Motion-coherent Affinities for Hypergraph based Motion Segmentation

CAIP 2017

Kai Cordes, Christopherus Ray'onaldo, and Hellward Broszio

VISCODA GmbH

22. Aug. 2017

VISCODA

VISCODA GmbH

- Foundation: August 2011
- Small to medium sized enterprise

Key Technologies

- Strong background in Computer Vision
- Structure from Motion (SfM)
- Real-time processing

→ Software development / series production VW/AUDI: ADAS (*zFAS*)
  - Real-time structure from motion
  - Moving obstacle detection

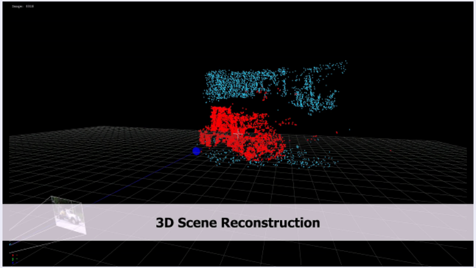
→ EU Project 5GCAR

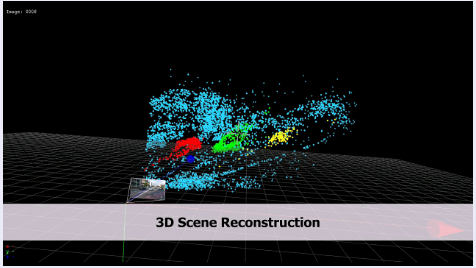
→ ...

Motivation

Scene Reconstruction including Moving Objects

- Motion Segmentation using feature trajectores
- *Multiple* Structure from Motion
- Composition of scenes

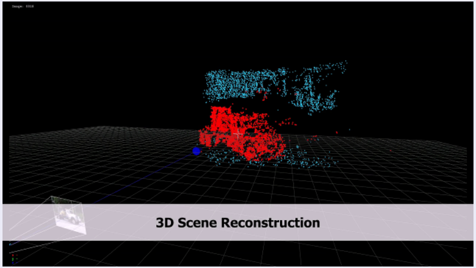


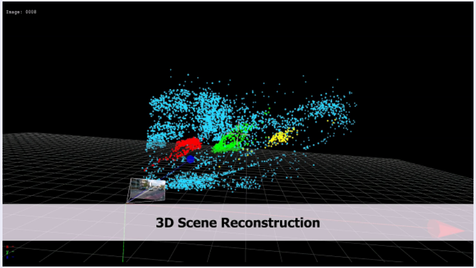


Motivation

Scene Reconstruction including Moving Objects

- Motion Segmentation using feature trajectores
- *Multiple* Structure from Motion
- Composition of scenes





Outline

Motion Segmentation

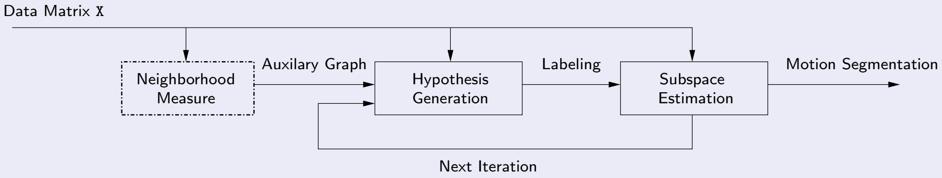
Goals:

- Classify trajectories to different motions
- Subspace estimation based on hypothesis generation and validation: (*sampling*)
- Hypergraphs incorporate higher order similarities
- Pulak et al. (ECCV 2014): large hyperedge degrees beneficial

Hopkins155 Benchmark

- 155 sequences
- Trajectories of features visible in all frames
- Ground truth annotations

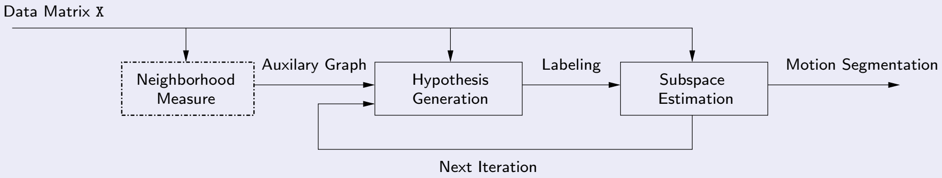
Reference Method (Pulak et al. ECCV 2014)



Auxiliary Graph: Guides sampling

Hypothesis Generation: Swedsen-Wang sampling using Random Cluster Model (RCM)

Subspace Estimation: Compute motion models and measure subspace error

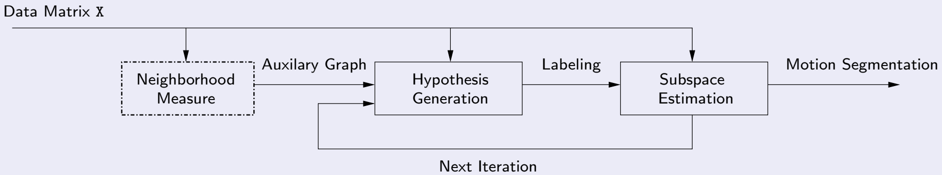


Auxiliary Graph

- Neighborhood structure: Graph G

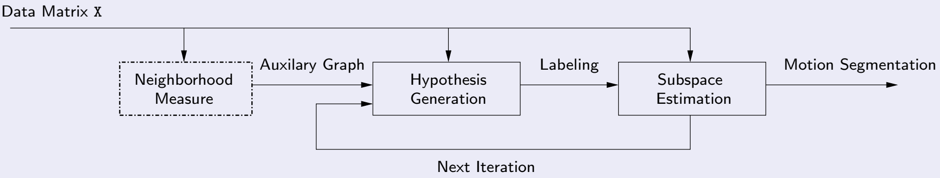
Neighborhood Measure: Spatial Proximity

- k-nearest neighbor algorithm (k=3)
- Vertices v_i, v_j in G are connected if v_i is a k-nearest neighbor of v_j



Hypothesis Generation

- Uses auxiliary graph
- Random Cluster Model / Swedsen-Wang sampling
  - Edge probabilities based on neighborhood distances
  - Bond variable switches edges on / off
  - Condition: random number $<$ edge weight



Subspace Estimation

- Compute motion models of the current labeling
- Determine subspace error

Auxiliary Graph

- Increases probability to generate a clean sample
- Reduces required number of samples

Hopkins155 Example

- Nicely separated subgraphs
- No edges between different motion models

- Usual case in Hopkins155 benchmark
- Provides only trajectories which are visible in every frame
- Short trajectories removed

Auxiliary Graph

- Increases probability to generate a clean sample
- Reduces required number of samples

MTPV Example

- MTPV more challenging
- Benchmark includes short trajectories as well

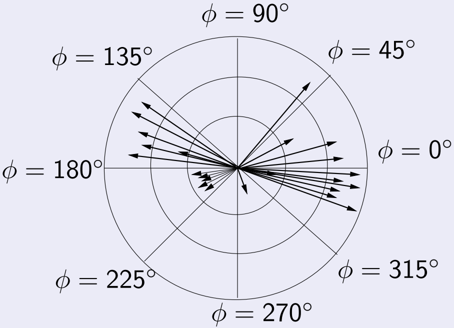
- Spatial distance not a good measure for building auxiliary graph

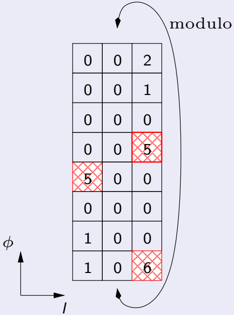
$\Rightarrow$ Auxiliary graph less beneficial

Outline

Design New Auxiliary Graph

- Incorporation of trajectory motion information
- Use two-dimensional motion histogram:
  - Avoid edges between different local maxima
  - Use spatial proximity for data points in a dominant motion





Design New Auxiliary Graph

- Incorporation of trajectory motion information
- Use two-dimensional motion histogram:
  - Avoid edges between different local maxima
  - Use spatial proximity for data points in a dominant motion

Example

Outline

- Segmentation error: wrongly classified trajectories
- Results given for different trajectory lengths f_s (generate subsequences)

Segmentation Error [%] on Benchmark Hopkins155

	Mean		Median	
f_s	reference	proposed	reference	proposed
3	7.96	**7.03**	0.84	**0.67**
4	5.76	**5.32**	**0.0**	**0.0**
5	4.94	**4.71**	**0.0**	**0.0**
6	4.64	**4.36**	**0.0**	**0.0**
7	3.79	**3.61**	**0.0**	**0.0**
8	4.03	**3.58**	**0.0**	**0.0**
9	3.76	**3.75**	**0.0**	**0.0**
10	3.87	**3.83**	**0.0**	**0.0**

- Segmentation error: wrongly classified trajectories
- Results given for different trajectory lengths f_s (generate subsequences)

Segmentation Error [%] on Benchmark MTPV

	Mean		Median	
f_s	reference	proposed	reference	proposed
3	26.96	**21.06**	32.27	**22.60**
4	21.30	**14.20**	22.83	**0.51**
5	19.91	**12.69**	17.87	**0.29**
6	18.27	**12.60**	9.74	**0.30**
7	19.01	**11.09**	12.30	**0.23**
8	18.66	**12.09**	10.75	**0.25**
9	17.46	**11.18**	1.21	**0.24**
10	16.83	**11.16**	1.87	**0.24**

Mean Computation Time in Seconds

	Mean		Median	
Benchmark	reference	proposed	reference	proposed
Hopkins	8.83s	**2.49s**	5.53s	**2.12s**
MTPV	7.98s	**4.30s**	4.26s	**2.75s**

- Lower graph complexity leads to smaller computation times

Outline

Analysis: Hypergraph-based Motion Segmentation

- State of the art approach (Pulak et al. ECCV 2014) provides general solution
- Good performance on Hopkins155
  - Spatial proximity used in auxiliary graph good choice
- Decreased performance on MTPV benchmark
  - Short trajectories $\rightarrow$ spatial proximity less beneficial

Proposal: Motion-Coherent Affinities

- Improves auxiliary graph

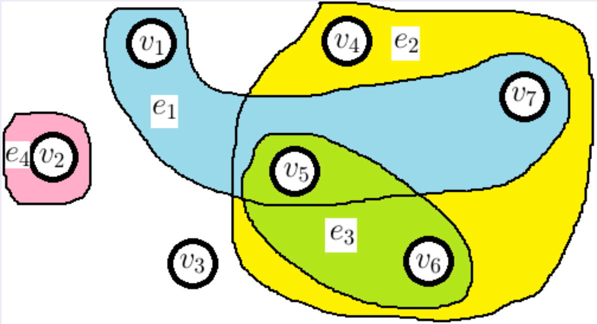
$\rightarrow$ Similar performance on Hopkin155

$\rightarrow$ Significantly improved performance on MTPV (up to 40%)

$\rightarrow$ Decreased computation time

VISCODA

Hypergraph

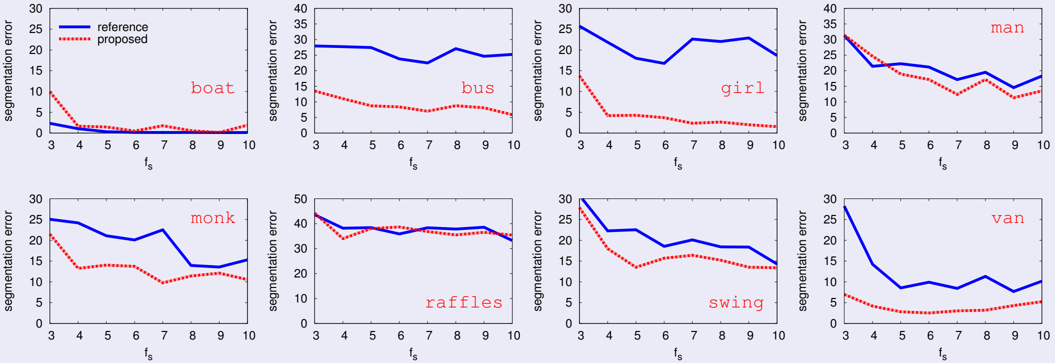


Vertices #	Degree
1	1
2	1
3	0
4	1
5	3
6	2
7	2

Edge #	Degree
1	3
2	4
3	2
4	1

$$H = (V, E)$$
$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$
$$E = \{e_1, e_2, e_3, e_4\}$$
$$= \{\{v_1, v_5, v_7\}, \{v_4, v_5, v_6, v_7\}, \{v_5, v_6\}, \{v_2\}\}$$

Mean Segmentation Error [%] on MTPV Benchmark



Median of Segmentation Error [%] on MTPV Benchmark