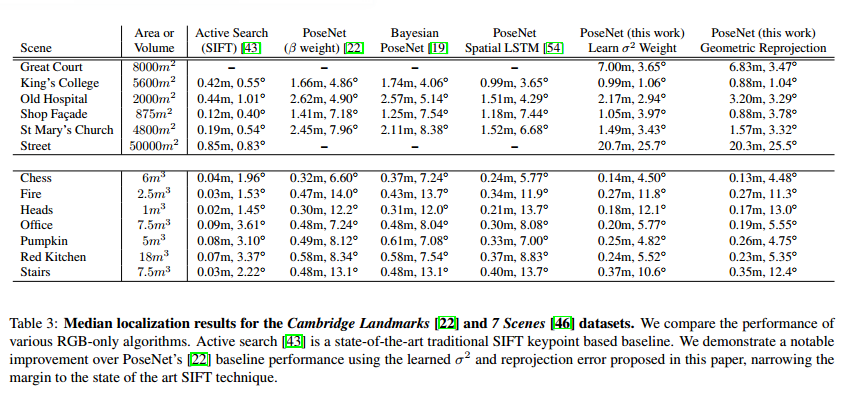
we chose quaternions as our orientation representation.

Loss Function

consequence:

dataset

Cambridge Landmarks [22], 7 Scenes [46], Cambridge Landmarks [22]

sequence

summary:

no code

no better than SIFT

1. Point Net: Deep Learning on Point Sets for 3D Classification and Segmentation (cvpr17)

pipeline:

Notation

A point cloud is represented as a set of 3D points fPij i = 1; :::; ng, where each point Pi is a vector of its (x; y; z) coordinate plus extra feature channels such as color, normal etc. For implicity and clarity, unless otherwise noted, we only use the (x; y; z) coordinate as our point’s channels.

Properties of Point Sets

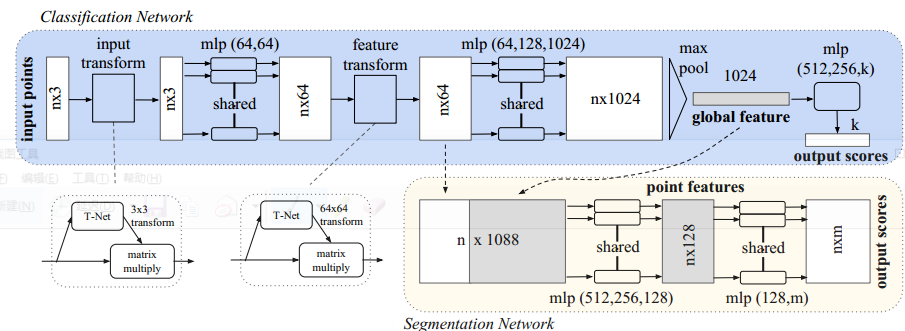
Unordered

Interaction among points.

points are not isolated, and neighboring points form a meaningful  
subset. Therefore, the model needs to be able to capture local structures from nearby points

Invariance under transformations

For example, rotating and translating points all together should not modify the global point cloud category

PointNet Architecture 

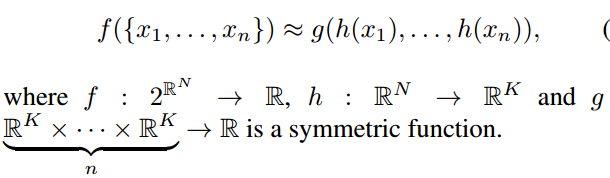
Classification

Our network has three key modules: the max pooling layer as a symmetric function to aggregate information from all the points, a local and global information combination structure, and two joint alignment networks that align both input points and point features

Symmetry Function for Unordered Input

Find that sort or RNN does not works

Our idea is to approximate a general function defined on a point set by applying a symmetric function on transformed elements



h by a multi-layer perceptron network and g by a composition of a single variable function and a max pooling function

Local and Global Information Aggregation

a vector [f1; : : : ; fK], which is a global signature of the input set

Joint Alignment Network

consequence:

Dataset

3D Object Classification

ModelNet40 [28]

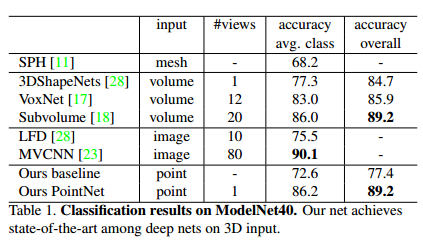
3D Object Part Segmentation

ModelNet40 [28]

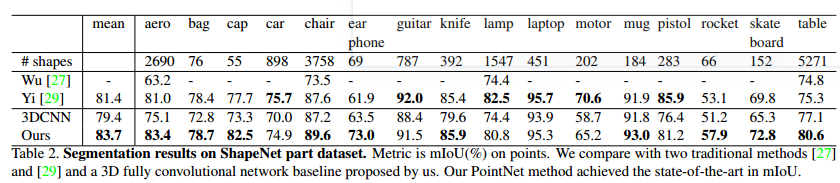
Stanford 3D semantic parsing data set [1]

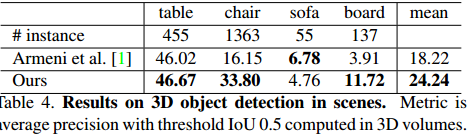
Sequence

3D Object Classification



**3D Object Part Segmentation**





summary:

# **2017 September 6 Wednesday**

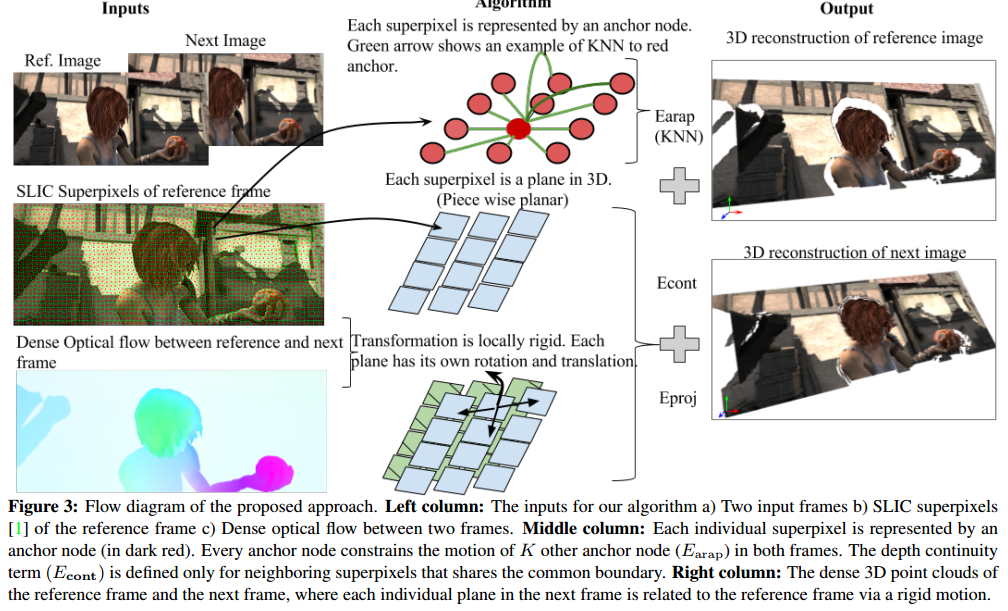
1. Monocular Dense 3D Reconstruction of a Complex Dynamic Scene from Two Perspective Frames (arXiv)

pipeline:

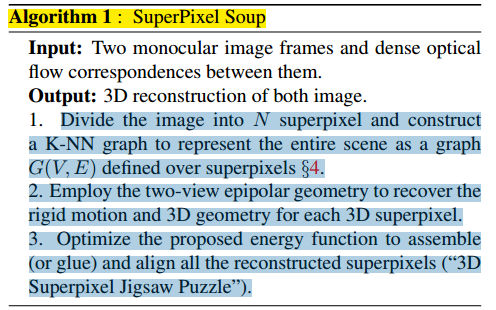
SuperPixel

define an anchor point xai for each superpixel, as the centroid point of the superpixel

K nearest neighbors, we build a K-NN graph G(V,E)



Pipline



Detail

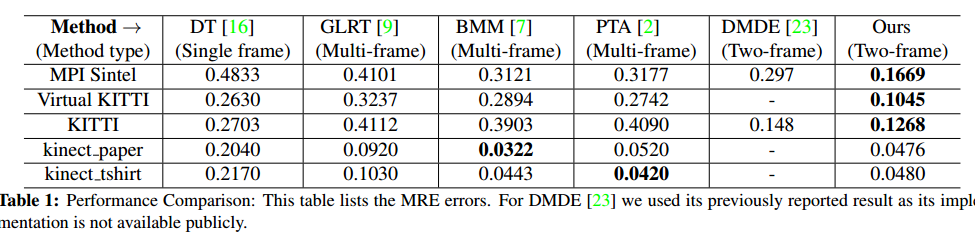
We partitioned a reference image into about 1,000-2,000 superpixels [1]. We used a state-of-the optical flow algorithm [3] to compute dense correspondences across two frames. Parameters like α1, α2, β, σ were tuned differently for different datasets. However, β = 3 and σ = 15 are fixed for all our tests on MPI Sintel and on VKITTI. To initialize the iteration, local rigid motion is estimated using traditional SfM pipeline [14]

consequence:

Dataset

KITTI dataset [12], the virtual KITTI [10], the MPI Sintel [6] and the YouTube-Objects [22]

Paper, T-shirts and Back sequence [28][27][11].

Sequence

summary:

# **2017 September 7 Thursday**

1. A Combinatorial Solution to Non-Rigid 3D Shape-to-Image Matching (cvpr17)

pipeline:

Template Shape Acquisition

calibrated automatically using an implementation (VisualSFM [36]) of standard rigid structure-frommotion (SfM)

extract a depth-map using the stereo method [5]

combine the depth-maps to recover a single watertight mesh S using the volumetric fusion technique of [35] combined with the probabilistic visibility approach of [12]

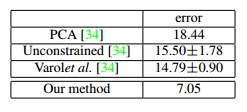
Non-Rigid Model Tracking

consequence:

Dataset

T-shirt ground truth dataset from CVLab [34]

Sequence



summary:

1. KillingFusion: Non-rigid 3D Reconstruction without Correspondences (cvpr17)

pipeline:

RGB-D data

consequence:

Dataset

Sequence

summary:

# **2017 September 8 Friday**

1. A Point Set Generation Network for 3D Object Reconstruction from a Single Image (cvpr17)

pipeline:

Network

Introduce our network progressively, we start from a simple version and gradually add components

The predictor generates the coordinates of *N* points through a fully connected network.

This version has two parallel predictor branches – a fully-connected (fc) branch and a deconvolution (deconv) branch

Distance Metric between Point Sets

？

Generation of Multiple Plausible Shapes

？

consequence:

Dataset

Shape Net dataset [4]

Sequence

summary:

# **2017 September 9 Saturday**

1. Template-based Monocular 3D Shape Recovery(tpami16)

pipeline:

consequence:

summary:

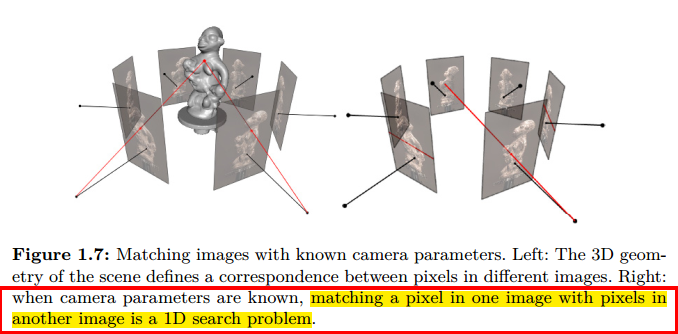
# **2017 September 11 12 13 Monday-Wednesday**

1. Template-based Monocular 3D Shape Recovery(tpami16)
2. Introduction

To see why, consider a 3D point belonging to the 3D scene geometry (See Figure 1.7 left). Projecting the 3D point into the set of visible cameras establishes a unique correspondence between the projected coordinates on each image. Given a pixel in an image, finding the corresponding pixels in other images needs two ingredients:

• An efficient way to generate possible pixel candidates in other images.  
• A measure to tell how likely a given candidate is the correct match

A pixel in an image generates a3D optic ray that passes through the pixel and the camera center of the image. The corresponding pixel on another image can only lie on the projection of that optic ray into the second image. T



1. Multi-view Photo-consistency

Photo-consistency: In [computer vision](https://en.wikipedia.org/wiki/Computer_vision) Photo-consistency determines whether a given [voxel](https://en.wikipedia.org/wiki/Voxel) is occupied. A voxel is considered to be photo consistent when its color appears to be similar to all the cameras that can see it.[[1]](https://en.wikipedia.org/wiki/Photo-consistency#cite_note-1) Most voxel coloring or space carving techniques require using photo consistency as a check condition in [Image-based modeling and rendering](https://en.wikipedia.org/wiki/Image-based_modeling_and_rendering) applications

A crucial requirement for photo-consistency measures is to compute photo-consistency on a set of images that see the same 3D geometry.

One needs the correct 3D-geometry to select which images to use in order to compute photo-consistency

2.1 Photo-consistency measures

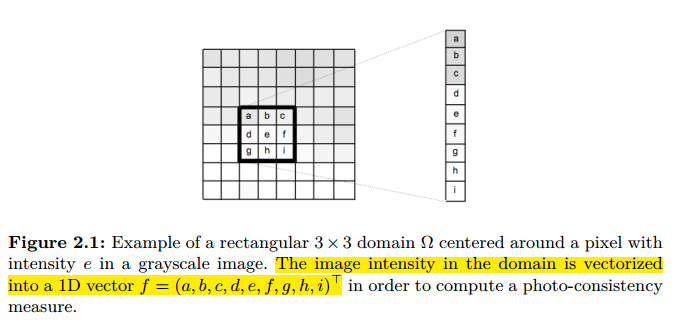
Given a set of N input images and a 3D point p seen by all the images, one can define the photo-consistency of p w.r.t. each pair of images Ii and Ij as:

Gij(p) = ρ(Ii(Ω(πi(p))), Ij(Ω(πj(p)))), (2.1)

where ρ(f, g) is a similarity measure that compares two vectors, πi(p) denotes the projection of p into image i, Ω(x) defines a support domain around point x, and Ii(x) denotes the image intensities

The main purpose of the support domain Ω is to define the size of a region where the appearance of the scene is expected to be unique and somewhat invariant to illumination and viewpoint changes

In MVS algorithms, the simplest way to define Ω for each image is to use a square grid of pixels with constant size across the images



2.1.1 Normalized Cross Correlation

*ρNCC*(*f, g*) = *∈* [*-*1*,* 1]*,*

2.1.7 Interval comparison

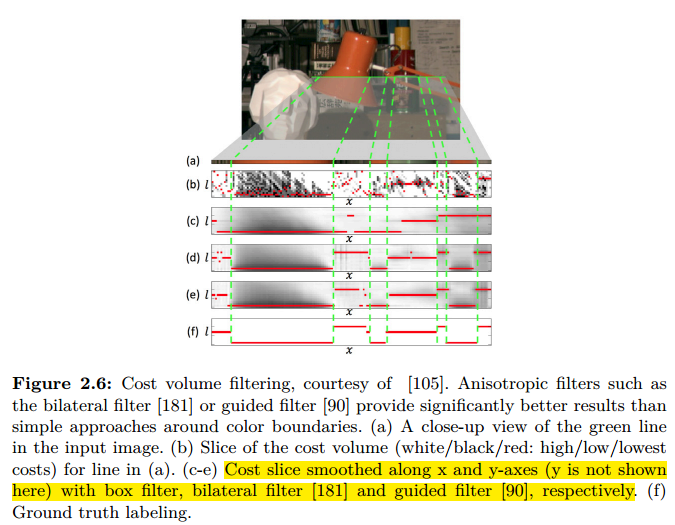
i) normalize different photo-consistency values to the same range,

ii) transform the original photo-consistency into something closer to "likelihood of geometry".

2.1.8 Photo-consistency aggregation

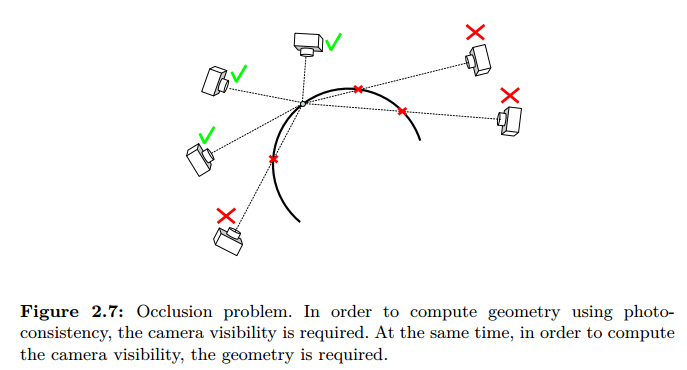
Photo-consistency is a noisy measure and often filtered before being used to compute 3D geometry

A related concept to the domain that is very common in stereo algorithms is photo-consistency aggregation which consists in spatially aggregating the photo-consistency measure to increase its robustness



2.1.9 Photo-consistency representation

2.2 Visibility estimation in state-of-the-art algorithms



2.2.1 Space-carving for visibility estimation

The main contribution of the work was the proposal of a geometric constraint on the camera centers such that there exists an ordinal visibility constraint on all the 3D voxels in the scene.

Visibility estimation for such large image collections typically happen  
in two phases:

First, visibility is estimated coarsely by clustering the initial set of images and reducing the large-scale MVS problem into a sequence of small sub-problems

Second. more fine-scale visibility estimation is conducted per 3D point basis

2.2.2 Coarse visibility estimation via pose clustering

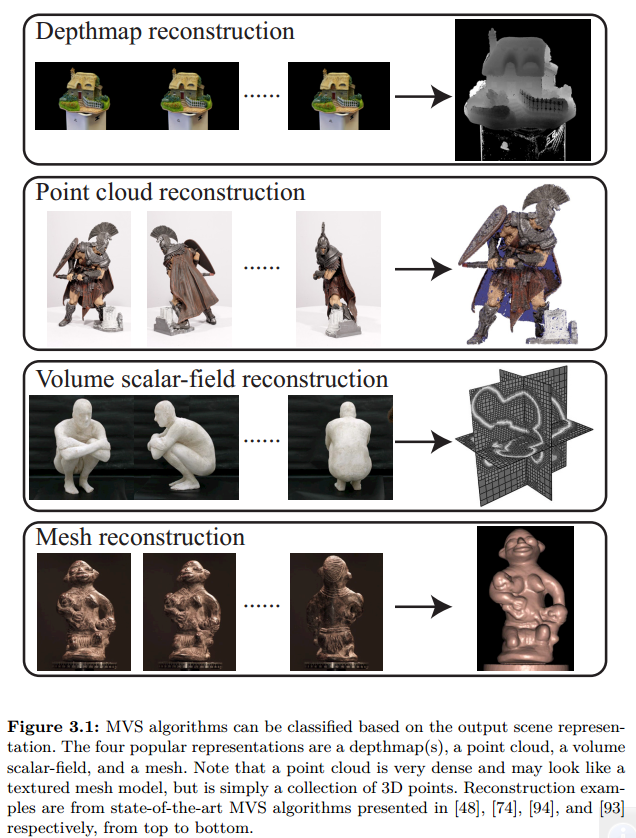
2.2.3 Fine-scale visibility estimation

1. Algorithms: From Photo-Consistency to 3D Reconstruction

provide details of recent popular multi-view stereo (MVS) algorithms that reconstruct 3D geometry by using photo consistency

In this article we use the output scene representation as an axis  
of taxonomy, because it often determines the range of possible applications

In this article we use the output scene representation as an axis  
of taxonomy, because it often determines the range of possible applications



3.1. Depthmap Reconstruction

3.2 Point-cloud Reconstruction

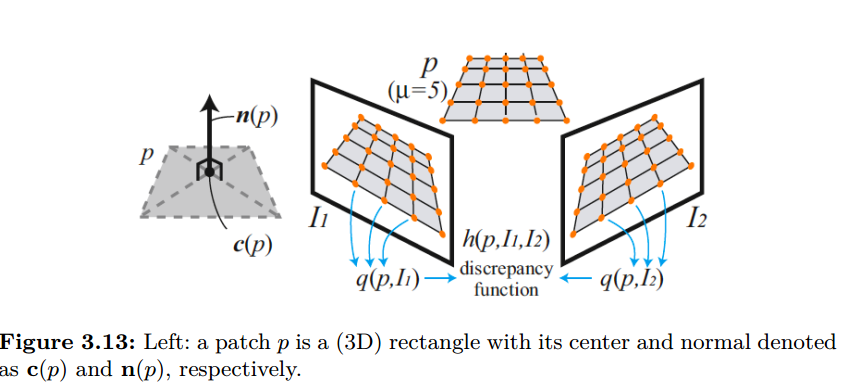
However, the depthmap quality tends to degrade significantly at depth discontinuities and occlusion boundaries

Researchers explored ways to estimate multiple depthmaps simultaneously while enforcing these approaches tend to make the optimization problem very large and computationally expensive

A common characteristic of point-cloud reconstruction algorithms is that they make use of an spatial consistency assumption and grow or expand the point-cloud on the surface of the scene during the reconstruction process, as opposed to reconstructing each point independently  
 3.2.1 Key Elements

Patch Model

A patch p is essentially a local tangent plane approximation of a surface, whose geometry is determined by its center c(p) and unit normal vector n(p)



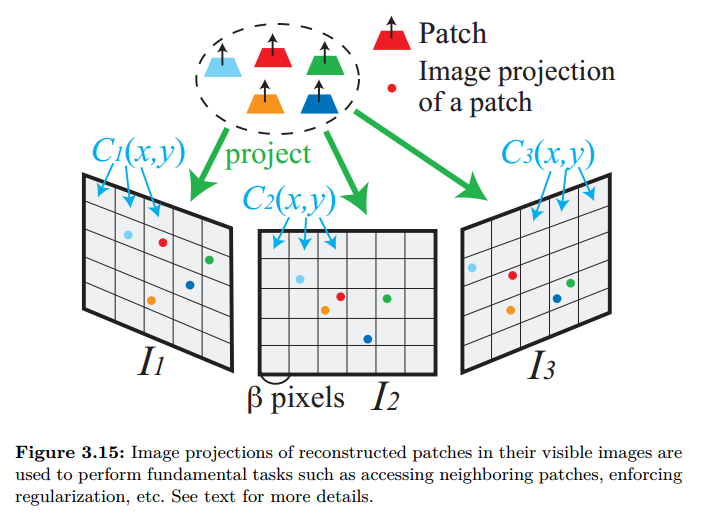
Having defined the photometric consistency measure for a patch as a function of its position and the normal, reconstructing a patch is simply achieved by maximizing the photo-consistency function with respect to those parameters.

Where one parameter for position and two parameters for normal are optimized via a standard non-linear least squares technique.

Image-based Data Structure

Main advantage of the patch based surface representation is its flexibility.

Due to the lack of connectivity information, it is not easy to just search or access neighboring patches, then enforce regularization



Main advantage of the patch based surface representation is its flexibility. However, due to the lack of connectivity information, it is not easy to just search or access neighboring patches, then enforce regularization,

Given a patch p and its visible images V (p), which is estimated as a part of the reconstruction process, they project p into each image in V (p) to identify the corresponding cell. Then, each cell Ci(x, y) remembers the set of patches Qi(x, y) that project into it. Neighboring patches can be collected by looking at neighboring cells in  
the visible images

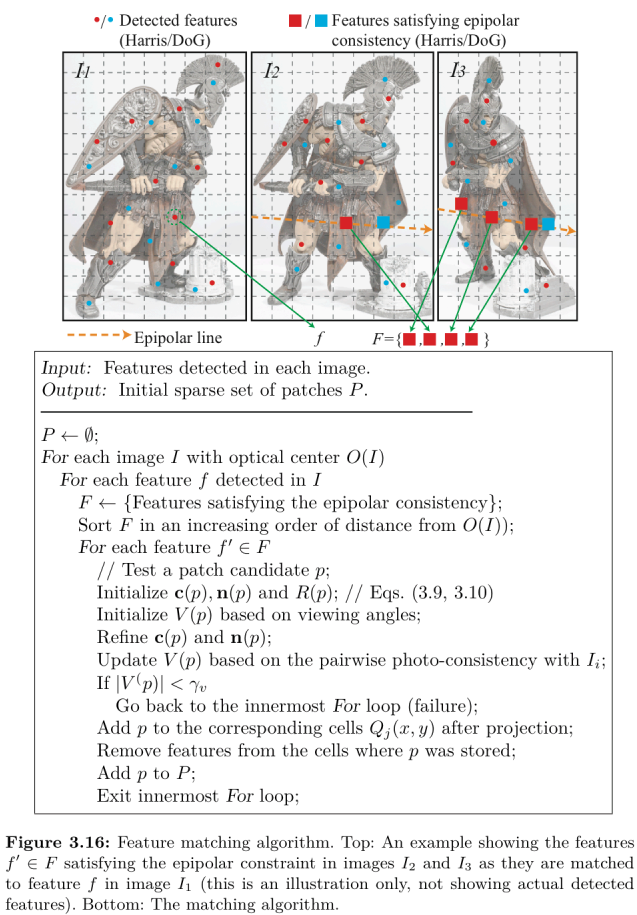
3.2.2 Algorithm

patch-based MVS algorithm attempts to reconstruct at least one patch in every image cell Ci(x, y). It is divided into three steps:

(1) initial feature matching, (2) patch expansion, and (3) patch filtering

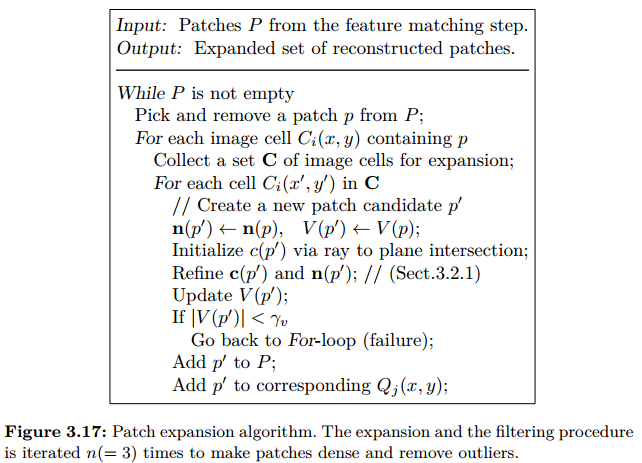
Initial Feature Matching

image Ii with its optical center denoted by O(Ii). For each feature f detected in Ii, they collect in the other images the set F of features f of the same type (Harris or DoG) that lie within two pixels from the corresponding epipolar line, and triangulate the 3D points associated with the pairs (f, f).



Expansion

The goal of the expansion step is to reconstruct at least one patch in every image cell Ci(x, y)



Filtering

First

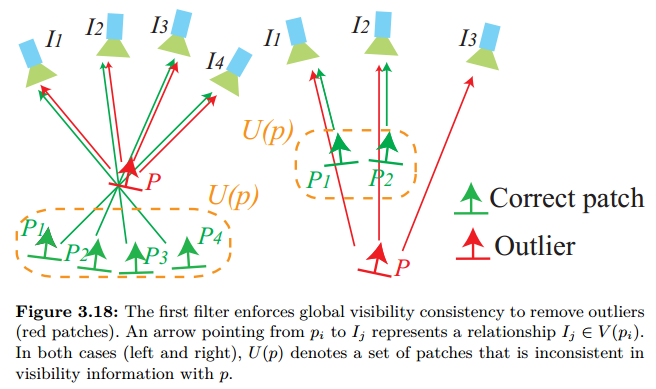
P and p’ are neighbors if their distance along the normals is less than a threshold:

|(c(p) - c(p)) · n(p)| + |(c(p) - c(p)) · n(p)| < γd. (3.13)

γd is the upper-bound on the allowed amount of vertical offset between the two patches.

Second

p and p are not neighbors, but are stored in the same cell of one of the images where p is visible



3.3 Volumetric data fusion

# **2017 September 14 Thursday**

1. Pixelwise View Selection for Unstructured (eccv16)

pipeline:

Notation

l : limit the description to a single image row with l as the column index : depth of a pixel

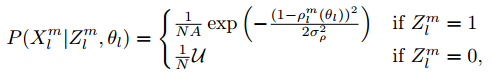
: reference image

: a set of unstructured source images

: the reference patch

: defines the set of non-occluded source images

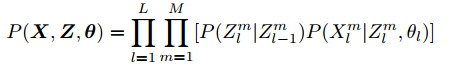
P (|; θl): infer the probability that the reference patch Xlref at depth θl is visible at the source patch Xlm



: describes the color similarity between the reference and source patch based on normalized cross-correlation (NCC)

σρ : determines a soft threshold for on the reference patch being visible in the source image

P() = : The state-transition matrix from the preceding pixel l - 1 to the current pixel l



Use generalized expectation maximization (GEM) algorithm [33] to solve Z

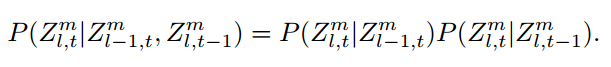
Algorithm

4.1 Normal Estimation

4.2 Geometric Priors for View Selection

4.3 View Selection Smoothness

In this new model, the state of Zl;t m depends not only on the state of its neighboring pixel l - 1 but also on its own state in the previous iteration t – 1



4.4 Photometric Consistency

consequence:

Dataset

South Building dataset [20] , Middlebury benchmark [38] , Strecha benchmark [48] , Fountain

Sequence

South Building dataset [20]

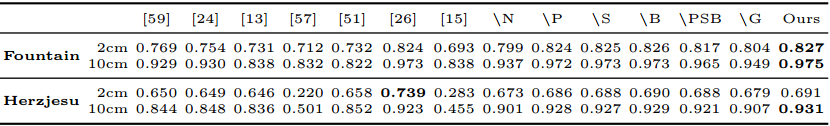
Not release

Middlebury benchmark [38]

Ranking online

Strecha benchmark [48]

None

Fountain

summary:

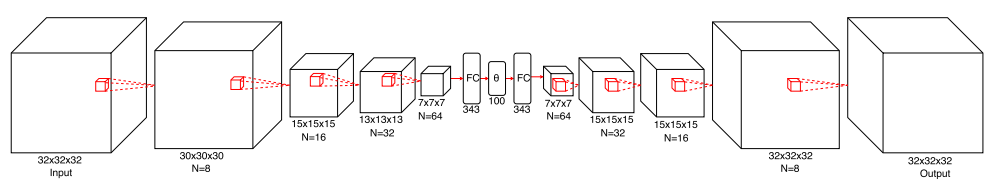
code

# **2017 September 15 Friday**

1. Generative and Discriminative Voxel Modeling with Convolutional Neural Networks (arXiv)

pipeline:

Network



Loss Function

L = -t log(o) - (1 - t) log(1 - o)

two key modifications to the BCE to improve training

……

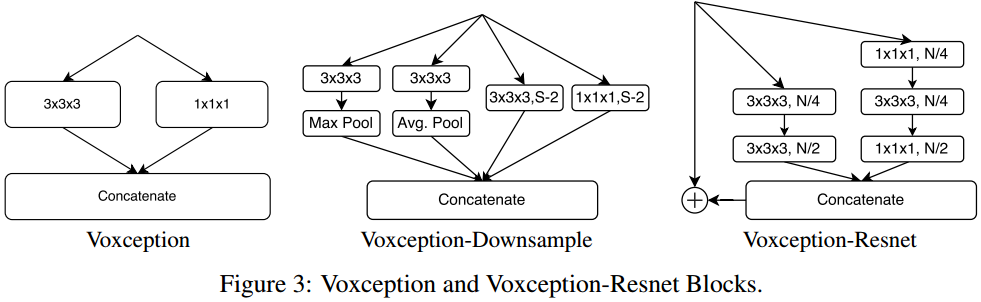
Training

The learning rate is set to 0.0001 for the first epoch, then increased to 0.001

data is augmented by adding random translations and horizontal flips to each training example

Architecture

Voxception



Voxception-ResNet

Not have a good insight

Data Augmentation and Training

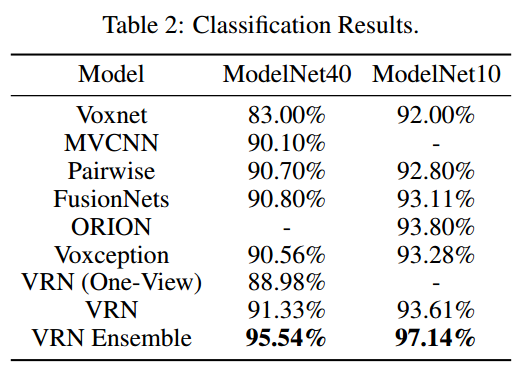
Just a report

consequence:

dataset

Shape Net

Consequence



summary:

code

# **2017 September 16 Saturday**

1. 3D Bounding Box Estimation Using Deep Learning and Geometry Networks (cvpr17)

pipeline:

**Intro**

3D object detection recovers both the 6 DoF pose and the dimensions of an object from an image.

**3D Bounding Box Estimation**

The 3D bounding box is described by its center T = [tx; ty; tz]T , dimensions D = [dx; dy; dz], and orientation R(θ; φ; α)

**Choice of Regression Parameters**

Apart from them, we choose to regress the box dimensions D rather than translation T because the variance of the dimension estimate is typically smaller

**Correspondence Constraints**

Each side of the 2D detection box can correspond to any of the eight corners of the 3D box which results in 84 = 4096 configurations

**CNN Regression of 3D Box Parameters**

MultiBin Orientation Estimation

Fig. 4 shows an example of a car moving in a straight line. Although the global orientation *R*(*θ*) of the car (its 3D bounding box) does not change, its local orientation *θl* with re spect to the ray through the crop center does, and generates changes in the appearance of the cropped image.

*Lθ* = *Lconf* + *w × Lloc*

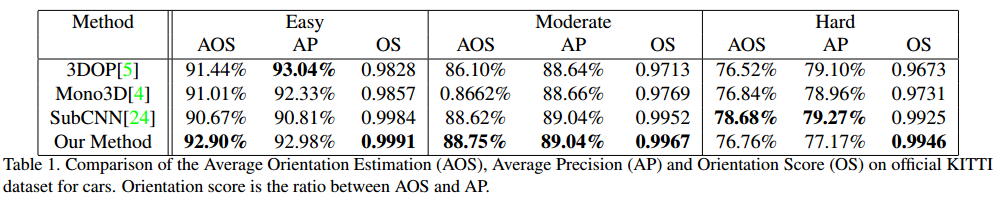
The confidence loss *Lconf* is equal to the softmax loss of the confidences of each bin. *Lloc* is the loss that tries to minimize the difference between the estimated angle and  
the ground truth angle in each of the bins that covers the ground truth angle, with adjacent bins having overlapping coverage. In the localization loss *Lloc*, all the bins that cover the ground truth angle are forced to estimate the correct angle.

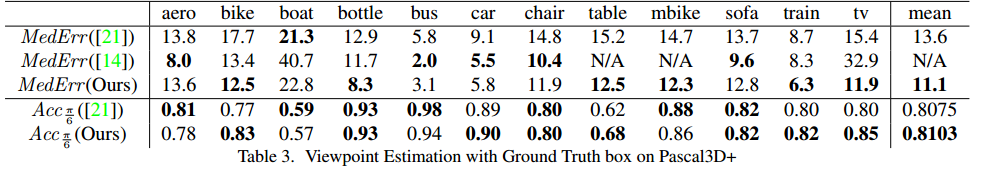
consequence:

**Dataset**

KITTI object detection benchmark [2], Pascal 3D+ dataset[26]

**Sequence**





summary:

learning rate of 0*:*0001.

1. Dominant Set Clustering and Pooling for Multi-View 3D Object Recognition (cvpr17)

pipeline:

Recurrent Clustering and Pooling Layer

A View Similarity Graph

We then construct a view similarity graph G = (V;E;w) where views  
i; j 2 V are distinct nodes and each edge E(i; j) has a weight w(i; j) corresponding to the  
similarity between the views i and j.

similarity between the appearance  
images of views i and j is therefore given by the inner product of the corresponding CNN  
relu feature vectors ri and r j:

w(i; j) = ri ·r j: (1

Dominant Set Clustering

Clustering, Pooling and Recurrence

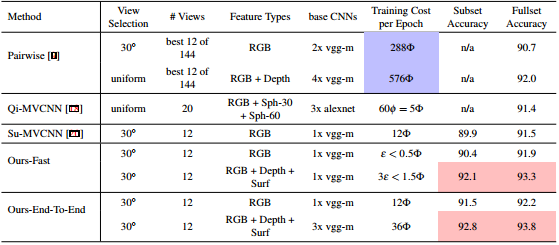
Back propagation

consequence:

Dataset

ShapeNet

Sequence



summary:

code

# **2017 September 17-23 Saturday-Saturday**

ShenZhen / Workshop / Phone Card / Zhong Lianban / Assignment /

# **2017 September 24 Sunday**

1. Multi-View 3D Object Detection Network for Autonomous Driving (cvpr17)

pipeline:

**3.0 Data input introduction**

**3.1 3D Point Cloud Representation**

We propose a more compact representation by projecting 3D point cloud  
to the bird’s eye view and the front view.

**Bird’s Eye View Representation**

discretize the projected point cloud into a 2D grid with resolution of 0.1m. For each cell, the height feature is computed as the maximum height of the points in the cell. intensity feature is the reflectance value of the point which has the maximum height in each cell

intensity and density features are computed for the whole point cloud while the height feature is computed for M slices, thus in total the bird’s eye view map is encoded as (M +2)-channel features.

**Front View Representation**

project it to a cylinder plane to generate a dense front view map as in [17]. Given a 3D point p = (x; y; z), its coordinates pfv = (r; c) in the front view map can be computed using

c = |atan2(y; x) / ∆θ]|

r = |atan2(z; px2 + y2) / ∆φ|

**3.2 3D Proposal Network**

3D box is parameterized by (x; y; z; l; w; h), which are the center and size (in meters) of the 3D box in LIDAR coordinate system

Anchor xy

the corresponding bird’s eye view anchor (xbv; ybv; lbv; wbv), (l; w) of prior boxes takes values in f(3:9; 1:6); (1:0; 0:6)g, and the height h is set to 1.56m. By rotating the bird’s eye view anchors 90 degrees, we obtain N = 4 prior boxes.

With a disretization resolution of 0.1m, object boxes in the bird’s eye view only occupy 5∼40 pixels. Detecting such extra-small objects is still a difficult problem for deep networks

Anchor

Empty

we remove all the empty anchors during both training and testing to reduce computation

Non-empty

the network generates a 3D box. To reduce redundancy, we apply Non-Maximum Suppression (NMS) on the bird’s eye view boxes

**3.3. Region-based Fusion Network**

**Multi-View ROI Pooling**

different resolutions, we employ ROI pooling [10] for each view to obtain feature vectors of the same length

**Deep Fusion**

employ a deep fusion approach, which fuses multi-view features hierarchically

**Oriented 3D Box Regression**

the regression targets are the 8 corners of 3D boxes

the category loss uses cross-entropy and the 3D box loss uses smooth L1.

**Network Regularization**

Not Know

**Network Architecture**

16-layer VGG net with modification

Modification :

Channels are reduced to half of the original network

Handle extra-small objects

**Input Representation.**

consequence:

summary:

# **2017 September 25 Monday**

1. 3D Bounding Box Estimation Using Deep Learning and Geometry Networks (cvpr17) - (twice read)

pipeline:

**Intro**

3D object detection recovers both the 6 DoF pose and the dimensions of an object from an image.

**3D Bounding Box Estimation**

The 3D bounding box is described by its center T = [tx; ty; tz]T , dimensions D = [dx; dy; dz], and orientation R(θ; φ; α)

**Choice of Regression Parameters**

Apart from them, we choose to regress the box dimensions D rather than translation T because the variance of the dimension estimate is typically smaller

**Correspondence Constraints**

Each side of the 2D detection box can correspond to any of the eight corners of the 3D box which results in 84 = 4096 configurations

**CNN Regression of 3D Box Parameters**

**MultiBin Orientation Estimation**

Fig. 4 shows an example of a car moving in a straight line. Although the global orientation *R*(*θ*) of the car (its 3D bounding box) does not change, its local orientation *θl* with re spect to the ray through the crop center does, and generates changes in the appearance of the cropped image.

*Lθ* = *Lconf* + *w × Lloc*

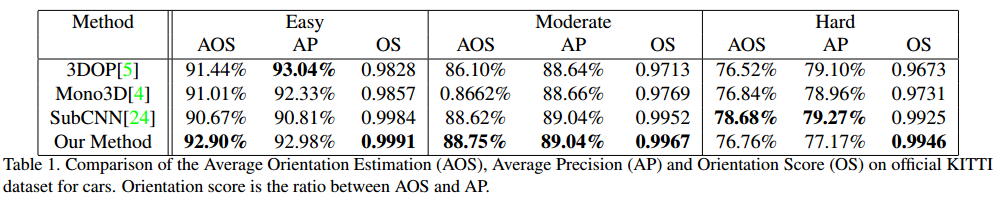
The confidence loss *Lconf* is equal to the softmax loss of the confidences of each bin. *Lloc* is the loss that tries to minimize the difference between the estimated angle and  
the ground truth angle in each of the bins that covers the ground truth angle, with adjacent bins having overlapping coverage. In the localization loss *Lloc*, all the bins that cover the ground truth angle are forced to estimate the correct angle.

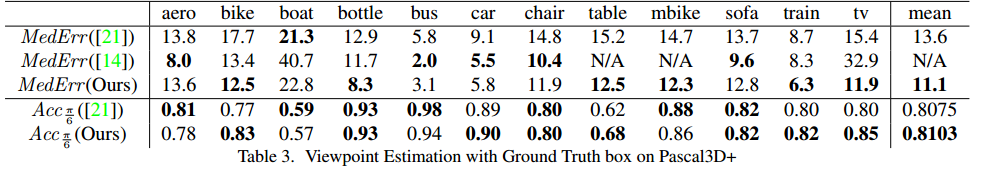
consequence:

**Dataset**

KITTI object detection benchmark [2], Pascal 3D+ dataset[26]

**Sequence**





summary:

learning rate of 0*:*0001.

# **2017 September 26-October 2 Tuesday-Tuesday**

Arxiv / PDC / Chat / GTA / TF & Linux / graph ass1 / theory hw1 / lost overwatch / graph hw2

# **2017 October 4 Wednesday**

1. Face-3D-Reconstruction\_Column

# **2017 October 5-7 Thursday – Saturday**

Wolf / Shen Zhen

# **2017 October 8 Sunday**

1. Face-3D-Reconstruction\_Column

# **2017 October 9 Monday**

1. Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network (arXiv)

pipeline:

consequence:

summary: