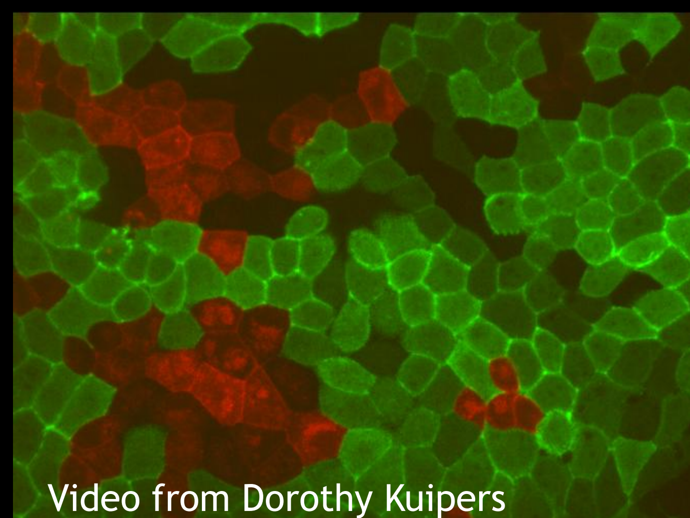
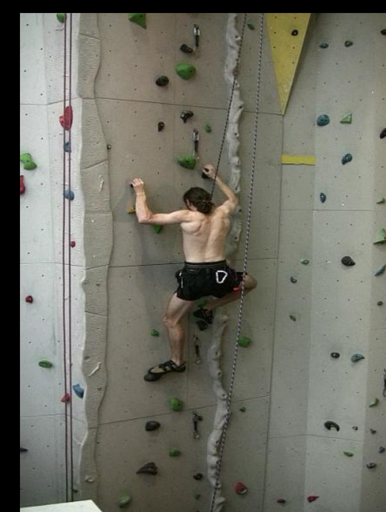
Segmenting Video Into Classes of Algorithm-Suitability

Oisin Mac Aodha (UCL)
Gabriel Brostow (UCL)
Marc Pollefeys (ETH)

Which algorithm should I (use / download / implement) to track things in <u>this</u> video?





The Optical Flow Problem

- #2 all-time Computer Vision problem (disputable)
- "Where did each pixel go?"

Optical Flow Solutions

- Compared against each other on the "blind" Middlebury test set

Optical flow evaluation results — **Statistics:** Average SD R0.5 R1.0 R2.0 A50 A75 A95 — **Error type:** angle end-point interpolation normalized interpolation

Average end-point error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive [23]	5.2	**0.09** 1	**0.26** 1	**0.06** 1	0.23 7	0.78 6	0.18 6	0.54 11	1.19 13	0.21 5	**0.18** 1	0.91 3	**0.10** 1	0.88 4	1.25 4	0.73 6	0.50 3	1.28 3	0.31 3	0.14 12	0.16 14	0.22 11	0.65 3	1.37 3	0.79 4
Complementary OF [24]	6.6	0.11 6	0.28 5	0.10 10	**0.18** 1	0.63 2	**0.12** 1	0.31 4	0.75 4	**0.18** 1	0.19 2	0.97 5	0.12 4	0.97 11	1.31 7	1.00 12	1.78 23	1.73 9	0.87 17	0.11 5	0.12 3	0.22 11	0.68 4	1.48 4	0.95 8

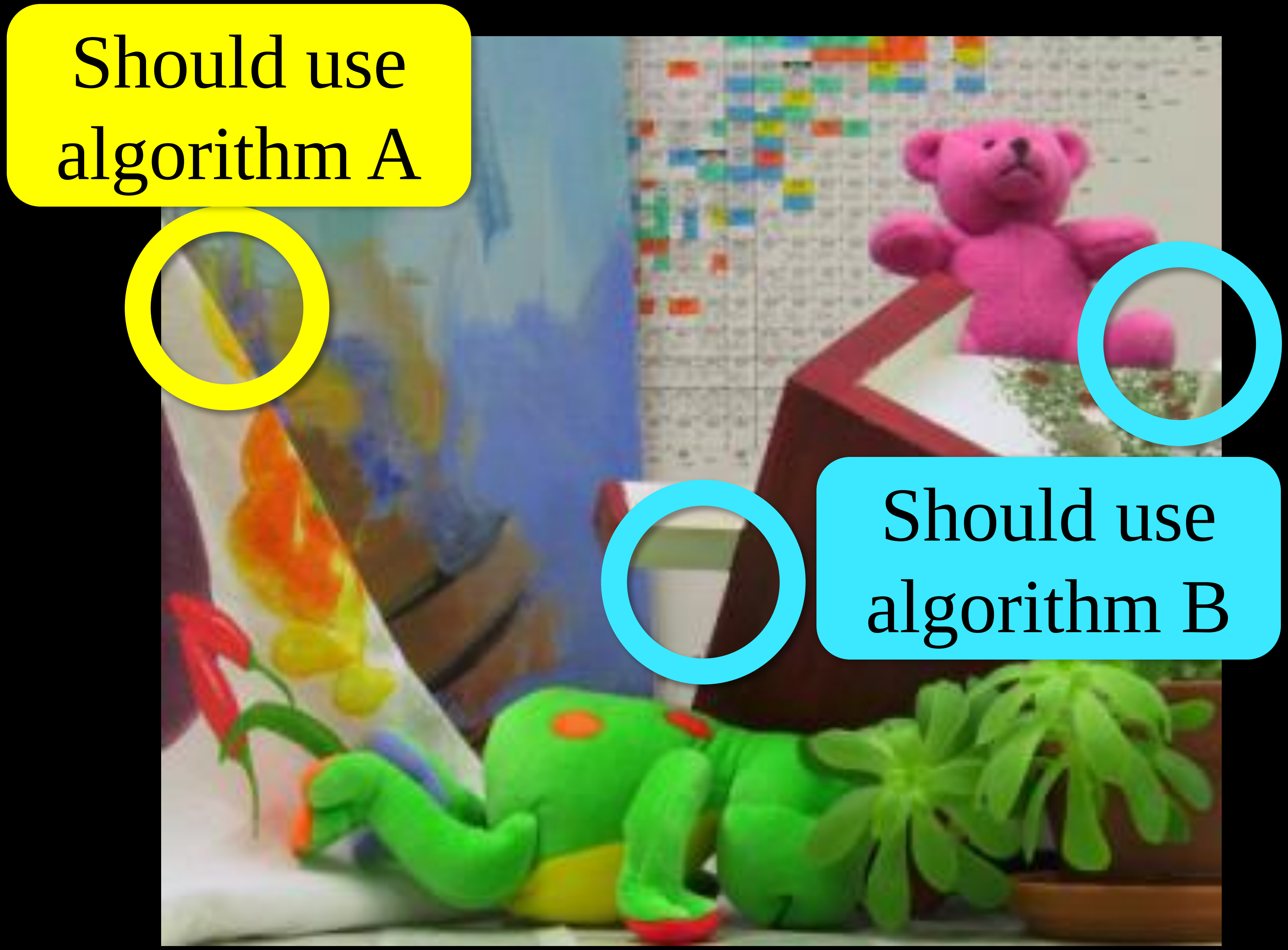
Average end-point error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive [23]	5.2	**0.09** 1	**0.26** 1	**0.06** 1	0.23 7	0.78 6	0.18 6	0.54 11	1.19 13	0.21 5	**0.18** 1	0.91 3	**0.10** 1
Complementary OF [24]	6.6	0.11 6	0.28 5	0.10 10	**0.18** 1	0.63 2	**0.12** 1	0.31 4	0.75 4	**0.18** 1	0.19 2	0.97 5	0.12 4
Aniso. Huber-L1 [25]	7.0	0.10 3	0.28 5	0.08 3	0.31 14	0.88 10	0.28 15	0.56 13	1.13 10	0.29 15	0.20 5	0.92 4	0.13 7
DPOF [18]	7.4	0.13 15	0.35 15	0.09 5	0.25 8	0.79 7	0.19 7	0.24 1	0.49 1	0.21 5	0.19 2	0.62 1	0.15 13

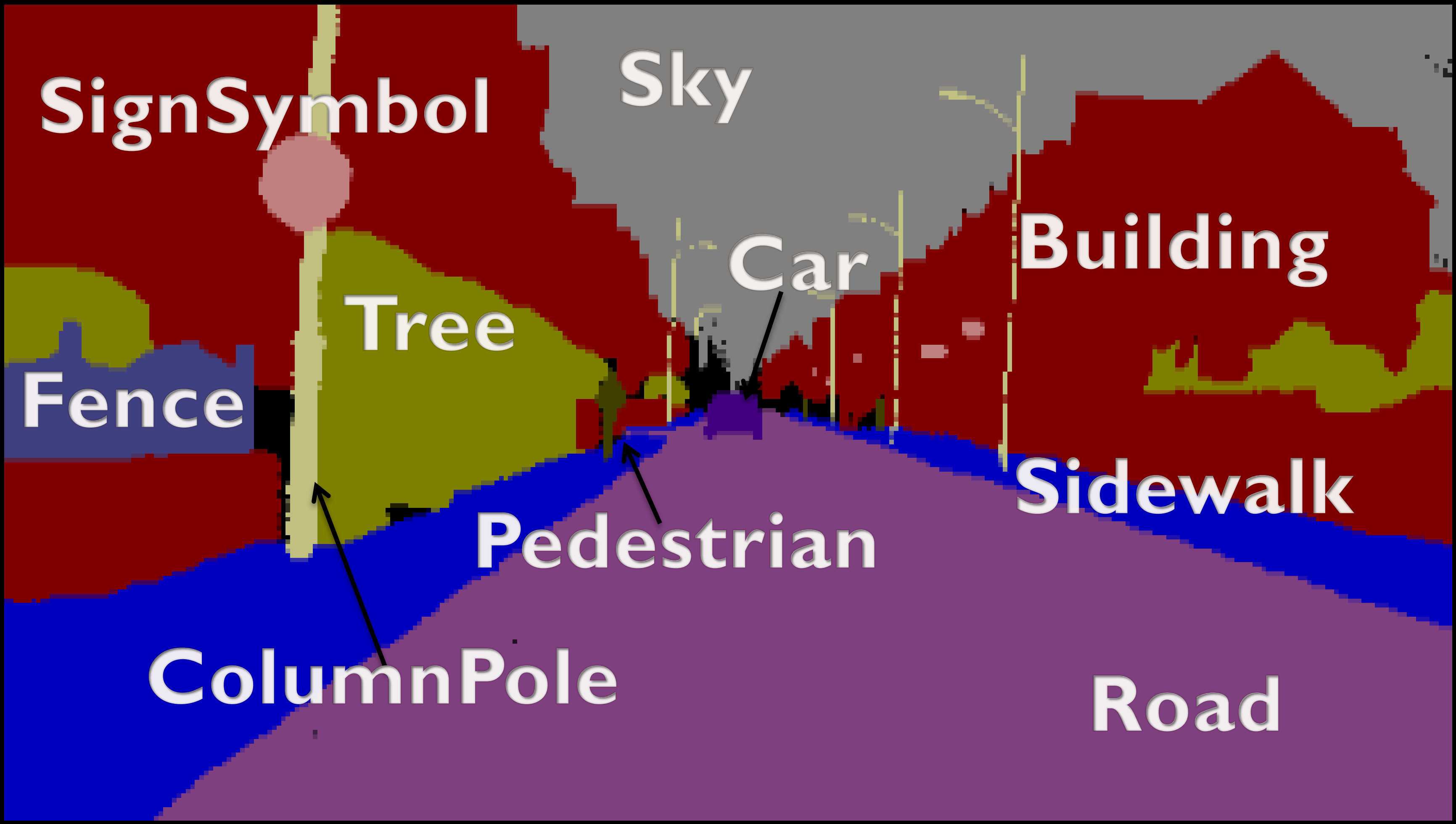
1st Best Algorithm (7th overall as of 17-12-2009)

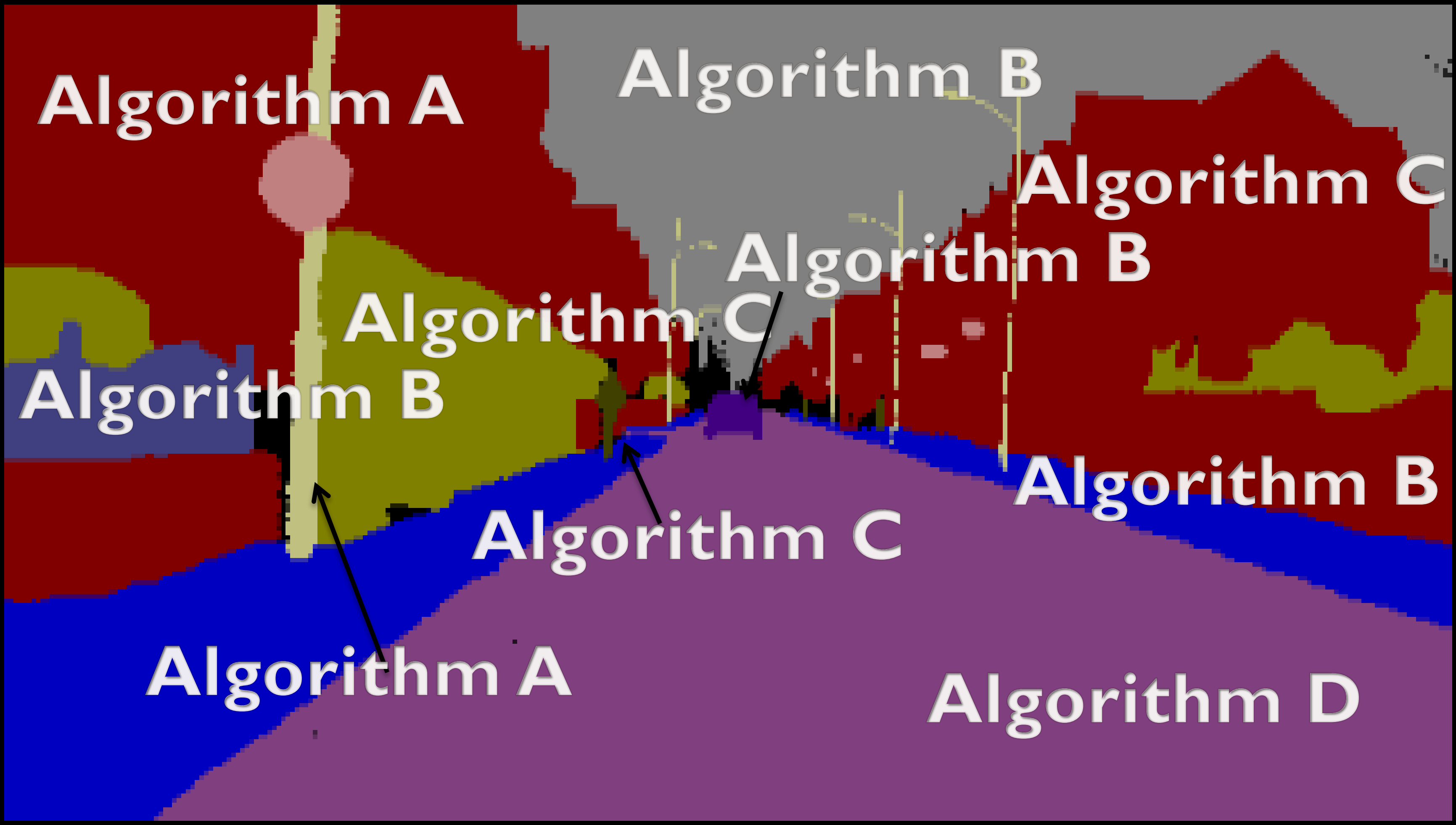
DPOF [18]: C. Lei and Y.-H. Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. ICCV 2009

(3rd overall as of 17-12-2009) 2nd Best Algorithm

Classic+Area [31]: Anonymous. Secrets of optical flow estimation and their principles. CVPR 2010 submission 477


Should use algorithm A
Should use algorithm B


Sky
SignSymbol
Building
Car
Tree
Fence
Sidewalk
Pedestrian
ColumnPole
Road



(Artistic version; object-boundaries don't interest us)

Hypothesis:

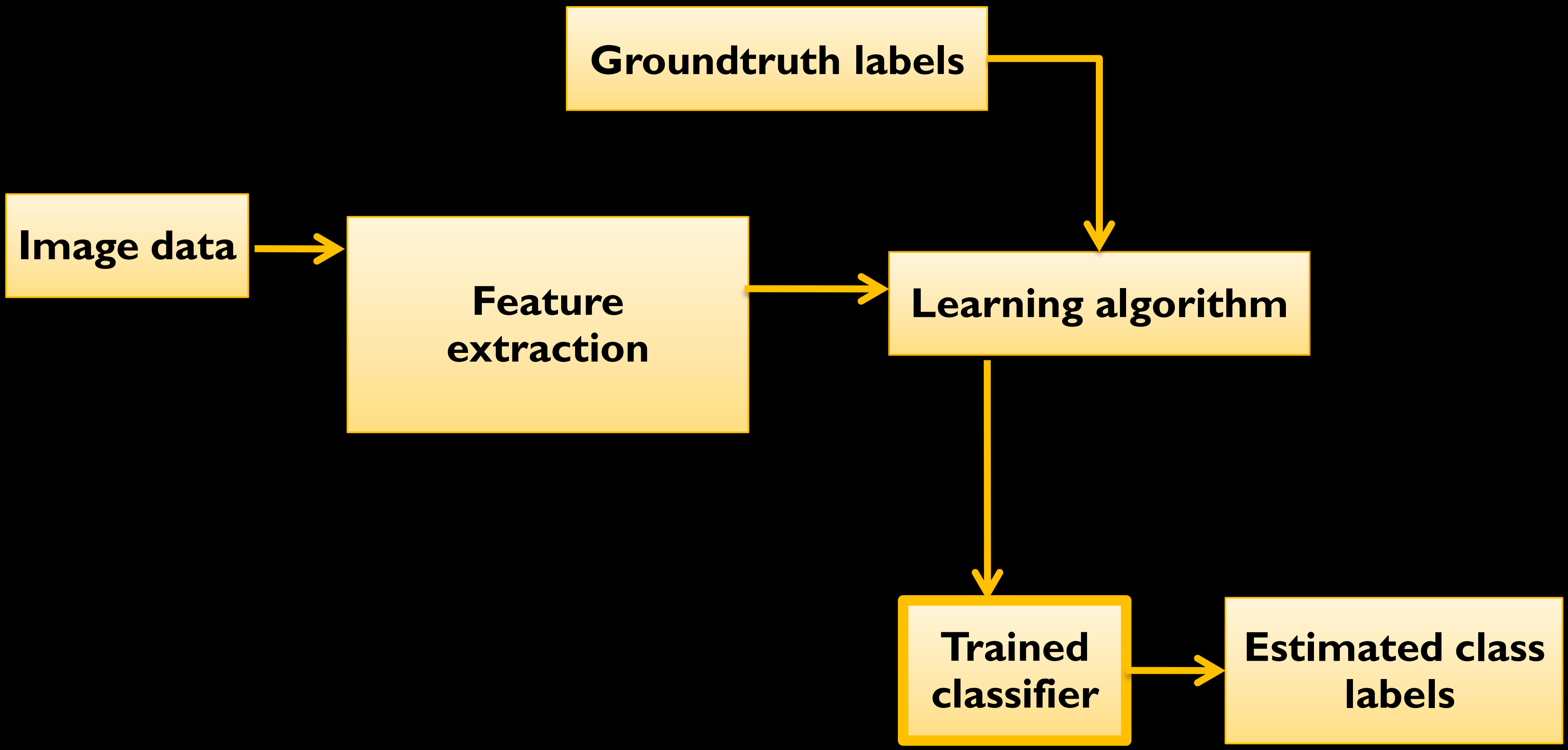
- that the most suitable algorithm can be chosen for each video automatically, through supervised training of a classifier

Hypothesis:

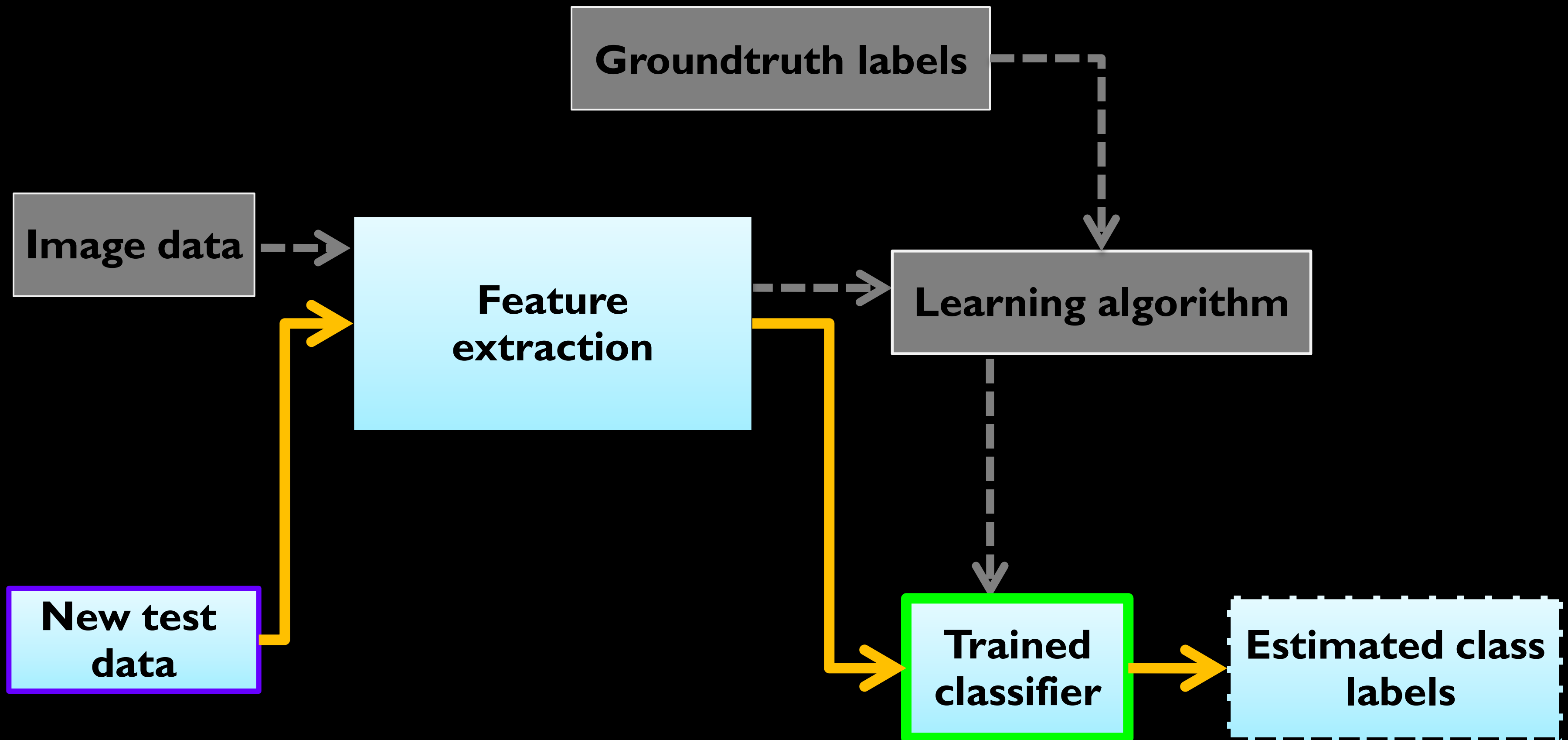
- ~~that the most suitable algorithm can be chosen for each video automatically, through supervised training of a classifier~~

- that one can predict the space-time segments of the video that are best-served by each available algorithm
  - (Can always come back to choose a per-frame or per-video algorithm)

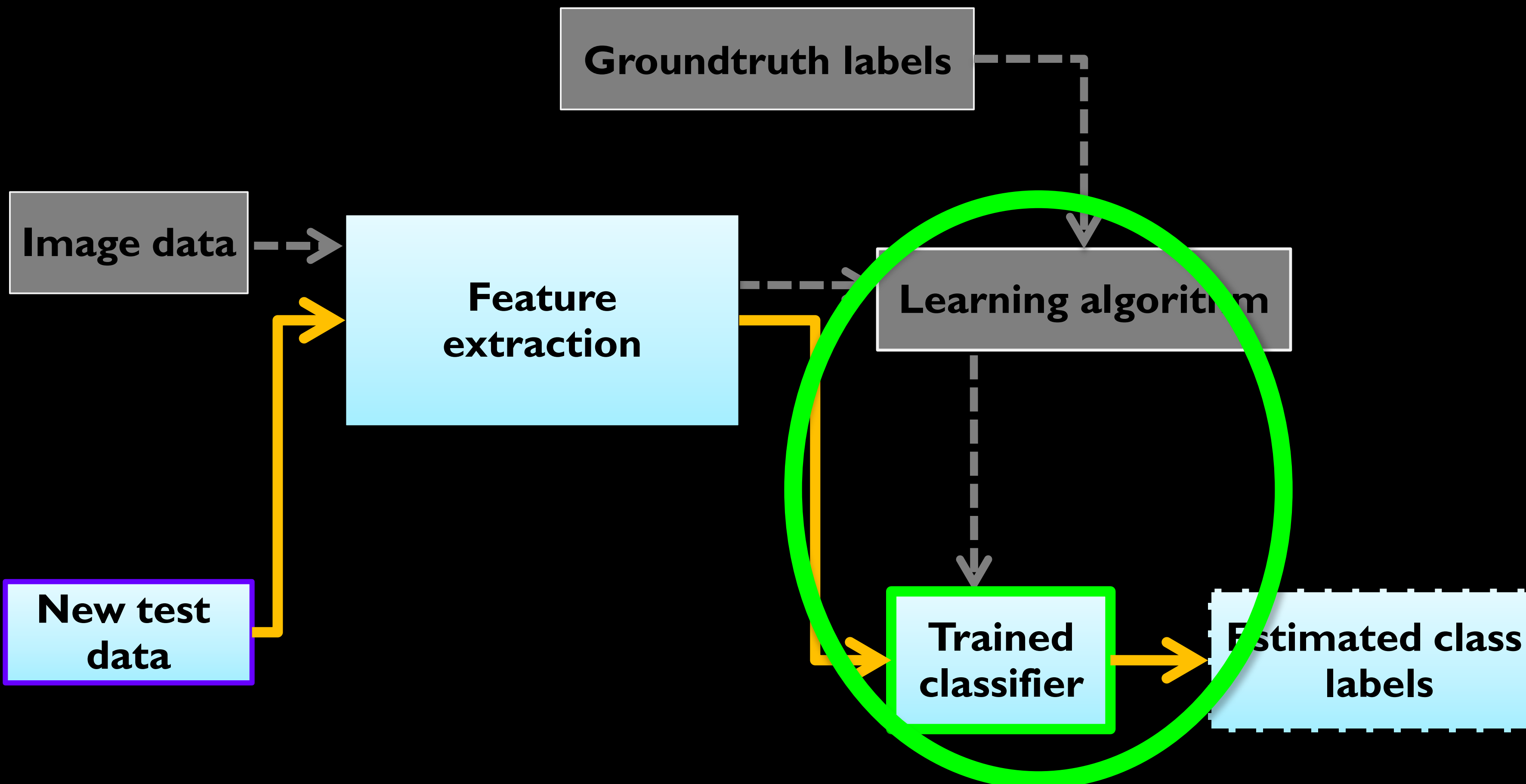
Experimental Framework



Experimental Framework

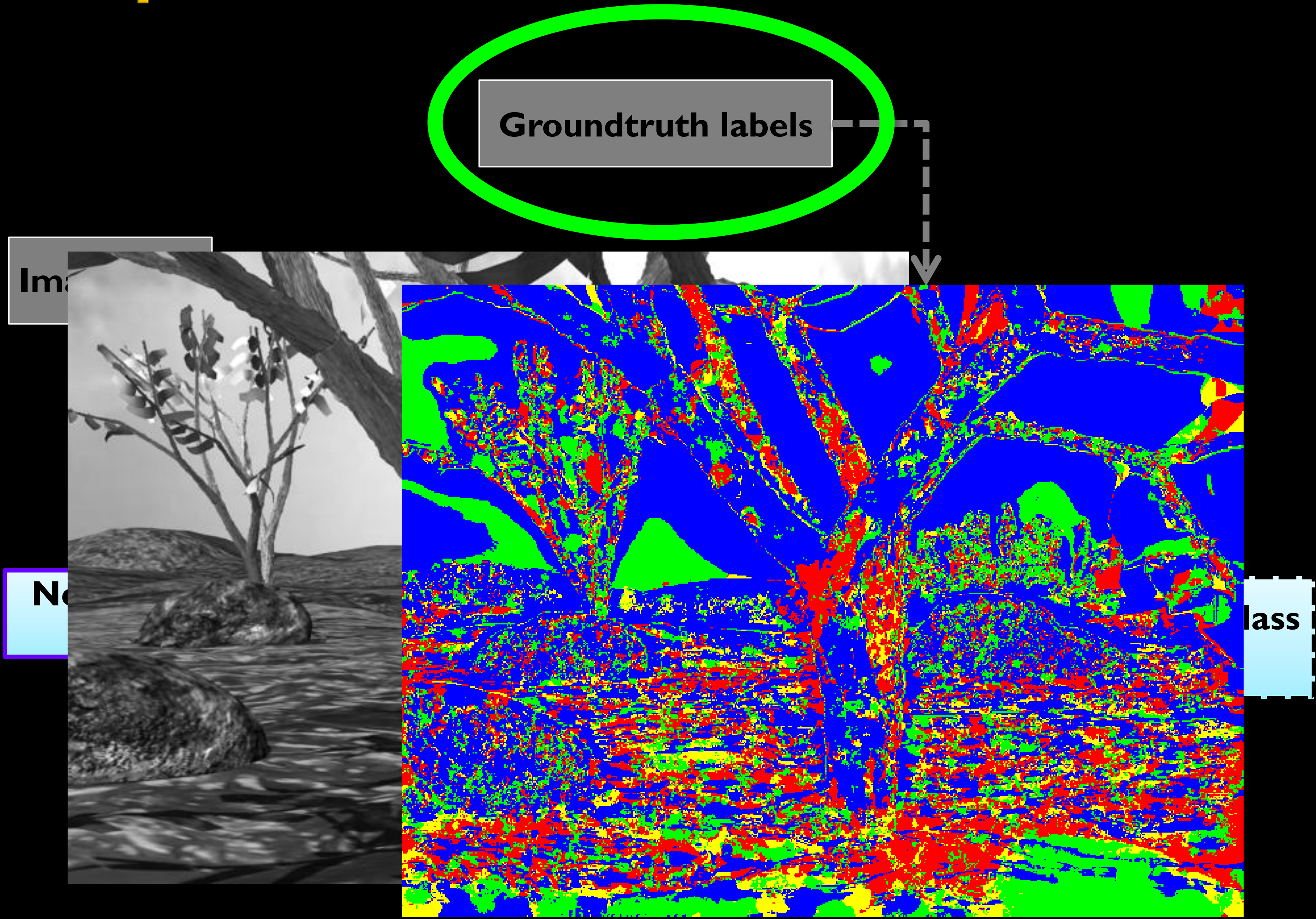


Experimental Framework



Random Forests
Breiman, 2001

Experimental Framework



“Making” more data

Formulation

$$\mathscr{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^d, c_i \in \mathbb{Z}^k\}_{i=1}^n$$

$$\mathbf{x}_i = \{g(x,y,[1,z]), d(x,y,[1,z]), t_x(x,y,[1,z]), t_y(x,y,[1,z]), r(x,y,[1,k])\}$$

- Training data $\mathscr{D}$ consists of feature vectors ***x*** and class labels *c* (i.e. best-algorithm per pixel)

- Feature vector x is multi-scale, and includes:
  - Spatial Gradient
  - Distance Transform
  - Temporal Gradient
  - Residual Error (after bicubic reconstruction)

Formulation

$$\mathscr{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^d, c_i \in \mathbb{Z}^k\}_{i=1}^{n}$$

$$\mathbf{x}_i = \{g(x,y,[1,z]), d(x,y,[1,z]), t_x(x,y,[1,z]), t_y(x,y,[1,z]), r(x,y,[1,k])\}$$

- Training data $\mathscr{D}$ c
  and class labels c

$$\boxed{g(x,y,z) = ||\nabla I_1||}$$

- Feature vector x is multi-scale, and includes:
  - Spatial Gradient
  - Distance Transform
  - Temporal Gradient
  - Residual Error (after bicubic reconstruction)

Formulation

$$\mathscr{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^d, c_i \in \mathbb{Z}^k\}_{i=1}^n$$

$$\mathbf{x}_i = \{g(x,y,[1,z]), d(x,y,[1,z]), t_x(x,y,[1,z]), t_y(x,y,[1,z]), r(x,y,[1,k])\}$$

$$\boxed{d(x,y,z) = disTrans(||\nabla I_1|| > \tau)}$$

- Feature vector x is multi-scale, and includes:
  - Spatial Gradient
  - Distance Transform
  - Temporal Gradient
  - Residual Error (after bicubic reconstruction)

Formulation Details

- Temporal Gradient

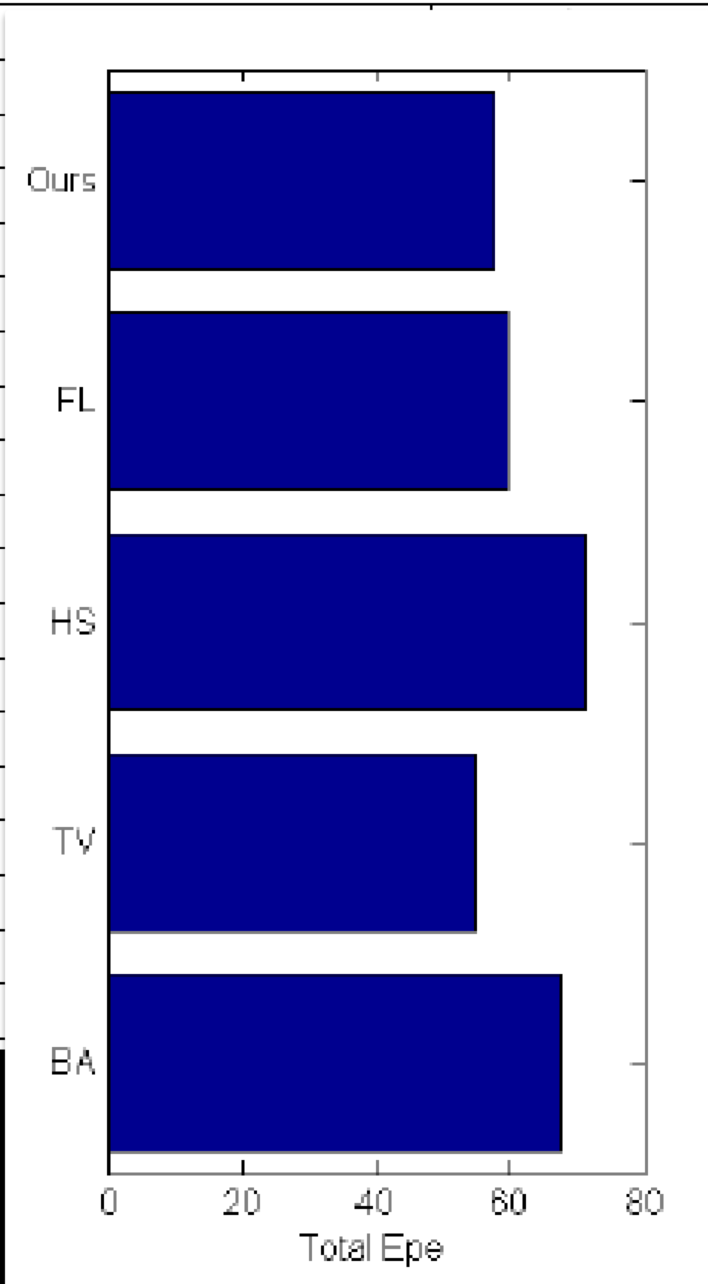
$$t_x = ||\nabla(x+\bar{u})||$$
$$t_y = ||\nabla(y+\bar{v})||$$

- Residual Error

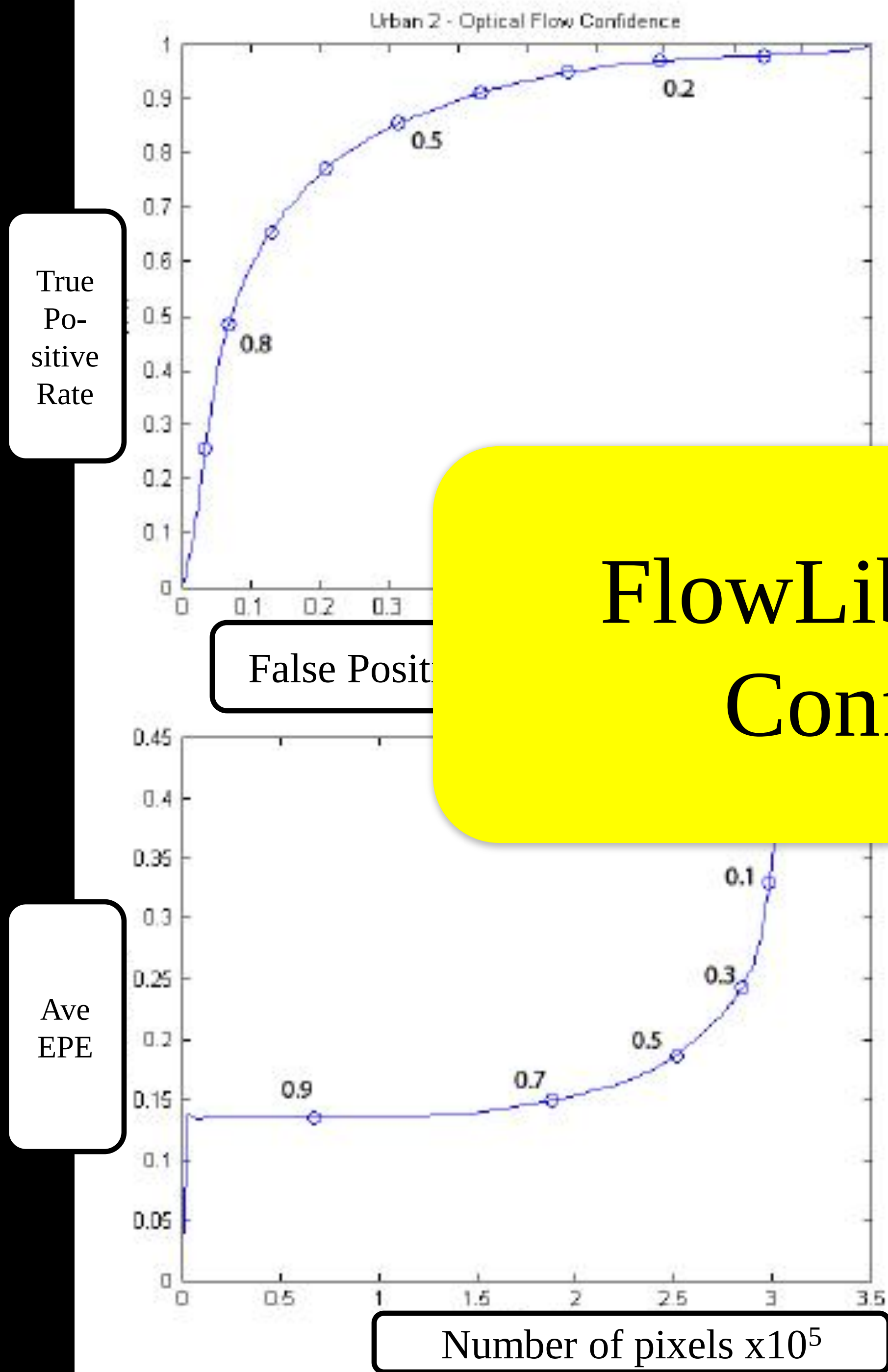
$$r_i(x,y,k) = I_1(x,y) - bicubic(I_2(x+u_i(k), y+v_i(k)))$$

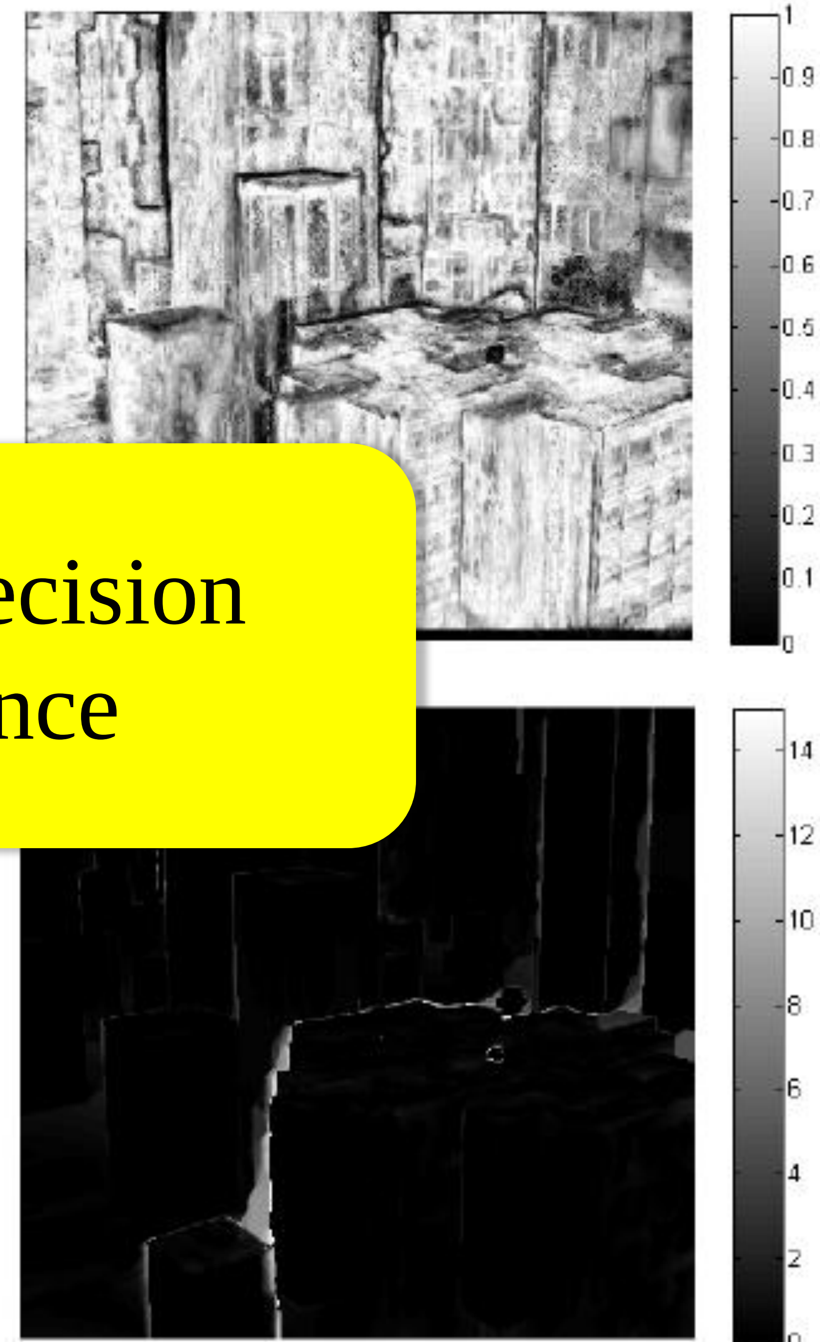
Application I: Optical Flow

Image Sequence	BA	TV	HS	FL	TrivComb	Ours	OptCombo
		08	0.549	0.350	0.387	**0.344**	0.271
		32	1.329	**0.527**	0.589	0.603	0.398
		06	0.772	**0.435**	0.428	0.436	0.256
		35	0.182	**0.096**	0.133	0.097	0.074
		96	0.276	**0.164**	0.191	0.165	0.123
		45	0.873	**0.622**	0.670	0.628	0.466
		20	0.285	**0.170**	0.189	0.171	0.111
		11	0.171	**0.144**	0.147	0.147	0.115
		64	5.195	3.724	4.582	3.748	2.448
		15	17.891	12.634	17.634	9.607	3.373
		25	7.129	**6.748**	7.045	6.791	6.606
		868	**21.643**	22.333	21.784	21.9	20.925
		31	2.702	**0.709**	2.177	1.833	0.443
		42	0.876	0.344	0.389	**0.342**	0.212
		10	0.267	0.250	0.221	0.252	0.186
		76	0.683	**0.403**	0.474	0.437	0.368
		59	**8.443**	8.607	8.52	8.612	8.11
		06	1.281	1.021	1.057	**0.988**	0.762
		96	0.464	0.473	0.456	0.467	0.296



FlowLib Decision Confidence





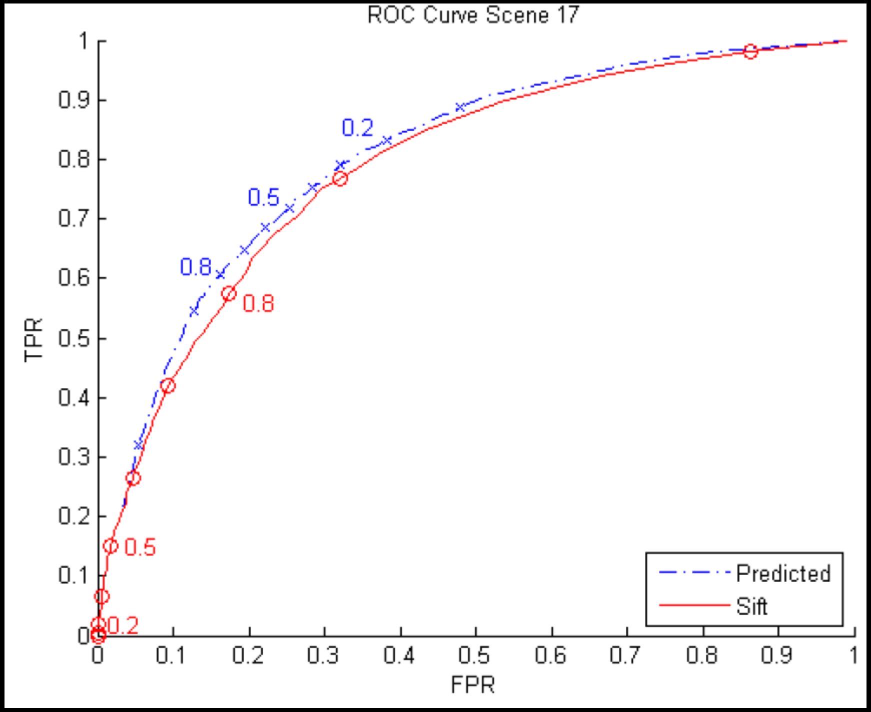
Application II: Feature Matching

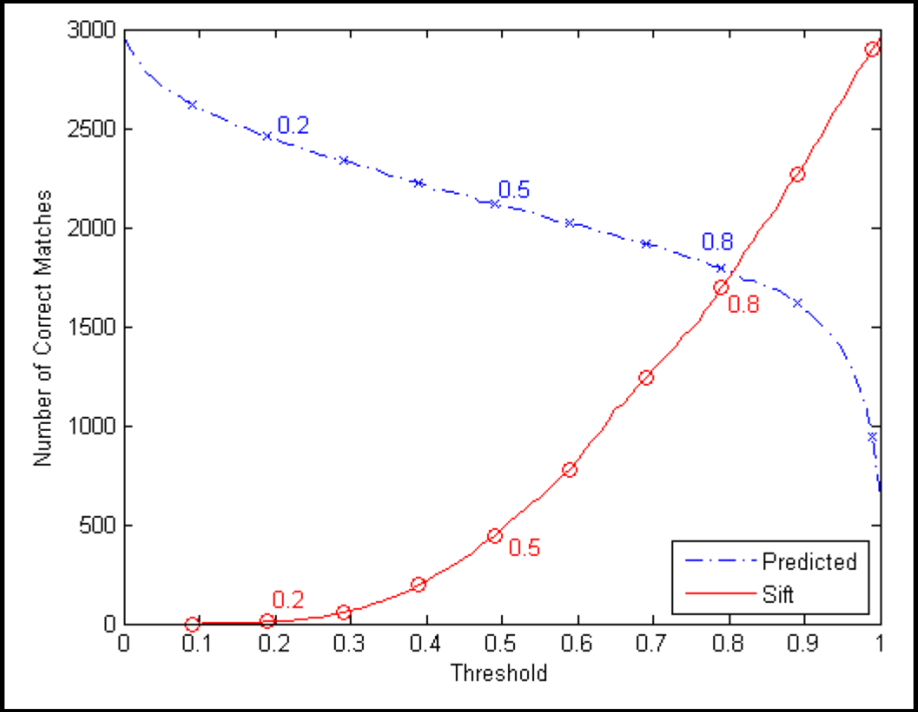
Comparing 2 Descriptions

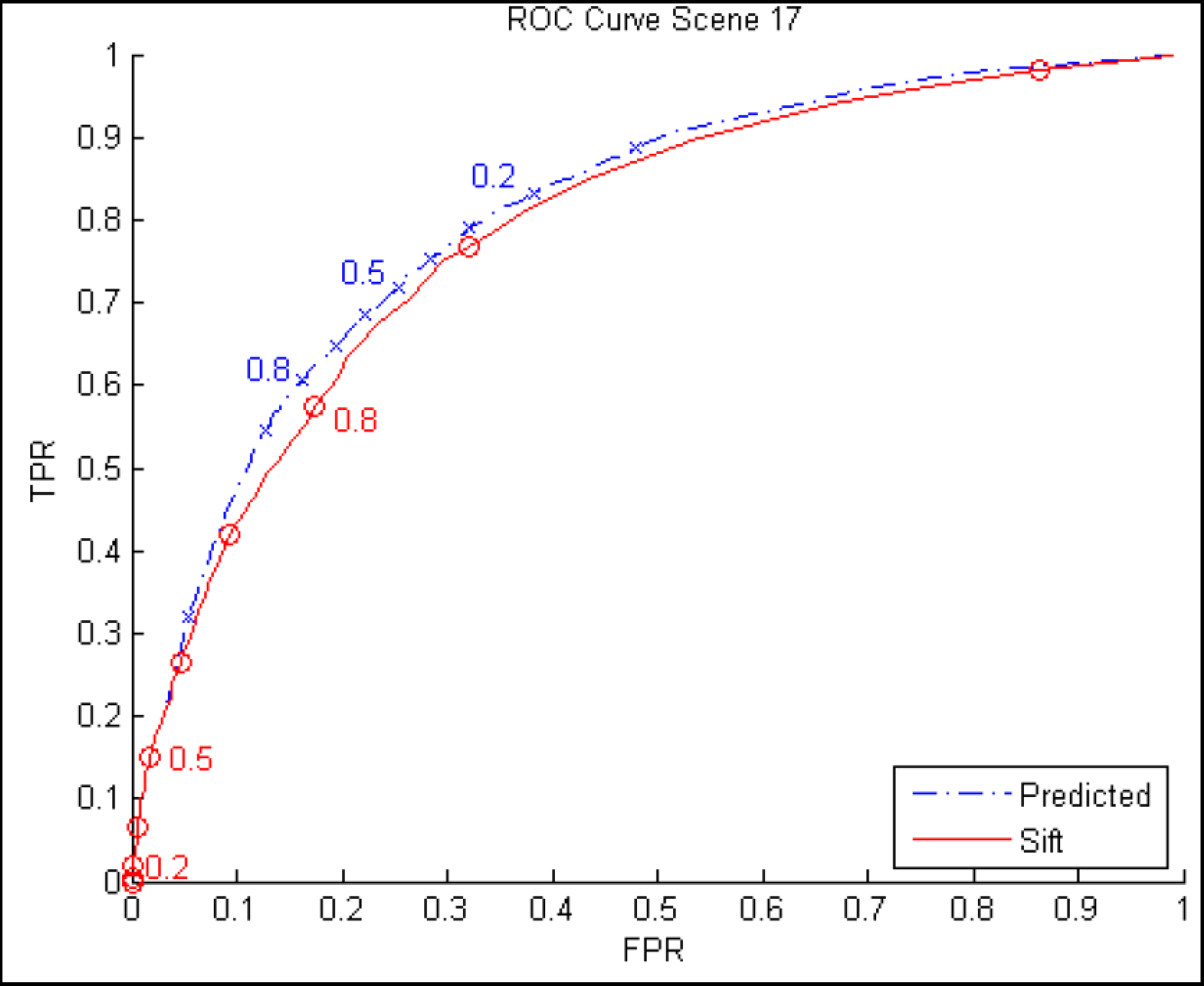
- What is a match? Details are important…
  - Nearest neighbor (see also [FLANN]())
  - Distance Ratio
  - PCA
- Evaluation: density, # correct matches, tolerance

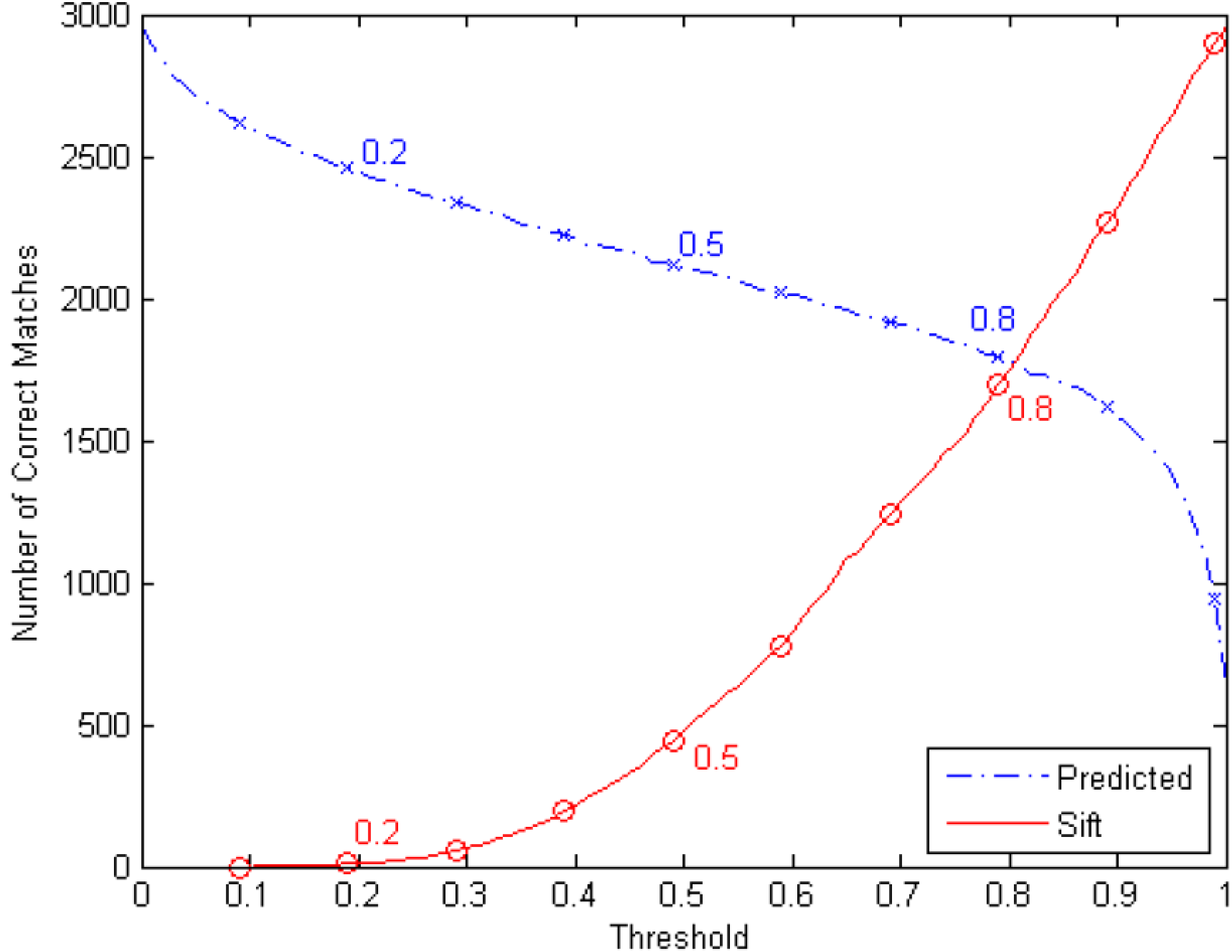
"192 correct matches (yellow) and 208 false matches (blue)"

SIFT Decision Confidence






ROC Curve Scene 17
TPR
FPR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.5
0.8
0.8
0.5
0.2
Predicted
Sift



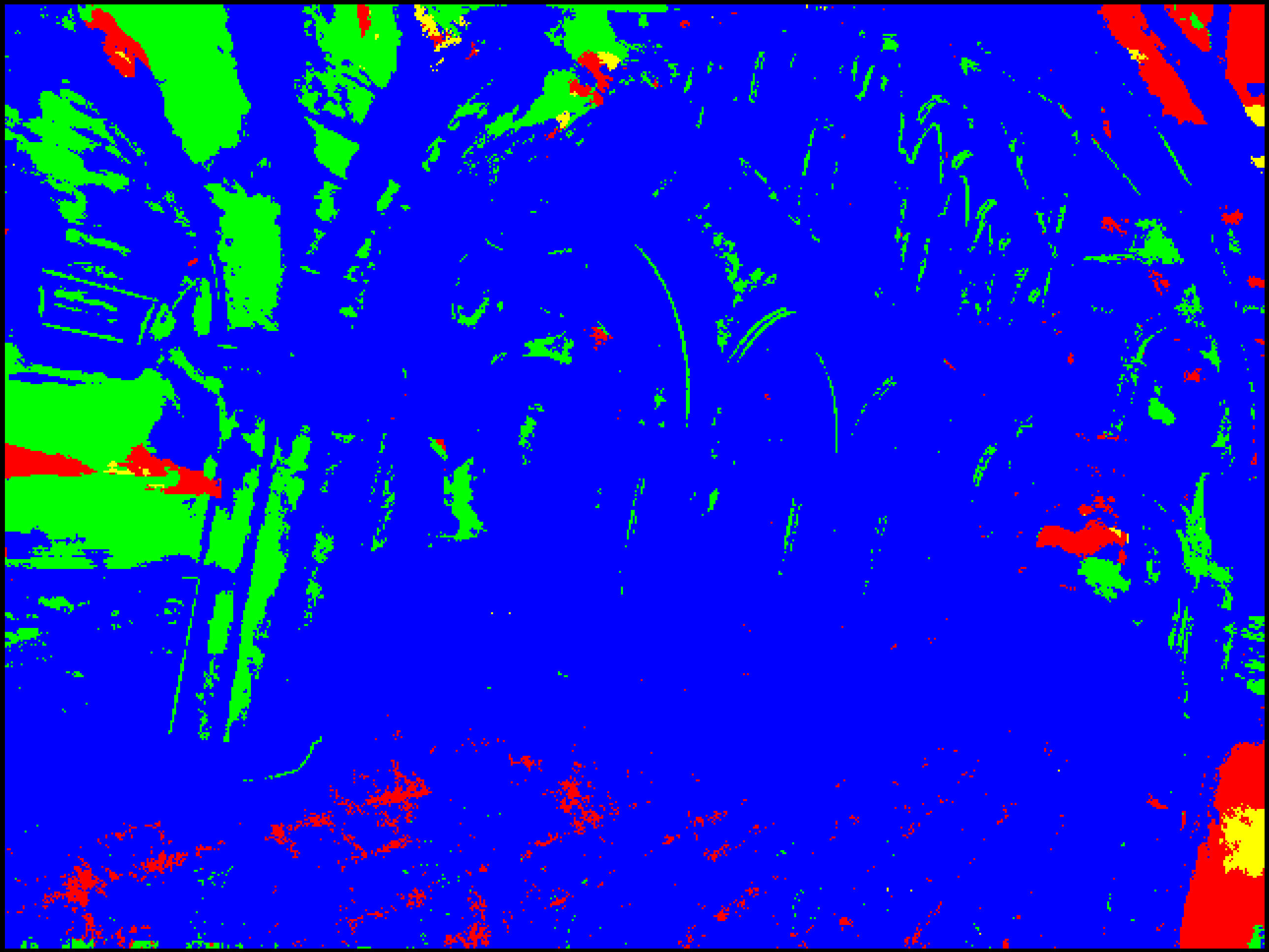
Number of Correct Matches
Threshold
3000
2500
2000
1500
1000
500
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.5
0.8
0.8
0.5
0.2
Predicted
Sift

Hindsight / Future Work

- Current results don’t quite live up to the theory:
  - Flaws of best-algorithm are the upper bound (ok)
  - Training data does not fit in memory (fixable)
  - “Winning” the race is more than rank (problem!)

Summary

- Overall, predictions are correlated with the best algorithm for each segment (expressed as Pr!)
- Training data where one class dominates is dangerous – needs improvement
- Other features could help make better predictions
  - Results don't yet do the idea justice
- One size does NOT fit all
  - At least in terms of algorithm suitability
  - Could use "bad" algorithms!

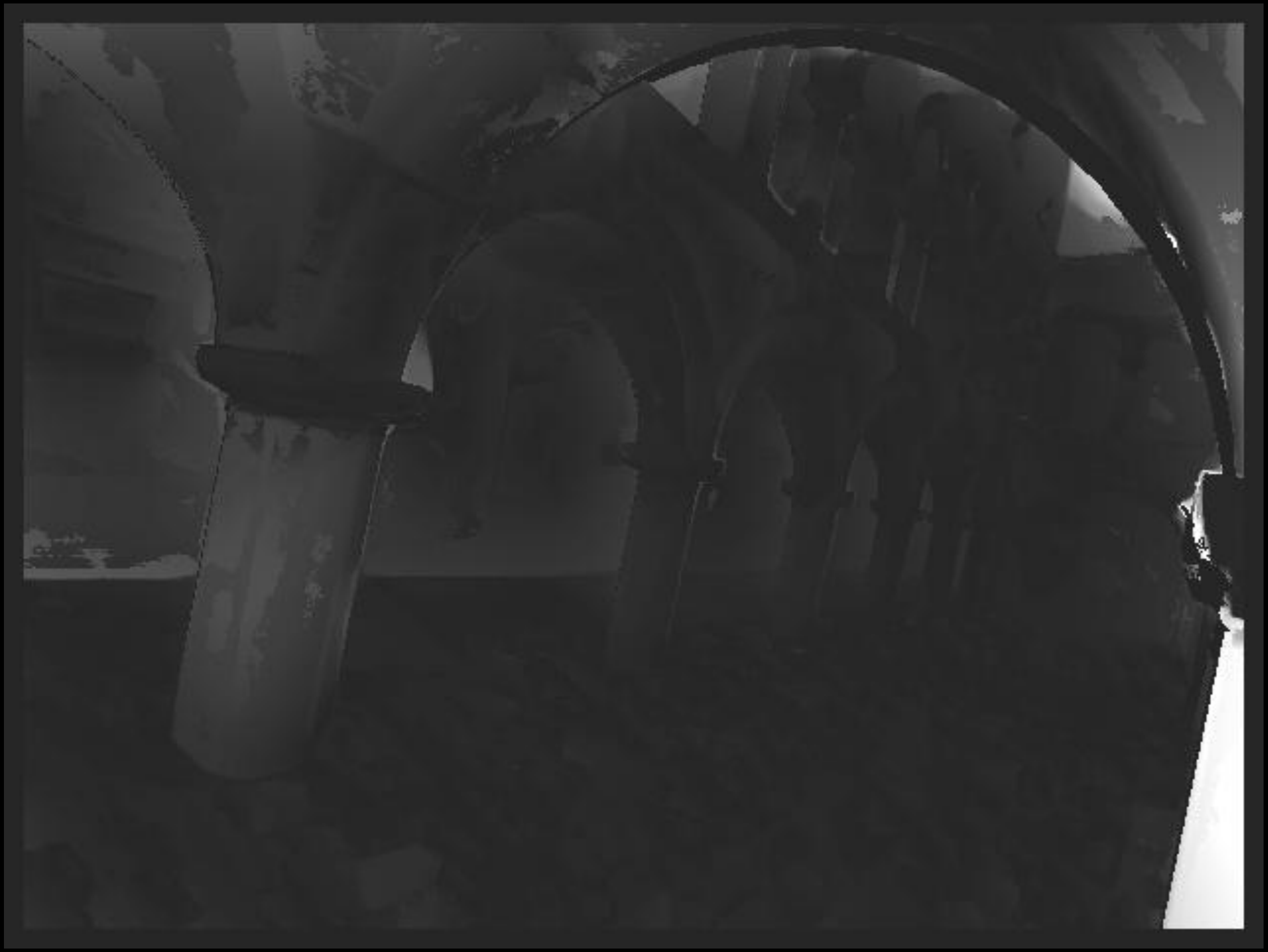


Ground Truth Best Based on Prediction

FlowLib

Based on Prediction

White = 30 pixel end point error



FlowLib Based on Prediction

(Contrast enhanced)