Learning Efficient Linear Predictors for Motion Estimation

Jiří Matas[1,2], Karel Zimmermann[1],
Tomáš Svoboda[3], Adrian Hilton[2]

[1]: Center for Machine Perception
Czech Technical University
Prague, Czech Republic

[2] : CVSSP
University of Surrey
Guildford, England

[3]: Center for Applied Cybernetics
Czech Technical University
Prague, Czech Republic

State-of-the-art: Tracking by detection

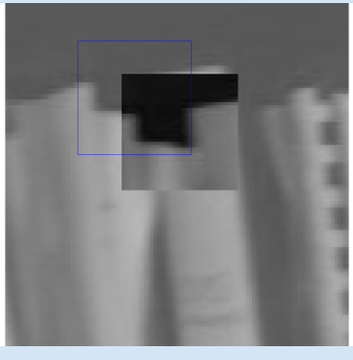
[1] I.Gordon and D.G.Lowe, Scene modelling, recognition and tracking with invariant image features, International Symposium on Mixed and Augmented Reality (ISMAR), pp.110-119, 2004

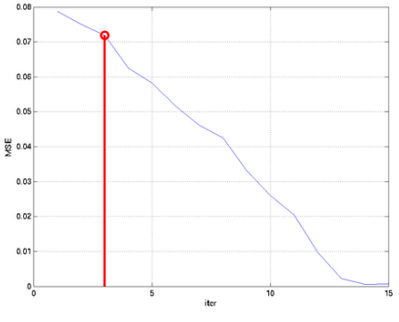
[2] V.Lepetit, P.Lagger and P.Fua, Randomized Trees for Real-Time Keypoint Recognition, Computer Vision and Pattern Recognition

- View point invariance but high computational complexity causing low frame-rate and inprecise position estimation.

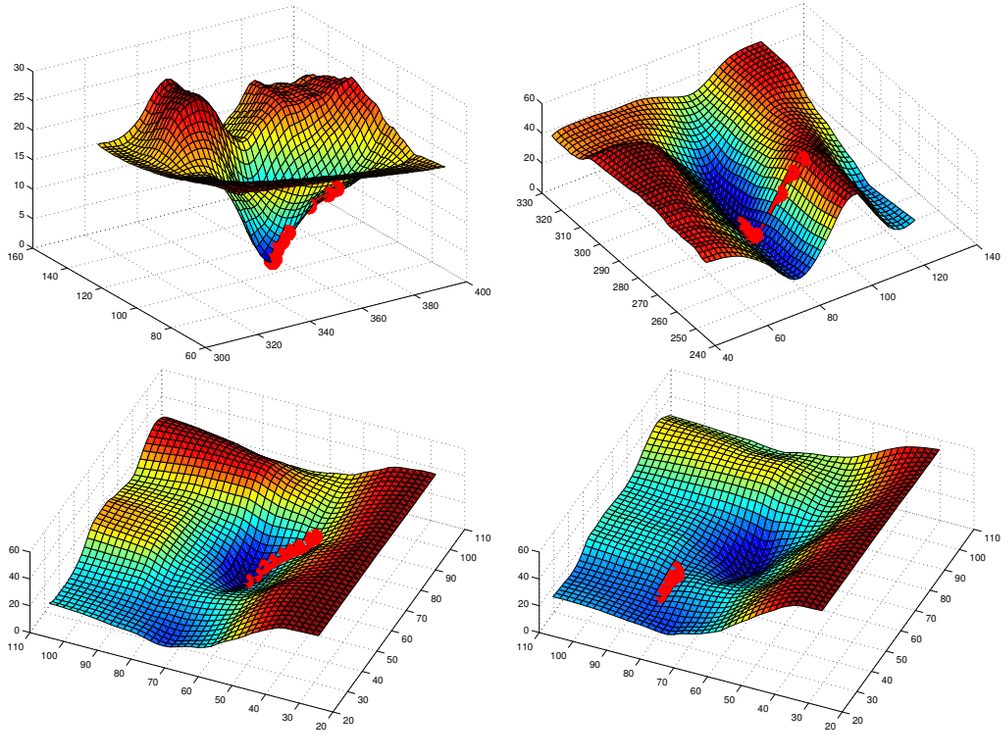
State-of-the-art: Tracking by displacement estimation

[3] S.Baker and I.Matthews, Lucas-Kanade 20 Years On: A Unifying Framework, International Journal of Computer Vision, pp.221-255, 2004





Video: example of KLT convergergence



$$\sum \Big(I(\mathbf{x}+\mathbf{t}) - J(\mathbf{x})\Big)^2 \quad \Rightarrow \quad \mathbf{t} = \mathtt{A}\Big(I(\mathbf{x}) - J(\mathbf{x})\Big)$$

State-of-the-art: Tracking by displacement estimation

- There is a hidden mapping:

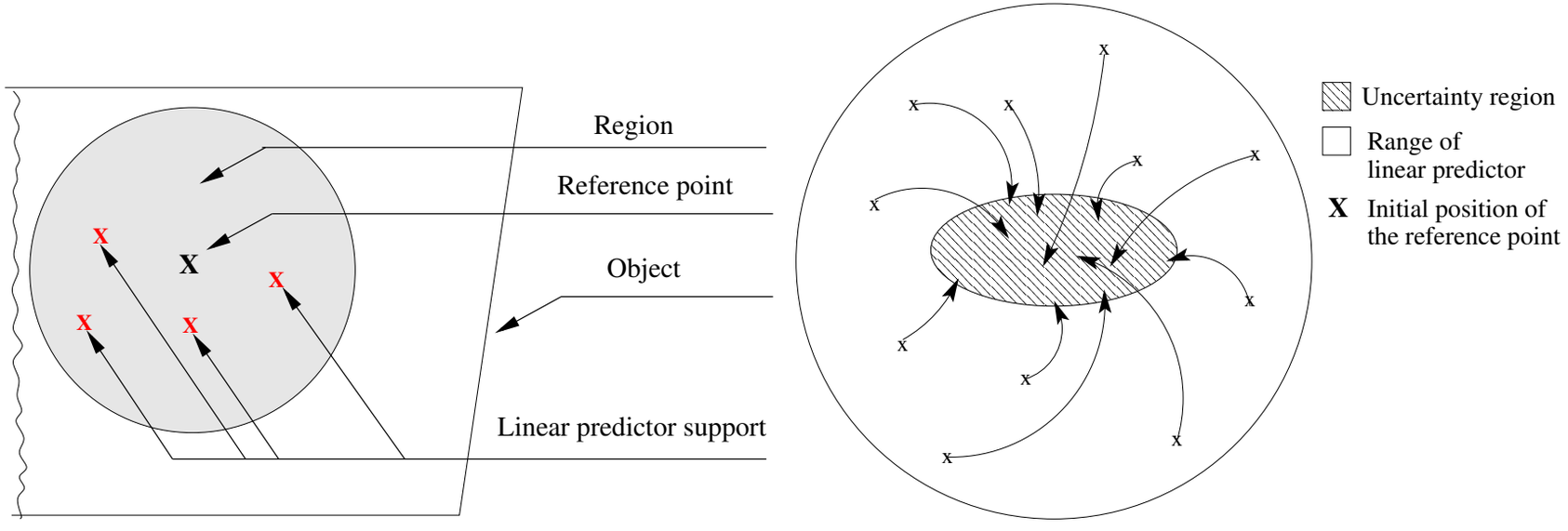
$$\Theta : intensities\ around\ the\ point \rightarrow motion$$

- [4] F.Jurie and M.Dhome, Real time robust template matching, British Machine Vision Conference, pp.123-131, 2002
  - Linear approximation: $\mathbf{t} = \mathtt{A}\big(I(\mathbf{x}) - J(\mathbf{x})\big) = \mathtt{Ad}$;
- [5] O.Williams, A.Blake and R.Cipolla, Sparse Bayesian Learning for Efficient Visual Tracking, Pattern Analysis and Machine Intelligence, pp.1292-1304, 2005
  - Non-linear approximation: RVM

Our Contributions:

- Relax the linearity in [4] without significant increase of computational complexity in [5].
- Maximizing the accuracy of tracking by explicitly considering the time-to-predict $\times$ prediction accuracy trade-off
- Selecting the points really suitable for tracking (unlike to [3]).

Complexity and uncertainty region

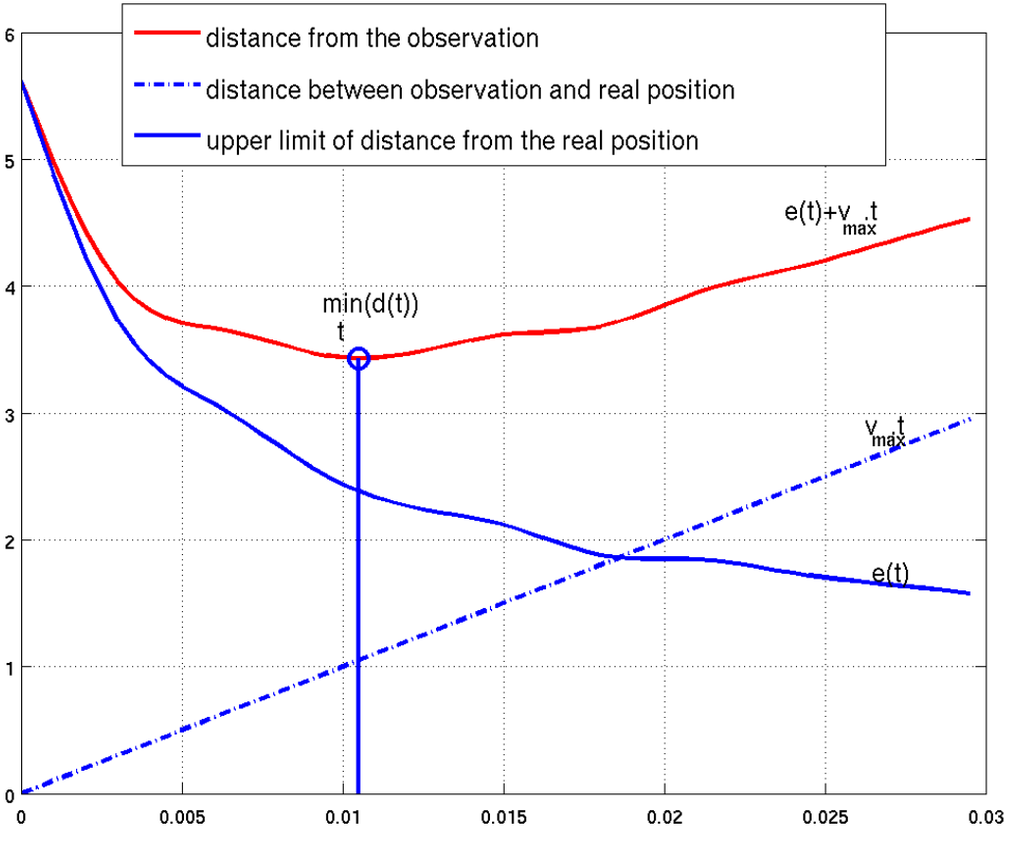


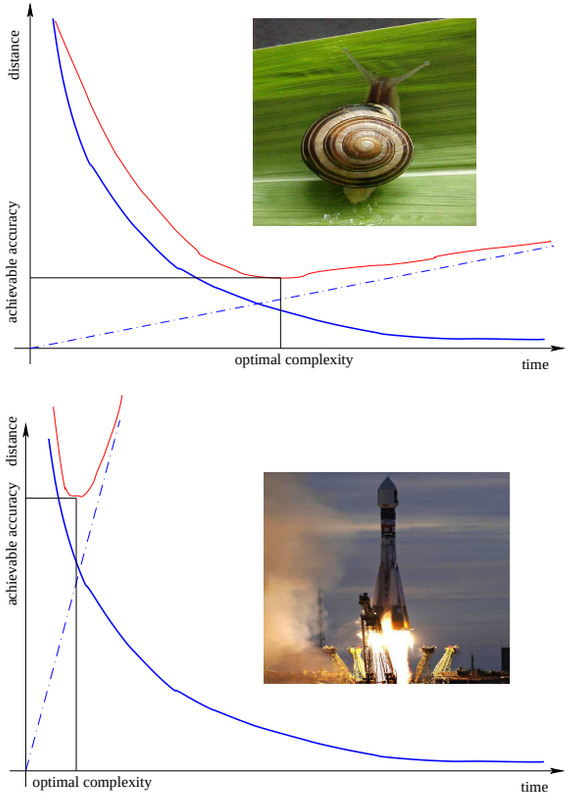
- $\mathtt{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_k], \mathtt{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \quad \Rightarrow$
- $\mathtt{A}^* = \arg\min_{\mathtt{A}} \sum_i \left(\mathbf{t}_i - \mathtt{A}\mathbf{d}_i\right)^2 \quad \Rightarrow \quad \mathtt{A}^* = \mathtt{T}\mathtt{D}^+$
- Complexity corresponds to the size of the support set.
- Randomized sampling selects a suitable support set for the tracking of the reference point.
- Range of the predictor is set of the admissible motions $\mathtt{T}$.
- Uncertainty region is the region within which all predictions lie.

Piece-wise linear approximation

- Linear predictor is replaced by a sequence of linear predictors, where uncertainty region of the preceding predictor includes range of the following one.
- Each iteration is performed by a different predictor specialized on the expected range.

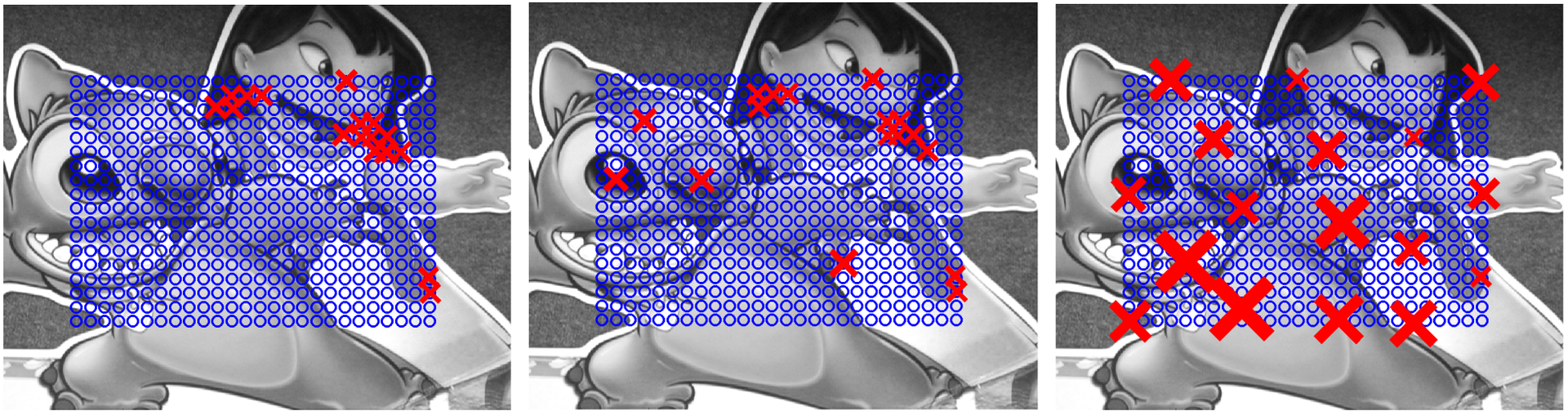
The maximal accuracy





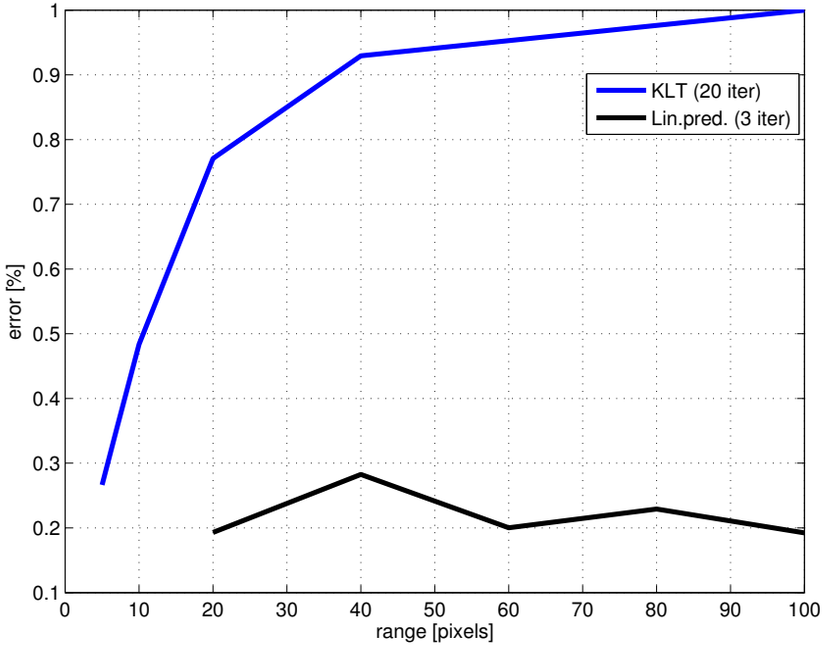
- Object is moving by the maximal velocity v_{max}.
- Motion estimation error in the observation $e(t)$ in time t.
- Distance between the prediction and true object position $d(t) = e(t) + v_{max}.t$

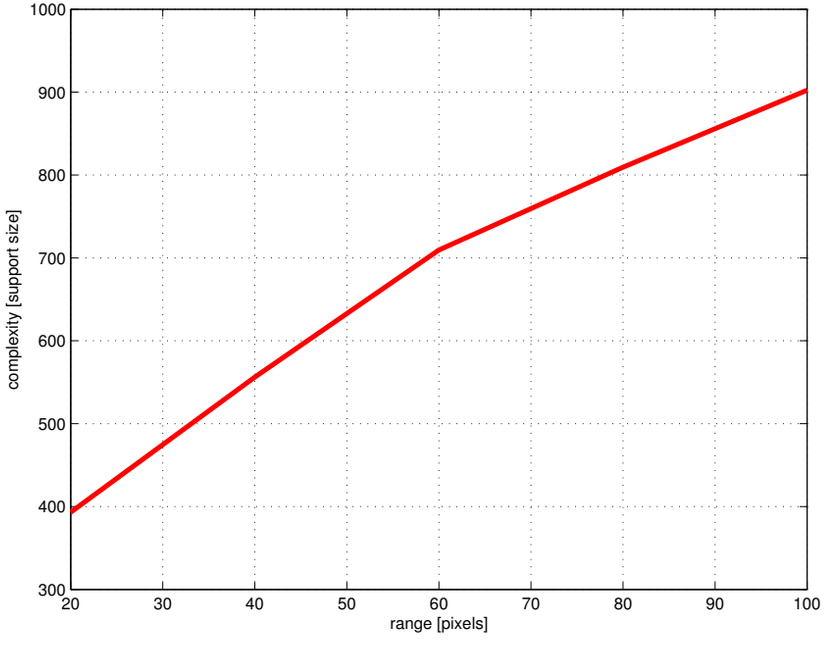
Selection of the points suitable for tracking



- The object is represented as a constellation of spatially localised linear predictors of the optimal complexity.
- The algorithm selects the subset of the points with low training error and a good coverage of the object.
- Even strong features (according to KLT definition) may not be suitable for tracking (see repetitive structure on teeth).

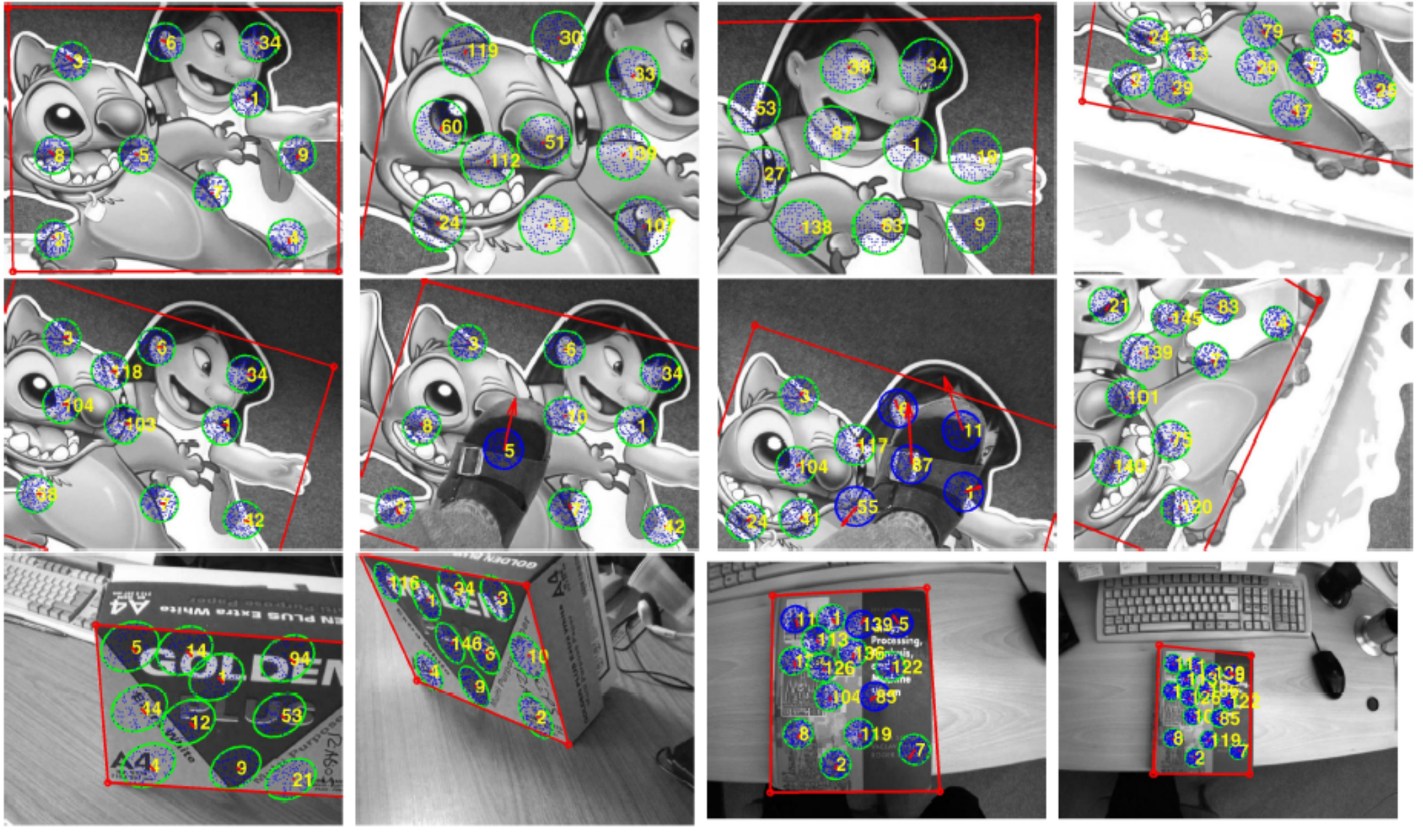
Experiment: Comparison with KLT tracker



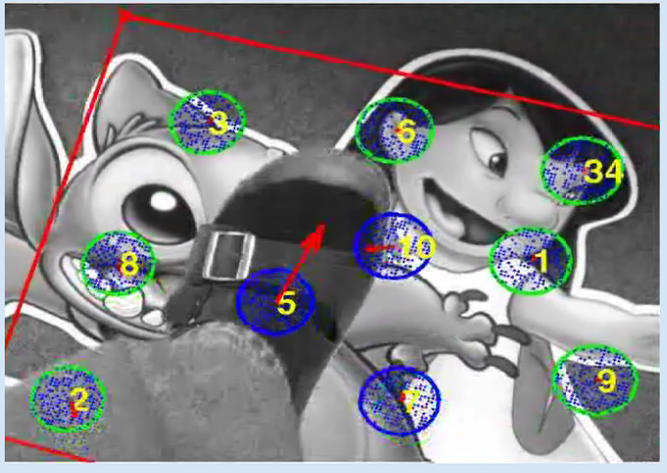


- Challenging testing sequence involving intrackable points like (blue sky or homogeneous walls, e.g.).
- 684 random different image points, in each 500 translations from each range were evaluated.
- Complexity of the linear tracker is essentially smaller than in KLT.
- Only the complexity grows with the range, error remains the same.

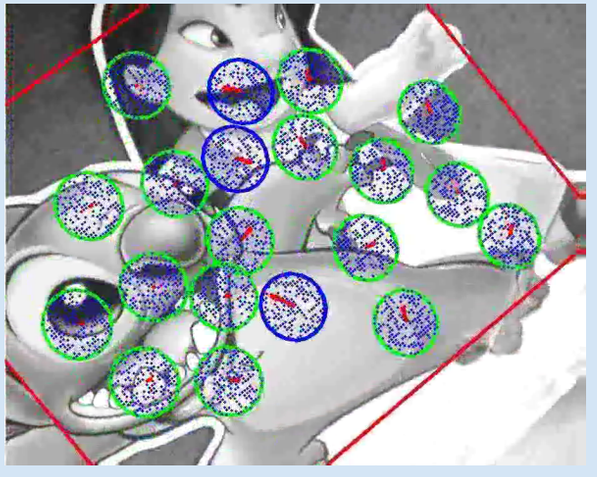
Reference points and their Supports



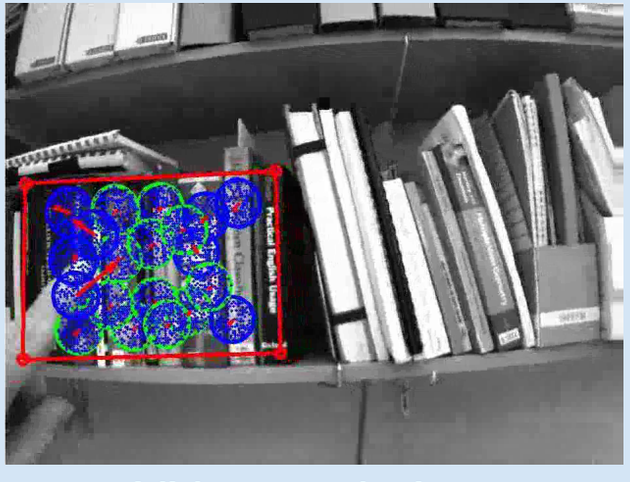
Reference points and their Supports — Videos



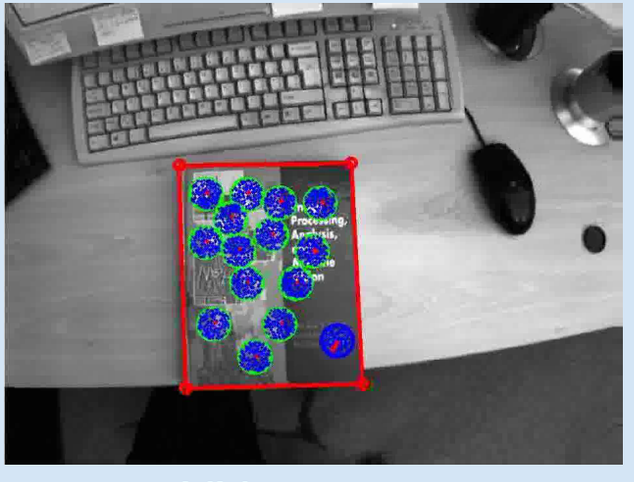
Video: occlusions



Video: shaking

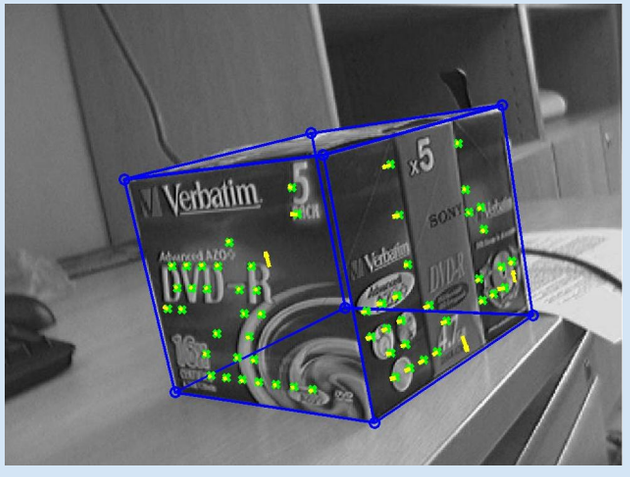


Video: occlusions



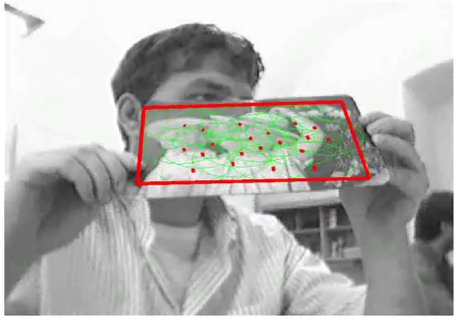
Video: zoom

Experiment: Application of the proposed method to object tracking



Video: 3D tracking

Video: fast motion



Video: out of plane rotations

Video: stability of the tracking

- Motion blur, fast motion, views from accute angles, discretization caused by scaling.

Experiment: Performance evaluation on real sequences with plane tracking

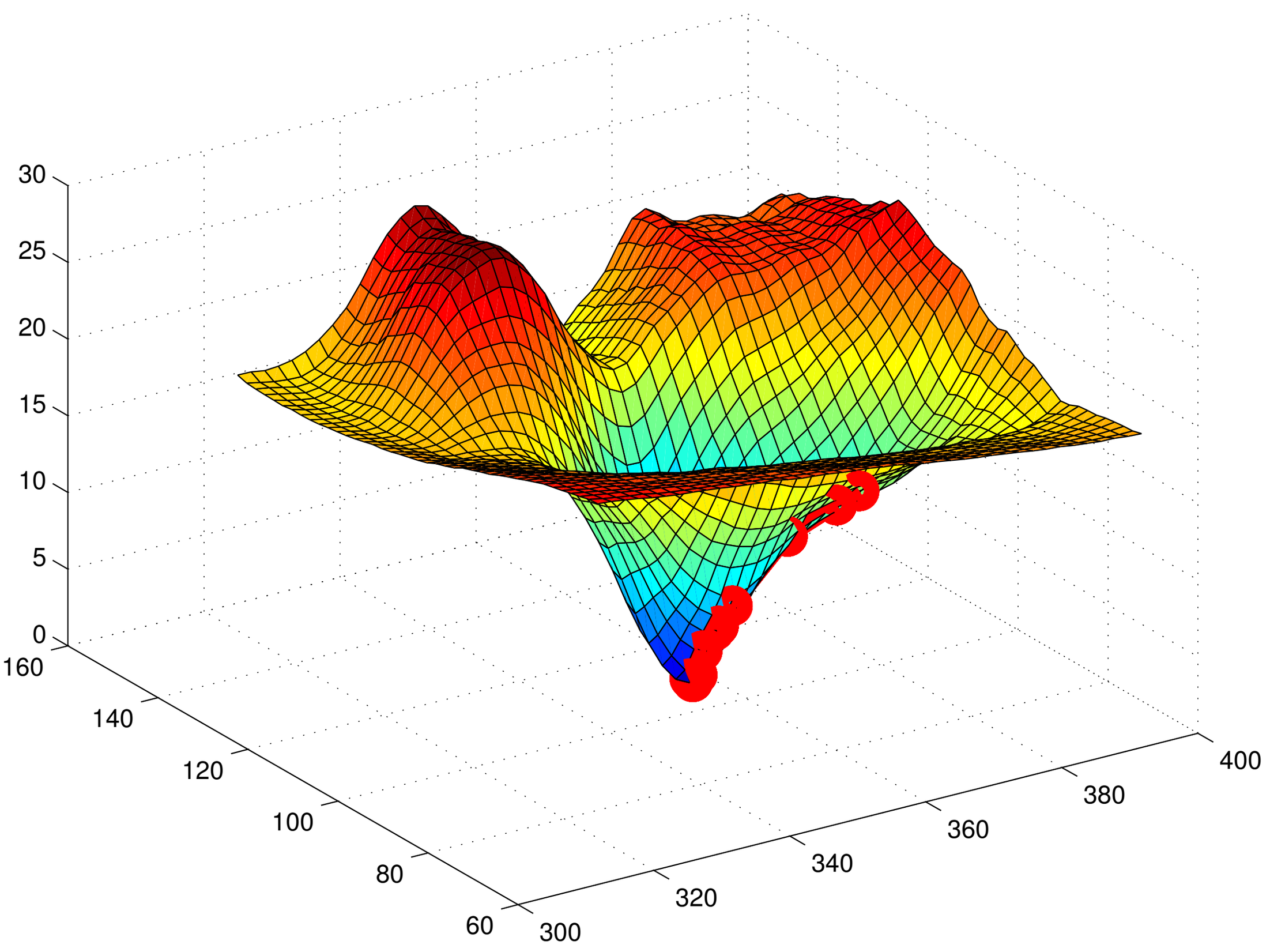
iter	examples	M=30	M=45
3	affine	6	7
3	affine+blur+discr	5	9
5	affine	4	3
5	affine+blur+discr	3	2

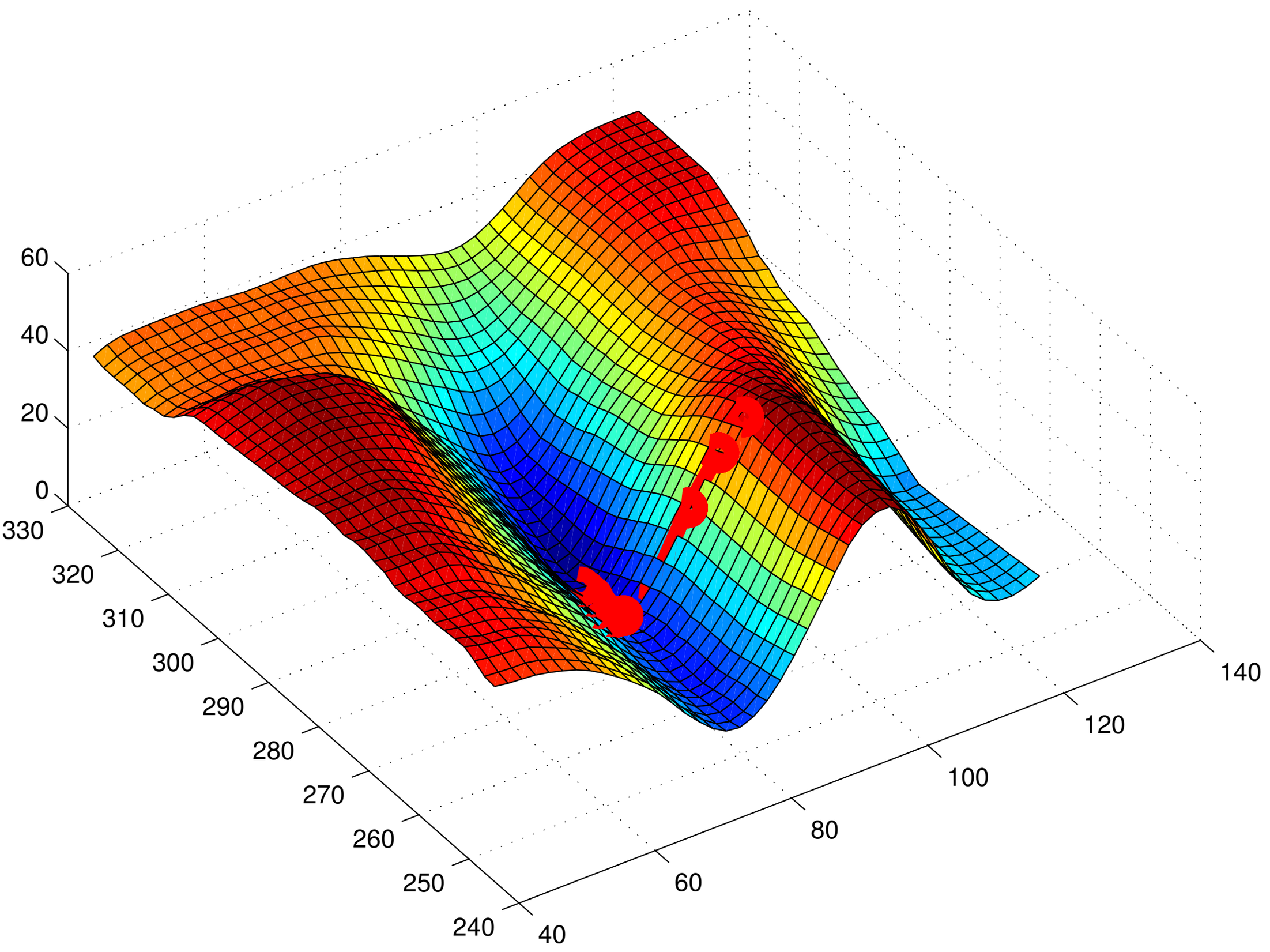
- Number of necessary re-detections $[-]$.
- Complexity might not be sufficient to approximate a complicated mapping (e.g. with high range or including blur, discretization etc.).

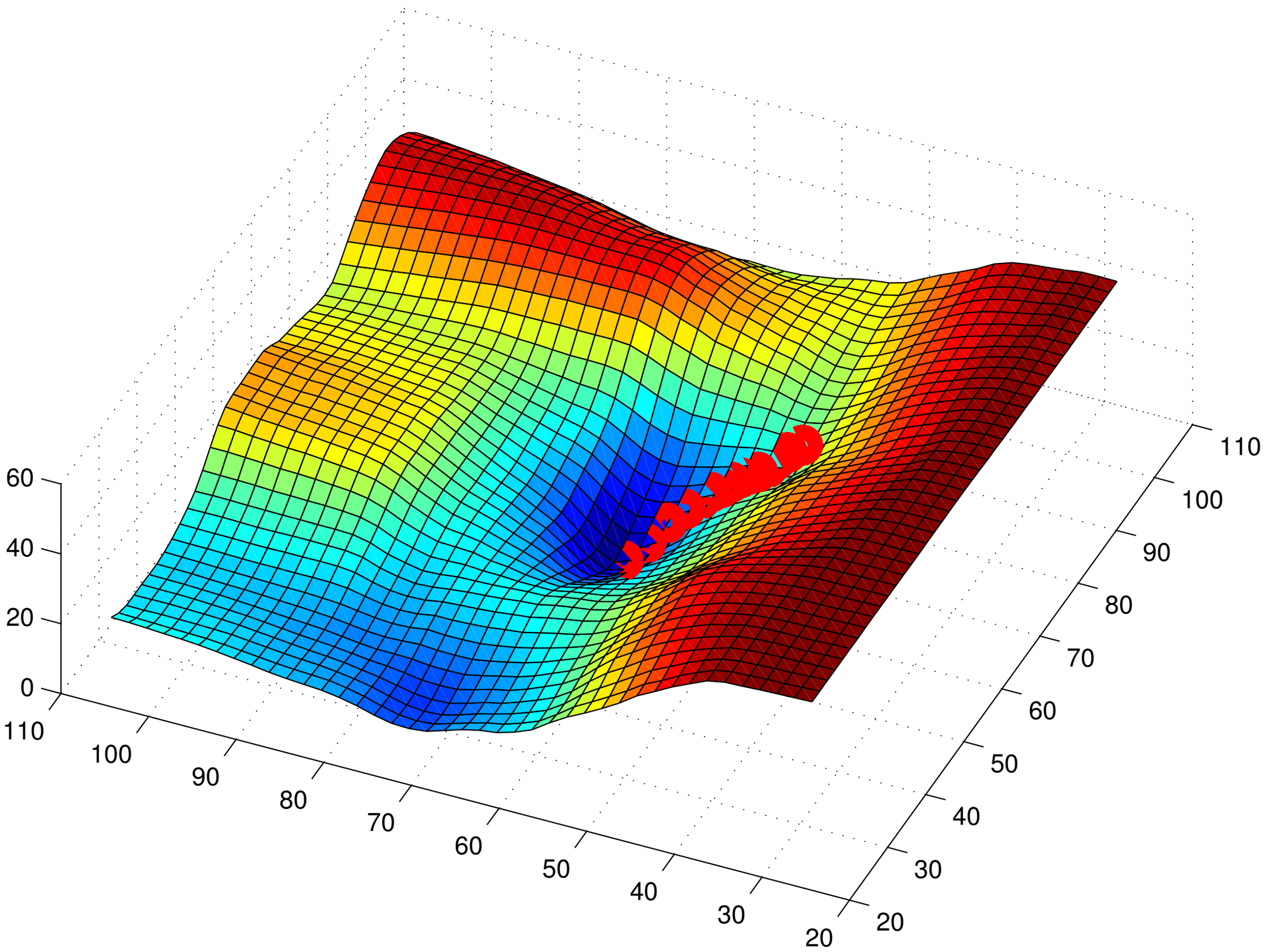
Conclusion

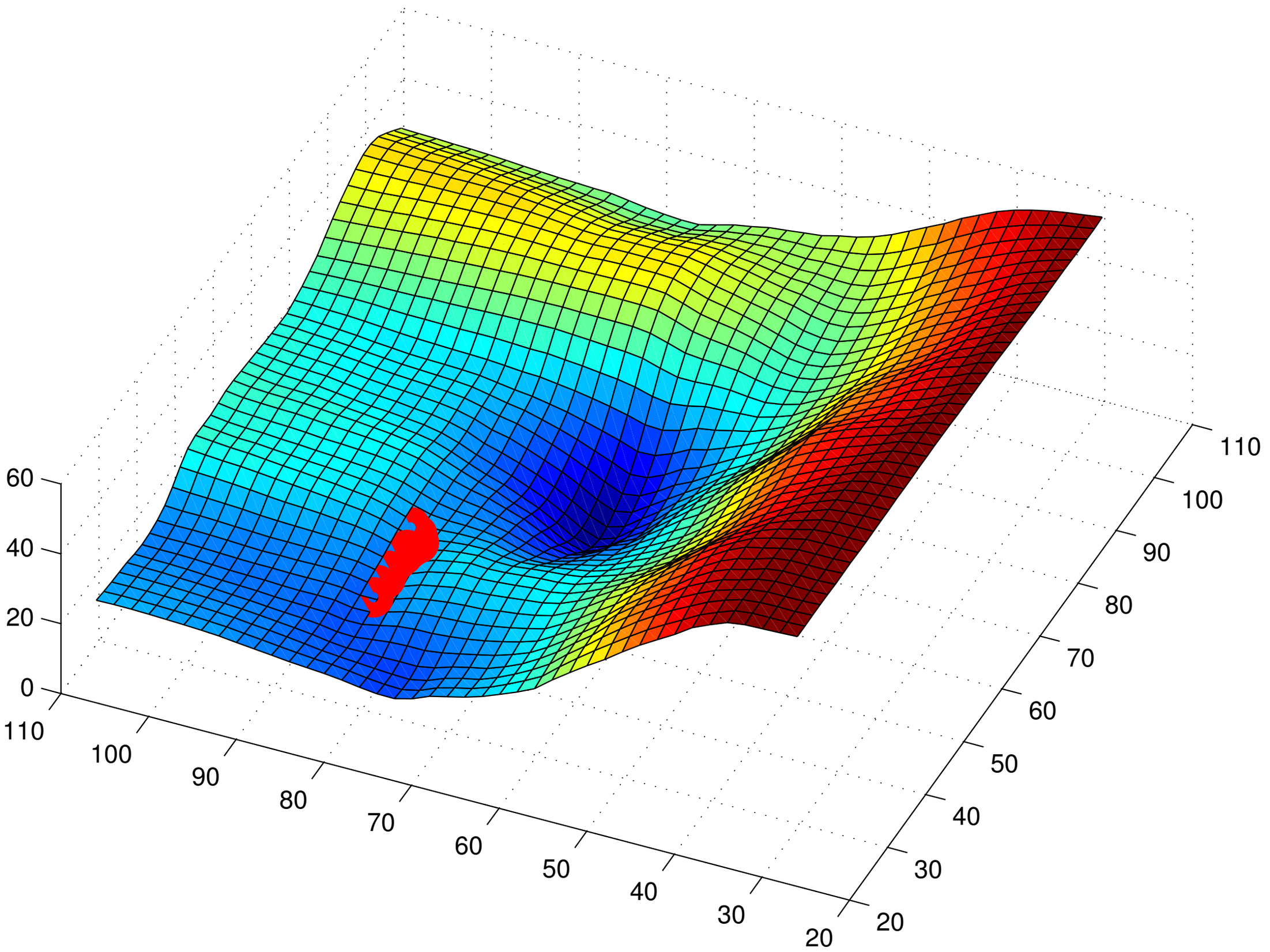
- We explicitly consider the time-to-predict $\times$ prediction accuracy trade-off, estimating the optimal complexity, given the maximal object velocity.
- Does not assume that neighbourhood of high gradient points are suitable for tracking (unlike [3],[4]).
- Simple implementation of the tracker: just dot-products + RANSAC.
- Non-optimized C++ implementation of the tracker takes about $\mathbf{30}\mu\mathbf{s}$ per point.
- Fast tracking allows robust motion estimation.
- One can smoothly change the tracker to the detector, by randomized sampling of the tracking space according to a motion prior.

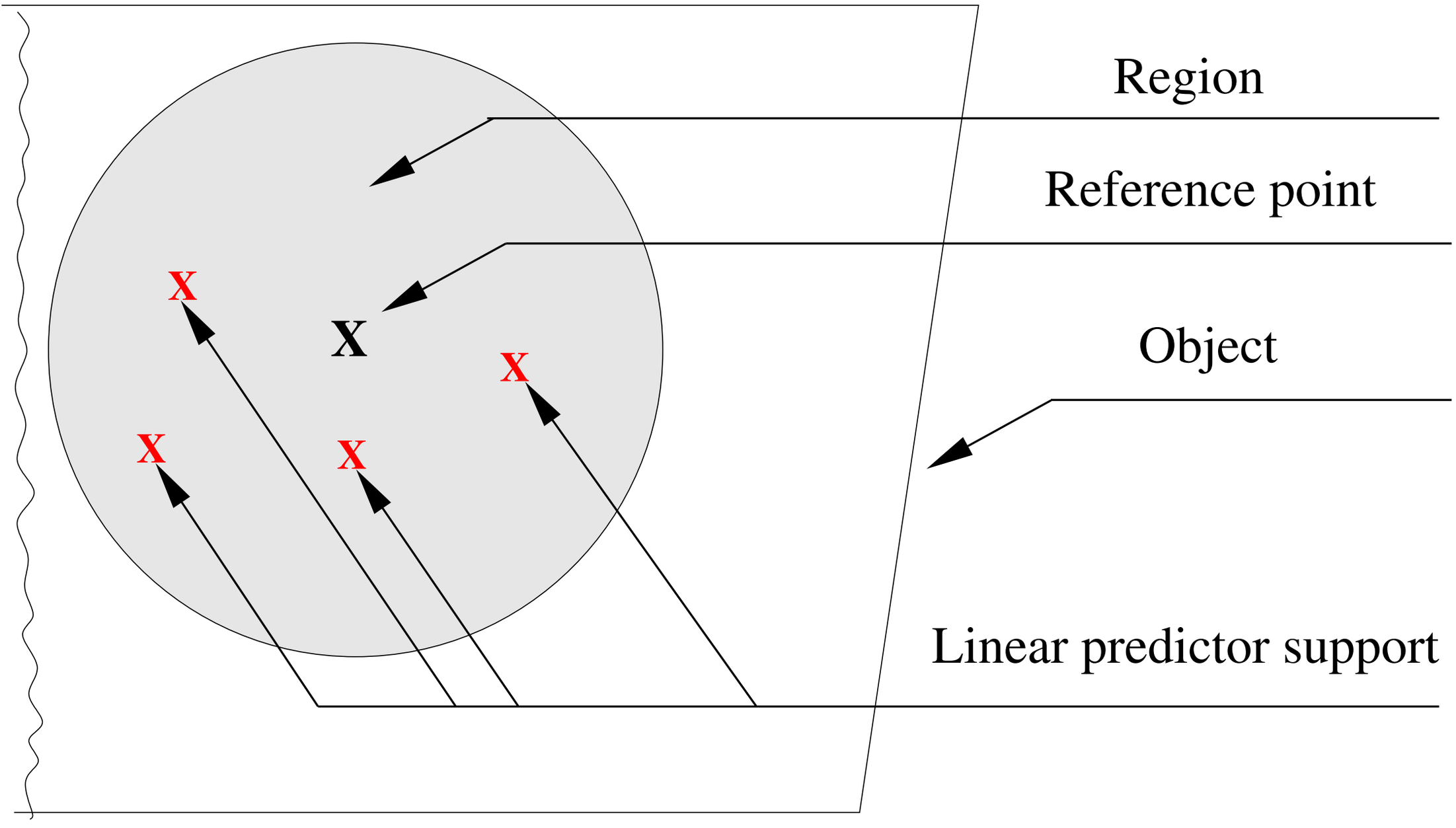
BASTONCINI
Crisp Almond Cookies
Bastoncini
alla Mandorla
PAN DUCALE
ATRI
Peso Netto 250 g
NET WT 8.8 OZ

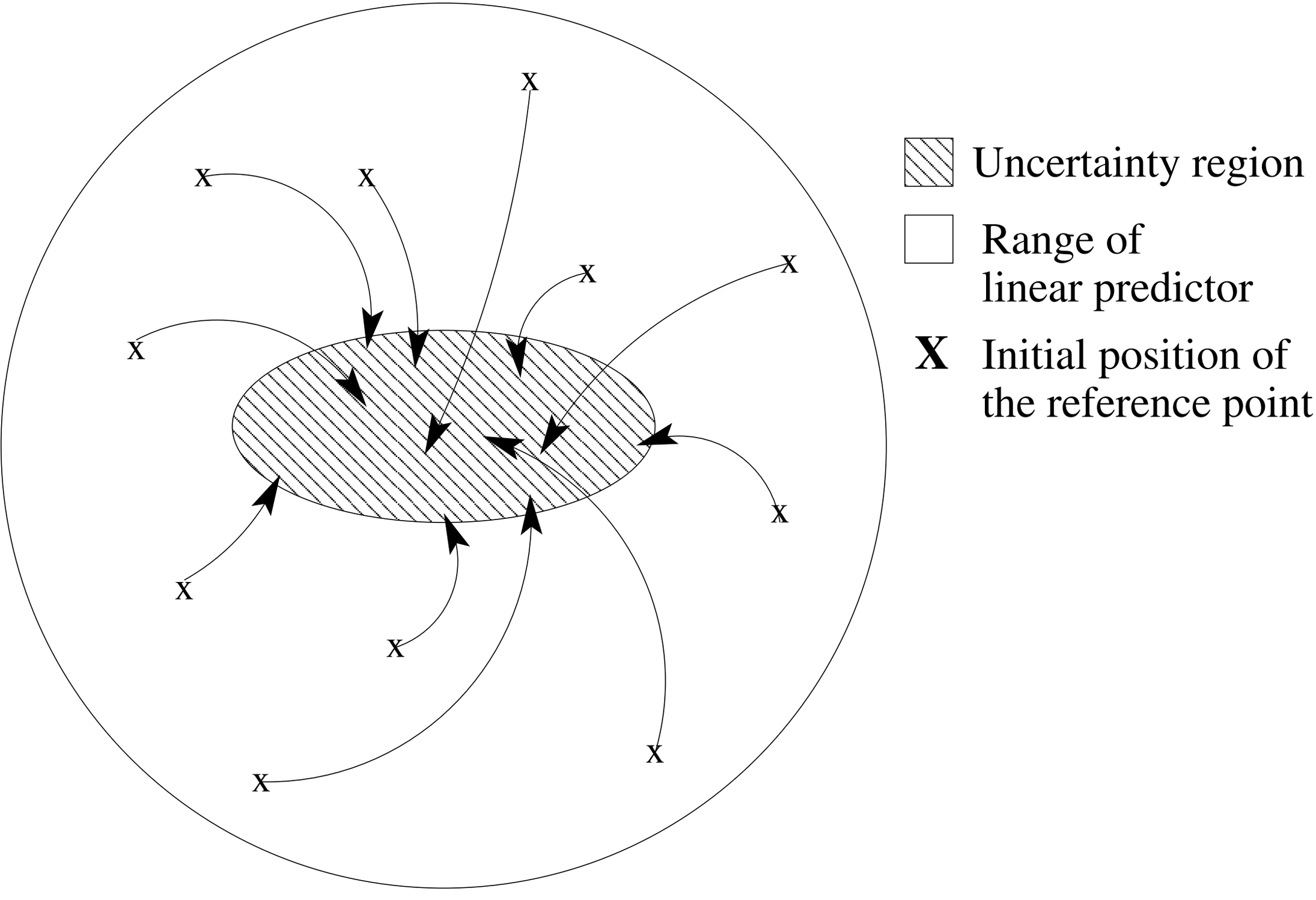

30
25
20
15
10
5
0
160
140
120
100
80
60
300
320
340
360
380
400

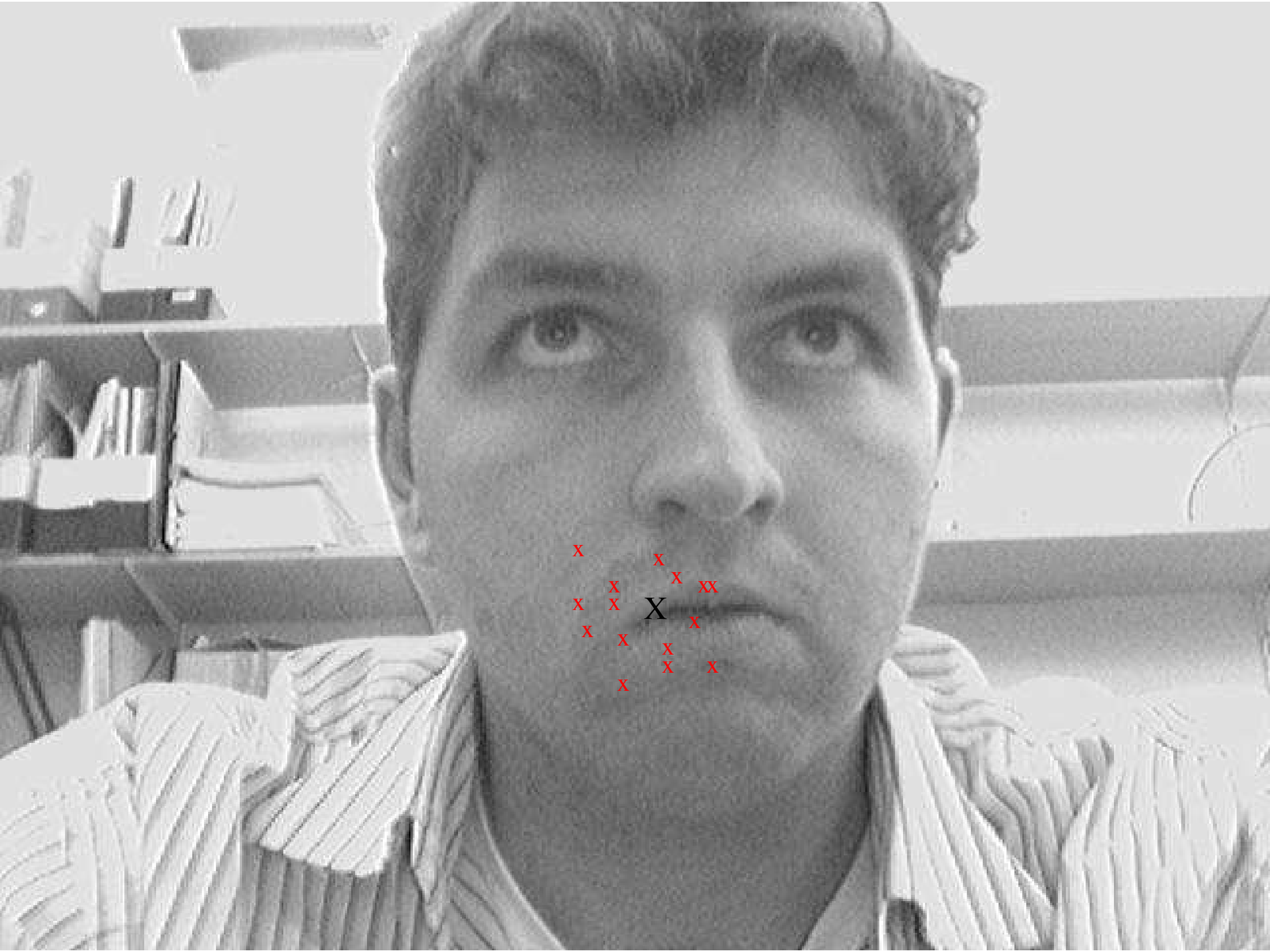

60
40
20
0
330
320
310
300
290
280
270
260
250
240
40
60
80
100
120
140

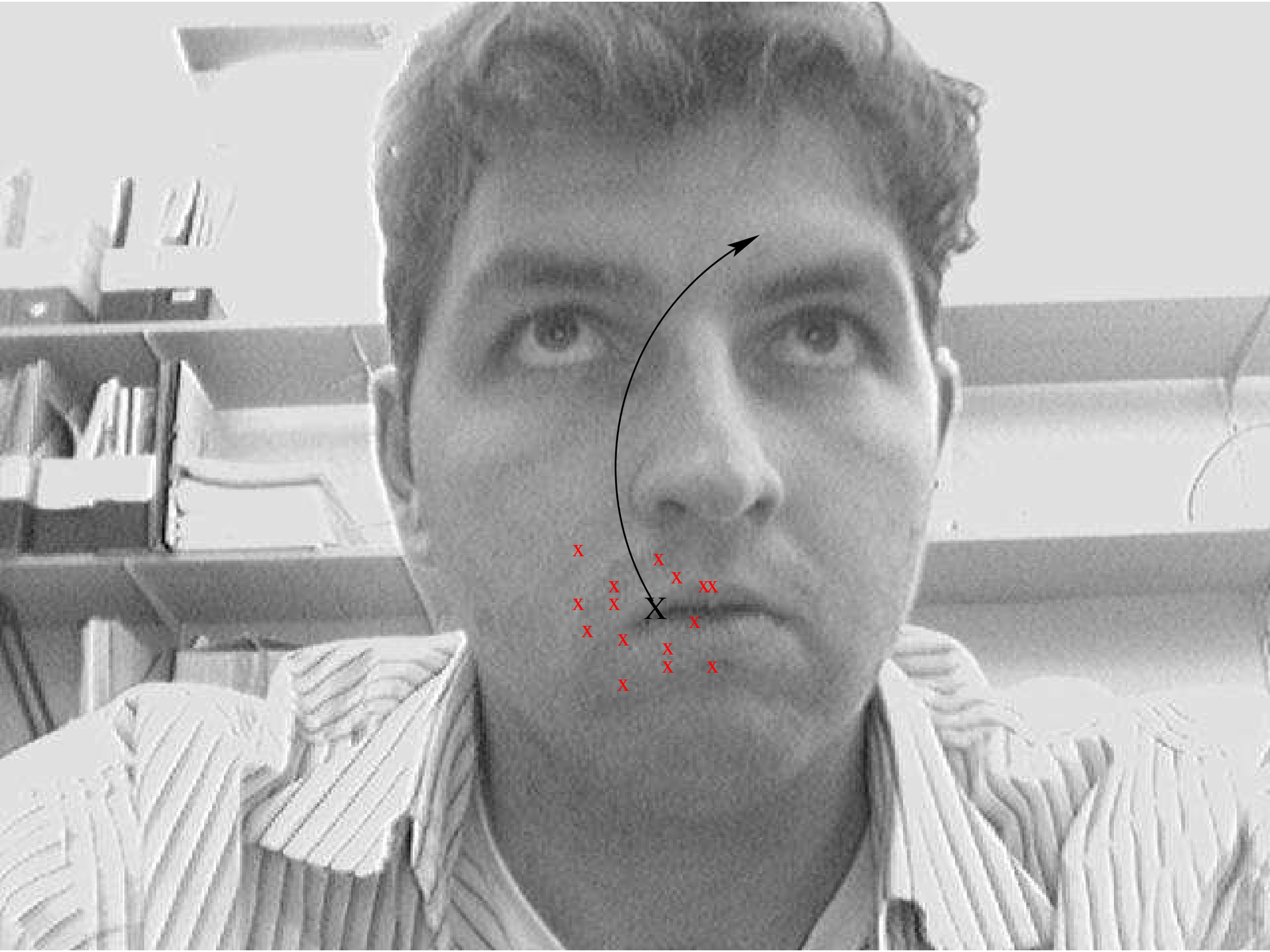

60
40
20
0
110
100
90
80
70
60
50
40
30
20
20
30
40
50
60
70
80
90
100
110


60
40
20
0
110
100
90
80
70
60
50
40
30
20
20
30
40
50
60
70
80
90
100
110


Region
Reference point
Object
Linear predictor support

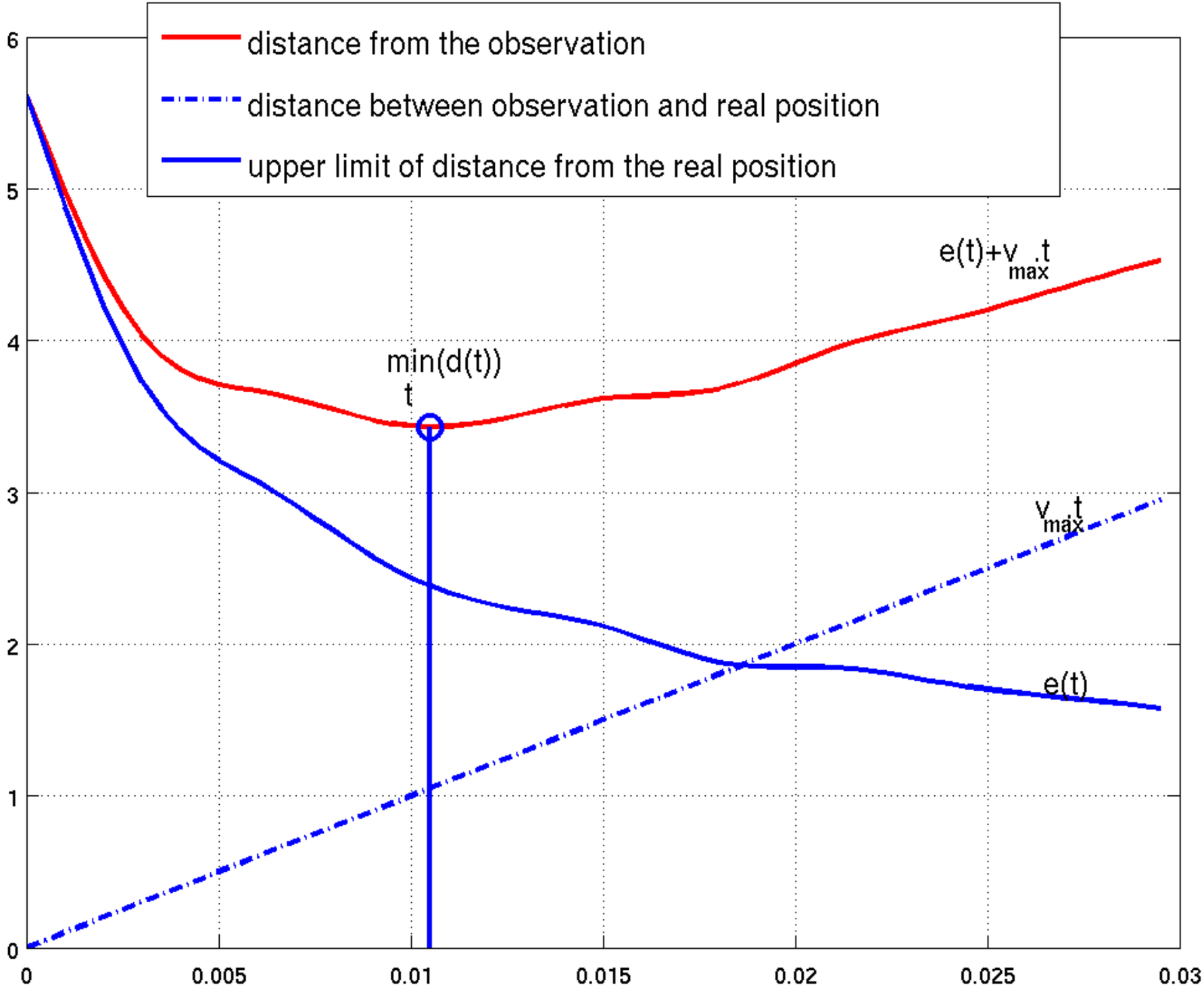

Uncertainty region
Range of linear predictor
X Initial position of the reference point

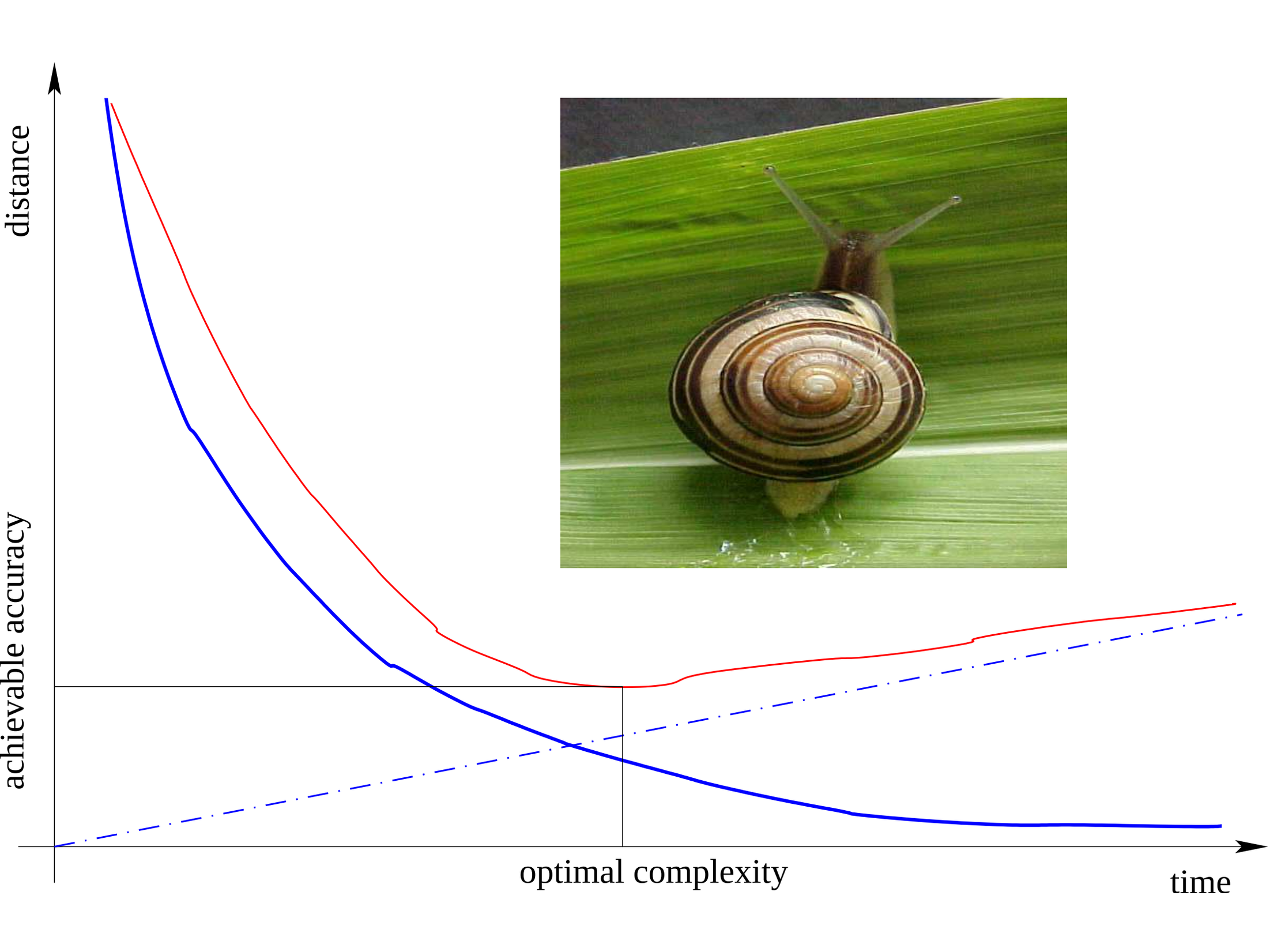


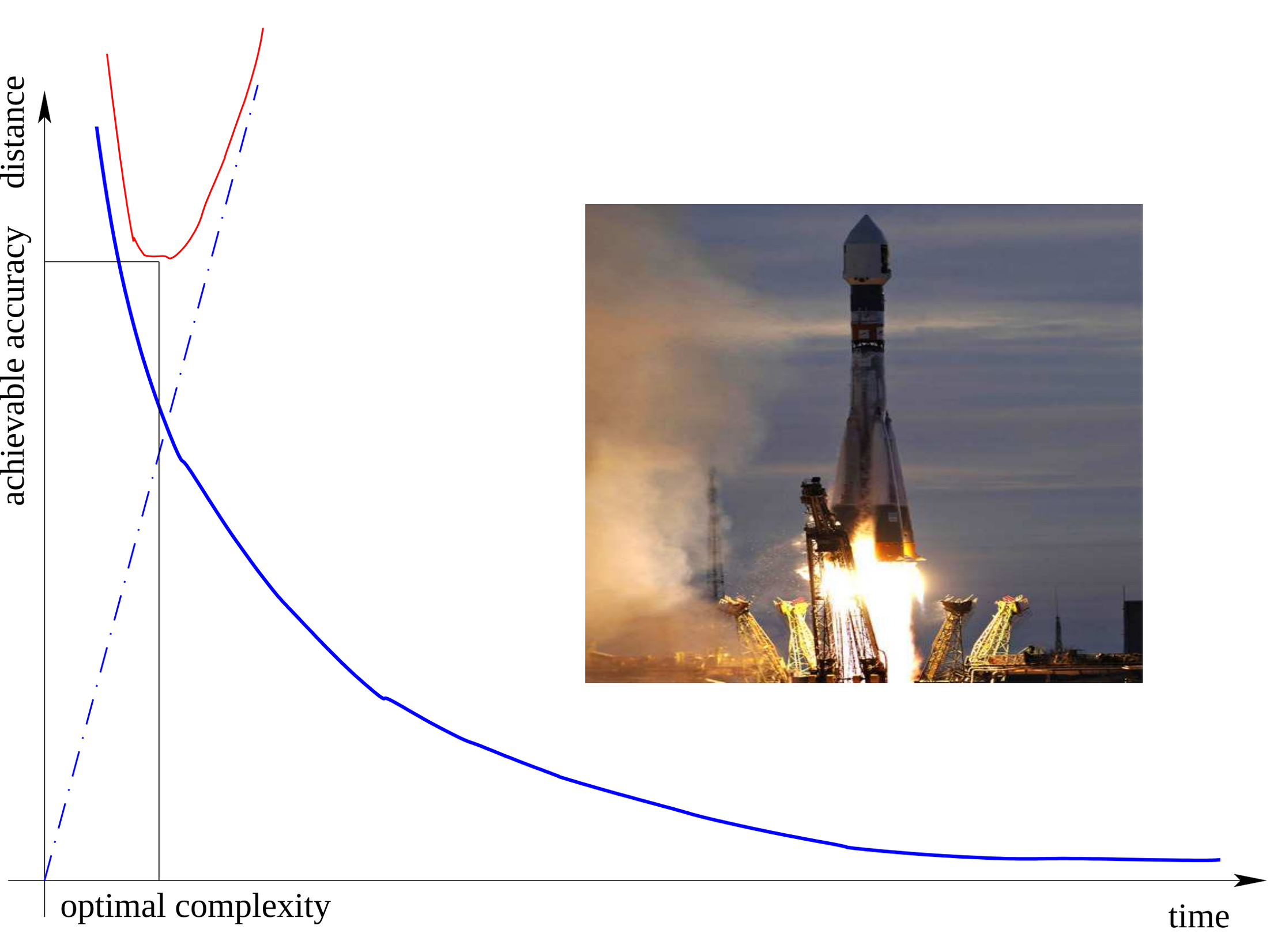


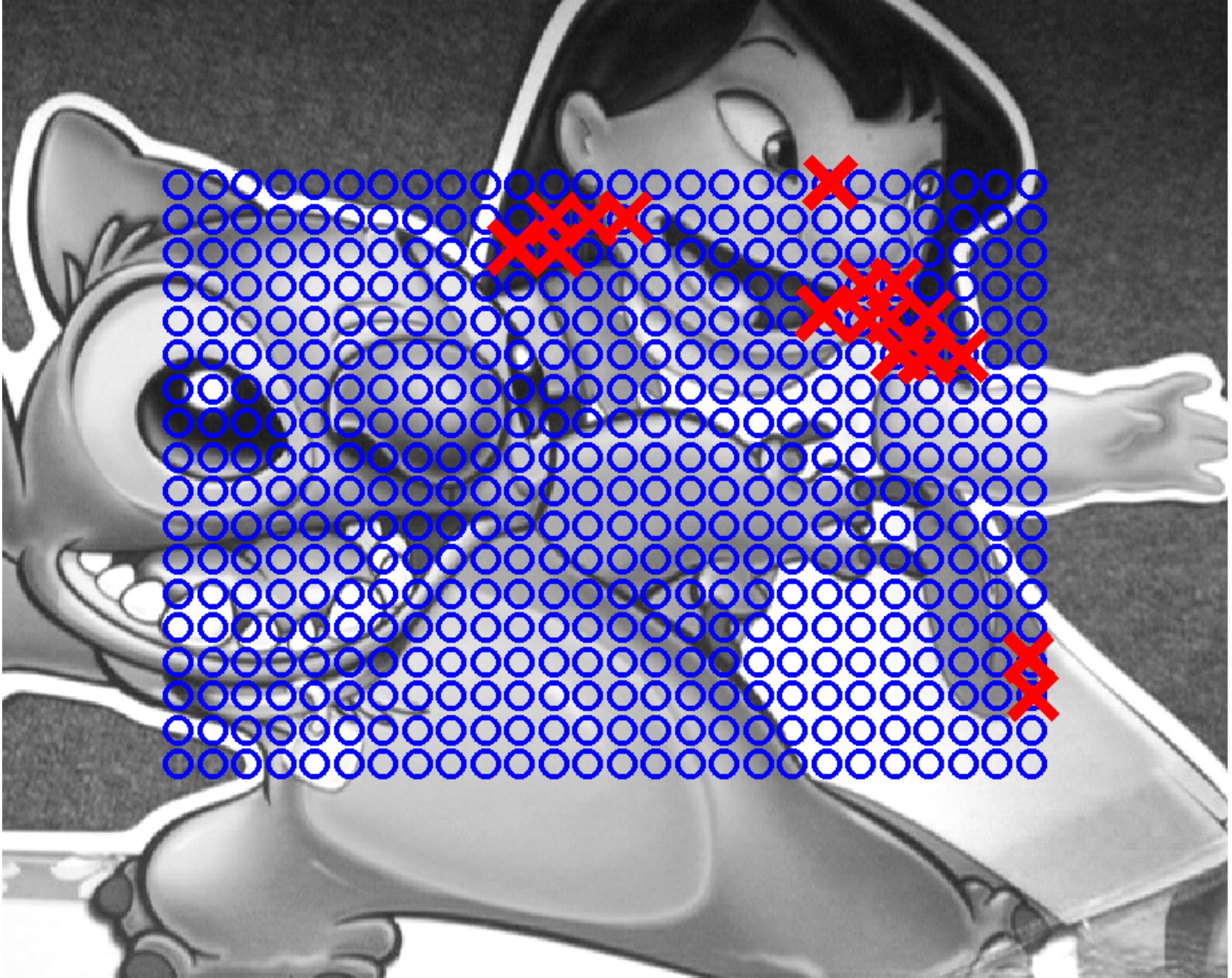
X
X

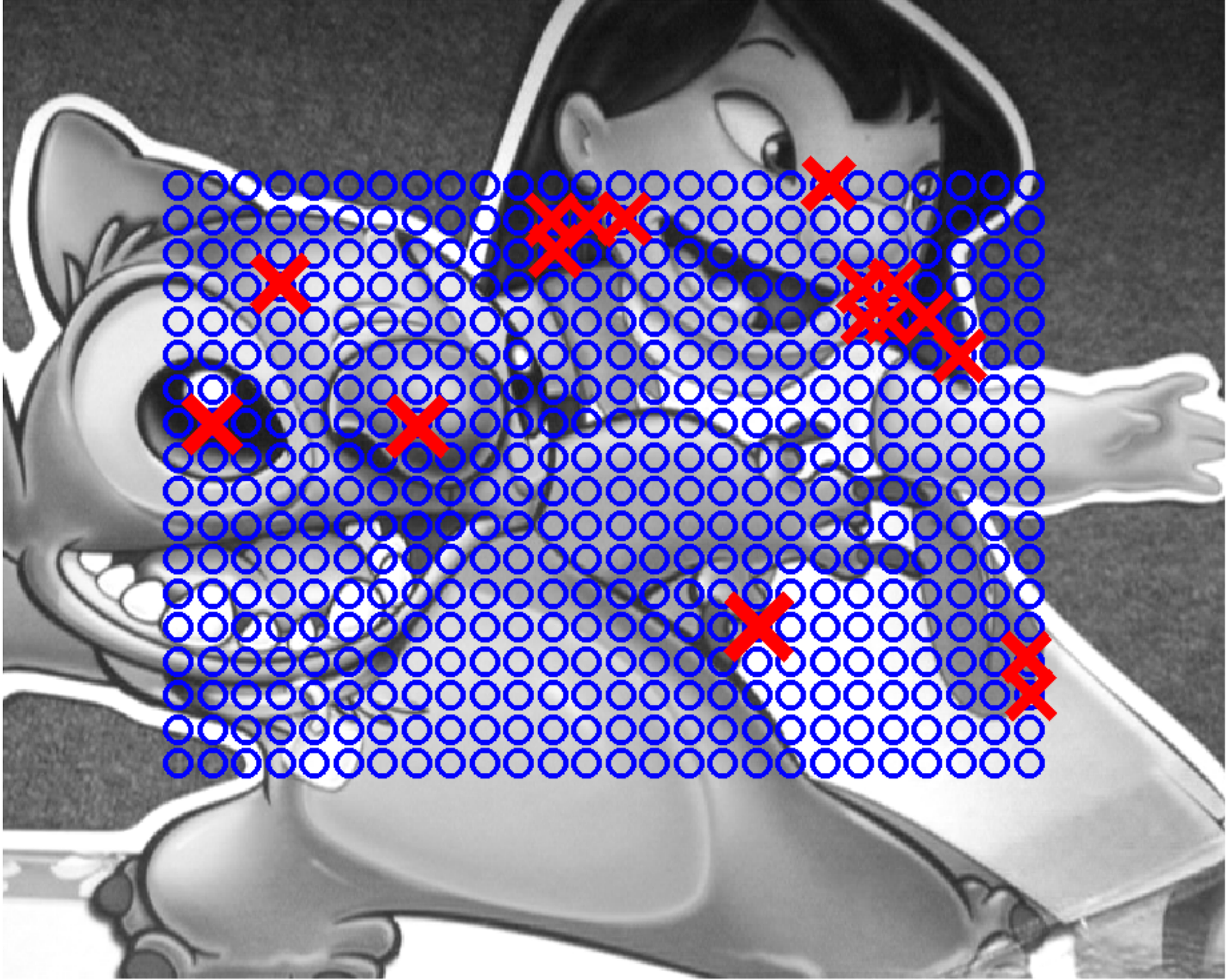
X

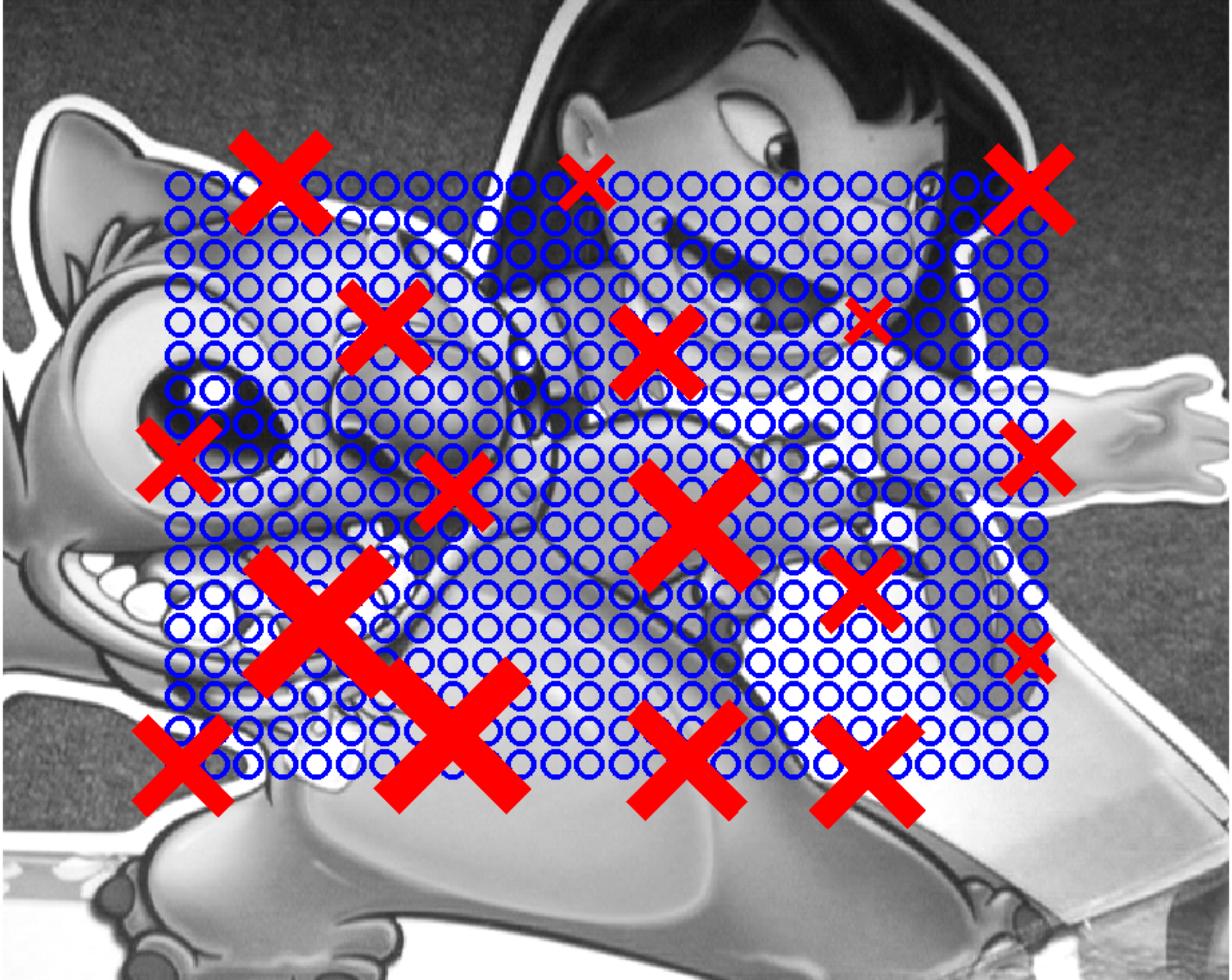

distance from the observation
distance between observation and real position
upper limit of distance from the real position
e(t)+v_max.t
min(d(t))
t
v_max.t
e(t)
6
5
4
3
2
1
0
0
0.005
0.01
0.015
0.02
0.025
0.03

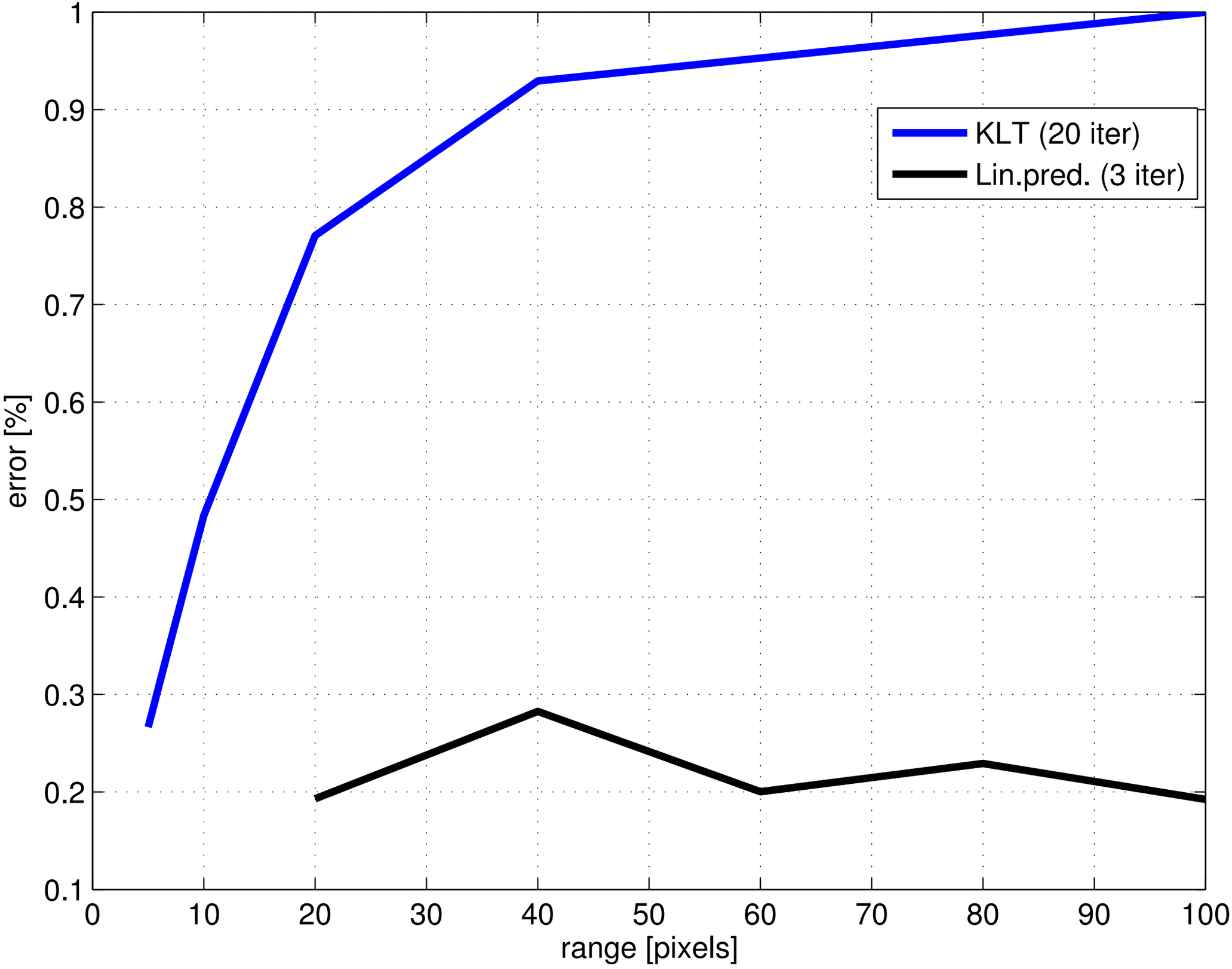

distance
achievable accuracy
optimal complexity
time

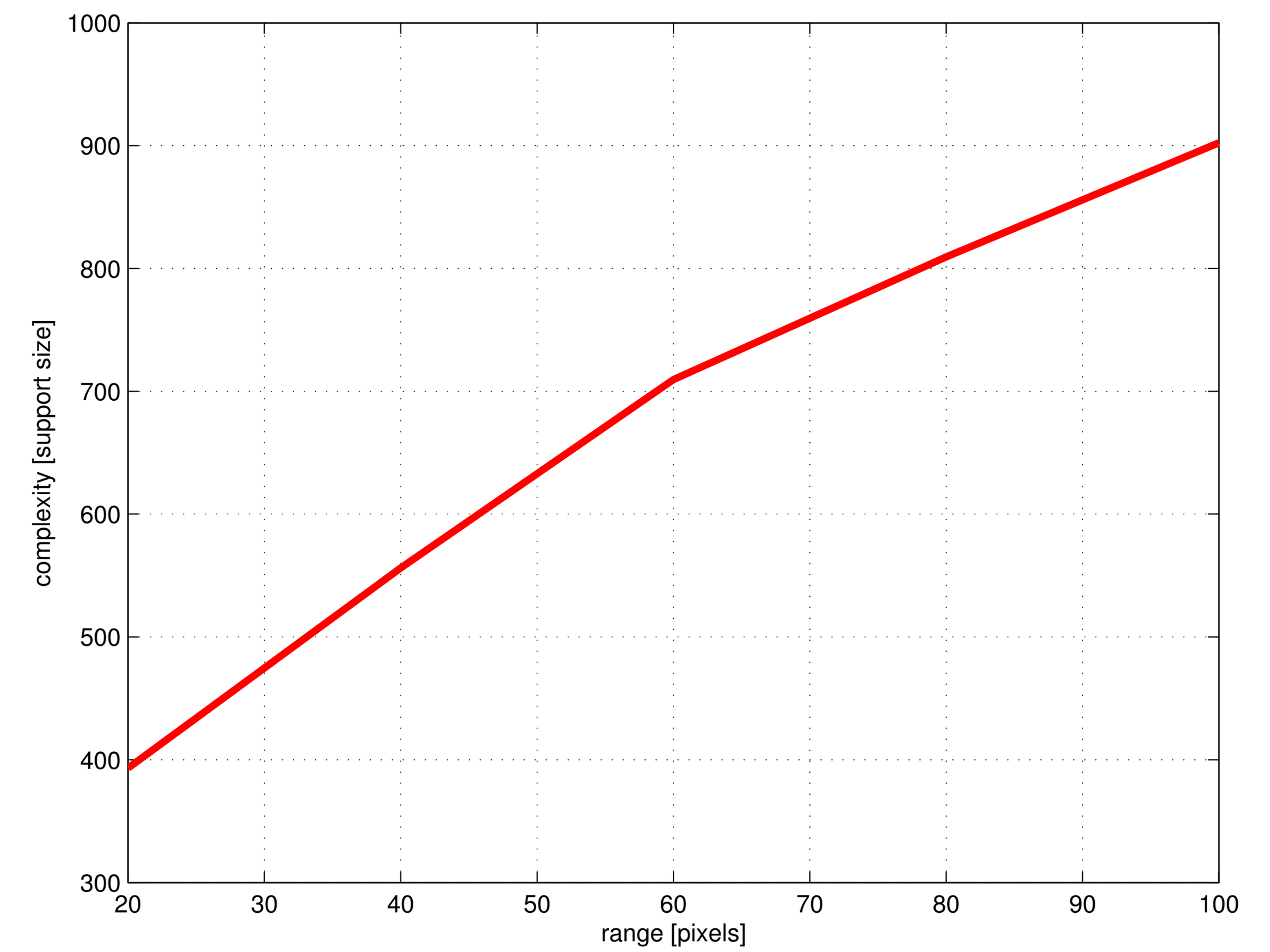
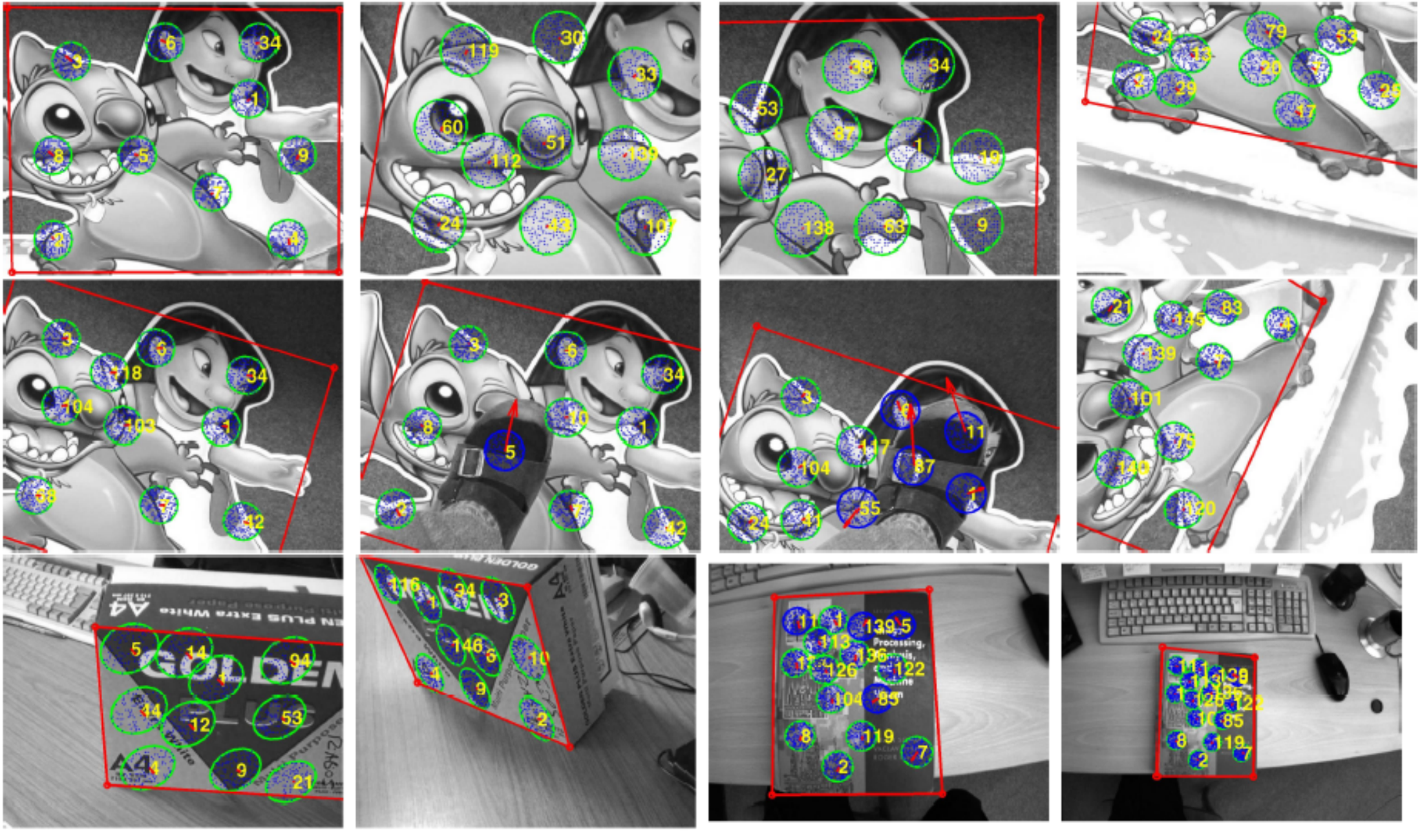

achievable accuracy
distance
optimal complexity
time








KLT (20 iter)
Lin.pred. (3 iter)
error [%]
range [pixels]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
60
70
80
90
100


1000
900
800
700
600
500
400
300
complexity [support size]
20
30
40
50
60
70
80
90
100
range [pixels]

81 VIA
NORD/NOORD (N)