An Optimal Sequence of Learned Motion Estimators

Karel Zimmermann\textsuperscript{1}, Jiří Matas\textsuperscript{1},
Tomáš Svoboda\textsuperscript{1,2}

\textsuperscript{1}: Center for Machine Perception
\textsuperscript{2}: Center for Applied Cybernetics
Czech Technical University
Prague, Czech Republic
Introduction

Tracking objectives:

- Fast
- Accurate
- Robust
State-of-the-art: Tracking by gradient optimization

- **Minimize dissimilarity:** \( t = \arg \min_t \sum (I(x + t) - J(x))^2 \)


- **Drawbacks:**
  - Convergence to a local minimum
  - Unknown basin of attraction
  - Criterial function
State-of-the-art: Tracking by regression

There is an inverse relation approximated by mapping

\[ \Phi : \text{intensities around a point} \rightarrow \text{motion} \]
State-of-the-art: Tracking by regression

◆ **Linear motion regression:** \( t = H(I(x) - J(x)) \)


◆ **Non-linear motion regression:** \( RVM \)

Our approach

- **Sequential motion regression:** \( t = \varphi_h\left( \ldots I(x + \varphi_1(I(x))) \right) \)

- We are looking for a sequence of predictors \( \Phi = [\varphi_1, \varphi_2, \ldots \varphi_h] \) with the lowest complexity.
  - How many iterations \( h \) are required?
  - How many pixels are necessary for each iteration?
  - What neighbouring pixels are used?
- **Range** $r$ the set of admissible motions.
- **Complexity** $c$ cardinality of support set.
- **Uncertainty region** $\lambda$ the region within which all the estimations lie.
Optimal sequence of optimal predictors

- **Predictors** $\phi_i(c, r, \lambda)$ lie in a subspace of the $(c, r, \lambda)$-space.
- **Optimal sequence of predictors** is a sequence $\Phi = [\varphi_1, \varphi_2, \ldots, \varphi_h]$ with the lowest total complexity $\sum c_i$ given:
  - range $r_1$ of the first predictor
  - uncertainty region $\lambda_h$ of the last predictor.
  - $r_{i+1} \geq \lambda_i$, $i = 1, \ldots, h - 1$. 
An optimal sequence

- Only those predictors lying on the $\lambda$-lower bound of the set of achievable predictors can create an optimal sequence $\hat{\Theta}$.
- Given $(c, r)$, minimax task is solved to find the predictor with the smallest uncertainty region.
- Color codes the size of the uncertainty region.
Searching for an optimal sequence.

- Dynamic programming searches for an optimal sequence of predictors.
- The algorithm searches for the cheapest path to a sufficiently small uncertainty region.
- In each state either complexity is increased or the next iteration initialized.
Greedy LSQ selection (red) of an efficient support set.

Much better than 1%-quantile (green) achievable by randomized sampling.
Online selection of an active predictor set

a) $w=0$  b) $w=0.1$  c) $w=0.5$  d) $w=1$

- Greedy online selection.
- Trade-off between abilities of local predictors and coverage of an object.
- Strong features may not provide good tracking results.
Probability of successful tracking as a function of number of ransac iterations and predictors.

We maximize the probability, given a time, we are allowed to spent with the motion estimation in the actual frame,
Motion blur, fast motion, views from acute angles and other image distortions.
Experiments: 3D fast blured tracking

a) slow motion  
b) fast blured motion  
c) close view
Experiments: Results on sequences 2000-7000 frames.

<table>
<thead>
<tr>
<th>object</th>
<th>processing</th>
<th>loss-of-locks</th>
<th>mean-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>mouse pad minmax</td>
<td>18.9fps</td>
<td>13/6935</td>
<td>[1.3%, 1.8%, 1.5%, 1.6%]</td>
</tr>
<tr>
<td>mouse pad sift</td>
<td>0.5fps</td>
<td>281/6935</td>
<td>[1.6%, 1.2%, 1.5%, 1.4%]</td>
</tr>
<tr>
<td>towel minmax</td>
<td>21.8fps</td>
<td>5/3229</td>
<td>[3.0%, 2.2%, 1.4%, 1.9%]</td>
</tr>
<tr>
<td>phone minmax</td>
<td>16.8fps</td>
<td>20/1799</td>
<td>[1.2%, 1.8%, 2.6%, 1.9%]</td>
</tr>
</tbody>
</table>

- Data captured at 22.7fps frame-rate.

- Comparison to SIFT detector.
Experiments: Comparison with KLT.

- Much lower complexity and substantially smaller error rate.
- If the number of iteration is constant than error rate is independent of the range.
Experiments: Application to a face detector.

<table>
<thead>
<tr>
<th></th>
<th>memory accesses</th>
<th>summations</th>
<th>multiplications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>15</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Detector</td>
<td>25</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>Align + Det</td>
<td>6.5</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>
Conclusion

◆ Drawbacks:
  - Learning required.
  - Predictor range is limited by the size of the object.

◆ Advantages:
  - Very fast motion estimation (30µs per predictor).
  - Ability to cover arbitrary cases (bluring, change of appearance).
  - Automatic setup of tracking procedure.
Support set
Object of interest
Reference point
Local motion
$$\Phi \left( \begin{array}{c} \text{image} \\ \text{image} \end{array} \right) = \left( \begin{array}{c} 0,0 \\ 12,7 \end{array} \right)^T$$

$$\Phi \left( \begin{array}{c} \text{image} \\ \text{image} \end{array} \right) = \left( \begin{array}{c} -14,2 \\ -9,18 \end{array} \right)^T$$

$$\Phi \left( \begin{array}{c} \text{image} \\ \text{image} \end{array} \right) = \left( \begin{array}{c} 14,0 \\ -16,14 \end{array} \right)^T$$

$$\Phi \left( \begin{array}{c} \text{image} \\ \text{image} \end{array} \right) = \left( \begin{array}{c} 16,14 \\ -12,7 \end{array} \right)^T$$
\( \Phi = (\phi_1, \phi_2, \phi_3) \)

**Ranges**

**New position**

**Motion**

\( t_1 = \hat{\phi}_1 \)

\( t_2 = \hat{\phi}_2 \)

\( t_3 = \hat{\phi}_3 \)

**Old position**
Inachievable estimators

Achievable estimators \( \omega \)

\( \lambda \)-lower bound

\( \lambda \) - uncertainty region

\( r \) - range

\( c \) - complexity
The graph illustrates the relationship between the range ($r$), complexity ($c$), and uncertainty region ($\lambda$). It distinguishes between achievable and inachievable estimators. The achievable estimators are represented by the red surface, while the inachievable estimators are shown by the blue surface. The $\lambda$-lower bound is marked with an arrow pointing towards the achievable estimators.
A graph showing the log of error against range in pixels for different methods: LP+DP, KLT, and LP.
Detection

Alignment + Detection

Detection Detection Detection Detection Detection
false positive detection rate [%]
detector +−4%
predictor+detector +−10% (5 iter)