Multi-Class Model Fitting by Energy Minimization and Mode-Seeking

Daniel Barath

joint work with Jiri Matas

Multi-class Multi-instance Fitting Problem

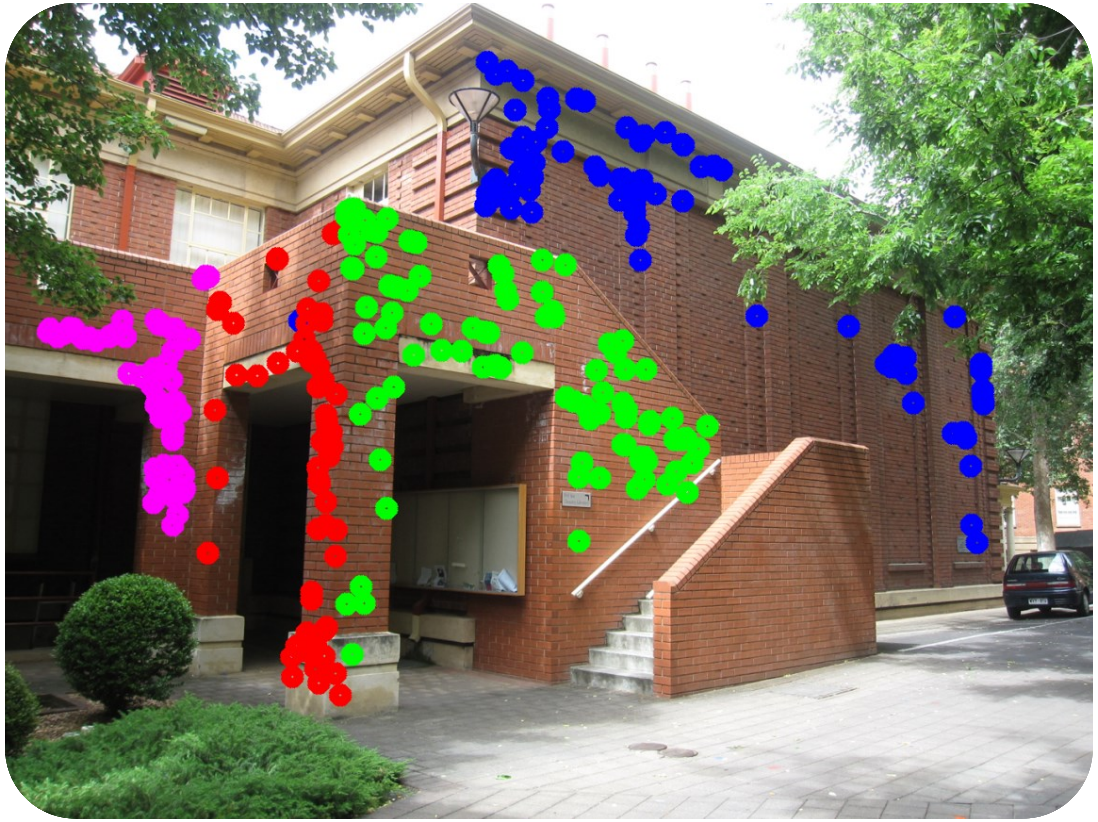
Interpreting the input data as a set of model instances of multiple classes.

Multi-class Multi-instance Fitting Problem

Interpreting the input data as a set of model instances of multiple classes.

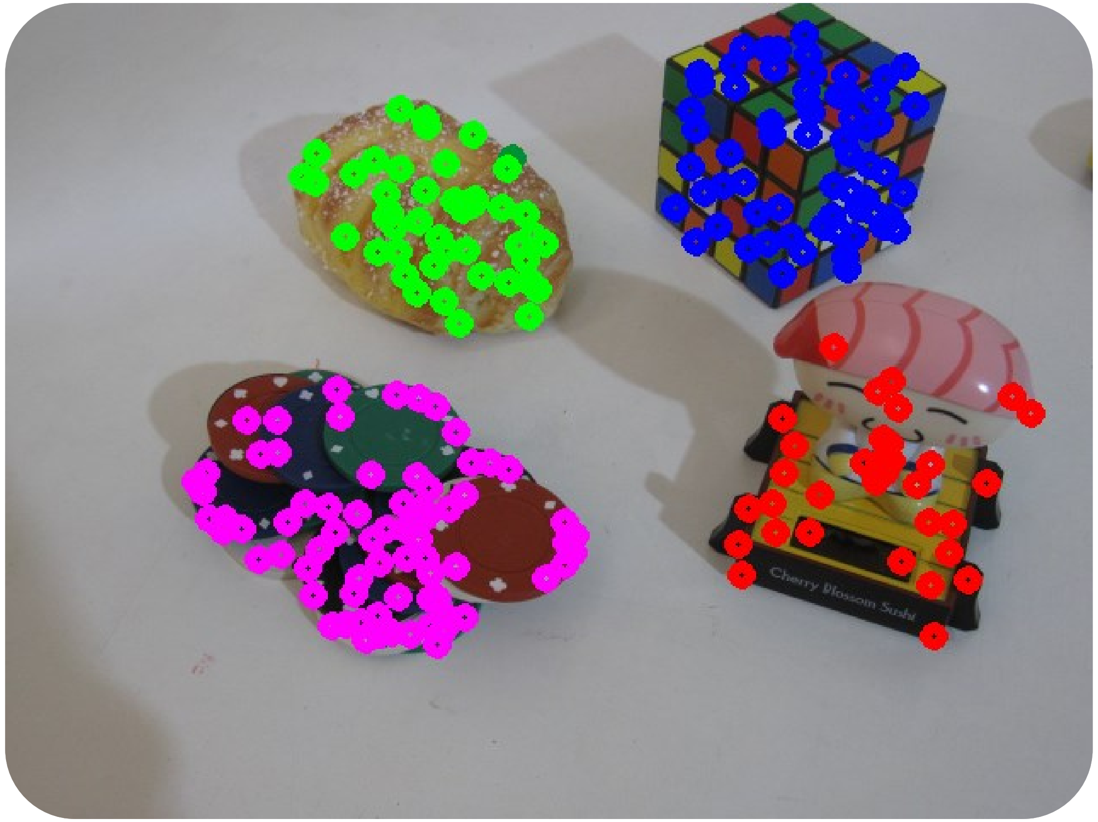
Instance of Single Class Multi Model Fitting:

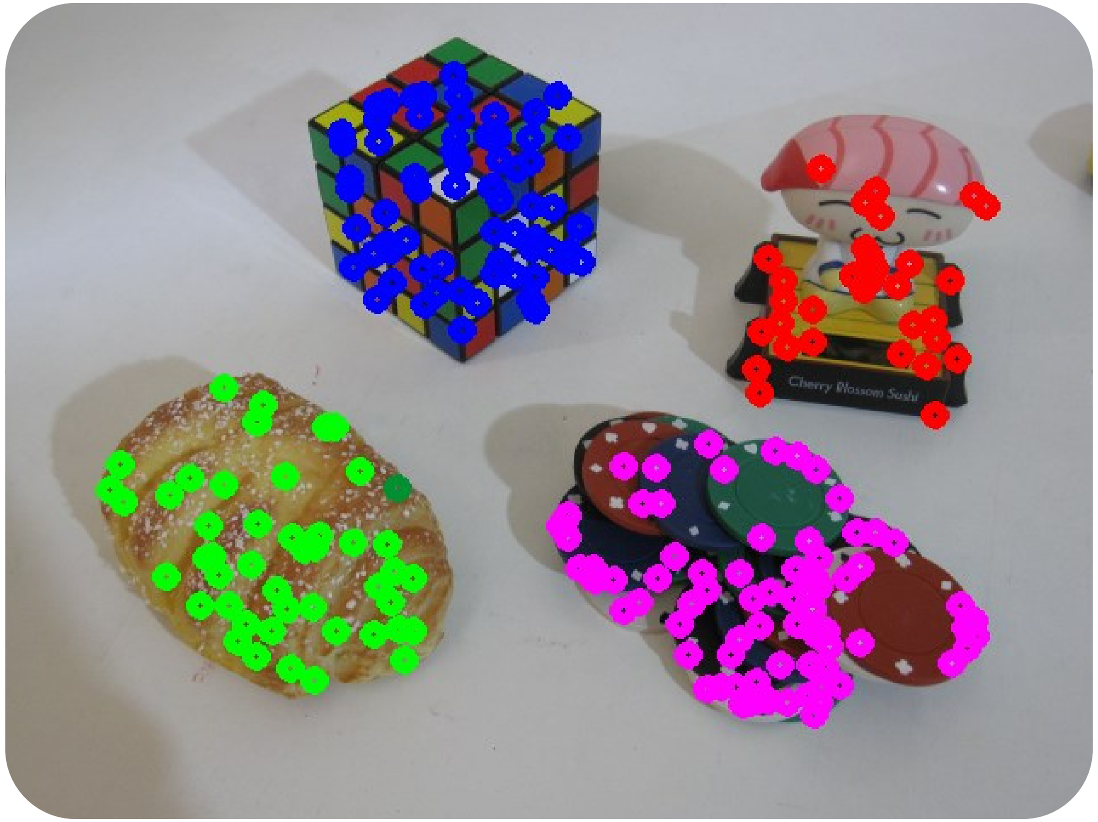
Fitting multiple **homographies.**



Instance of Single Class Multi Model Fitting:

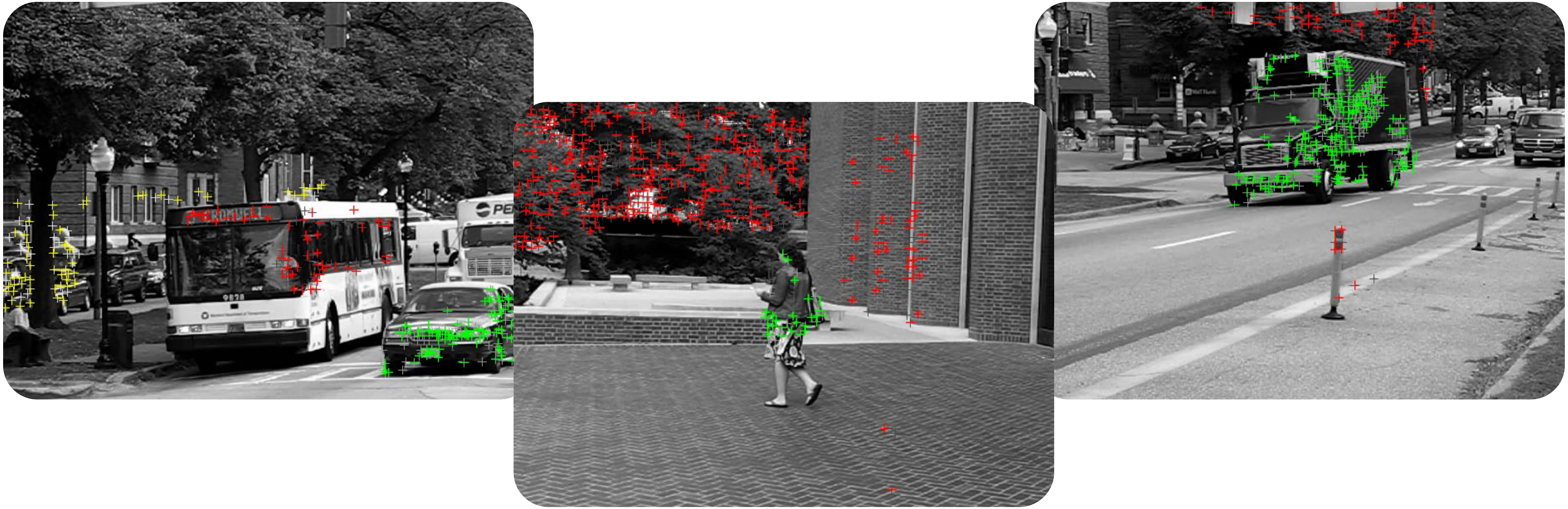
Fitting multiple two-view **rigid motions.**





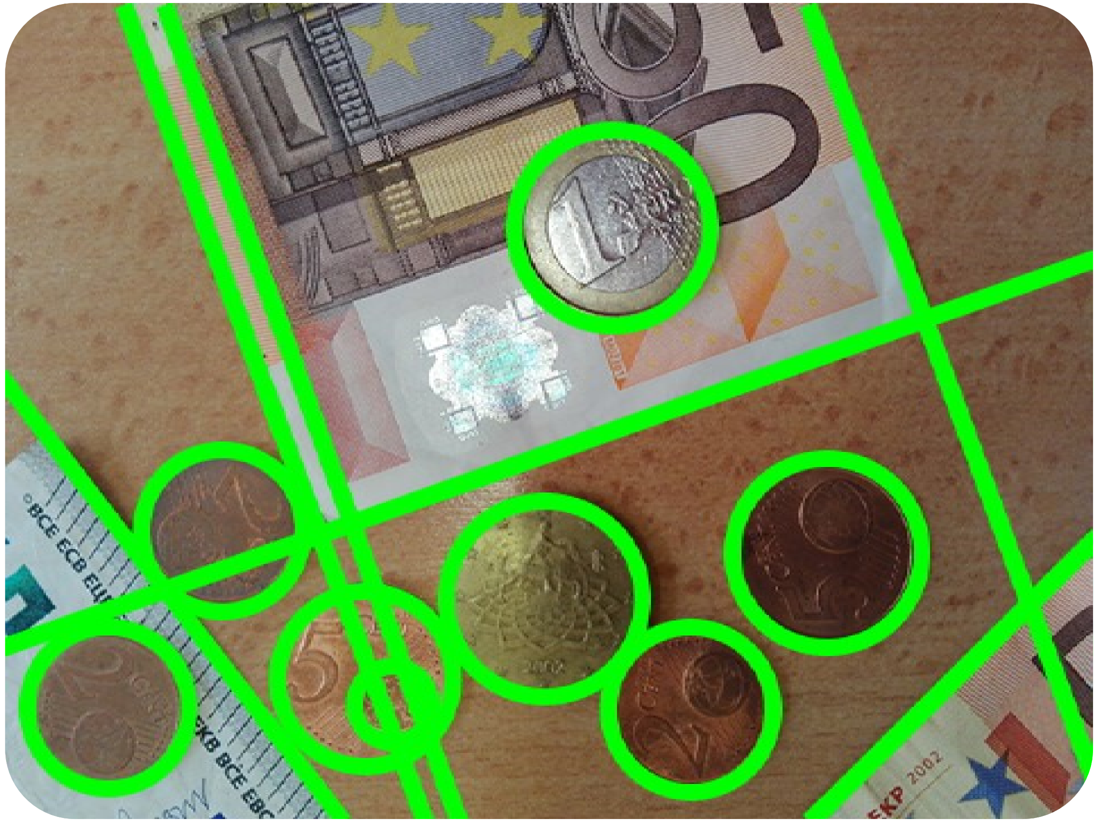
Instance of Single Class Multi Model Fitting:

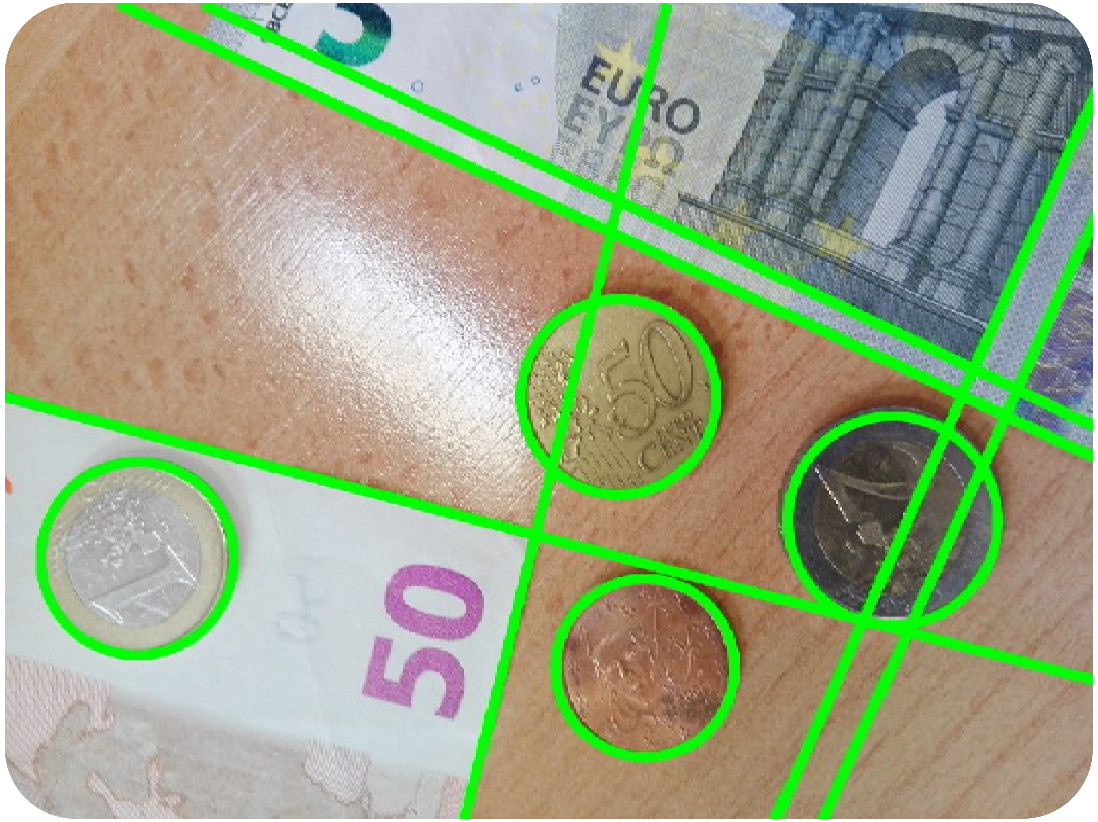
Fitting multiple **motions in video sequences.**



Instance of Multi Class Multi Model Fitting:

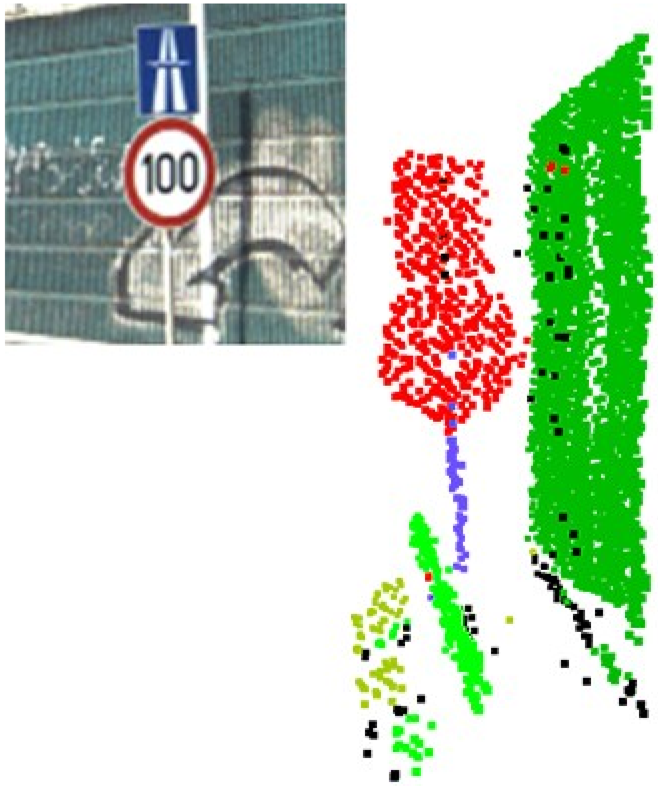
Fitting **lines and circles** (or other 2D shapes) on edge map.

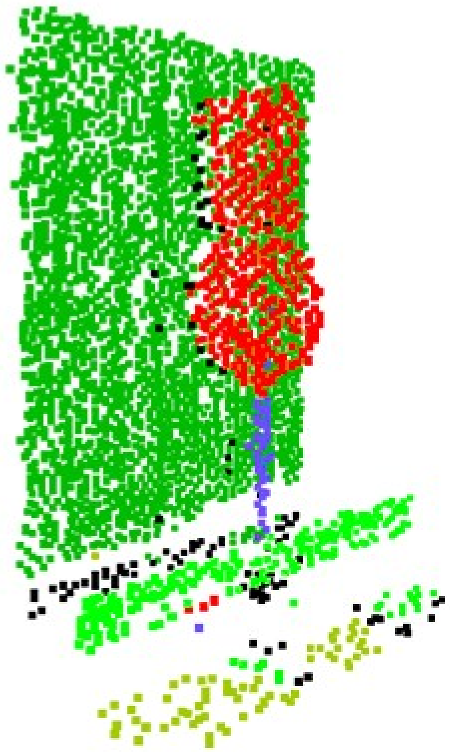




Instance of Multi Class Multi Model Fitting:

Fit **planes and cylinders** to detect traffic signs and columns in LIDAR data





It is and Active and Old Problem

Multi-model fitting **of a single class** is still an open problem.

Publications from the last few years:

- D., Barath, L., Hajder, and J., Matas [BMVC 2016]
- L., Magri and A., Fisuello: [ECCV 2008, CVPR 2014, BMVC 2015, CVPR 2016]
- H. Wang, G. Xiao, Y. Yan, and D. Suter: [ICCV 2015]
- T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter: [TIP 2014]
- H. Isack and Y. Boykov: [IJCV 2012]
- E. Elhamifar and R. Vidal: [CVPR 2009]
- J.-P. Tardif: [ICCV 2009]
- N. Lazic, I. Givoni, B. Frey, and P. Aarabi: [ICCV 2009]

Multi-model fitting **of multiple classes???**

- No recent publications in the literature.

It is and Active and Old Problem

Multi-model fitting **of multiple classes???**

- No recent publications in the literature

I have two interpretations:

- Even the single-class case is barely solved: **good results, but for the per-test-tuned case.** (Parameters tuned separately for each test case.)
- It becomes important in 3D and cheap 3D sensors have only been available for the last few years.

Energy Minimization for single class multi instance fitting

PEARL: H. Isack and Y. Boykov: [IJCV 2012]
MFIGP: T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter [TIP 2014]
Multi-H: D., Barath, L., Hajder, and J., Matas [BMVC 2016]

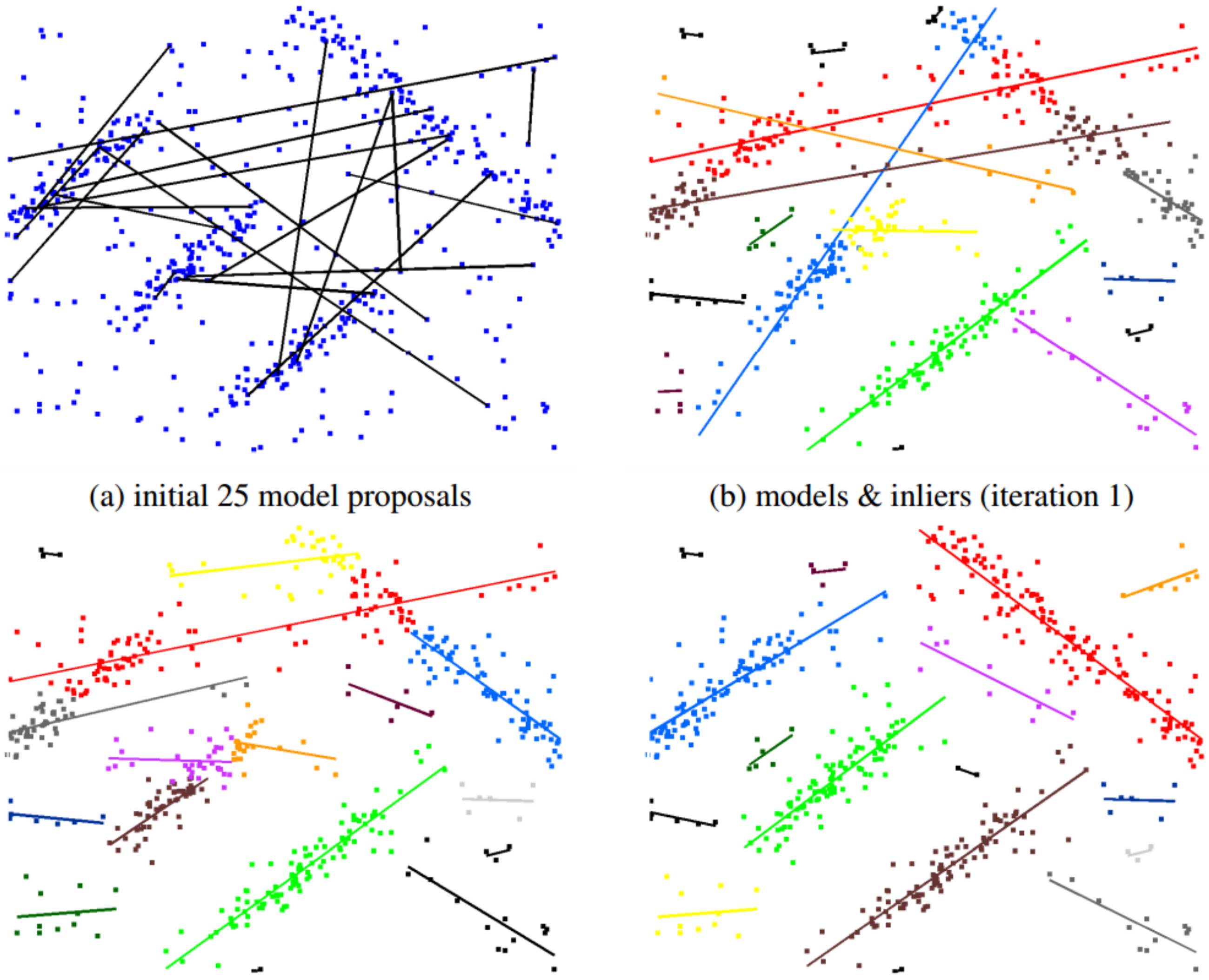
PEARL

H. Isack and Y. Boykov: [IJCV 2012]

A global energy term consisting of three terms:

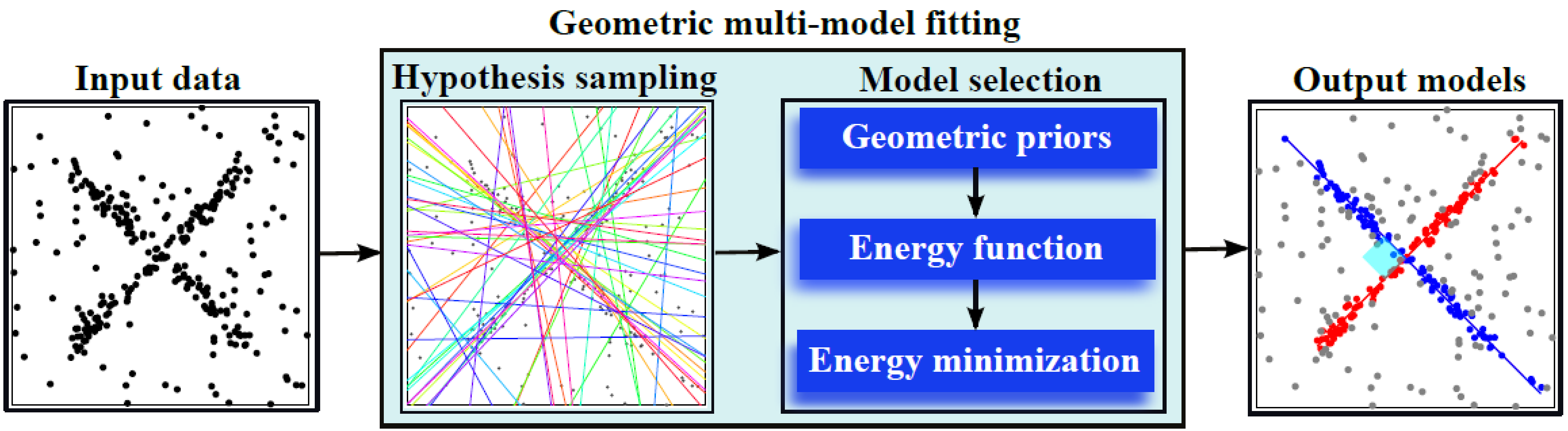
1. **Data term:** Penalize point-to-model assignment.
2. **Spatial Regularization term:** Close points are more likely belong to the same model instance.
3. **Complexity term:** Penalize the introduction of new labels.

PEARL algorithm: iteration of labeling and model refitting.


(a) initial 25 model proposals (b) models & inliers (iteration 1)
(c) models & inliers (iteration 2) (d) at convergence (iteration 5)

MFIGP

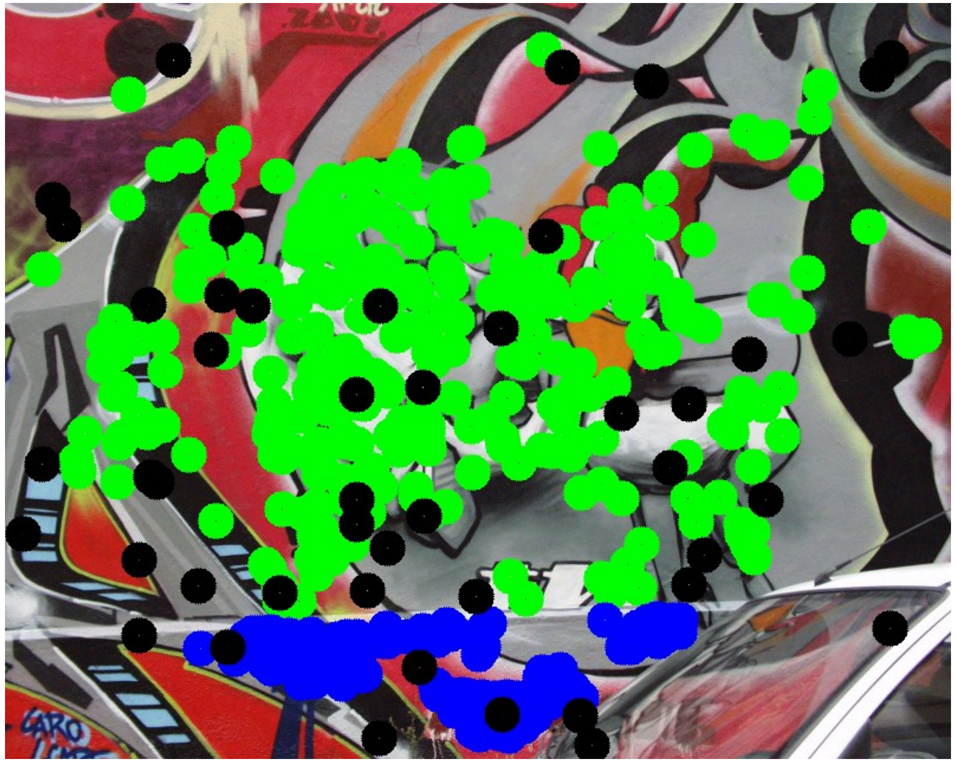
T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter [TIP 2014]



Multi-H

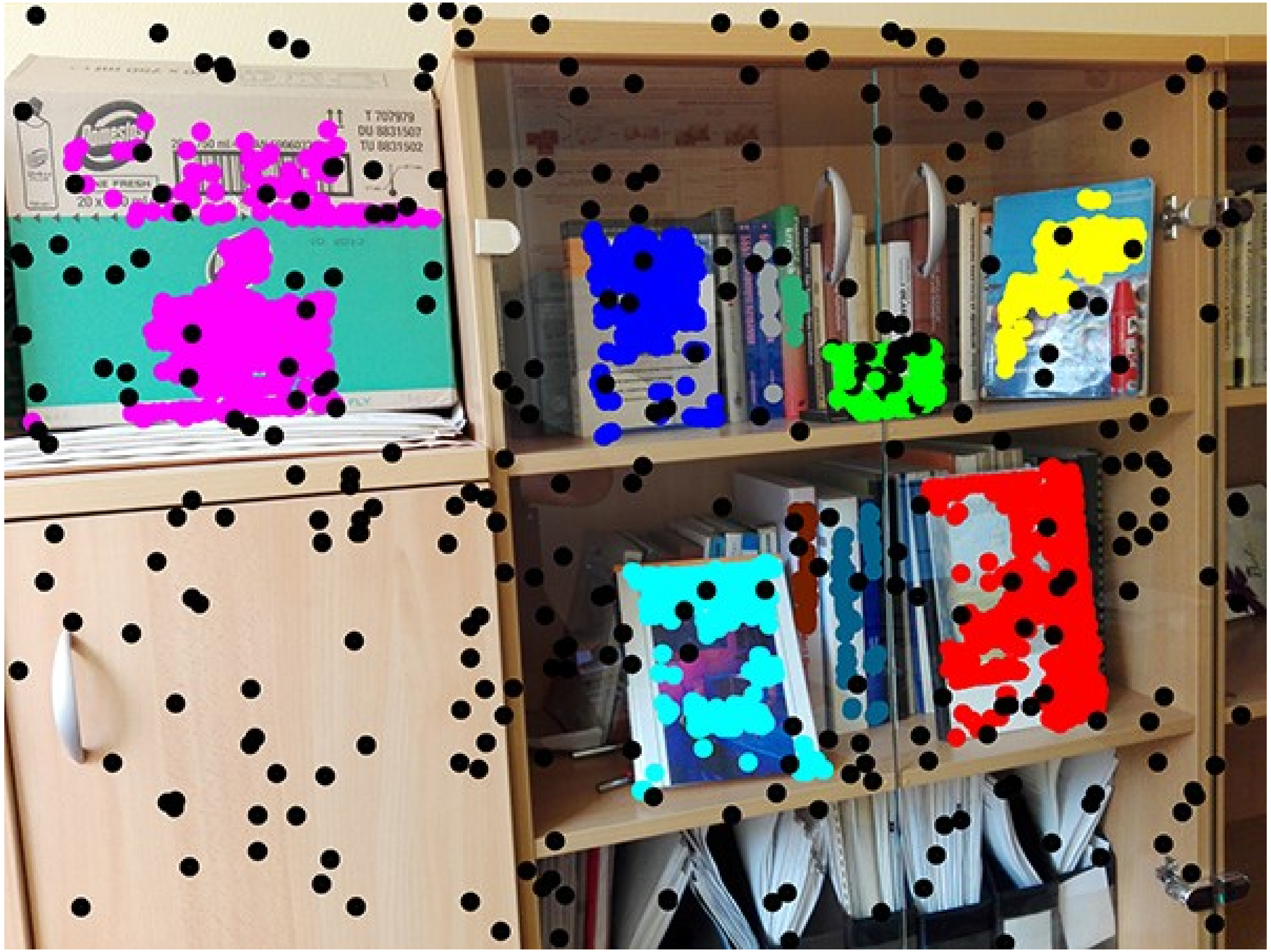
D., Barath, L., Hajder, and J., Matas [BMVC 2016]

1. Concentrating on **multi-homography estimation**.
2. Achieves more accurate results than state-of-the-art multi-homography estimation methods using **mode-seeking and energy minimization.**
3. Doesn’t consider the general case, only homographies are fitted.



Multi-H

D., Barath, L., Hajder, and J., Matas [BMVC 2016]



Multi-X
for multi class multi instance fitting

Goals

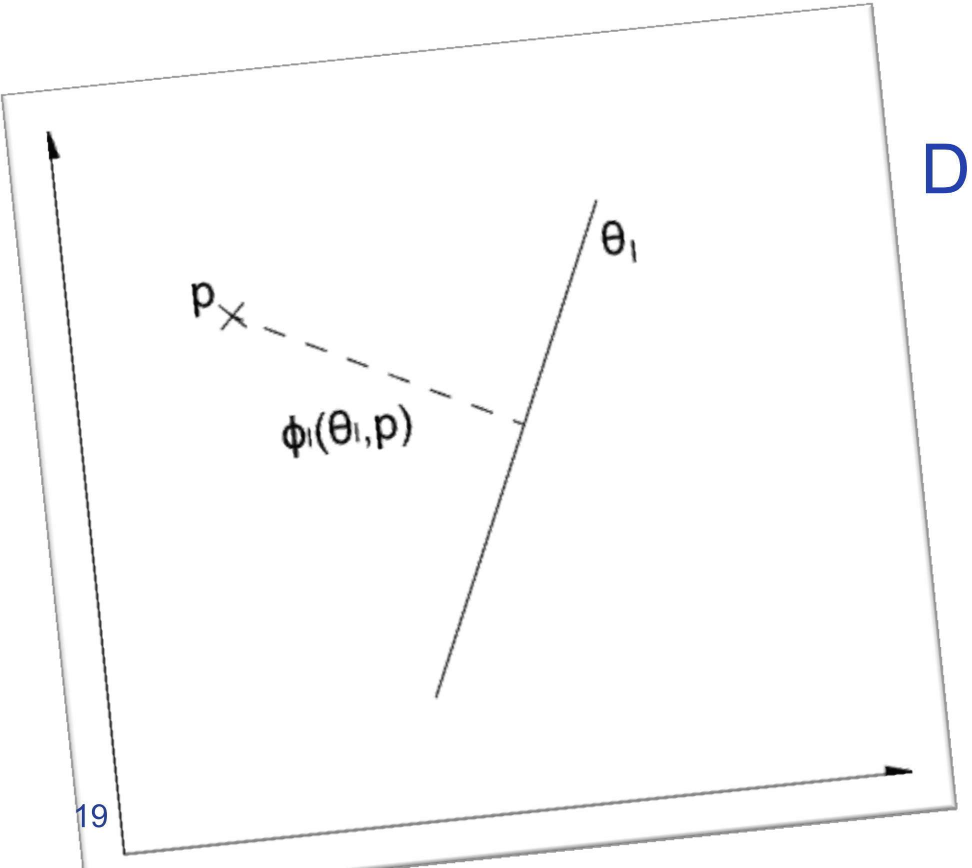
1. Fit multiple model instances of different classes.
2. Having accurate results without tuning the parameters problem-by-problem.

Problem Formulation

Example Model: Line Model

Line model: $\mathcal{H}_l = (\theta_l, \phi_l)$

Line model instance: $h \in \mathcal{H}_l$



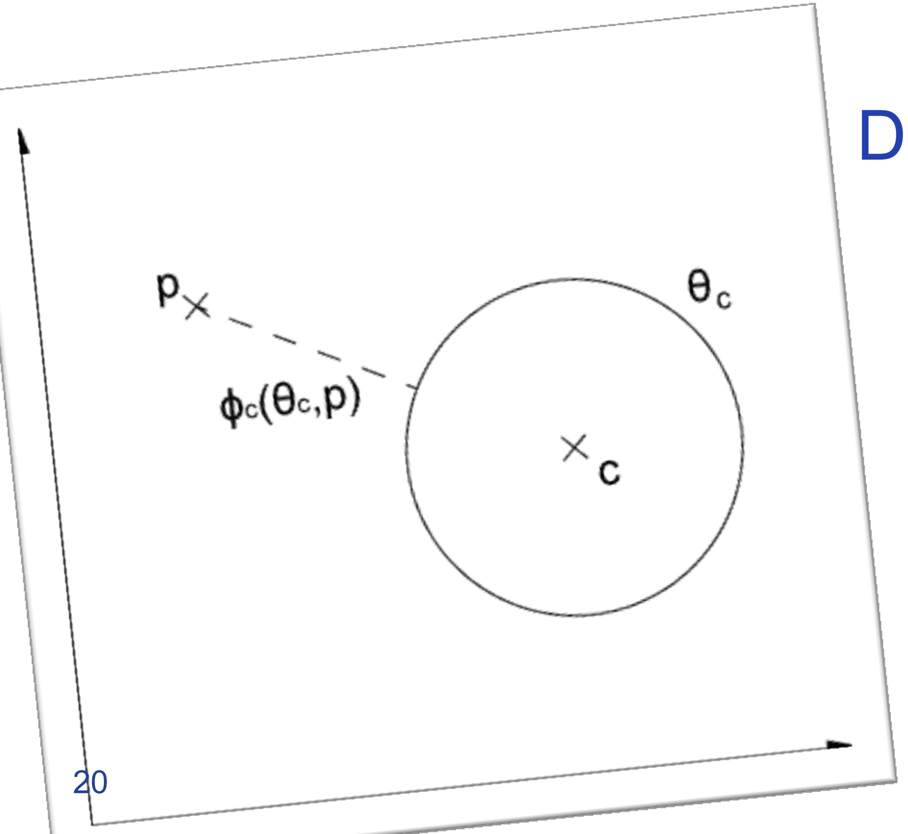
Distance function: $\phi_l(\theta_l, p) = \frac{ax + by + c}{\sqrt{a_l^2 + b_l^2}}$

Parameter vector: $\theta_l = [a \quad b \quad c]^T$

Example Model: Circle Model

Circle model: $\mathcal{H}_c = (\theta_c, \phi_c)$

Circle model instance: $h \in \mathcal{H}_c$



Distance function: $\theta_c = |r - \sqrt{(c_x - x)^2 + (c_y - y)^2}|$

Parameter vector: $\phi_c = [c_x \quad c_y \quad r]^T$

Definition 1 (Multi-Class Model) *The multi-class model is a set* $\mathcal{H}^*$ *consisting of all models* $\mathcal{H}^* = \{(\theta, \phi) \mid d \in \mathbb{N}, \theta \in \mathbb{R}^d, \phi \in \mathcal{P} \times \mathbb{R}^d \rightarrow \mathbb{R}\}$*, where* $\mathcal{P}$ *is the set of data points and d is the dimension of parameter vector* θ*.*

Parameter vector

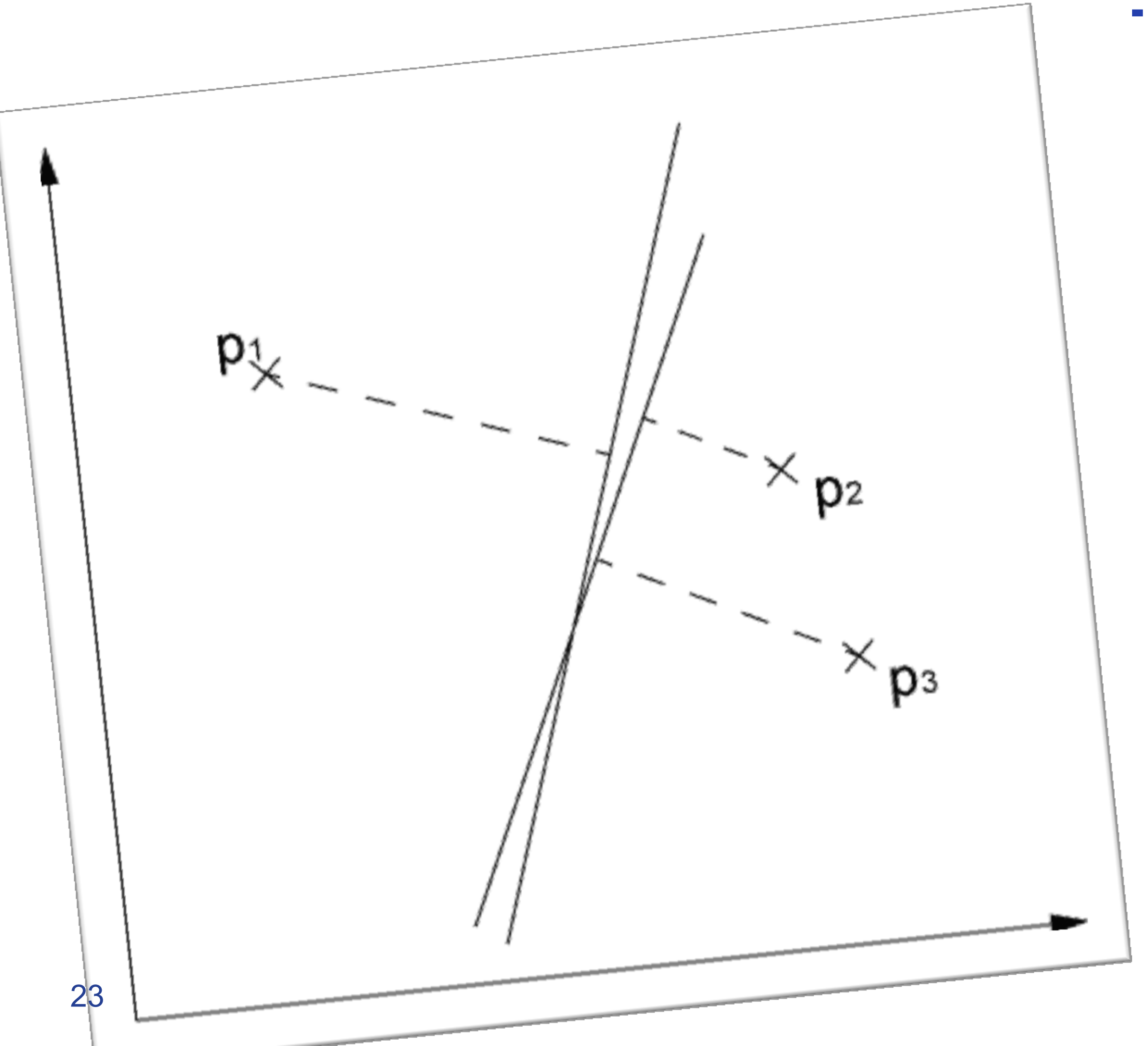
Distance function

Multi-class Multi-instance Fitting Problem

Given:
- the input data $\mathcal{P}$
- the multi class model $\mathcal{H}^*$

Output:
- model instances $\mathcal{G} \subset \mathcal{H}^*$
- the labelling *L* assigning points from $\mathcal{P} \rightarrow \mathcal{G}$ minimizing an energy *E*.

Energy – Data Term



- The term penalizing the **point-to-model** assignment used in the literature:

$$E_d(L) = \sum_{p \in \mathcal{P}} \phi_{L(p)}(\theta_{L(p)}, p)$$

Energy – Data Term

Assumption: randomly generated model instances form modes around the ground truth instances in the model parameter space.

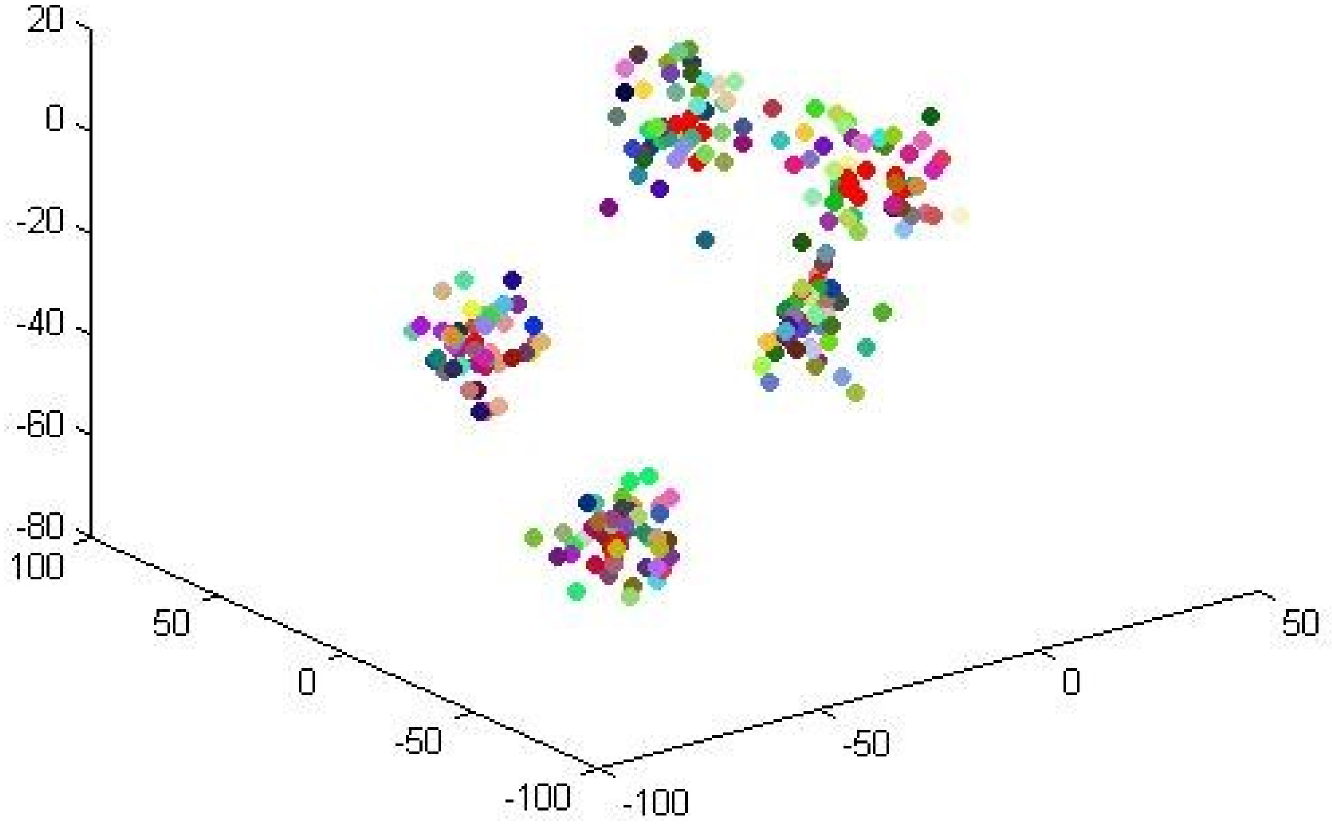
Example:

A **2D** line can be represented by a 3D vector

$$\theta_l = [a \quad b \quad c]^T$$

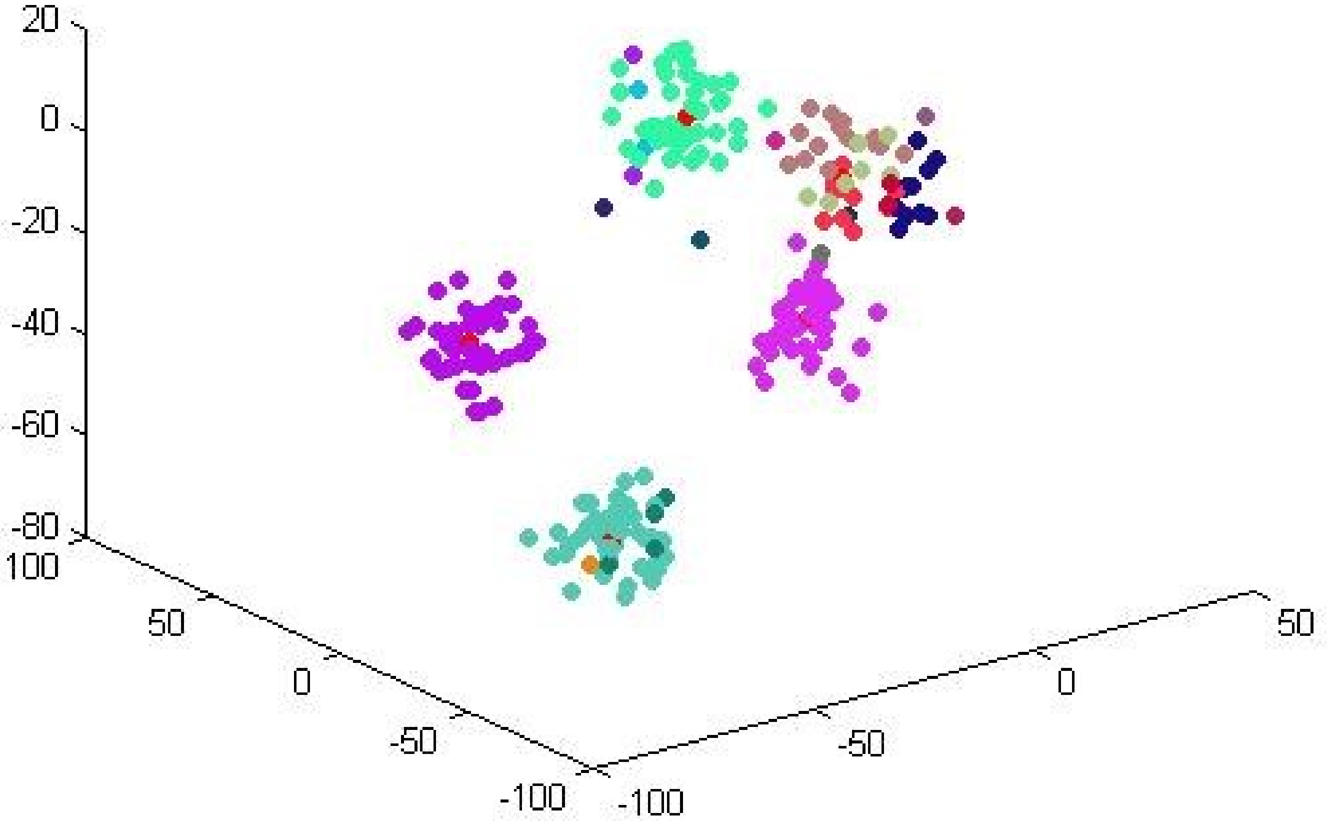
Represent a set of line instances in the model parameter space...

Energy – Data Term



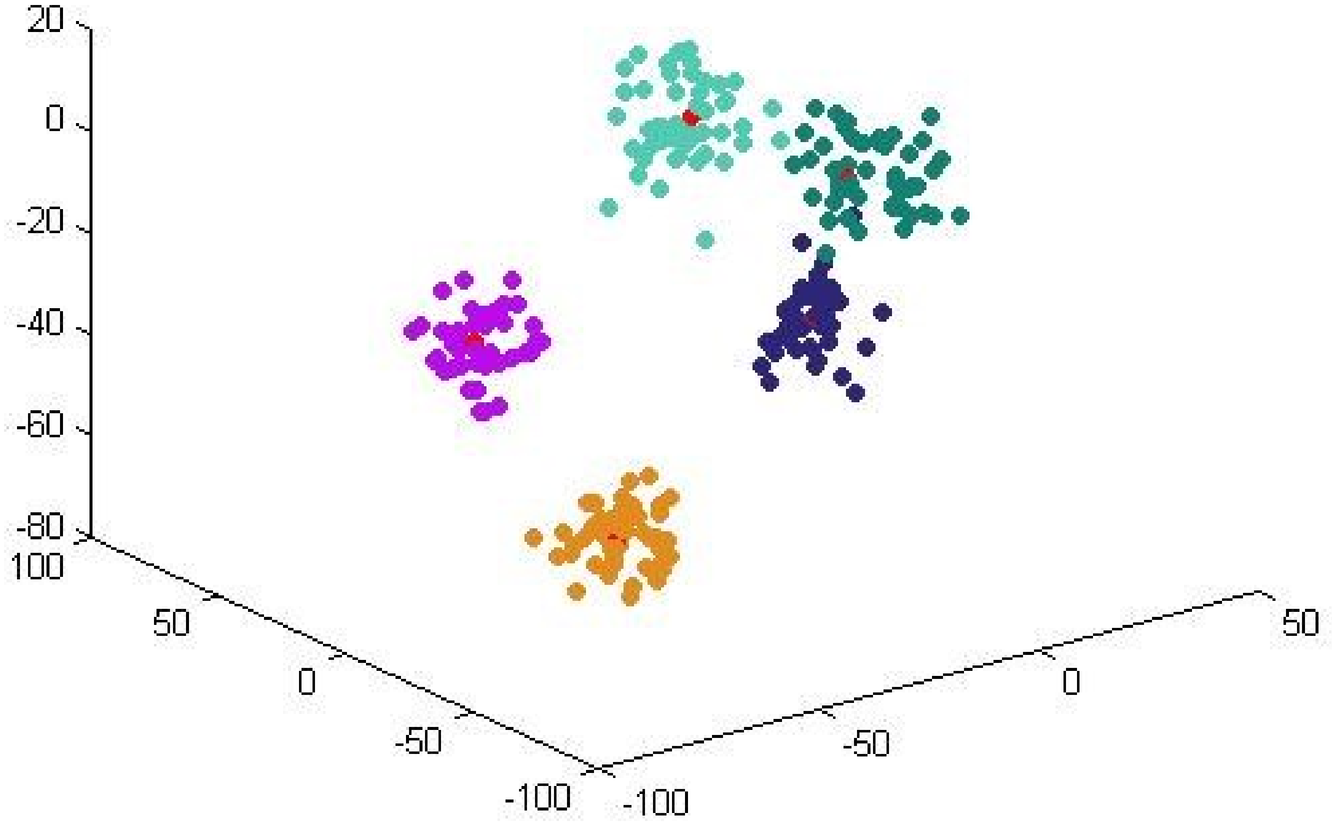
Line instances in their 3D space. Median-Shift, iteration #1

Energy – Data Term



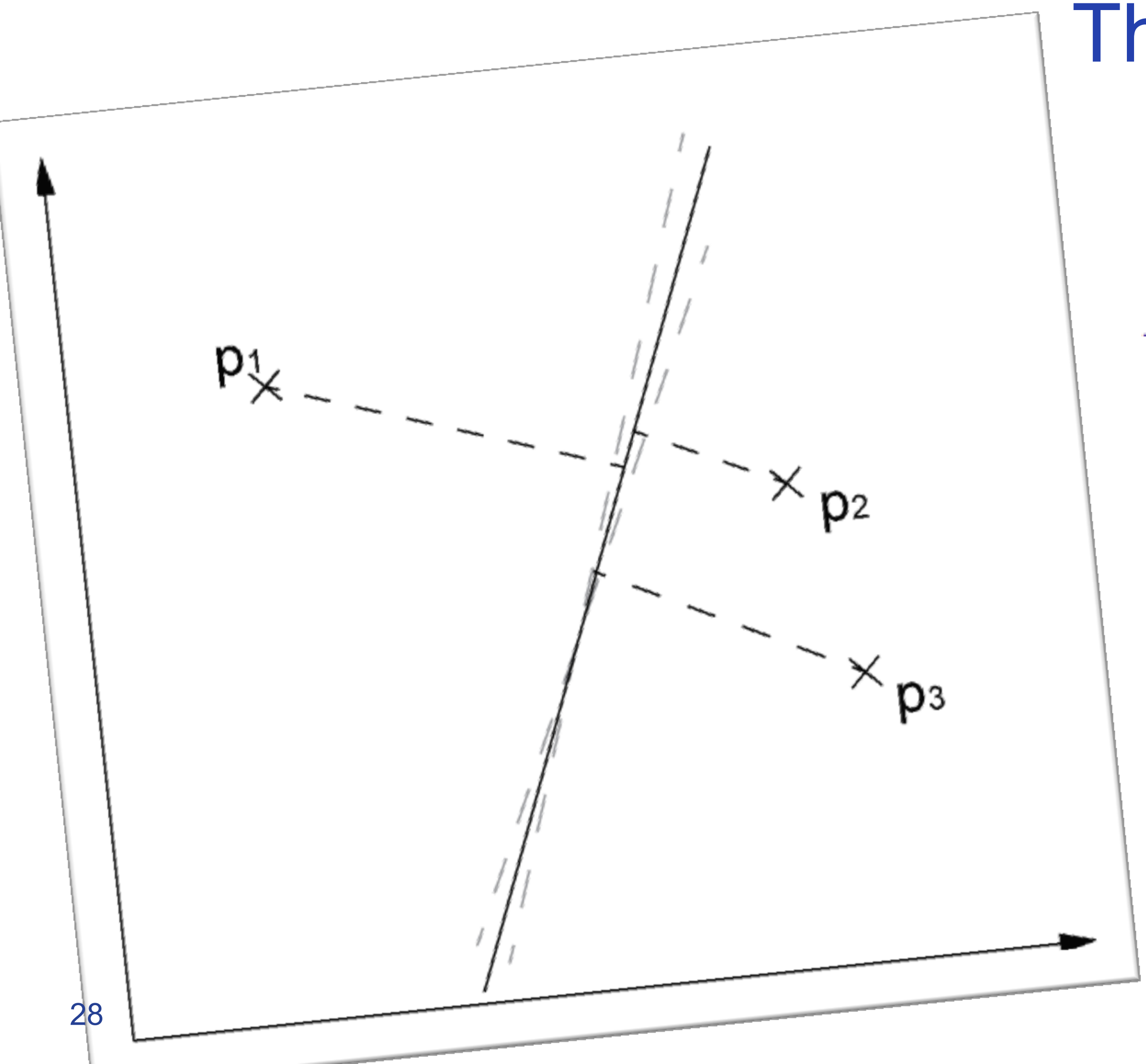
Line instances in their 3D space. Median-Shift, iteration #2

Energy – Data Term



Line instances in their 3D space. Median-Shift, iteration #3

Energy – Data Term



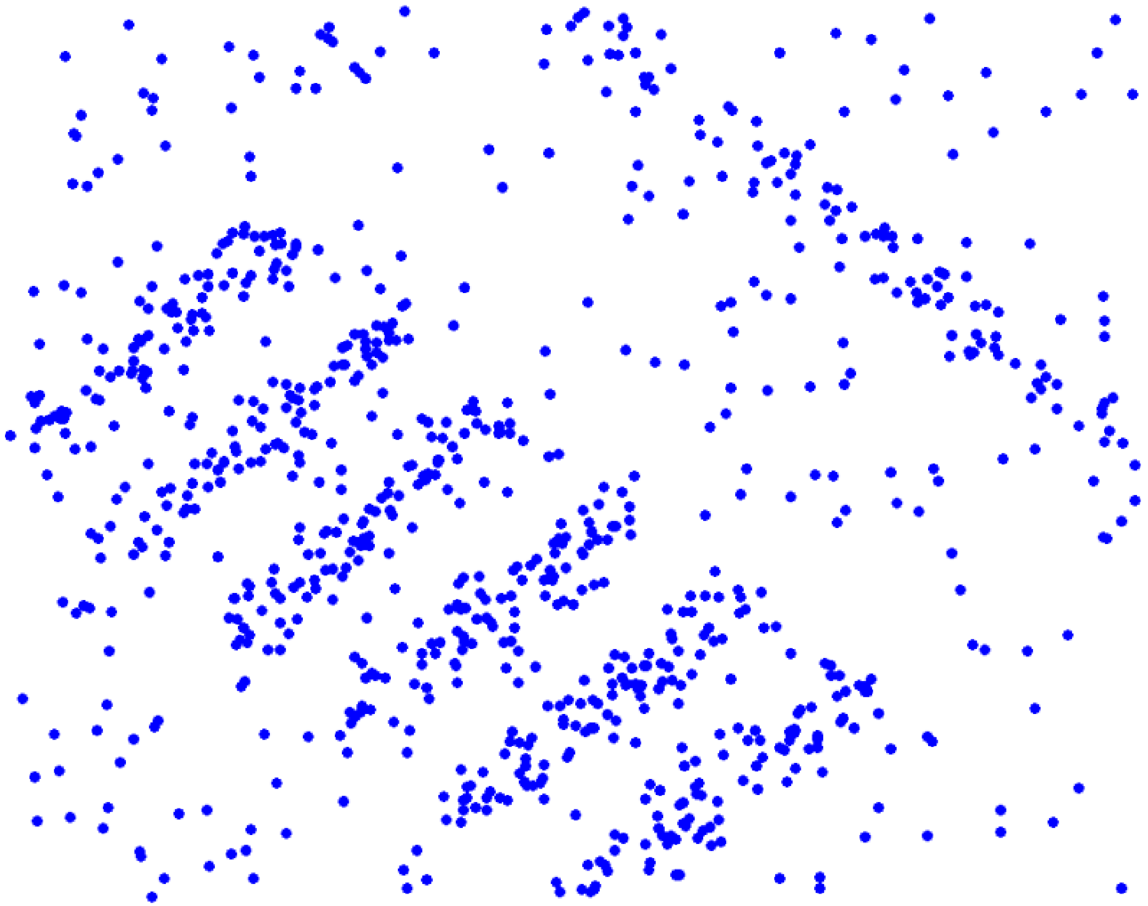
The term penalizing the **point-to-mode** assignment:

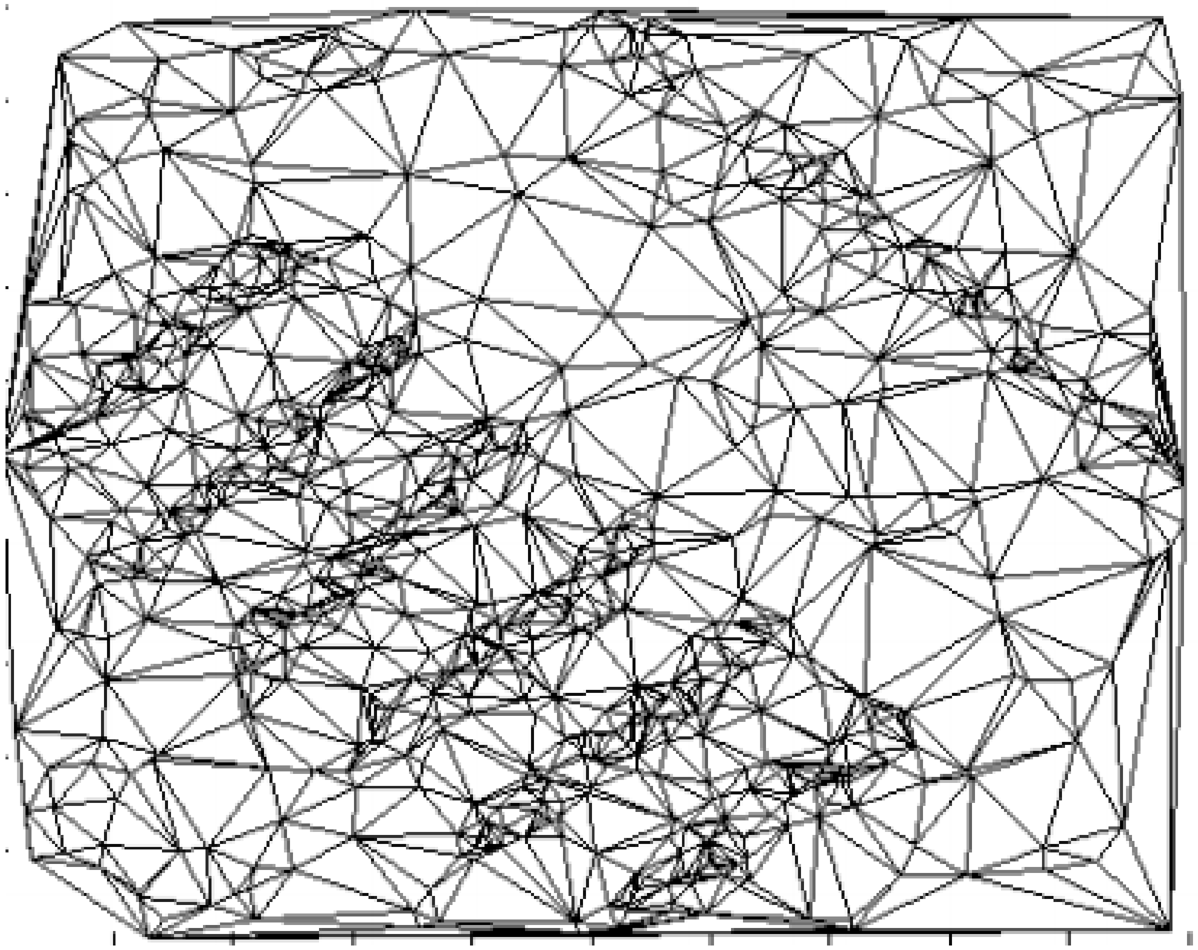
$$\widehat{E}_d(L) = \sum_{p \in \mathcal{P}} \phi^{\Theta}_{L(p)}(\theta^{\Theta}_{L(p)}, p)$$

Θ is a mode-seeking function.

$(\theta^{\Theta}_{L(p)}, \phi^{\Theta}_{L(p)})$ is the mode assigned to point p.

Energy – Spatial Coherence Term





Isack, Hossam, and Yuri Boykov. "Energy-based geometric multi-model fitting." *International journal of computer vision* 97.2 (2012): 123-147.

Energy – Spatial Coherence Term

The term penalizing neighbors with different labels:

Iverson bracket

$$E_s(L) = \sum_{(p,q)\in N} w_{pq} [\![L(p) \neq L(q)]\!]$$

Edges in the neighborhood graph

Weighting parameter

Labels of point *p* and *q*

Isack, Hossam, and Yuri Boykov. "Energy-based geometric multi-model fitting." *International journal of computer vision* 97.2 (2012): 123-147.

Energy – Complexity

The term to suppress weak model instances by penalizing the introduction of new labels.

We propose a term having different cost for each model classes:

$$\widehat{E}_c(L) = \sum_{l \in \mathcal{L}_L} \psi_l$$

Penalty of class

Set of distinct labels

Overall Energy

Spatial Coherence term
(close points belong to the same instance)

$$\widehat{E}(L) = \widehat{E}_d(L) + \lambda E_s(L) + \beta \widehat{E}_c(L)$$

Data term
(point-to-mode assignment)

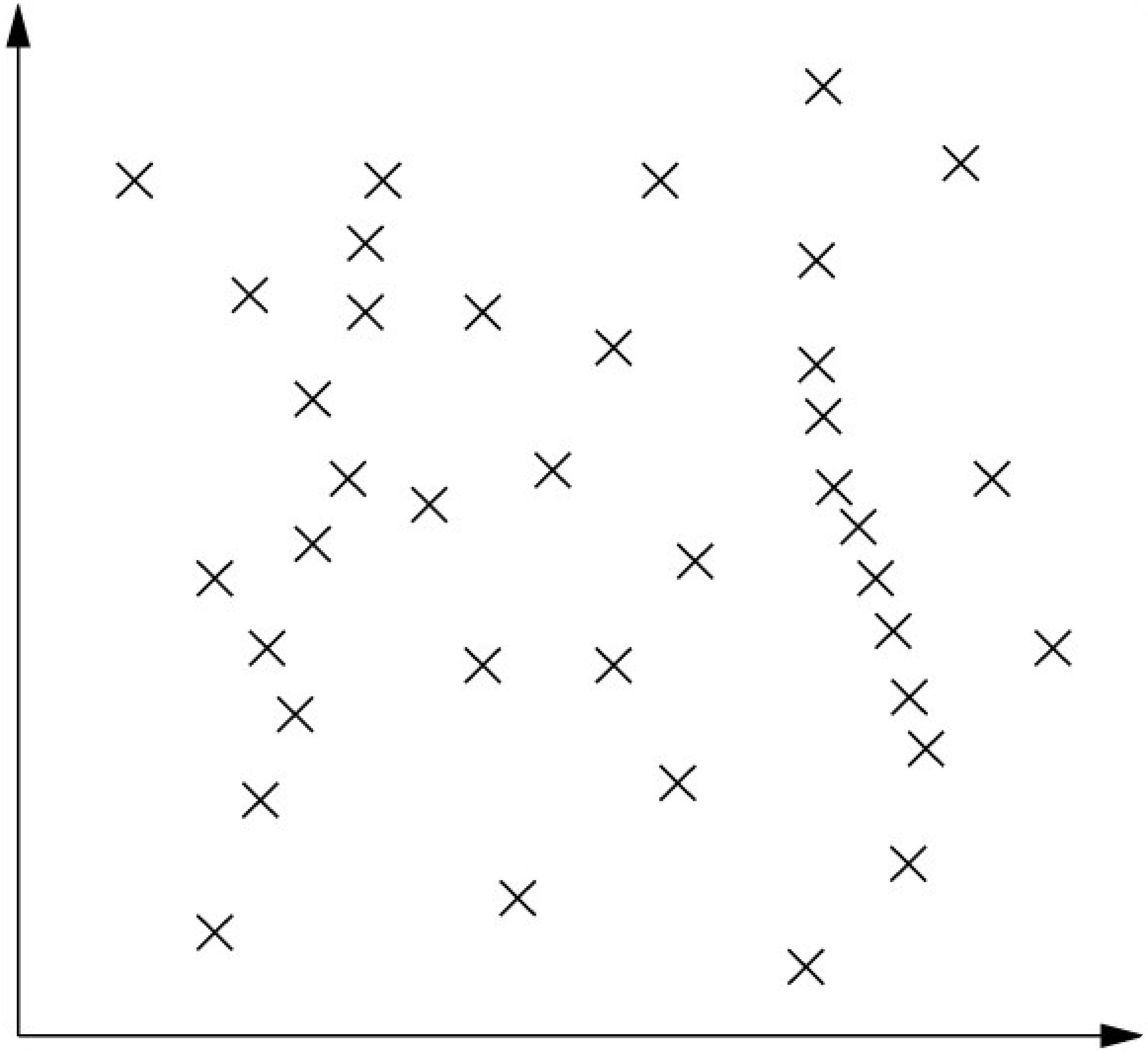
Regularization term
(penalize new instances)

Algorithm

Input: P – data points
Output: H^* – model instances, L^* – labeling

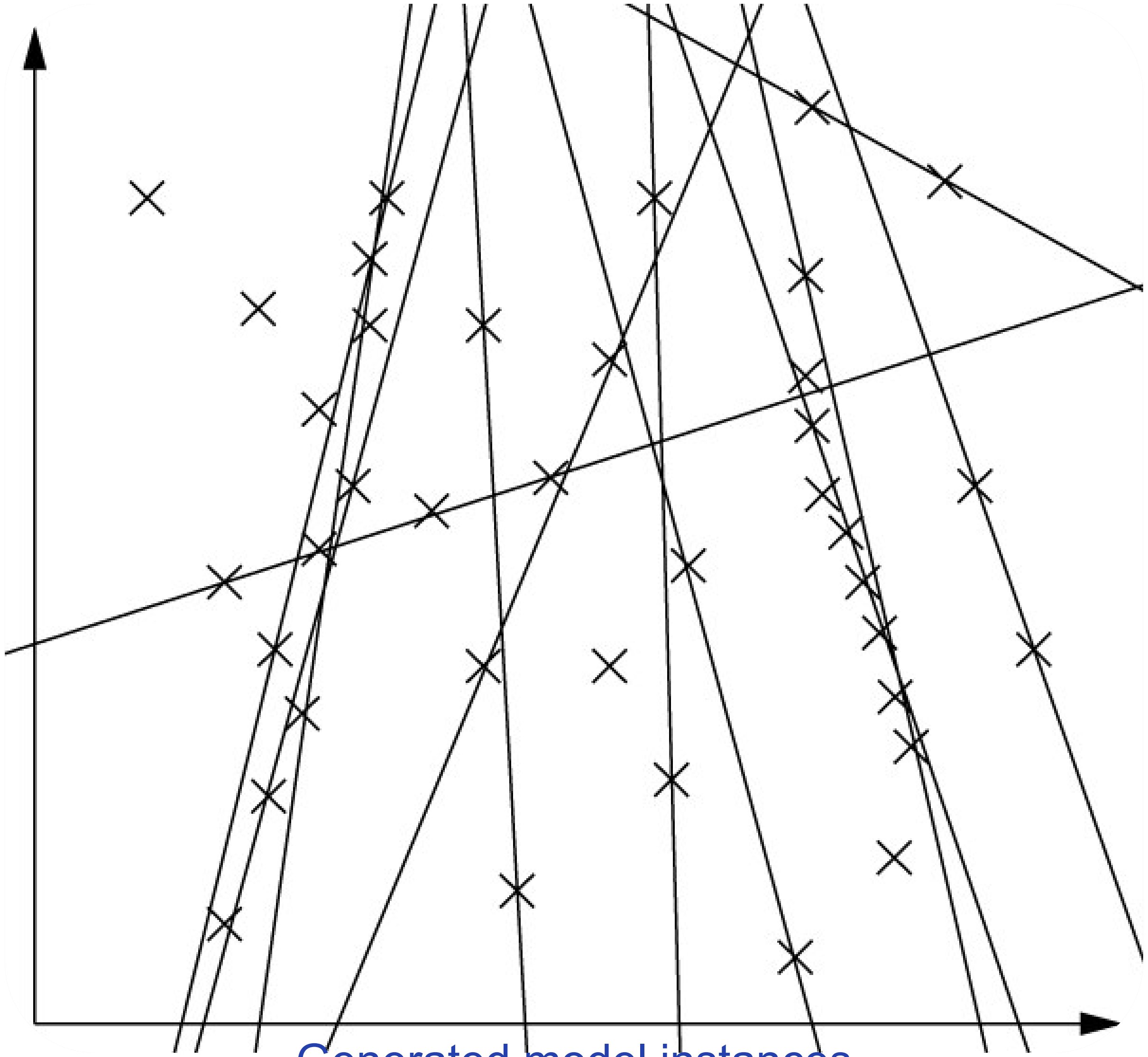
$H_0 :=$ InstanceGeneration(P); $i := 1$;
repeat
  $H_i :=$ ModeSeeking(H_{i-1}); ▷ by Median-Shift
  $L_i :=$ LabelingToMode(H_i, P); ▷ by α-expansion
  $L_i :=$ OutlierRemoval(H_i, L_i, γ);
  $H_i :=$ ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
  $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}$, $L^* := L_{i-1}$;
H^*, $L^* :=$ RemoveUnstableModels(H^*, L^*)

Algorithm: Input Points



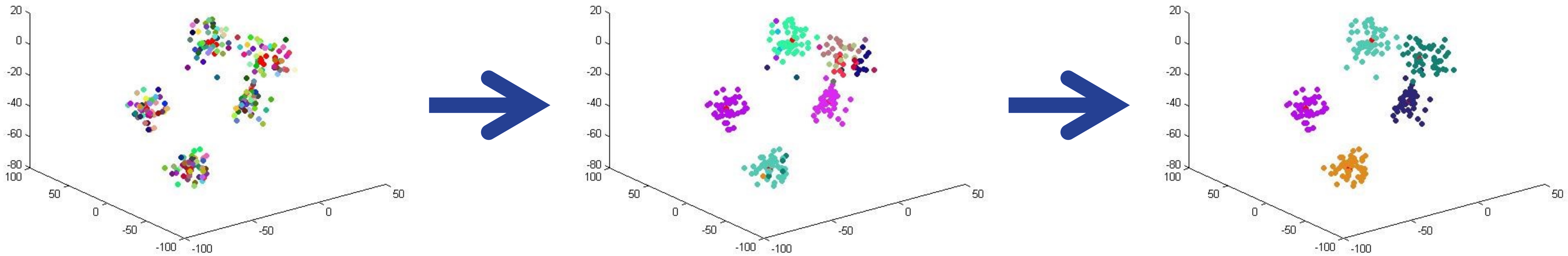
Input 2D points

Algorithm: Model Instance Generation



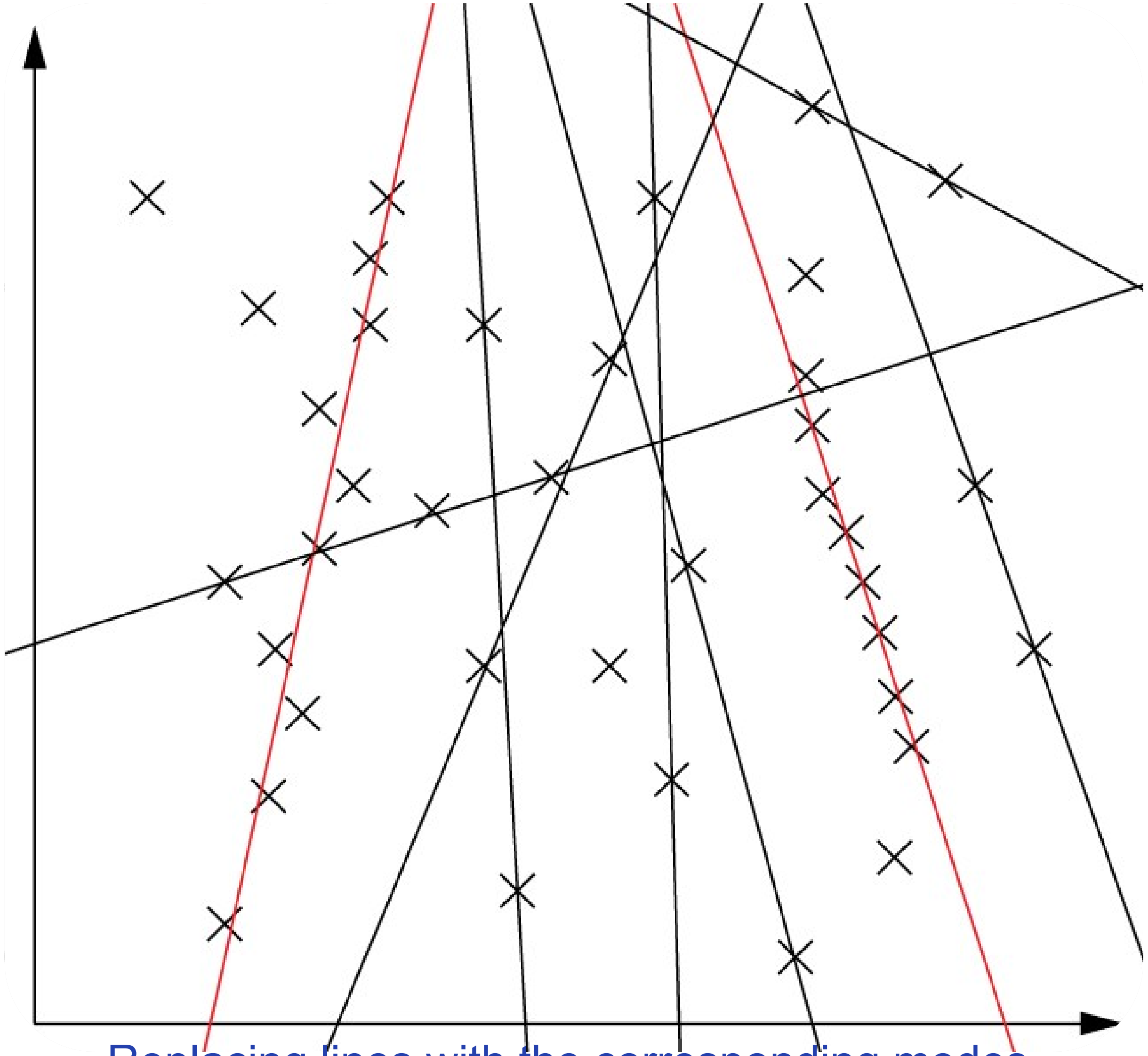
Generated model instances

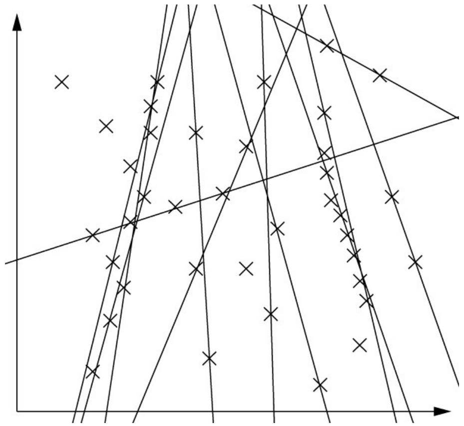
Algorithm: Mode-Seeking



Mode-Seeking in the line space

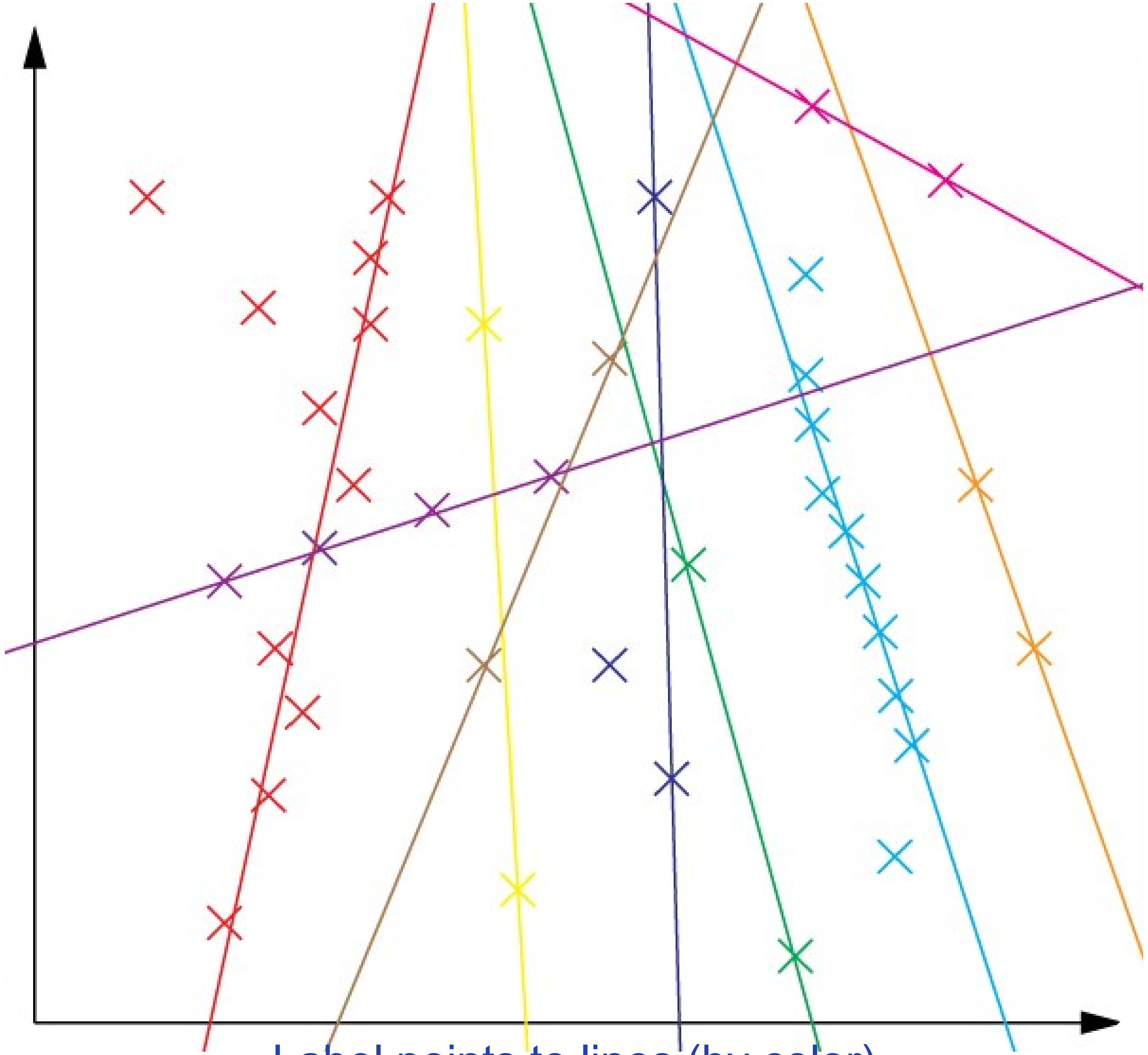
Algorithm: Replacing with Mode





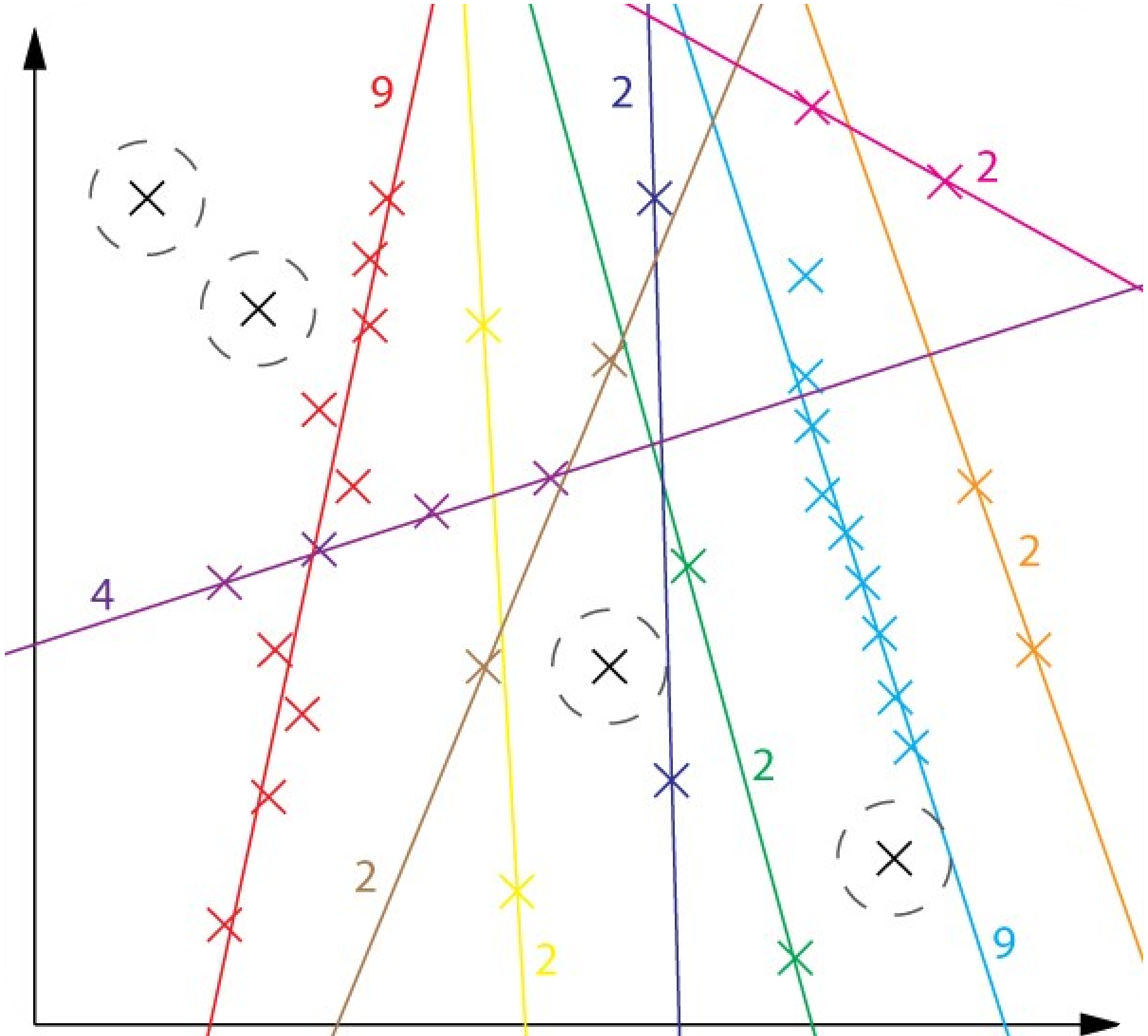
Replacing lines with the corresponding modes.

Algorithm: Labeling



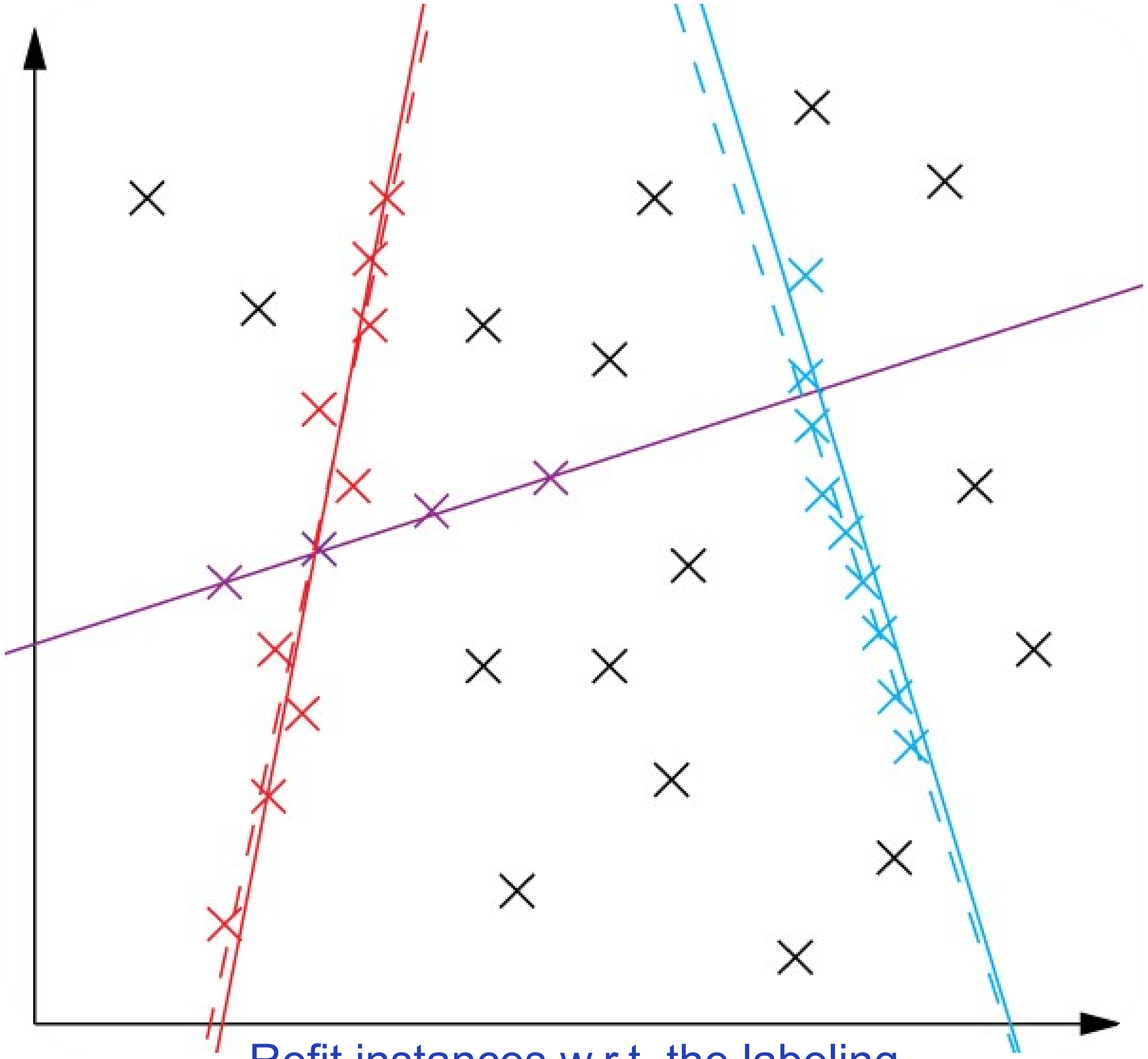
Label points to lines (by color)

Algorithm: Outlier Removal



Remove outliers and instances which have not enough inliers.

Algorithm: Instance Refitting



Refit instances w.r.t. the labeling

Model Description and Generation

$H_0 :=$ InstanceGeneration(P) ; $i := 1$;
repeat
    $H_i :=$ ModeSeeking(H_{i-1}); ▷ by Median-Shift
    $L_i :=$ LabelingToMode(H_i, P); ▷ by α-expansion
    $L_i :=$ OutlierRemoval(H_i, L_i, γ);
    $H_i :=$ ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
    $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}, L^* := L_{i-1}$;
$H^*, L^* :=$ RemoveUnstableModels(H^*, L^*)

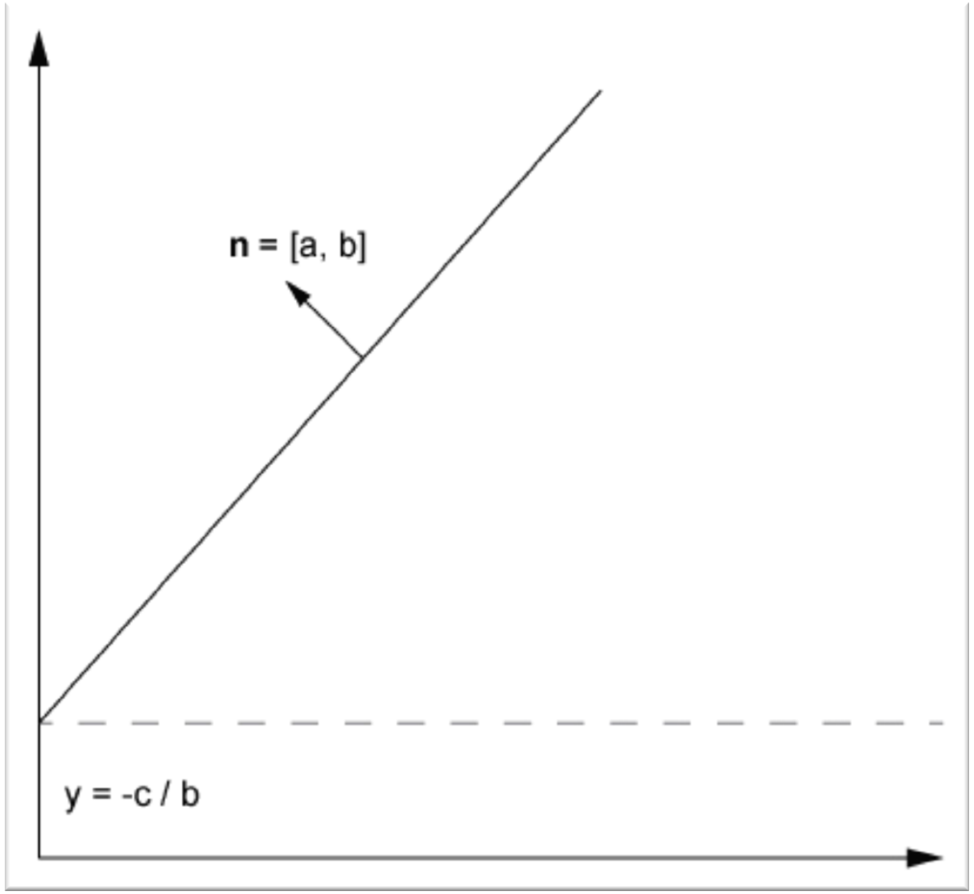
Model Representation (2D Line Example)

Line model 1:

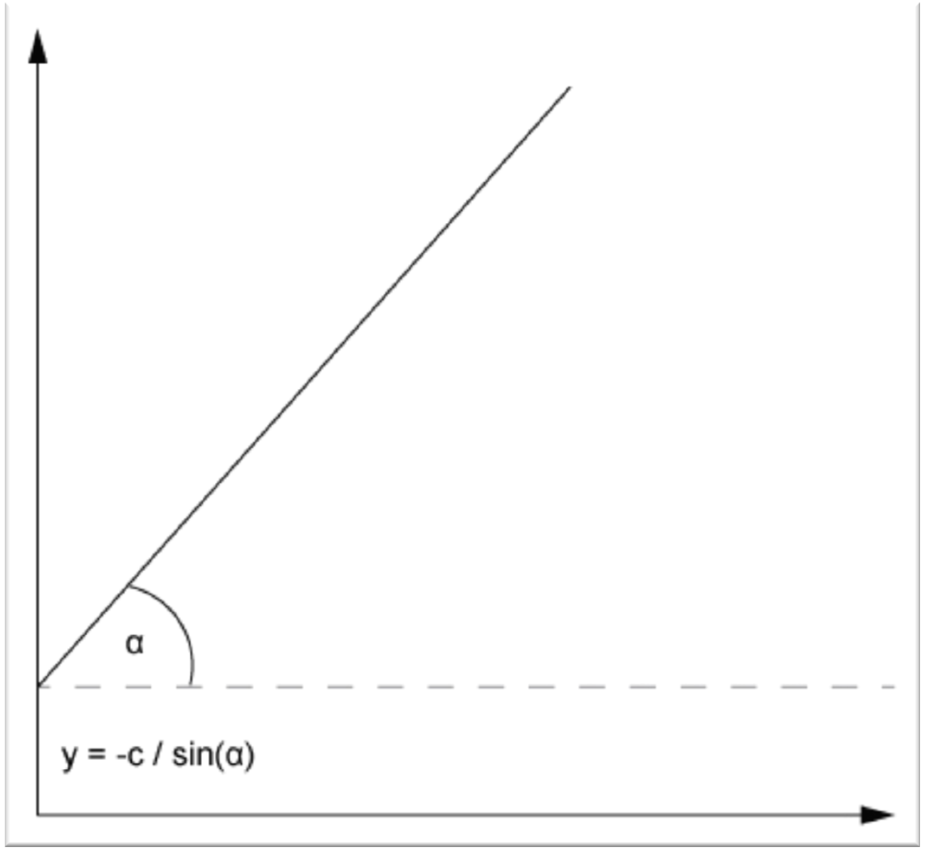
$$(\theta_1, \phi_1)$$

$$\theta_1 = [a \quad b \quad c]^T$$

$$\phi_1(\theta_1, p) = \frac{ax + by + c}{\sqrt{a^2 + b^2}}$$



Model Representation (2D Line Example)



Line model 2:

$$(\theta_2, \phi_2)$$

$$\theta_2 = [\alpha \quad c]^T$$

$$\phi_2(\theta_2, p) = \cos(\alpha)x + \sin(\alpha)y + c$$

Model Representation

Definition 1 (Equivalence of Multi-Class Models) *Multi-Class models* $(\theta_1, \phi_1), (\theta_2, \phi_2) \in \mathcal{H}^*$ *are equivalent over a set of points* $\mathcal{P}$ *if and only if* $\forall p \in \mathcal{P} : \phi_1(p, \theta_1) = \phi_2(p, \theta_2)$.

Model Representation: Two Rules

1. Represent in an **orthonormal coordinate system**, e.g. a 2D line by two points.

2. A **minimal representation** which satisfies the first criterium.

Model Generation

Stochastic Sampling (like RANSAC):

1. Selecting a minimal subset (MSS), e.g. 2 points for a line.

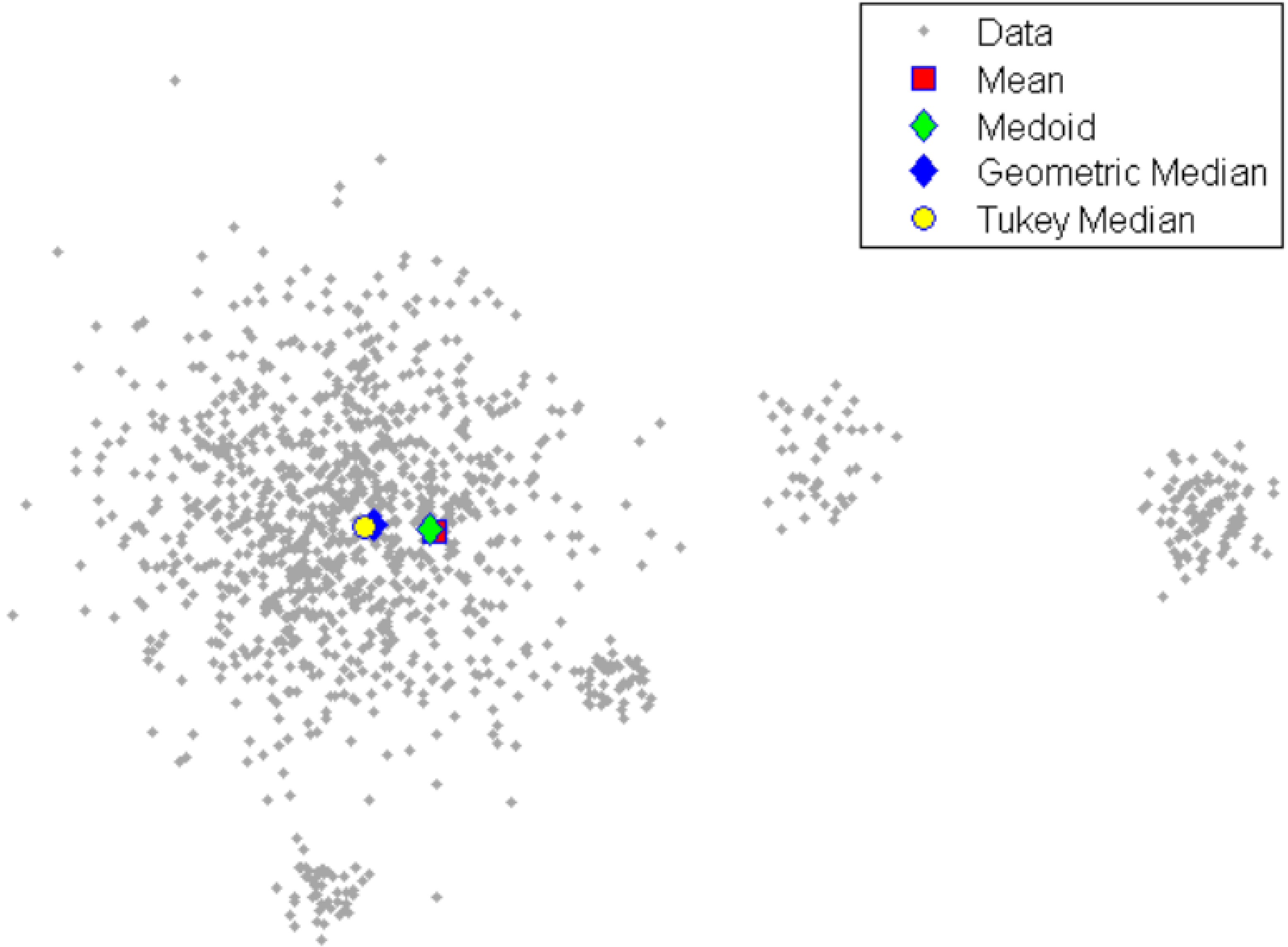
2. Fit the model to the MSS.

3. Start from 1.

Mode-Seeking

$H_0 :=$ InstanceGeneration(P) ; $i := 1$;
repeat
    $H_i :=$ ModeSeeking(H_{i-1}); ▷ by Median-Shift
    $L_i :=$ LabelingToMode(H_i, P); ▷ by α-expansion
    $L_i :=$ OutlierRemoval(H_i, L_i, γ);
    $H_i :=$ ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
    $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}$, $L^* := L_{i-1}$;
H^*, $L^* :=$ RemoveUnstableModels(H^*, L^*)

Mode-Seeking: Mode Types



Shapira, Lior, Shai Avidan, and Ariel Shamir. "Mode-detection via median-shift." *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009.

Mode-Seeking: Clustering Algorithm

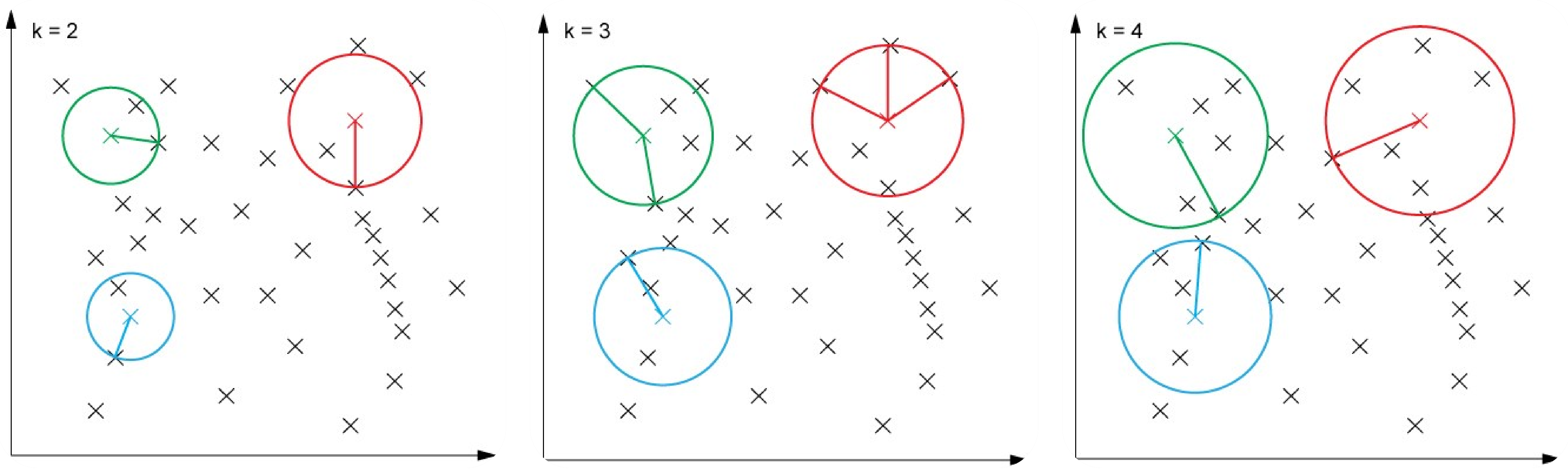
Clustering in arbitrary dimensions:

- **K-Means** is not applicable since the number of modes is unknown.
- **Mean-Shift** is a good choice.
- **Median-Shift** is more robust than Mean-Shift. << we chose this

Median-Shift is applied using Tukey-median.

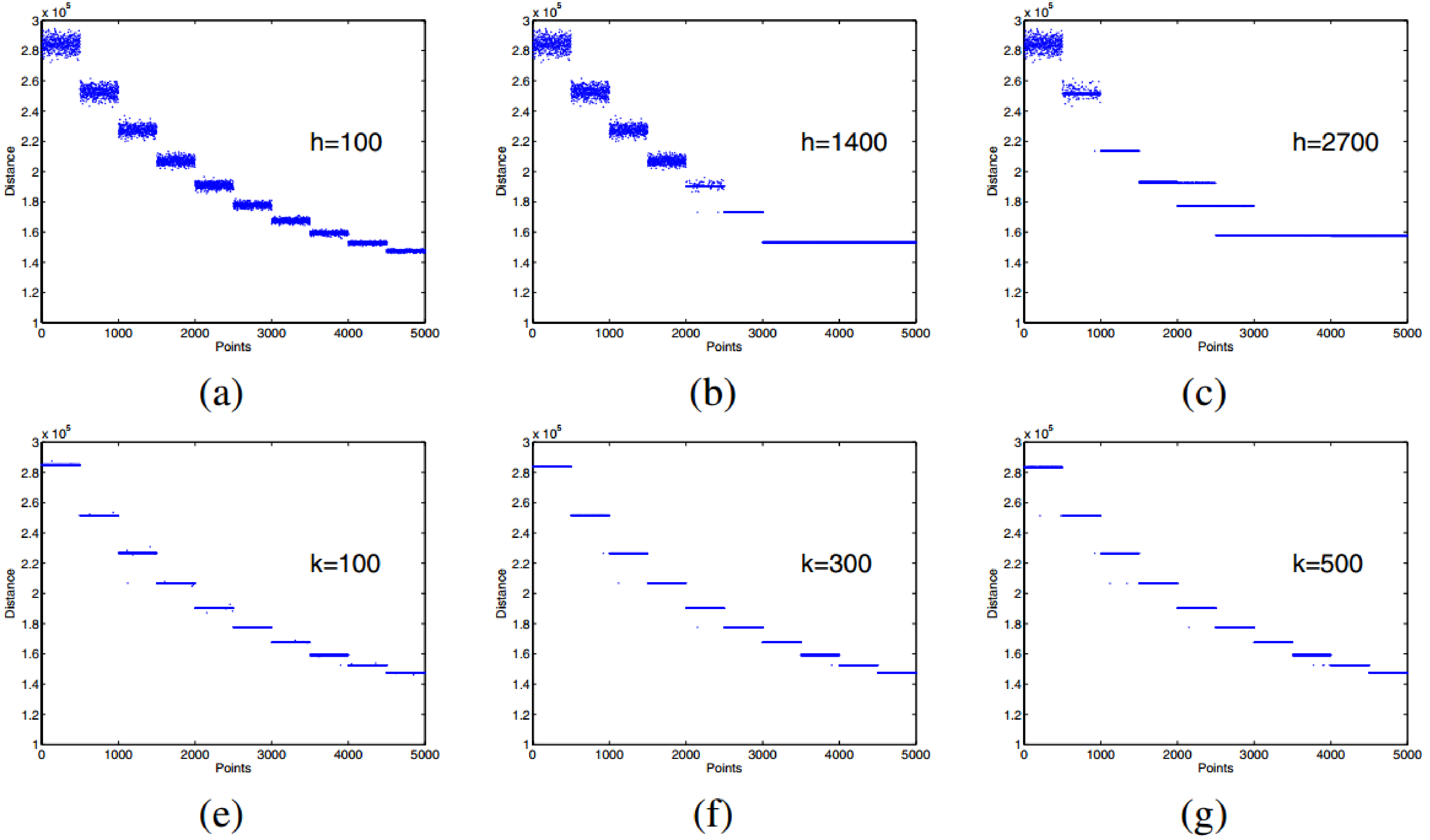
Mode-Seeking: Automatic Parameter Setup

Different bandwidth for all data points determined as the distance from the **k-th nearest neighbor.**



Georgescu, Bogdan, Ilan Shimshoni, and Peter Meer. "Mean shift based clustering in high dimensions: A texture classification example." *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003.

Mode-Seeking: Automatic Parameter Setup



Georgescu, Bogdan, Ilan Shimshoni, and Peter Meer. "Mean shift based clustering in high dimensions: A texture classification example." *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003.

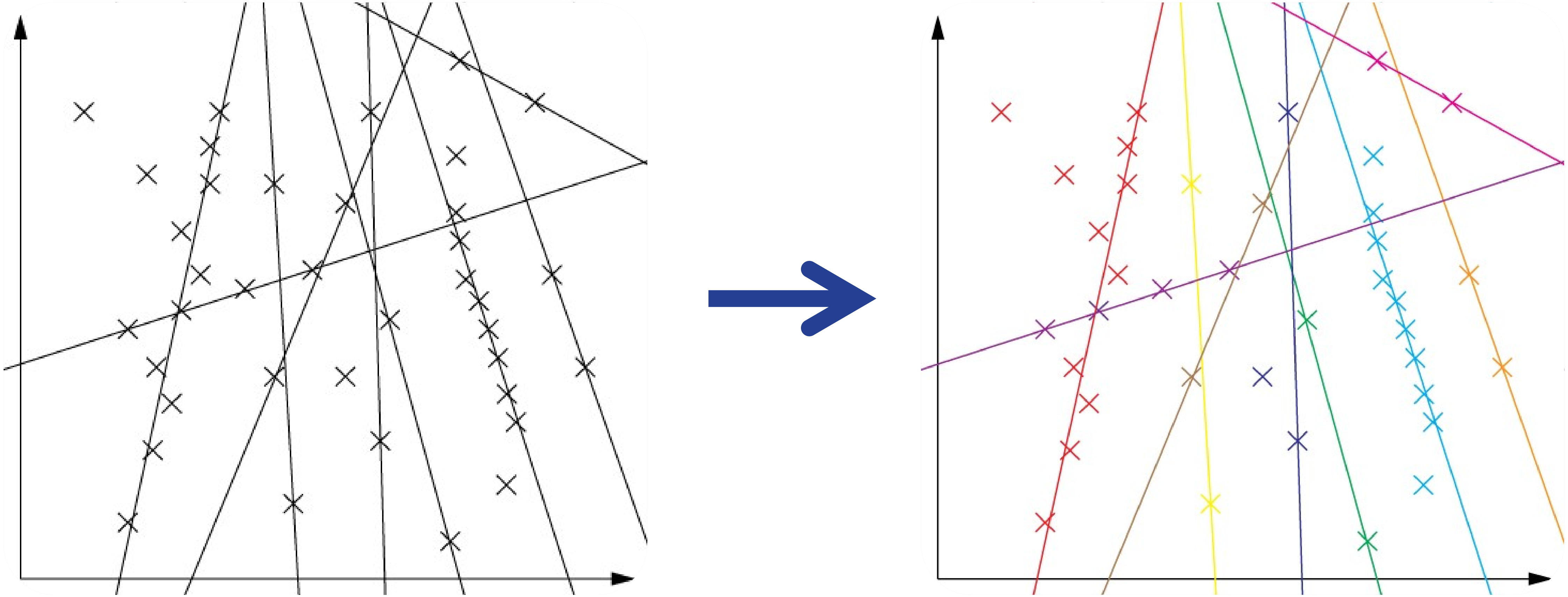
3. Labeling

```
H_0 := InstanceGeneration(P) ; i := 1;
repeat
    H_i := ModeSeeking(H_{i−1});            ▷ by Median-Shift
    L_i := LabelingToMode(H_i, P);          ▷ by α-expansion
    L_i := OutlierRemoval(H_i, L_i, γ);
    H_i := ModelFitting(H_i, L_i, P);       ▷ by Weiszfeld
    i := i + 1;
until !Convergence(H_i, L_i)
H* := H_{i−1}, L* := L_{i−1};
H*, L* := RemoveUnstableModels(H*, L*)
```

Labeling

Each point is labeled to a model instance using α-expansion algorithm minimizing energy

$$\widehat{E}(L) = \widehat{E}_d(L) + \lambda E_s(L) + \beta \widehat{E}_c(L)$$

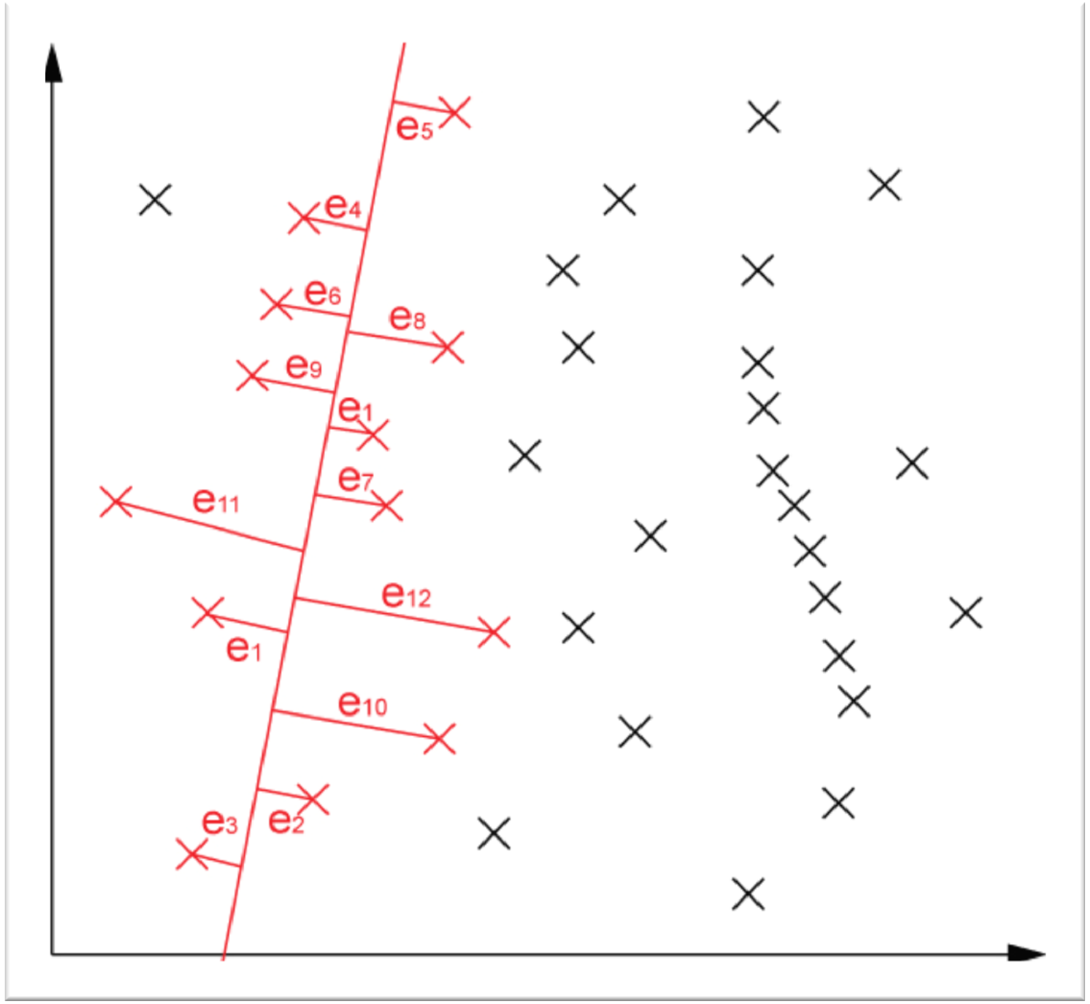


Adaptive Outlier Removal

H_0 := InstanceGeneration(P) ; $i := 1$;
repeat
    H_i := ModeSeeking(H_{i-1}); ▷ by Median-Shift
    L_i := LabelingToMode(H_i, P); ▷ by α-expansion
    L_i := OutlierRemoval(H_i, L_i, γ);
    H_i := ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
    $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}$, $L^* := L_{i-1}$;
H^*, L^* := RemoveUnstableModels(H^*, L^*)

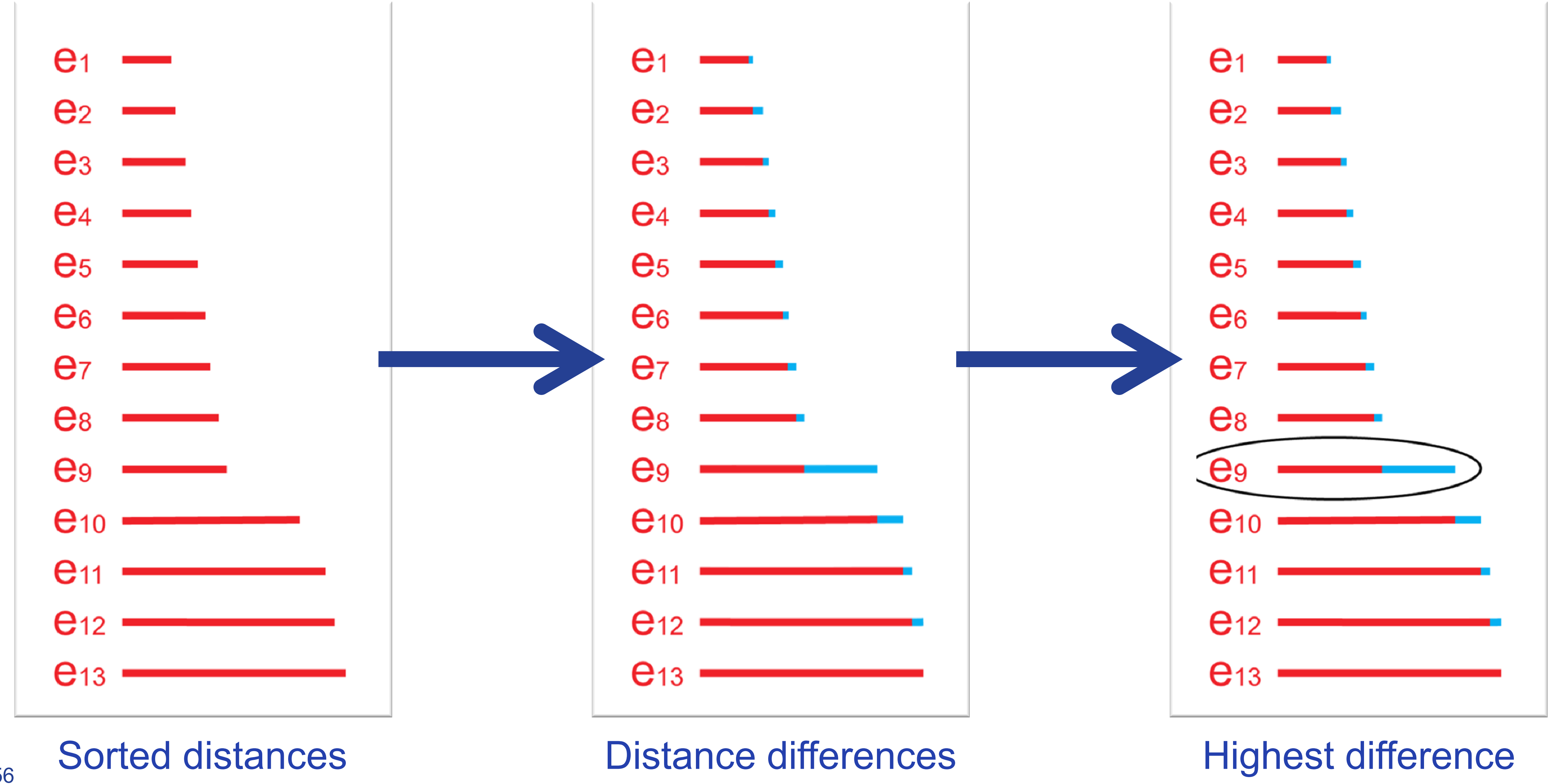
Adaptive Outlier Removal

Removal of data points too far from the assigned model.

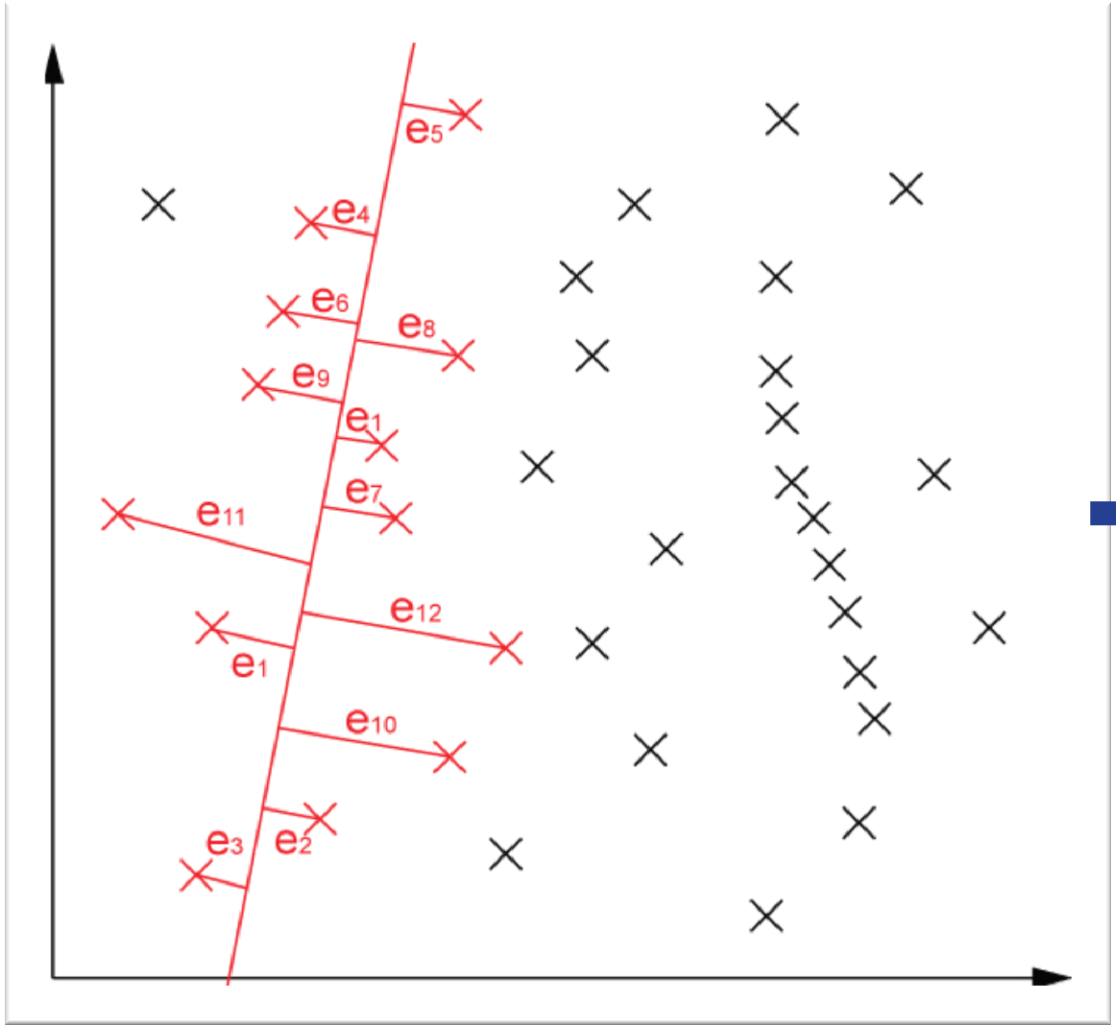


Original labeling

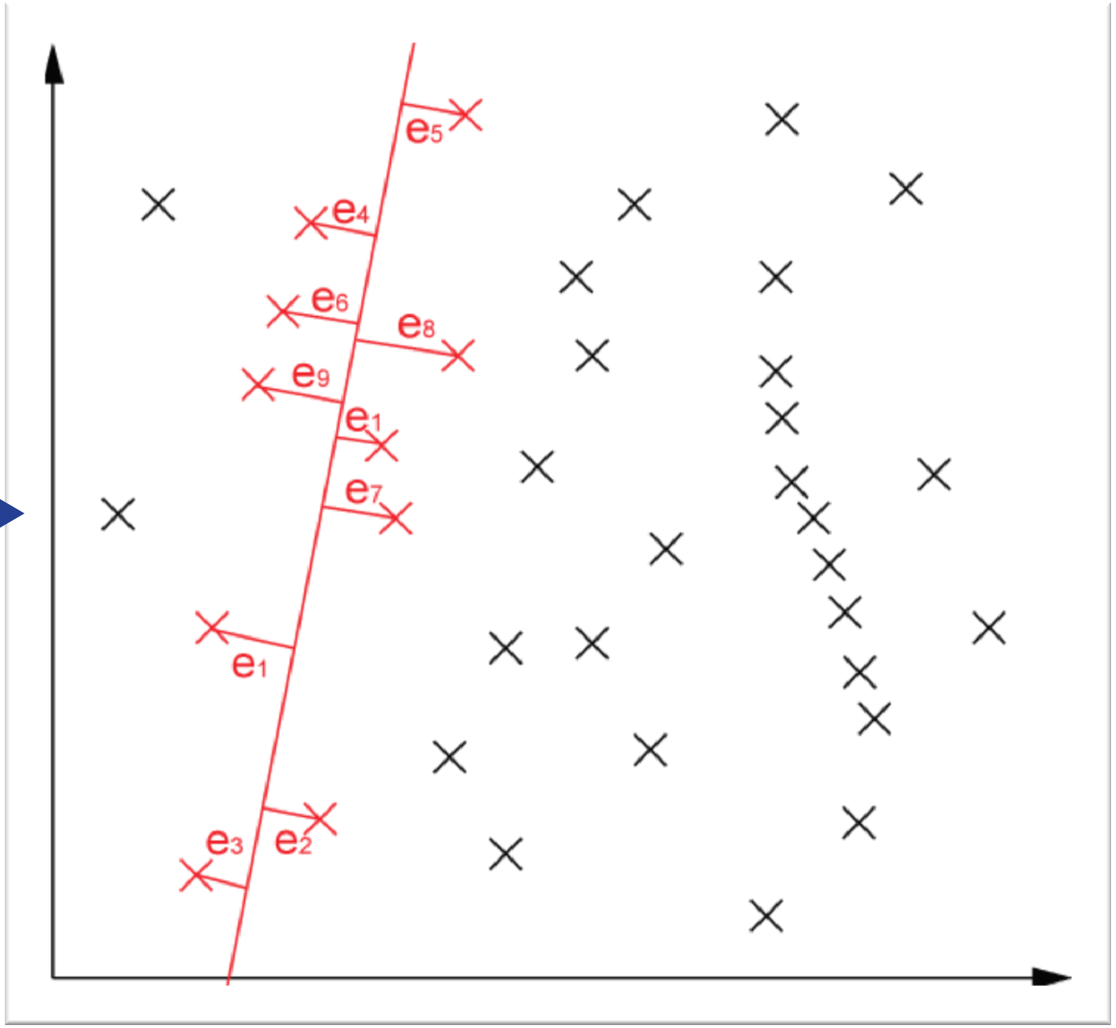
Adaptive Outlier Removal



Adaptive Outlier Removal



Original labeling



Labeling without outliers

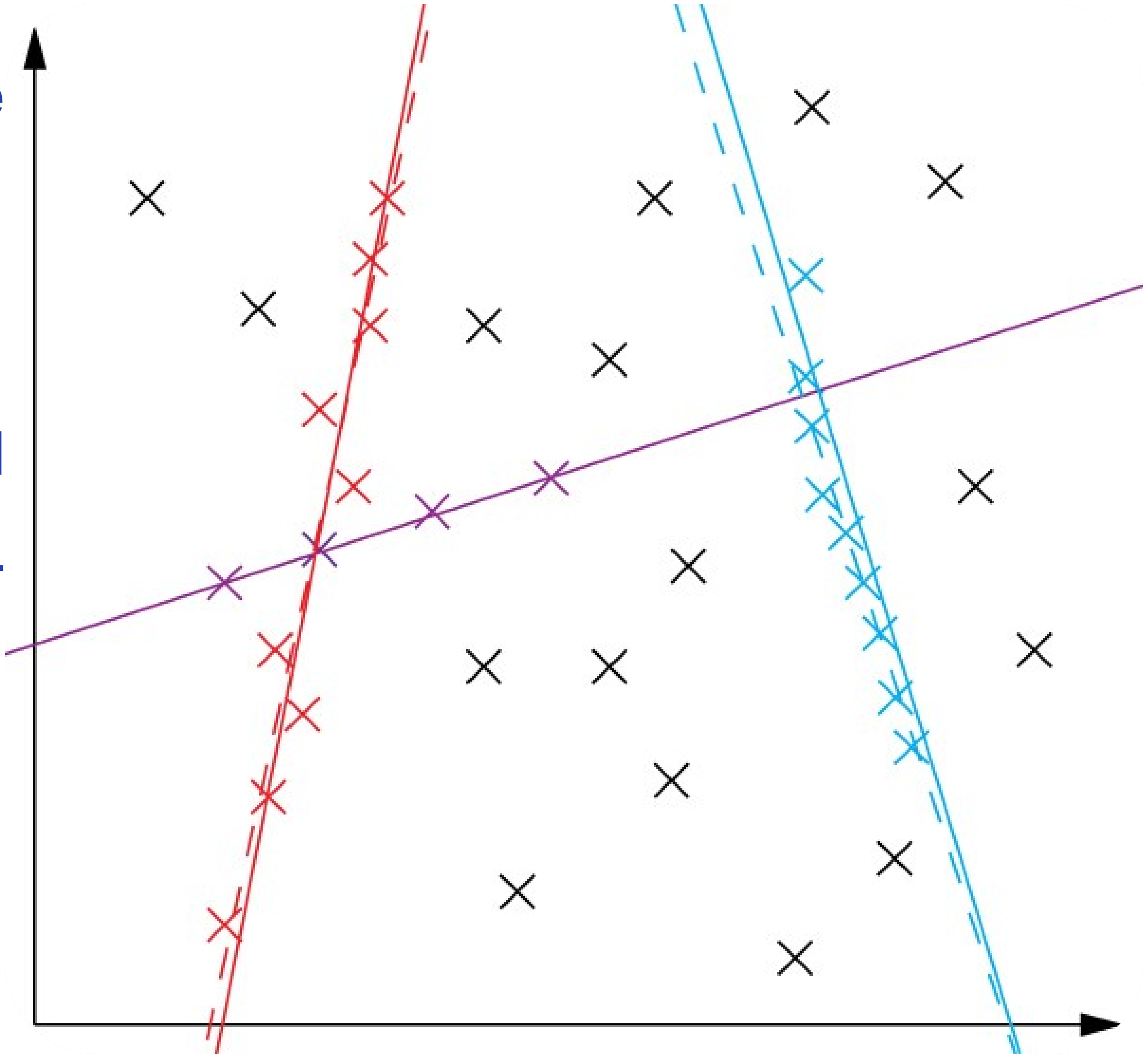
Model Fitting

$H_0 :=$ InstanceGeneration(P) ; $i := 1$;
repeat
    $H_i :=$ ModeSeeking(H_{i-1}); ▷ by Median-Shift
    $L_i :=$ LabelingToMode(H_i, P); ▷ by α-expansion
    $L_i :=$ OutlierRemoval(H_i, L_i, γ);
    $H_i :=$ ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
    $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}, L^* := L_{i-1}$;
$H^*, L^* :=$ RemoveUnstableModels(H^*, L^*)

Model Fitting

The task is to **update the instance parameters** using the obtained labeling.

L_1 model fitting using Weiszfeld algorithm (iteratively re-weighted least-squares).



Convergence

H_0 := InstanceGeneration(P) ; $i := 1$;
repeat
  H_i := ModeSeeking(H_{i-1}); ▷ by Median-Shift
  L_i := LabelingToMode(H_i, P); ▷ by α-expansion
  L_i := OutlierRemoval(H_i, L_i, γ);
  H_i := ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
  $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}$, $L^* := L_{i-1}$;
H^*, L^* := RemoveUnstableModels(H^*, L^*)

Convergence

Due to the mode-seeking **the energy can increase**, thus the **convergence** have to be defined **over the full state** of the algorithm.

Definition 1 (State) *The state* $\in \mathbb{N} \times \mathbb{R}$ *of Multi-X is a pair of numbers, where* $\mathbb{N}$ *and* $\mathbb{R}$ *represent the number of instances* $|\mathcal{H}_i|$ *and the value of the energy* $E(L_i)$*), respectively.*

Convergence

1. **Mode-Seeking:**
   1. Instance number must decrease or hold.
   2. The energy can increase.

2. **Labeling:**
   1. Instance number does not change.
   2. Energy must decrease or hold.

3. **Outlier Removal:**
   1. Instance number does not change.
   2. Energy can’t increase.

4. **Model Fitting:**
   1. Instance number does not change.
   2. Energy must decrease or hold.

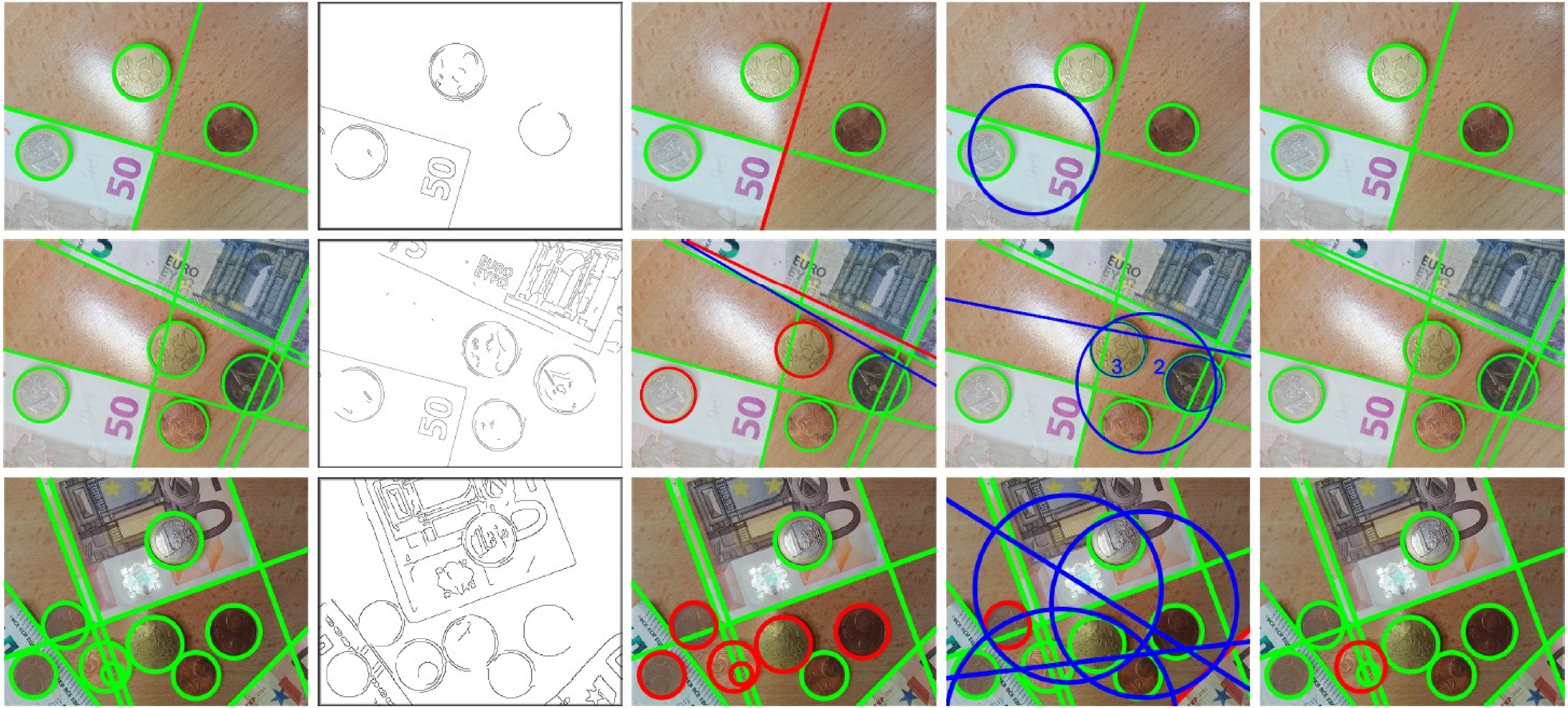
Convergence is ensured since the **number** of possible **labelings is finite** and the model **instance number** monotonically **decrease**.

Convergence is reached when

$$(n_i, e_i) = (n_{i+1}, e_{i+1})$$

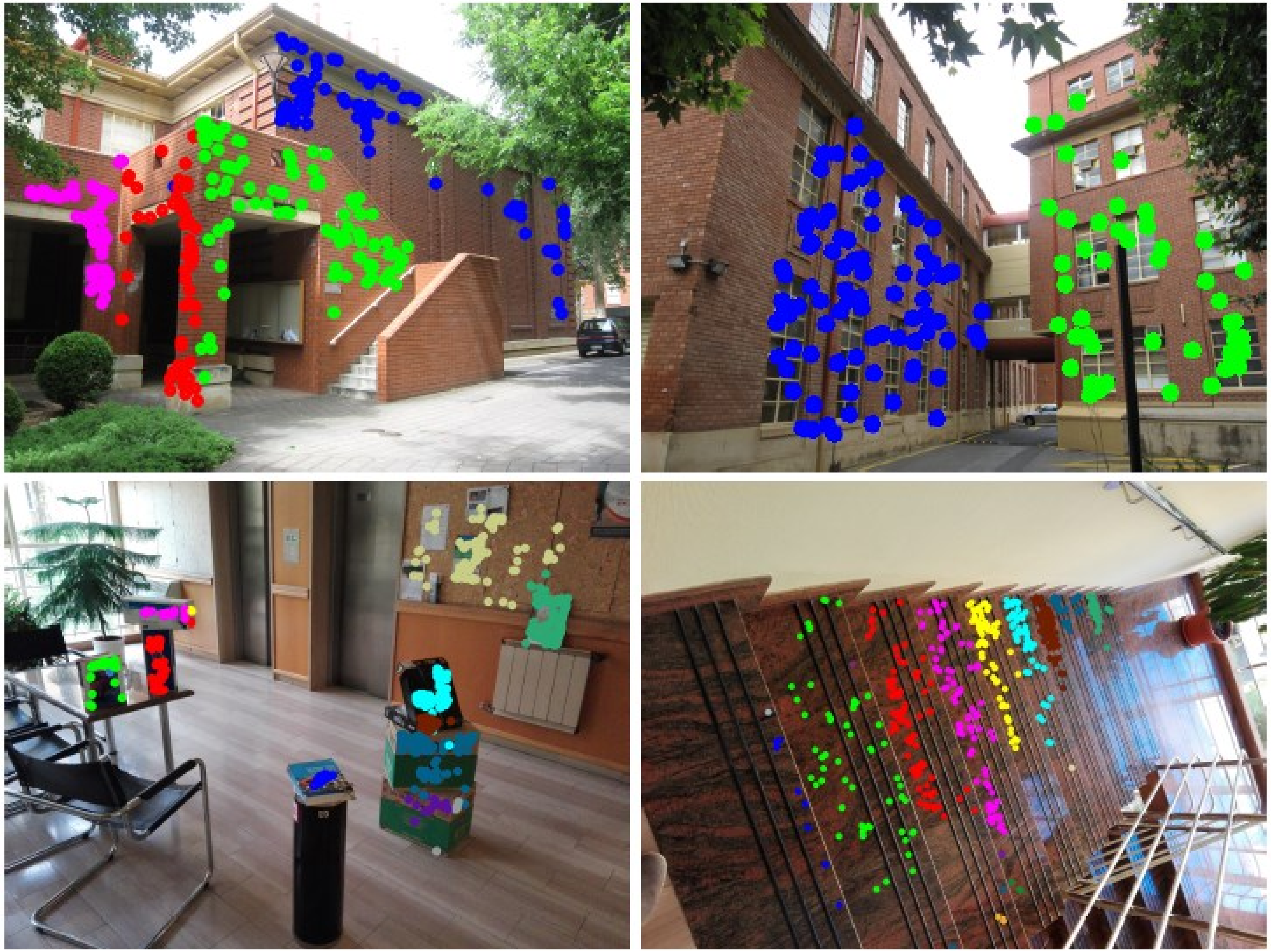
Experimental Results

Tests – Lines and Circles



(a) Ground truth (b) Edge map (c) T-Linkage (d) PEARL (e) **Multi-X**

Tests – Homographies



Top row: AdelaideRMF dataset, bottom row: Multi-H dataset.
Points assigned to planes by color.

Tests – Homographies

	# of planes	PEARL	FLOSS	T-Lnkg	ARJMC	RCMSA	J-Lnkg	**Multi-X**
(1)	4	4.02	4.16	4.02	6.48	5.90	5.07	**3.75**
(2)	6	18.18	18.18	18.17	21.49	17.95	18.33	**4.46**
(3)	2	5.49	5.91	5.06	5.91	7.17	9.25	**0.00**
(4)	3	5.39	5.39	3.73	8.81	5.81	3.73	**0.00**
(5)	2	1.58	1.85	0.26	1.85	2.11	0.27	**0.00**
(6)	2	0.80	0.80	0.40	0.80	0.80	0.84	**0.00**
Avg.		5.91	6.05	5.30	7.56	6.62	6.25	**1.37**
Med.		4.71	4.78	3.87	6.20	5.86	4.40	**0.00**

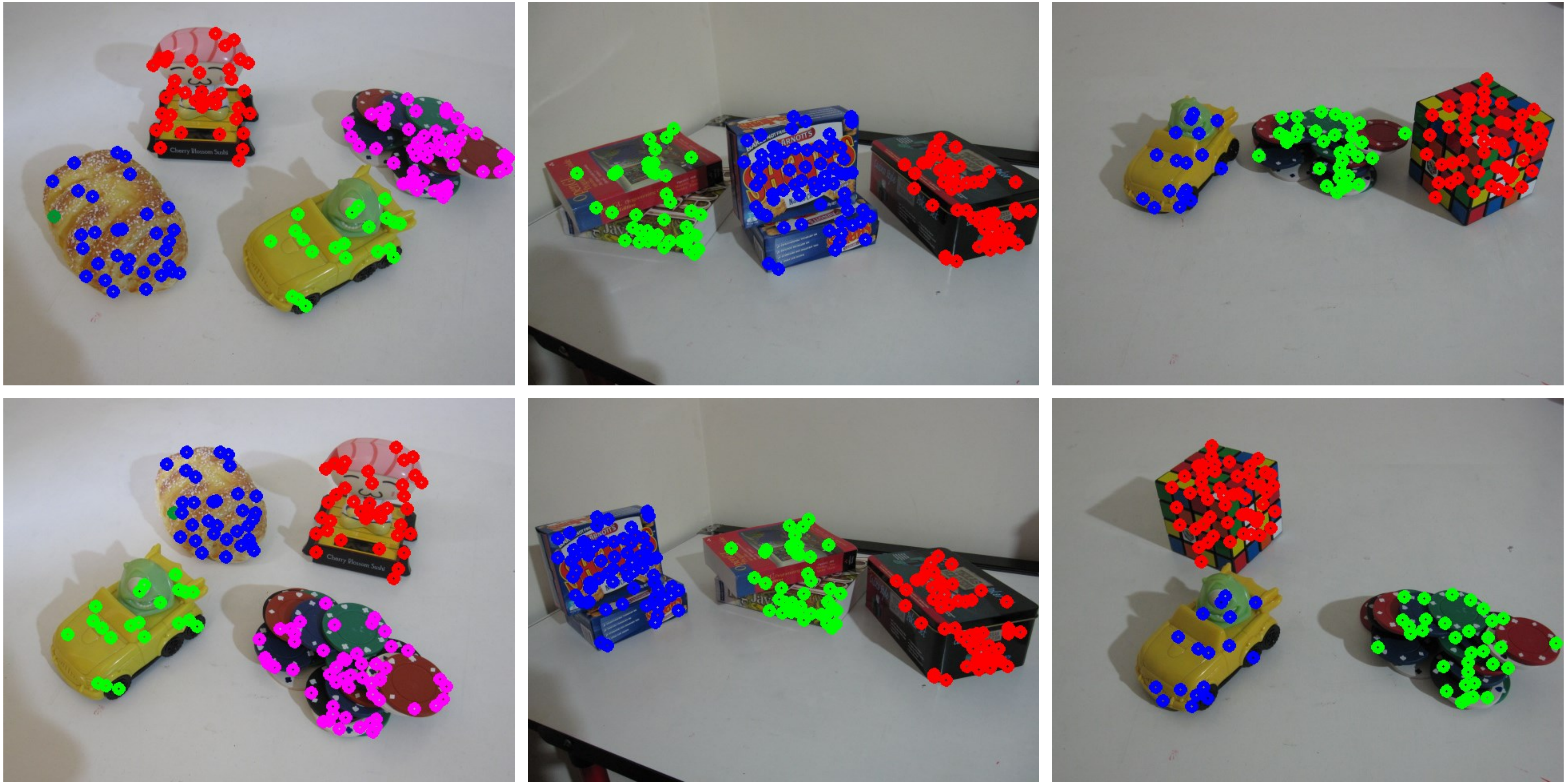
Misclassification error (%) for the two-view plane segmentation on AdelaideRMF test pairs.
All methods, including Multi-X, are tuned separately for each test.

Tests – Homographies

	T-Lnkg	RCMSA	RPA	Multi-H	**Multi-X**
Avg.	44.68	23.17	15.71	14.35	**9.72**
Med.	44.49	24.53	15.89	9.56	**2.49**

Misclassification errors (%, average and median) for two-view plane segmentation on all the 19 pairs from AdelaideRMF test pairs using fixed parameters.

Tests – Two-view Motions



Tests – Two-view Motions

	KF		RCG		T-Lnkg		AKSWH		MSH		**Multi-X**	
	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.
(1)	8.42	4.23	13.43	9.52	5.63	2.46	4.72	2.11	3.80	2.11	**3.45**	**1.41**
(2)	12.53	2.81	13.35	10.92	5.62	4.82	7.23	4.02	3.21	1.61	**2.27**	**0.40**
(3)	14.83	4.13	12.60	8.07	4.96	1.32	5.45	1.42	2.69	0.83	**1.45**	**0.41**
(4)	13.78	5.10	9.94	3.96	7.32	3.54	7.01	5.18	3.72	1.22	**0.61**	**0.30**
(5)	16.87	14.55	26.51	19.54	**4.42**	4.00	9.04	8.43	6.63	4.55	5.24	**1.80**
(6)	16.06	14.29	16.87	14.36	1.93	1.16	8.54	4.99	1.54	1.16	**0.62**	**0.00**
(7)	33.43	21.30	26.39	20.43	**1.06**	0.86	7.39	3.41	1.74	0.43	5.32	**0.00**
(8)	31.07	22.94	37.95	20.80	3.11	3.00	14.95	13.15	4.28	3.57	**2.63**	**1.52**

Misclassification errors (%) for two-view motion segmentation on the AdelaideRMF dataset.

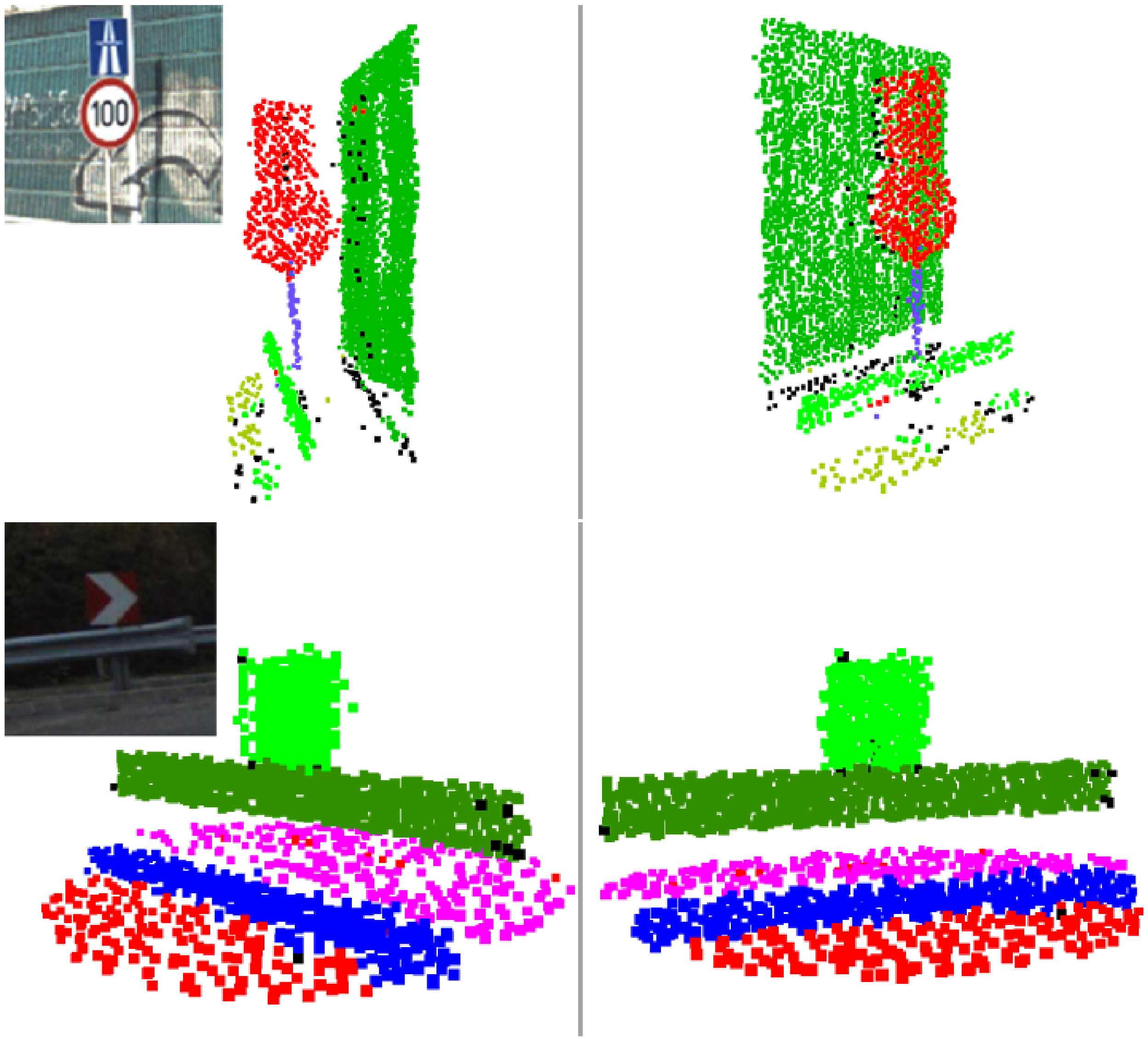
All methods, including Multi-X, are tuned separately for each test.

Tests – Two-view Motions

	RPA	RCMSA	T-Lnkg	AKSWH	**Multi-X**
Avg.	5.62	9.71	43.83	12.59	**2.97**
Med.	4.58	8.48	39.42	11.57	**0.00**

Misclassification errors (%, average and median) for two-view motion segmentation on all the 21 pairs from the AdelaideRMF dataset using fixed parameters.

Tests – Planes and Cylinders



Tests – Planes and Cylinders

	PEARL	T-Lnkg	RPA	**Multi-X**
(1)	10.63	57.46	46.83	**8.89**
(2)	10.88	41.79	53.39	**4.72**
(3)	37.34	52.97	61.64	**2.84**
(4)	38.13	38.91	41.41	**19.38**
(5)	17.20	51.83	53.34	**16.83**
(6)	**17.35**	61.77	51.21	21.72
(7)	6.12	12.49	80.45	**5.72**

Misclassification error (%) of simultaneous plane and cylinder fitting to LIDAR data.
All methods, including Multi-X, are tuned separately for each test.

Tests – Motions in video sequences

Tests – Motions in video sequences

		(1)	(2)	(3)	(4)	(5)
SSC	Avg.	0.06	0.76	3.95	2.13	1.08
	Med.	**0.00**	**0.00**	**0.00**	2.13	**0.00**
T-Lnkg	Avg.	1.31	0.48	6.47	5.32	2.47
	Med.	**0.00**	0.19	2.38	5.32	**0.00**
RPA	Avg.	0.14	0.19	4.41	9.11	1.42
	Med.	**0.00**	**0.00**	2.44	9.11	**0.00**
Grdy-RC	Avg.	7.48	28.65	8.75	14.89	10.91
	Med.	**0.00**	1.53	0.20	14.89	**0.00**
ILP-RC	Avg.	0.54	0.35	2.40	2.13	0.98
	Med.	**0.00**	0.19	1.30	2.13	**0.00**
J-Lnkg	Avg.	1.75	1.58	5.32	6.91	2.70
	Med.	**0.00**	0.34	1.30	6.91	**0.00**
Multi-X	Avg.	**0.05**	**0.09**	**0.32**	**1.06**	**0.16**
	Med.	**0.00**	**0.00**	**0.00**	**1.06**	**0.00**

Misclassification errors (%, average and median) for multi-motion detection on 51 videos of Hopkins dataset.
All methods, including Multi-X, are tuned separately for each test.

Processing Time

#	(1)		(2)		(3)		(4)		(5)	
	M	T	M	T	M	T	M	T	M	T
100	0.05	0.39	0.11	0.25	0.12	0.26	0.02	0.19	0.08	0.42
500	1.97	14.00	3.22	8.42	2.05	8.36	0.78	6.96	3.81	15.86
1000	5.13	102.76	-	-	-	-	-	-	7.45	120.91

Processing times (sec) of Multi-X (M) and T-Linkage (T) for the problem of fitting (1) lines and circles, (2) homographies, (3) two-view motions, (4) video motions, and (5) planes and cylinders. The number of data point is shown in the first column.

Conclusions

1. Simultaneous fitting of models is an old open problem.

2. A novel method for the multi-class multi-instance method was proposed.

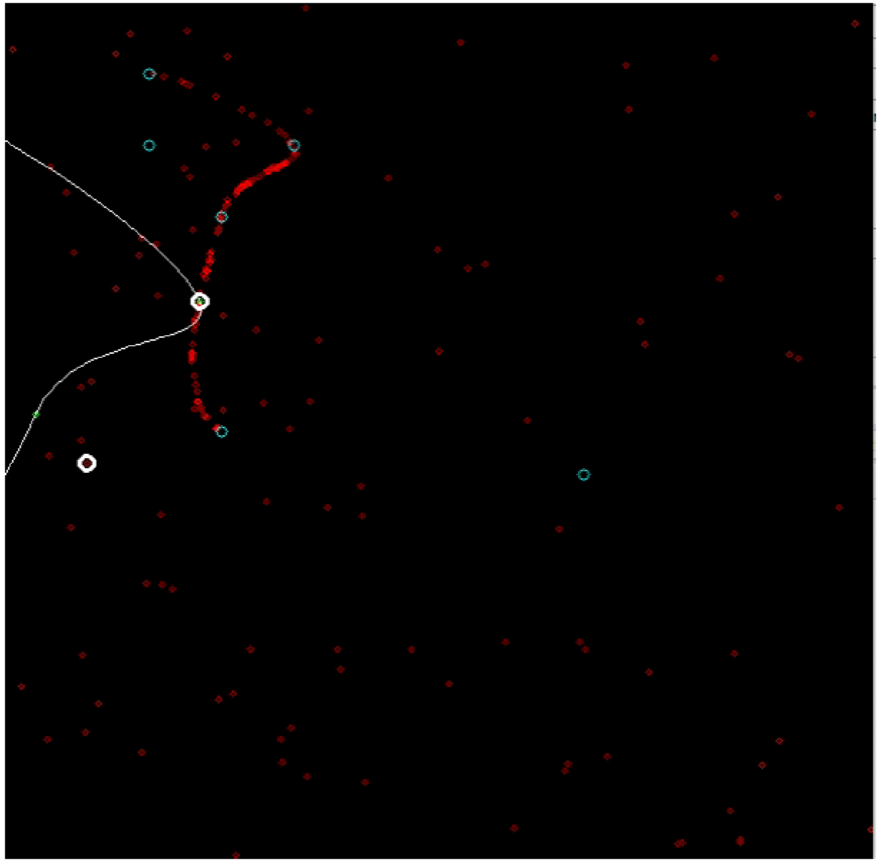
3. Energy minimization combined with mode seeking for multi model fitting outperforms the state of the art on several problems.

4. Automatic parameter setting makes the proposed method applicable to real world tasks without high effort on manual parameter tuning.

Work in Progress

Multiple free-form surface (3D) and curve (2D) fitting.

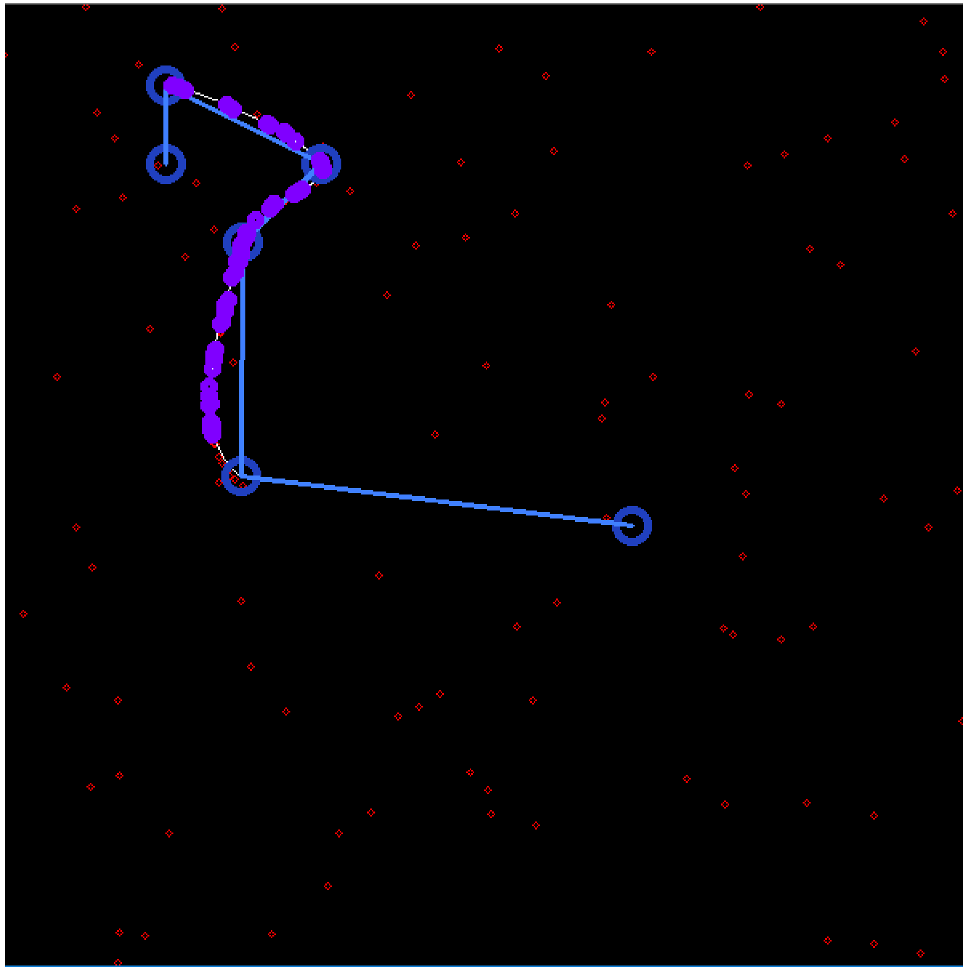
A possible application: car fitting to LIDAR point cloud.



Work in Progress

Multiple free-form surface (3D) and curve (2D) fitting.

A possible application: car fitting to LIDAR point cloud.



Thank you for your attention!

Questions, please?

Paper will be available on arXiv later today.