Video/Demo

current vision based localisation systems
sparse feature method
dense method
Types

• Geometry based Methods: extract geometric constraints from imagery to estimate motion
  • Sparse feature based methods
  • Direct methods

• Learning based Methods
  • Traditional learning based methods
  • Deep Learning based methods

several decades
everseveral years
excellent accuracy
very popular
rare
Learning based Methods

• Learning based methods aim to derive motion model and infer VO from sensor data by using ML techniques without explicitly applying geometric theory.
  • Traditional learning based methods
  • Deep Learning based methods
Traditional learning based methods

- Sparse optical flow
  - K Nearest Neighbour (KNN)
  - Gaussian Processes (GP)
  - Support Vector Machines (SVM) regression

- difficult to use raw images
  - high-dimensional
  - redundant information
Why Deep Learning

• curse of data
  • high-dimensional: raw images
  • big: sensor data from hundreds of thousands of robots/devices

© Andrew Ng
Hand-Crafted Features by Human

Pervasive Data
- time-series data (vision)
- point cloud

Feature Extraction (hand-crafted)

Inference
- Activities, Context, ...
- Objects, Scene types, ...
- Objects, Structure, Semantics, ...
End-to-End Learning

Pervasive Data
- time-series data
- vision
- point cloud

End-to-End Learning

Inference
- Activities,
- Context, ...
- Objects, Scene
types, ...
- Structure,
- Semantics, ...

automatically learn most effective feature representation to solve the problem
Deep Learning

Training (supervised learning)

Testing

- big data
- test data
- trained DNN
- prediction
- error
- data label
- label

forward
backward

(supervised learning)
Convolutional Neural Network - CNN

• recap previous courses
• similar to feed-forward Neural Networks

• Merits:
  • more efficient to implement
  • vastly reduce the amount of parameters

• Important Ideas:
  • local connectivity
  • parameter sharing

• Main Type of Layers:
  • convolutional layer
  • activation layer
  • pooling layer
CNN Layers

- Local Connectivity
- Parameter Sharing

courtesy of Andrej Karpathy
Recurrent Neural Network - RNN

- temporal processing and learn sequences
- sequential data: voice, text, language, video, sensor data, etc.
- flexible configurations: one to one, one to many, many to many

- memory

courtesy of Andrej Karpathy
Recurrent Neural Network - RNN

- loops along time: memory to save information
- unfold over time
- dependence and connection
- Backpropagation Through Time (BPTT)

Image courtesy of Christopher Olah
Simple Recurrent Neural Networks

- a sequence of \( x \) at every time step

\[
h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)
\]

\[
y_t = W_{hy} h_t
\]

image courtesy of Christopher Olah
Example: Language Model

- vocabulary: [h, e, l, o]
- input: one-hot encoding

\[ h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \]

courtesy of Andrej Karpathy
Long Short-Term Memory - LSTM

- LSTM to avoid long-term dependency problem
- gates: control information flow, cell states
- forget gate: “remember” or “forget” states

Image courtesy of Christopher Olah
LSTM Cell: Step by Step

\[ i_k = \sigma(W_{li}I_k + W_{hi}h_{k-1} + b_i) \]
\[ f_k = \sigma(W_{lf}I_k + W_{hf}h_{k-1} + b_f) \]
\[ g_k = \tanh(W_{lg}I_k + W_{hg}h_{k-1} + b_g) \]
\[ c_k = f_k \odot c_{k-1} + i_k \odot g_k \]
\[ o_k = \sigma(W_{lo}I_k + W_{ho}h_{k-1} + b_o) \]
\[ h_k = o_k \odot \tanh(c_k) \]
Bidirectional-LSTM

- forward and backward layers
- future information is reachable from current state
- output can access information from past and future states
LSTM Variants

Gated Recurrent Units (GRUs)


eight LSTM variants


1. No Input Gate (NIG)
2. No Forget Gate (NFG)
3. No Output Gate (NOG)
4. No Input Activation Function (NIAF)
5. No Output Activation Function (NOAF)
6. No Peepholes (NP)
7. Coupled Input and Forget Gate (CIFG)
8. Full Gate Recurrence (FGR)

\[
MUT1:
\begin{align*}
z &= \text{sigmoid}(W_{xz}x_t + b_z) \\
r &= \text{sigmoid}(W_{xr}x_t + W_{hr}h_t + b_r) \\
h_{t+1} &= \text{tanh}(W_{hh}(r \odot h_t) + \text{tanh}(x_t) + b_h) \odot z \\
&+ \ h_t \odot (1 - z)
\end{align*}
\]

\[
MUT2:
\begin{align*}
z &= \text{sigmoid}(W_{xz}x_t + W_{hz}h_t + b_z) \\
r &= \text{sigmoid}(x_t + W_{hr}h_t + b_r) \\
h_{t+1} &= \text{tanh}(W_{hh}(r \odot h_t) + W_{zh}x_t + b_h) \odot z \\
&+ \ h_t \odot (1 - z)
\end{align*}
\]

\[
MUT3:
\begin{align*}
z &= \text{sigmoid}(W_{xz}x_t + W_{hz} \tanh(h_t) + b_z) \\
r &= \text{sigmoid}(W_{xr}x_t + W_{hr}h_t + b_r) \\
h_{t+1} &= \text{tanh}(W_{hh}(r \odot h_t) + W_{zh}x_t + b_h) \odot z \\
&+ \ h_t \odot (1 - z)
\end{align*}
\]
Deep RNN

- multiple layers: high level features
- more complex connections between inputs and features
End-to-End Visual Odometry
Architecture

• achieve monocular Visual Odometry (VO) in an end-to-end, sequence-to-sequence manner based on Deep Learning, i.e., directly estimating poses from a sequence of raw RGB images

• leverage a large number of images

• feature extraction + sequential learning
CNN based Feature Extraction

- Convolutional Neural Network (CNN)
- 2 consecutive images as input
- feature extraction on raw RGB images
- generalise to new environments, NOT confined to trained ones
- geometric feature representation
RNN based Sequential Learning

• sequential dependence and motion dynamics of an image sequence are automatically learnt by Recurrent Neural Network (RNN)

• convolutional feature is passed to RNN for sequential modelling

• LSTM as RNN

• multiple layers of LSTM as deep structure
Novel SE(3) Layer

- Special Euclidean Group SE(3) for transformation
- Pose composition: matrix multiplication in the context of SE(3)
- No hyper-parameter to be learned
- Direct pose feedback

\[ T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \]

\[ T_{WC_k} = T_{WC_{k-1}} T_{C_{k-1} C_k} \]
Experiments

- Testing scenarios: outdoor driving, indoor Micro Aerial Vehicle, indoor pedestrian, outdoor fixed-wing UAV

- Comparison (VO version):
  - VISO2: sparse feature based VO
  - ORB-SLAM: state-of-the-art sparse feature based visual SLAM
  - LSD-SLAM: state-of-the-art direct method based visual SLAM

- Scale recovery:
  - VISO2: fixed camera height
  - ORB-SLAM and LSD-SLAM: similarity transformation to ground truth
  - Deep-VO: automatic
Outdoor Driving

- KITTI VO/SLAM Benchmark and Raw Dataset
- Trained on
  - 5 of VO training sequences: test on other training sequences
  - all training sequences: test on testing and raw sequences
- Results:
  - monocular VISO2: big drift
  - ORB-SLAM: scale drift without loop closure detection
  - LSD-SLAM: lose tracking (low frame rate and high speed)
Indoor Flying

• ETH EuRoC dataset: indoor Micro Aerial Vehicle
• quadrotor in an indoor large machine hall
• trained on 4 sequences of Machine Hall and tested on another one
Indoor Motion: Test in Office Building

- Training data: Wolfson Building of Department of Computer Science
- Test:
  - Robert Hooke Building of Department of Computer Science
  - Natural History Museum
- Geometry based methods
  - prone to lose tracking
  - texture-less: corridor, white wall
  - agile motion: fast turn

ORB-SLAM: lose tracking

Deep-VO

Ground Truth – Tested VO

Sensor and Actuator Networks - AIMS CDT - Sensor Networks Group
Indoor Motion: Test in Museum

- Natural History Museum
  - Challenging lighting conditions
- Café with walking people
- Deep-VO: see motion in video

ORB-SLAM: scale problem

LSD-SLAM: lose tracking
Supermarket

Demo
Deep Learning based VIO

• Visual Inertial Odometry: visual and inertial sensors
• multi-rate Recurrent Neural Network
• robust to
  • extrinsic calibration errors
  • time synchronisation errors

End-to-End Global Localisation
Global Localisation

• global localisation:
  • not only place recognition
  • 6 DoF pose in the environments, maps

• loop closure detection for SLAM

• re-localisation: recovery after losing tracking
Perceptual Aliasing

• different locations have similar appearance
• same location has different appearances
• how to distinguish, then localise?

Images from RobotCar Dataset
Methods
End-to-End Global Localisation

- learn most effective features
- spatial and **temporal** models
- a sequence of images (video)

\[ \text{Time} \]

[CVPR'17] “6-DoF video-clip re-localisation.”
Image Features: CNN

- images -> features
- extract features by CNN
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNet
  - .......

![Diagram of CNN processing](image-url)
Temporal Modelling: Bidirectional RNN

• a sequence of image features -> temporal model
• video clip:
  • dynamics, motion
  • robot, car, pedestrian, etc.
• bi-directional LSTM
  • connection between steps
Global Poses

- a pose for each image -> sequence
- DNN model
  - “compressed map”
  - feature matching
  - loop closure detection
  - pose estimation
Test on RobotCar Dataset

without with temporal modelling
Pros and Cons?

- geometry based methods
- learning based methods