One-shot-learning gesture recognition using 3D MoSIFT and bag of features

Jun Wan

Institute of Information Science

Beijing Jiaotong University, China

joewan10@gmail.com

Shuang Deng

China machinery TDI international engineering co., ltd

daisy_shuang@hotmail.com

Outline

- Background
  - Challenge problems for one-shot learning gestures recognition
  - Bag of features model
- Our approach
- Results
- Discussion and feature works
- Reference

Background

- Challenge problems for one-shot learning gestures recognition
  - One training sample per gesture class

Background

- Challenge problems for one-shot learning gestures recognition
  - One training sample per gesture class
  - How to extract distinctive features?

Background

- Challenge problems for one-shot learning gestures recognition
  - One training sample per gesture class
  - How to extract distinctive features?
    - motion trajectory
    - spatio-temporal features: Cuboid , Harri3D + HOG/HOF, MoSIFT, 3D MoSIFT

Background

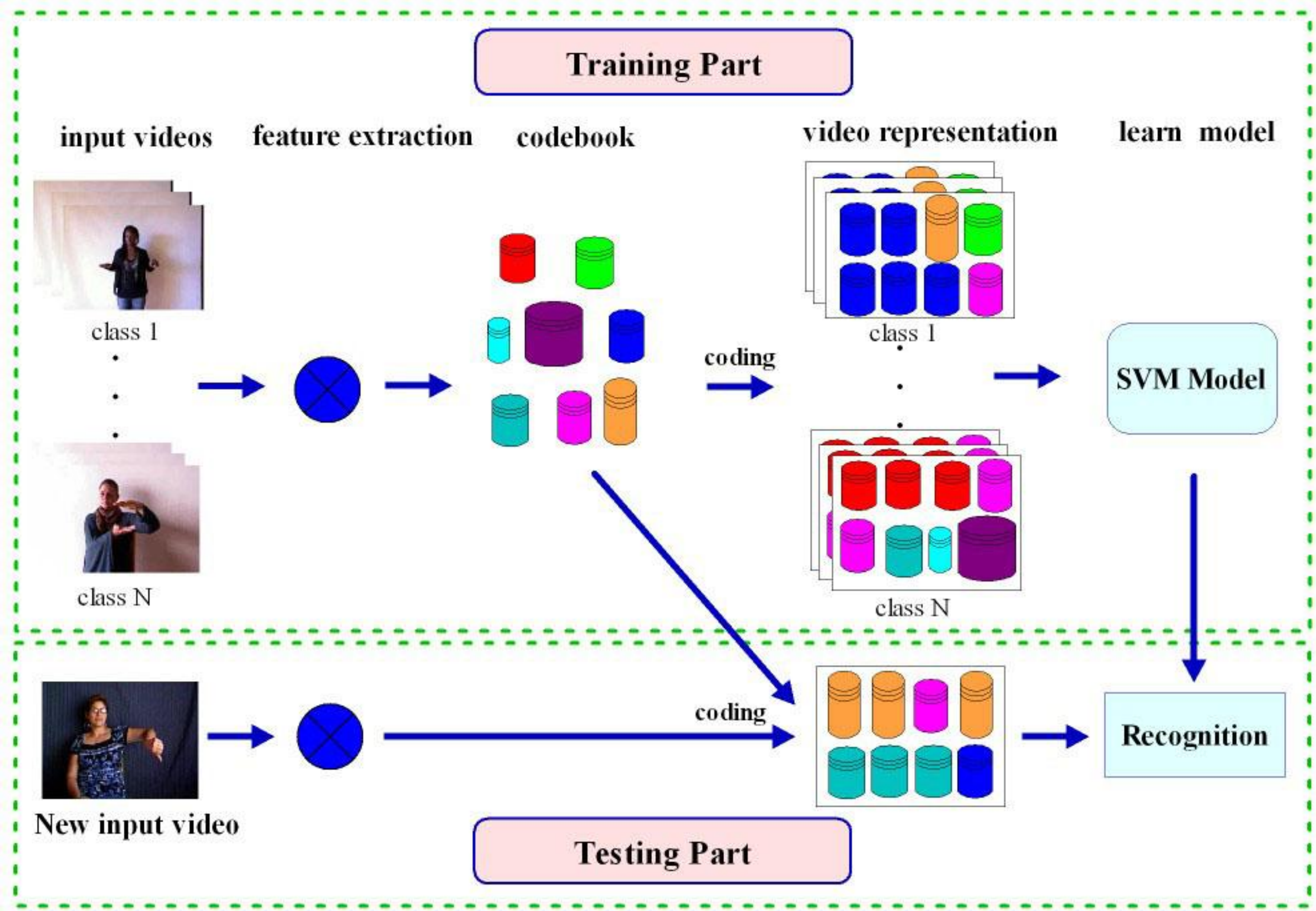
- Challenge problems for one-shot learning gestures recognition
  - One training sample per gesture class
  - How to extract distinctive features?
    - motion trajectory
    - spatio-temporal features: Cuboid , Harri3D + HOG/HOF, MoSIFT, 3D MoSIFT
  - How to select a suitable model?

Background

- Challenge problems for one-shot learning gestures recognition
  - One training sample per gesture class
  - How to extract distinctive features?
    - motion trajectory
    - spatio-temporal features: Cuboid , Harri3D + HOG/HOF, MoSIFT, 3D MoSIFT
  - How to select a suitable model?
    - Hidden Markov Model
    - Conditional Random Field
    - Bag of features Model

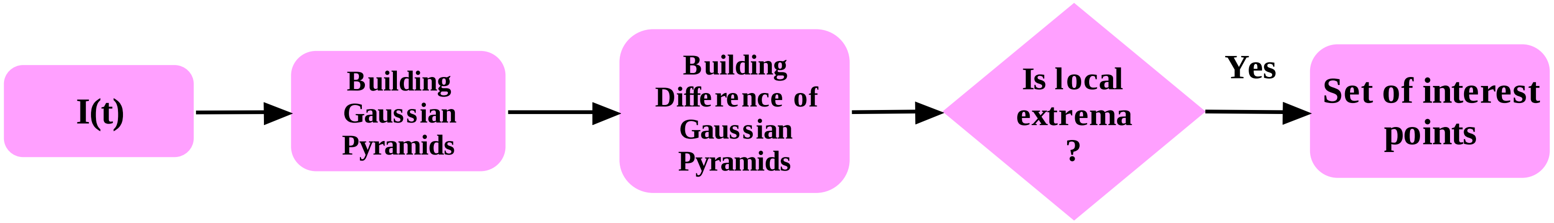
Background

- Bag of features model



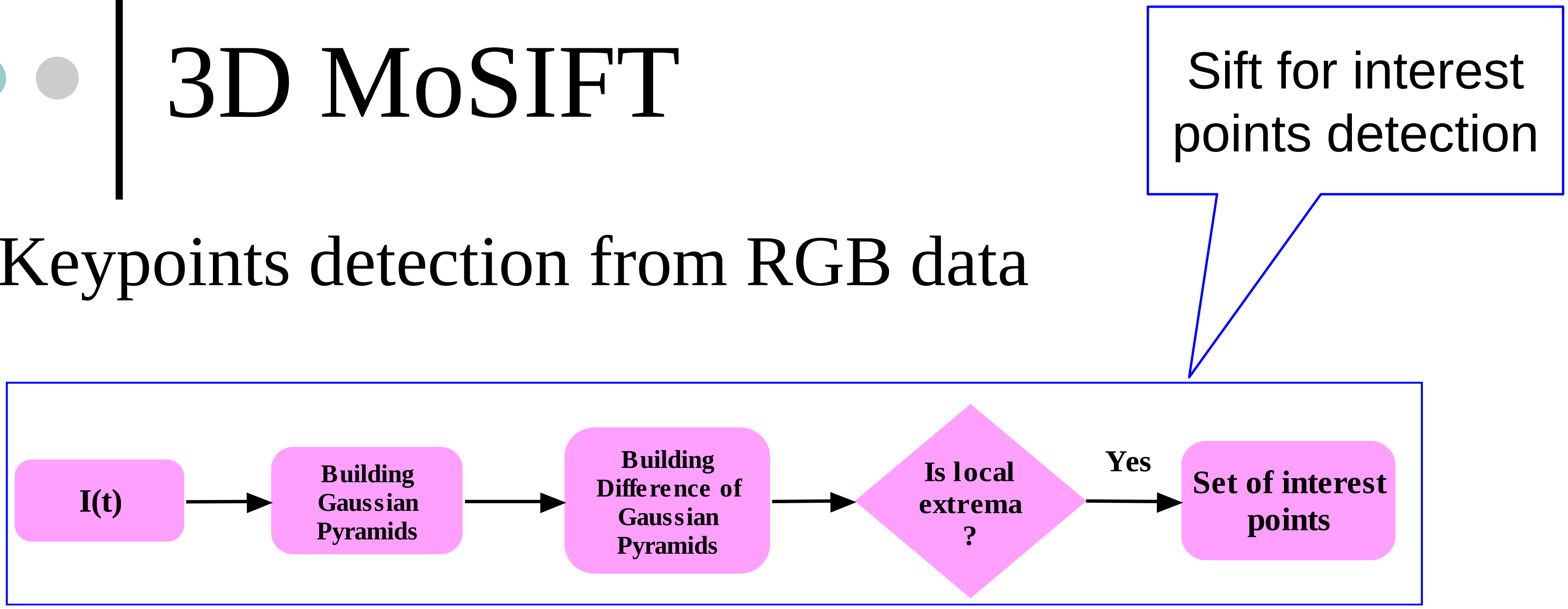
3D MoSIFT

- Keypoints detection from RGB data



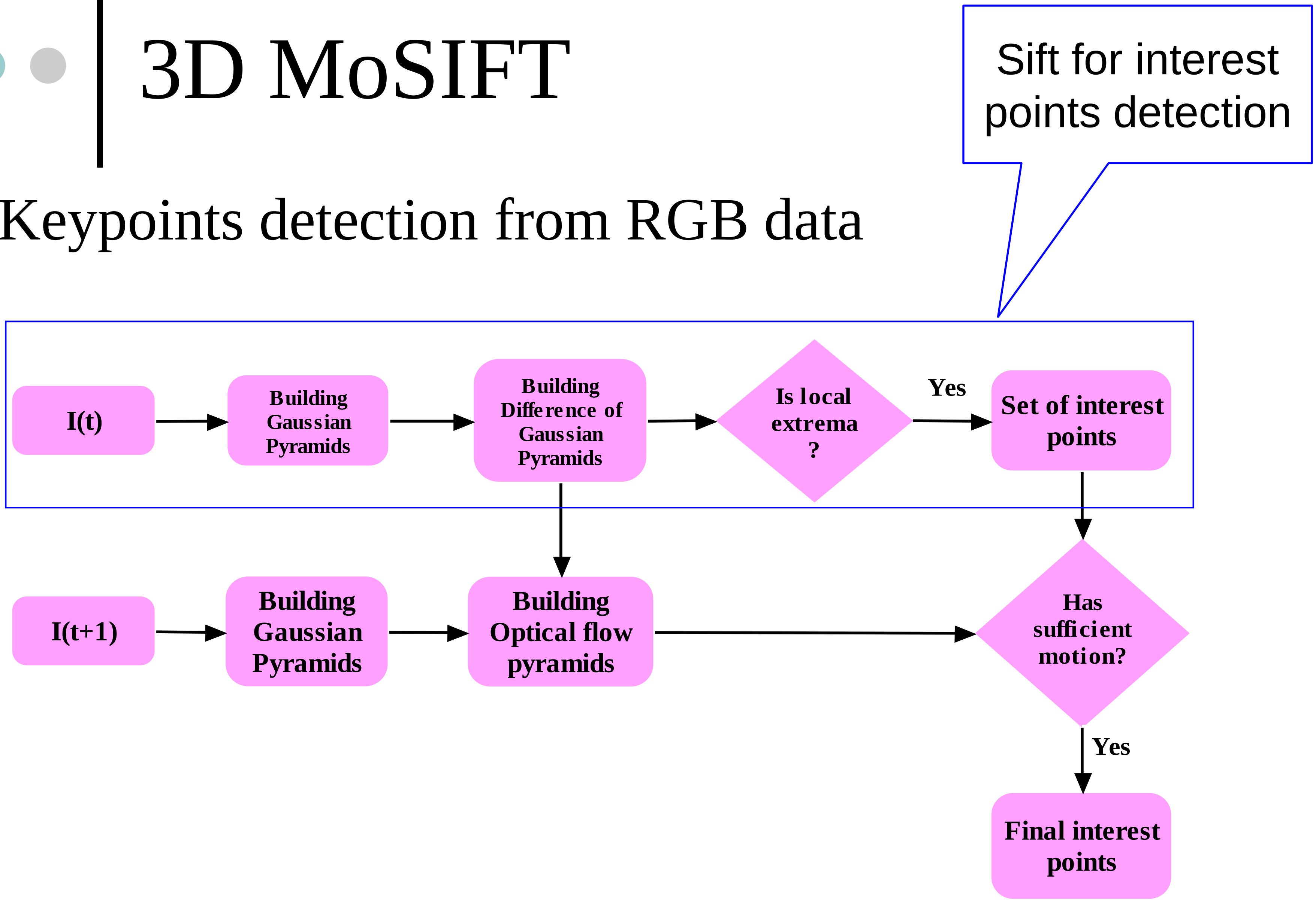
3D MoSIFT

- Keypoints detection from RGB data



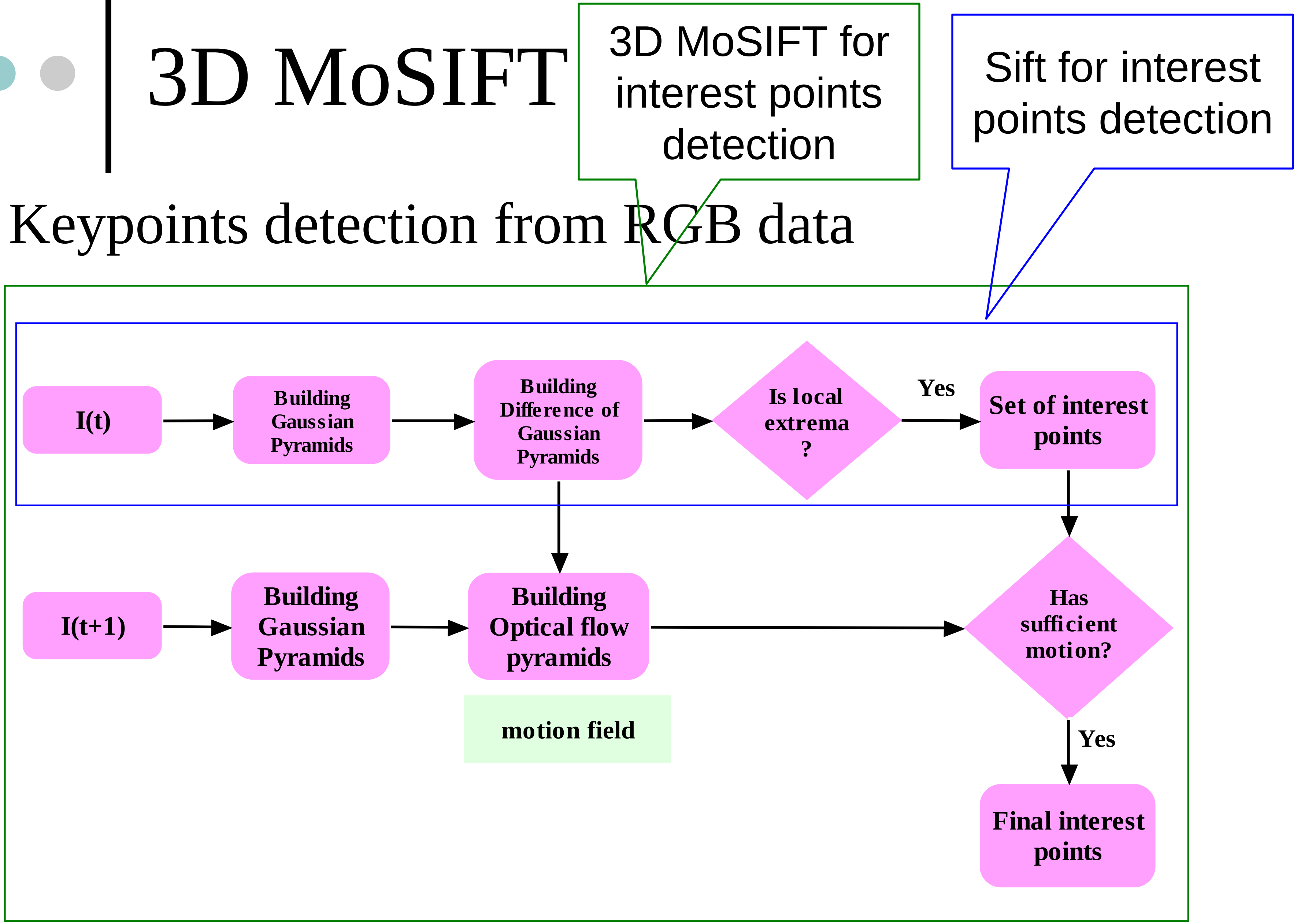
3D MoSIFT

- Keypoints detection from RGB data



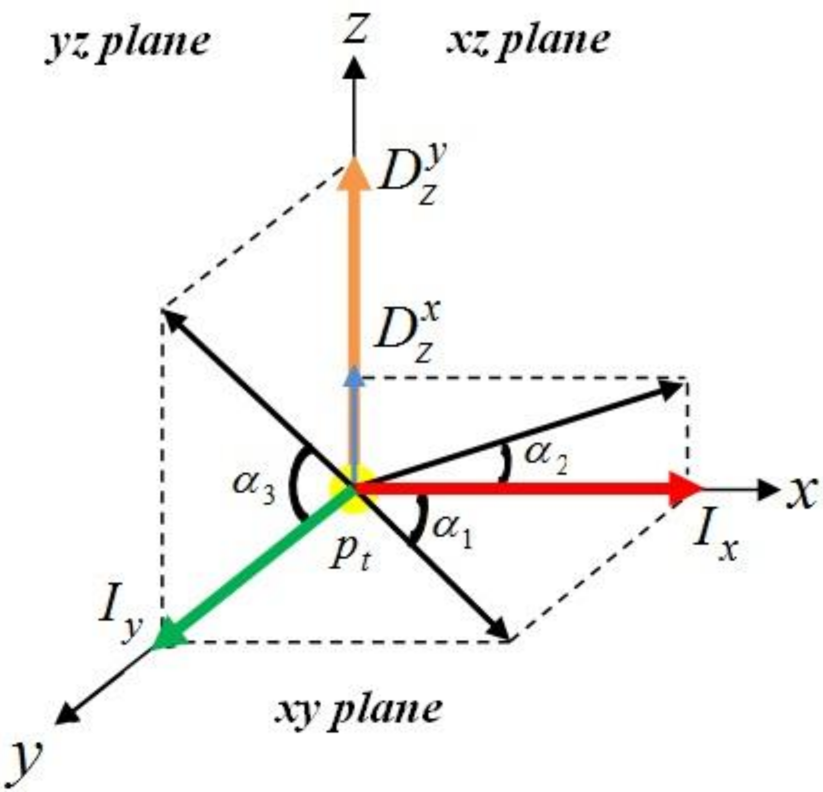
3D MoSIFT

- Keypoints detection from RGB data



3D MoSIFT

- Computing feature descriptors from RGB-D data
  - 3D Gradient space construction



3D Gradient Space

$$I_x = \nabla_x(I) = \frac{\partial I}{\partial x}$$

$$I_y = \nabla_y(I) = \frac{\partial I}{\partial y}$$

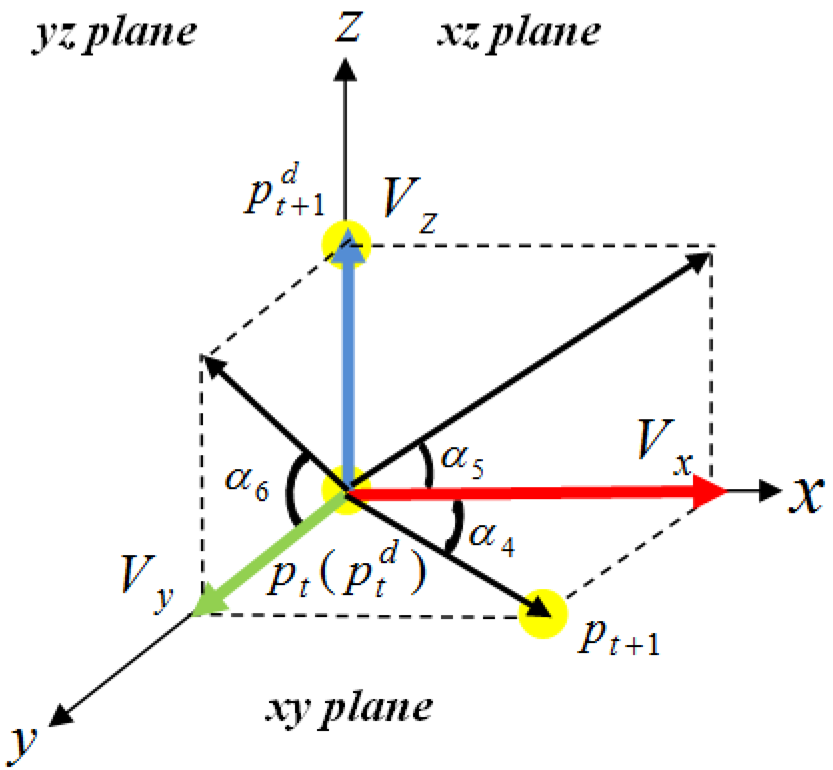
$$D_z^x = \nabla_x(D) = \frac{\partial D}{\partial x}$$

$$D_z^y = \nabla_y(D) = \frac{\partial D}{\partial y}$$

where $\frac{\partial(\cdot)}{\partial x}$ and $\frac{\partial(\cdot)}{\partial y}$ are the gradient in the x and y direction, respectively.

3D MoSIFT

- Computing feature descriptors from RGB-D data
  - 3D Motion space construction



3D Motion Space

V_x and V_y are calculated by optical flow.

$$V_z = D_{t+1}(p^d_{t+1}) - D_t(p^d_t)$$

where D_t deontes the depth frame at time t; p^d_t is the interest point at time t;

3D MoSIFT

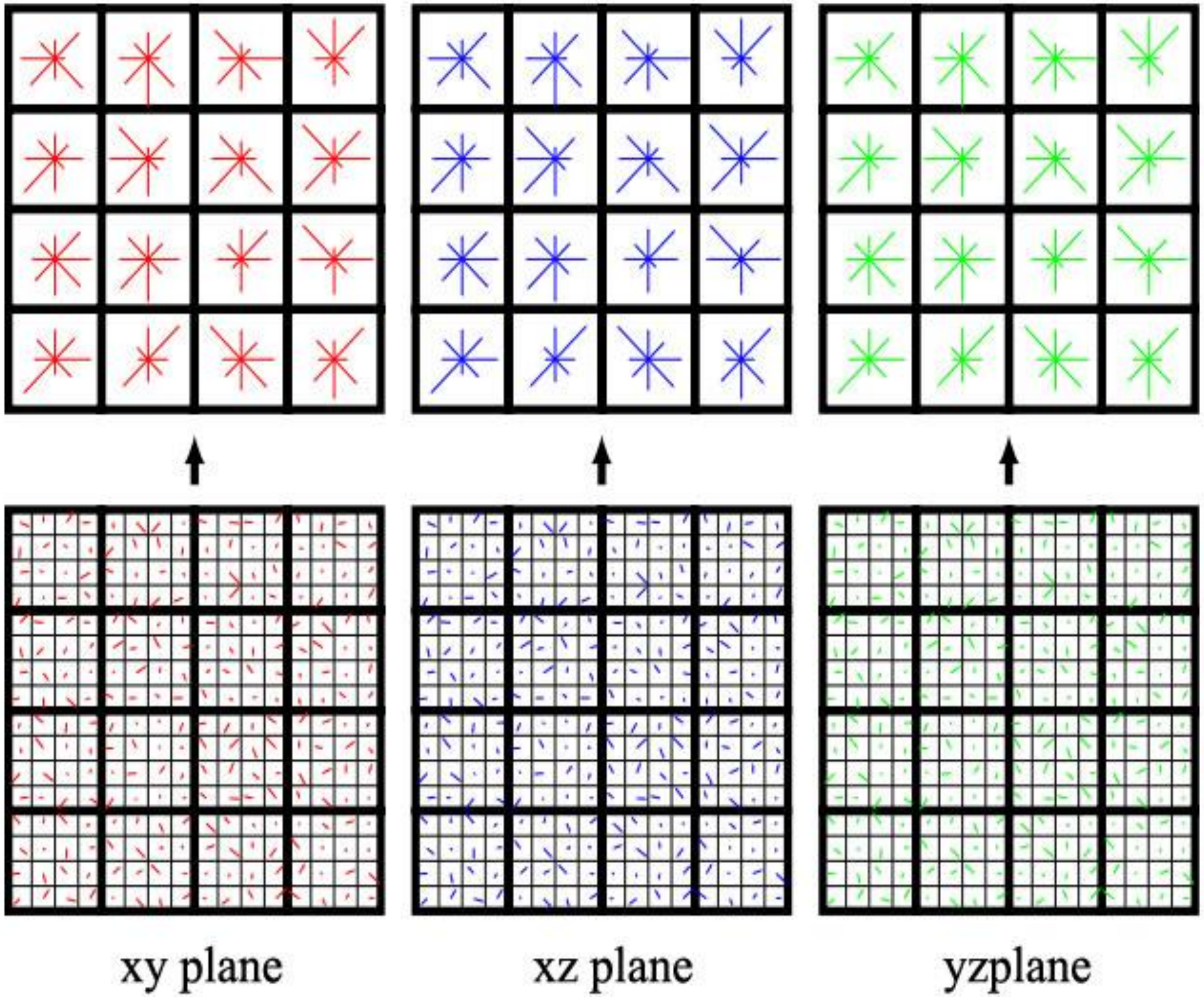
*Local patch around a interest point :16*16;*

*Each Grid size: 4*4;*

Calculate orientation histogram in each grind: 8 bins;

*Descriptors for a 2D plane: 4*4*8 = 128;*

*3D MoSIFT size: 128*6 = 768*



Bag of features

- Codebook learning and vector quantization (VQ) coding
  - Codebook learning (in the training stage)

------ by K-means algorithm

Let X be the set of descriptors with D dimensional feature space, $X = [x_1, x_2, ..., x_N] \in \mathbb{R}^{D\times N}$. Suppose the codebook B with M entries, $B = [b_1, b_2, ..., b_M] \in \mathbb{R}^{D\times M}$, the set of codes for X can be denoted as $C = [c_1, c_2, ..., c_N] \in \mathbb{R}^{M\times N}$

Then, K-means algorithm is solving the problems to seek the optimal B and C:

$$\min_{B,C} \|X - BC\|_F^2 \qquad s.t. \ \|c_i\|_0 = 1, \|c_i\|_1 = 1, c_i \geq 0, \forall i \quad (1)$$

Bag of features

- Codebook learning and vector quantization (VQ) coding
  - VQ coding (in the testing stage)

Let Y be the set of descriptors with D dimensional feature space, $Y = [y_1, y_2, ..., y_N] \in \mathbb{R}^{D \times N}$. Given the codebook B with M entries, $B = [b_1, b_2, ..., b_M] \in \mathbb{R}^{D \times M}$. Then, VQ method is solving the problems to seek the optimal C:

$$\min_C \|Y - BC\|_F^2 \quad s.t. \ \|c_i\|_0 = 1, \|c_i\|_1 = 1, c_i \geq 0, \forall i \quad (2)$$

In practice, the single non-zero element of c_i is found by searching the nearest neighbor.

Video representations

After obtaining codebook $B = [b_1, b_2, ..., b_M] \in \mathbb{R}^{D \times M}$ and a set of descriptors $Y = [y_1, y_2, ..., y_N] \in \mathbb{R}^{D \times N}$ from a sample, Then the set of codes $C = [c_1, c_2, ..., c_N] \in \mathbb{R}^{M \times N}$ for Y can be calculated via VQ method.

- Histogram calculation:

$$h = \frac{1}{N} \sum_{i=1}^{N} c_i$$

So, each sample can be denoted by a histogram vector $h \in \mathbb{R}^M$

Classification

- KNN classification

Because there is only one sample per gesture class, KNN classification is used to recognize any input sample.

Overview

Algorithm 2 The unified framework for one-shot learning gesture recognition

The condition for one-shot learning: given K training samples (RGB-D data) for K class (each sample per class).

Input:

Training samples (RGB-D data): $T_r = [t_{r1}, ..., t_{rK}]$

A learned codebook: B (compute from training stage)

Histogram vectors of training samples: $H_r = [h_{r1}, h_{r2}, ..., h_{rK}]$ (compute from training stage)

A test sample (RGB-D data): t_e

Output:

The recognition results: $class$

```
Initialization: class = [ ]
Temporal gesture segmentation: [t_{e_1}, t_{e_2}, ..., t_{e_N}] = DTW(T_r, t_e)
for i = 1; i ≤ N; i++ do
    Spatio-temporal feature extraction: X_{t_e} = 3D_MoSIFT(t_{e_i})
    For X_{t_e}, calculate the codes C over the pre-trained codebook B
    min_C ||X_{t_e} - BC||_F^2    s.t.  ||c_j||_0 = 1, ||c_j||_1 = 1, c_j ⩾ 0, ∀ j
    Calculate the histogram vector h_{t_e}
    Recognization: tmp_calss = knn_classify(H_r, h_{t_e})
    class = [class tmp_calss]
end for
return class
```

Results

- Features analysis and Parameter discussion

Batch name	**N_{tr}**	**L_{tr}**	**A_{tr}**
final21	10	18116	1812
final22	11	19034	1730
final23	12	11168	931
final24	9	10544	1172
final25	11	**8547**	**777**
final26	9	9852	1095
final27	10	29999	3000
final28	11	16156	1469
final29	8	30782	3848
final30	10	20357	2036
final31	12	22149	1846
final32	9	12717	1413
final33	9	**42273**	**4697**
final34	8	24099	3012
final35	8	39409	4926
final36	9	9206	1023
final37	8	22142	2768
final38	11	26160	2378
final39	10	16543	1654
final40	12	11800	983

N_{tr} : number of training samples

L_{tr} : number of features

A_{tr} : average number of features ($A_{tr} = L_{tr} / N_{tr}$)

Batch name	**N_{tr}**	**L_{tr}**	**A_{tr}**
final21	10	18116	1812
final22	11	19034	1730
final23	12	11168	931
final24	9	10544	1172
final25	11	**8547**	**777**
final26	9	9852	1095
final27	10	29999	3000
final28	11	16156	1469
final29	8	30782	3848
final30	10	20357	2036
final31	12	22149	1846
final32	9	12717	1413
final33	9	**42273**	**4697**
final34	8	24099	3012
final35	8	39409	4926
final36	9	9206	1023
final37	8	22142	2768
final38	11	26160	2378
final39	10	16543	1654
final40	12	11800	983

N_{tr} : number of training samples

L_{tr} : number of features

A_{tr} : average number of features ($A_{tr} = L_{tr} / N_{tr}$)

Codebook size: M

Traditional strategy: a given const value (e.g. 500 ,1000 2000)

But it may not be suitable for one-shot learning.

Batch name	N_{tr}	L_{tr}	A_{tr}
final21	10	18116	1812
final22	11	19034	1730
final23	12	11168	931
final24	9	10544	1172
final25	11	**8547**	**777**
final26	9	9852	1095
final27	10	29999	3000
final28	11	16156	1469
final29	8	30782	3848
final30	10	20357	2036
final31	12	22149	1846
final32	9	12717	1413
final33	9	**42273**	**4697**
final34	8	24099	3012
final35	8	39409	4926
final36	9	9206	1023
final37	8	22142	2768
final38	11	26160	2378
final39	10	16543	1654
final40	12	11800	983

N_{tr} : number of training samples

L_{tr} : number of features

A_{tr} : average number of features ($A_{tr} = L_{tr} / N_{tr}$)

Codebook size: M

Traditional strategy: a given const value (e.g. 500 ,1000 2000)

But it may not be suitable for one-shot learning.

Reason:

1) The number of features is varied for 20 batches. A given value may not be suitable for all 20 patches;

Batch name	N_{tr}	L_{tr}	A_{tr}
final21	10	18116	1812
final22	11	19034	1730
final23	12	11168	931
final24	9	10544	1172
final25	11	**8547**	**777**
final26	9	9852	1095
final27	10	29999	3000
final28	11	16156	1469
final29	8	30782	3848
final30	10	20357	2036
final31	12	22149	1846
final32	9	12717	1413
final33	9	**42273**	**4697**
final34	8	24099	3012
final35	8	39409	4926
final36	9	9206	1023
final37	8	22142	2768
final38	11	26160	2378
final39	10	16543	1654
final40	12	11800	983

N_{tr} : number of training samples

L_{tr} : number of features

A_{tr} : average number of features ($A_{tr} = L_{tr} / N_{tr}$)

Codebook size: M

Traditional strategy: a given const value (e.g. 500 ,1000 2000)

But it may not be suitable for one-shot learning.

Reason:

1) The number of features is varied for 20 batches. A given value may not be suitable for all 20 patches;

2) k-means condition: M ≦ the number of cluster

For example, if M=10000, it is failed in some batches (final 25, final 26, final 36)

Results

- Features analysis and Parameter discussion

Therefore, we use a new parameter $\gamma \in (0,1)$ to replace the codebook size M, then the codebook size M can be calculated:

$$M = L_{tr} * \gamma$$

Advantages:

1) Different batches have different codebook size.

2) γ satisfies the k-means condition.

$$(M = L_{tr} * \gamma \leqq L_{tr})$$

Results ---- comparison

ratio / Methods	0.1	0.2	0.3	0.4	0.5
Cuboid (R)	0.36717	0.36495	0.34332	0.33111	**0.31392**
Coboid (R+D)	0.33666	0.31559	0.30948	0.30782	**0.28064**
Harri3D hog (R)	0.30061	0.26012	0.25014	0.23516	**0.23461**
Harri3D hog (R+B)	0.24903	0.22795	**0.22407**	0.22795	0.22684
Harri3D hof (R)	0.34831	0.32668	0.31281	0.29895	**0.29063**
Harri3D hof (R+B)	0.32169	0.29174	0.28508	0.27898	**0.27121**
Harri3D hog/hof (R+B)	0.24237	0.21963	0.20022	0.19468	**0.18857**
Harri3D hog/hof (R+B)	0.20965	0.18802	0.18303	0.18747	**0.18192**
MoSIFT(R)	0.41653	0.39601	0.35885	0.36606	**0.335**
MoSIFT(R+B)	0.44426	0.4426	0.43594	0.42318	**0.40488**
3D MoSIFT (R+B)	0.19135	0.16694	0.16195	***0.14476***	0.14642

Discussion and feature work

- 3D MoSIFT merits:

1) Scale and rotation invariant

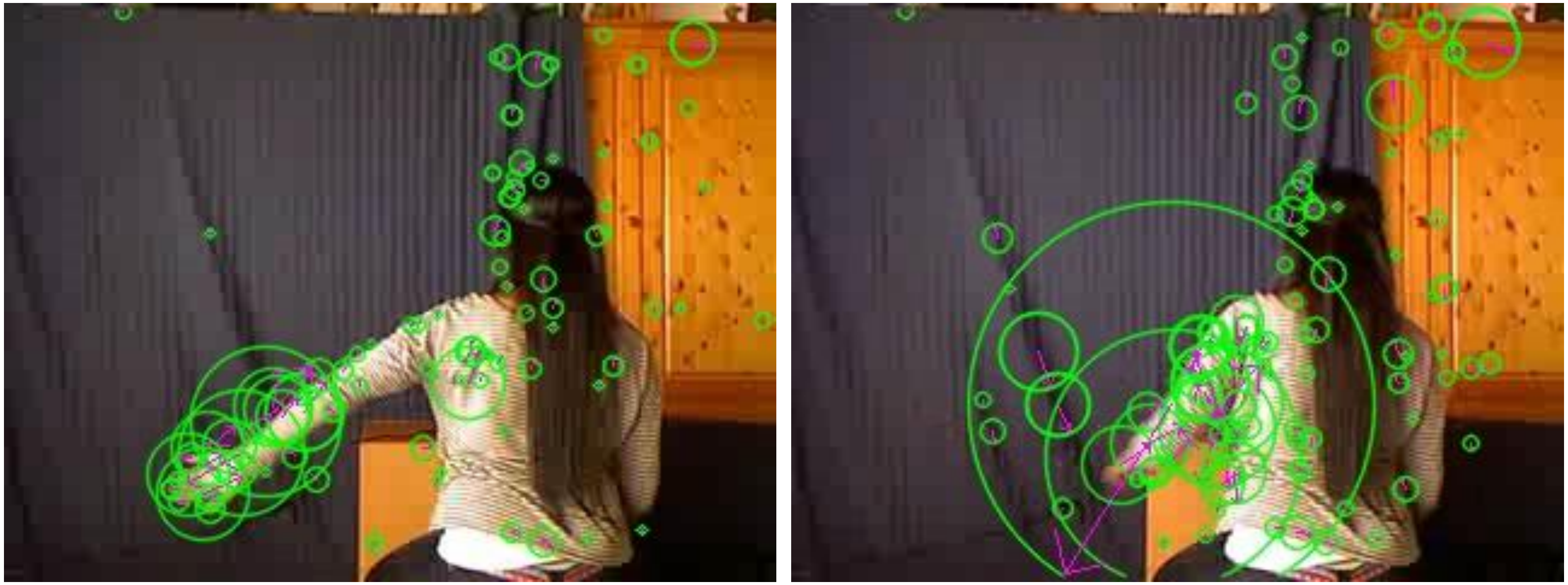
2) Capture more compact and richer representation

According to experimental results, 3D MoSIFT is suitable for one-shot learning gesture recognition.

Discussion and feature work

- 3D Mosift drawbacks:

1) Processing time: about 200ms/f.
2) Still capture redundant features in the background.



Green circles indicate the scale value of interest points and purple arrow shows the direction of movement.

Reference

(1) I. Laptev. On space-time interest points. International Journal of Computer Vision, 64(2):107–123, 2005.

(2) P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, pages 65–72, 2005.

(3) M. Chen and A. Hauptmann. “Mosift: Recognizing human actions in surveillance videos”. 2009.

(4) Y. Ming, Q. Ruan, and A.G. Hauptmann. “Activity recognition from rgb-d camera with 3d local spatio-temporal features”. In Multimedia and Expo (ICME), 2012 IEEE International Conference on, pages 344–349. IEEE, 2012.

Thank you!