Simultaneous learning of motion and appearance

Karel Zimmermann$^{1,3}$, Tomáš Svoboda$^{1,2}$, Jiří Matas$^{1}$

$^1$CMP/CTU-Prague $^2$CAK/CTU-Prague $^3$Visics/KU-Leuven
Tracking of objects with variable appearance

- **Tracking** - iterative estimation of object pose (e.g., human head).

- **Variable appearance** - the way how the object looks like in the camera changes due to illumination, non-rigid deformation, out-of-plane rotation, ...
Tracking of objects with variable appearance

- **Tracking** - iterative estimation of object pose (e.g., human head).
- **Variable appearance** - the way how the object looks like in the camera changes due to illumination, non-rigid deformation, out-of-plane rotation, ...
Tracking of objects with variable appearance

- **Tracking** - iterative estimation of object pose (e.g., human head).

- **Variable appearance** - the way how the object looks like in the camera changes due to illumination, non-rigid deformation, out-of-plane rotation, ...
Motivation
State-of-the-art
Theory
Experiments
Conclusions


\[
\argmin_t \| I(t) - J \|^2
\]

image \( I \) → alignment \( t \)

\( J \leftarrow I(t) \)

optional template update

Lucas–Kanade

\[ H(I - J) \]

Image alignment

\[ J \leftarrow I(t) \]

Optional template update

Jurie–Dhome
Learning alignment

$\varphi(0, 0) = (0, 0)^T$

$\varphi(-25, 0) = (-25, 0)^T$

$\varphi(25, -15) = (25, -15)^T$
Learning alignment

\[ \phi(0, 0)^T \]
\[ \phi(-25, 0)^T \]
\[ \phi(25, -15)^T \]

\[ \phi(-25, 0)^T \]
\[ \phi(25, -15)^T \]
Learning alignment

\[ \varphi(\text{[Image]}) = (0, 0)^T \]
\[ \varphi(\text{[Image]}) = (-25, 0)^T \]
\[ \varphi(\text{[Image]}) = (25, -15)^T \]
Learning alignment

- \( \varphi(\text{face}) = (0, 0)^T \)
- \( \varphi(\text{mouth}) = (-25, 0)^T \)
- \( \varphi(\text{eyes}) = (25, -15)^T \)
Learning alignment

- \( \varphi(\text{neutral}) = (0, 0)^T \)
- \( \varphi(\text{angry}) = (-25, 0)^T \)
- \( \varphi(\text{surprised}) = (25, -15)^T \)
- \( \varphi(\text{frightened}) = (0, 0)^T \)
- \( \varphi(\text{scared}) = (-25, 0)^T \)
- \( \varphi(\text{happy}) = (25, -15)^T \)
Learning alignment

\[ \varphi(\text{neutral}) = (0, 0)^\top \]

\[ \varphi(\text{smile}) = (-25, 0)^\top \]

\[ \varphi(\text{surprise}) = (25, -15)^\top \]

\[ \varphi(\text{neutral}) = (0, 0)^\top \]

\[ \varphi(\text{smile}) = (-25, 0)^\top \]

\[ \varphi(\text{surprise}) = (25, -15)^\top \]
Learning appearance

image $I$ $\xrightarrow{H(I - J)}$ alignment $t$ $\xleftarrow{J \leftarrow I(t)}$

optional template update

**Jurie–Dhome**
Learning appearance

\[ H(I - J(\theta)) \]

\[ \theta \]

\[ \text{PCA}_J \]

Cootes–Edwards

State-of-the-art

Theory

Experiments

Conclusions
Learning appearance – our approach

image I → \( H(I - J(\theta)) \) → \( I(t) \)

Cootes–Edwards

\( \theta \)

PCA_J

image I → \( \varphi(I; \theta) \) → alignment \( t \)

\( \theta \) appearance parameters

Our algorithm

\( \gamma(I(t)) \) → \( I(t) \) image
Learning appearance – our approach

Motivation
State-of-the-art
Theory
Experiments
Conclusions

Our algorithm

\[ \hat{H}(I - J(\theta)) \]

Cootes–Edwards

PCA

ϕ(I; \theta)

γ(I(t))

image I

\[ \varphi(I; \theta) \]

\[ \gamma(I(t)) \]

motion estimator

appearance encoder

θ appearance parameters

Our algorithm

image I

I(t) image

I(t)

\[ \theta \]

\[ t \]
Learning the appearance encoder $\gamma$

- Current appearance encoded in low-dim parameters.

$\gamma() = \theta_1$

$\gamma() = \theta_2$
Learning the appearance encoder $\gamma$

- Current appearance encoded in low-dim parameters.

$\gamma(\text{face}) = \theta_1$

$\gamma(\text{mouth}) = \theta_2$
Learning the appearance encoder $\gamma$

- Current appearance encoded in low-dim parameters.

\[ \gamma(\text{current}) = \theta_1 \]

\[ \gamma(\text{changed}) = \theta_2 \]
Learning the tracker $\varphi(\mathbf{l}; \mathbf{\theta})$

$\varphi(\mathbf{l}; \mathbf{\theta}_1) = (0, 0)^T$

$\varphi(\mathbf{l}; \mathbf{\theta}_1) = (-25, 0)^T$

$\varphi(\mathbf{l}; \mathbf{\theta}_1) = (25, -15)^T$

$\varphi(\mathbf{l}; \mathbf{\theta}_2) = (0, 0)^T$

$\varphi(\mathbf{l}; \mathbf{\theta}_2) = (-25, 0)^T$

$\varphi(\mathbf{l}; \mathbf{\theta}_2) = (25, -15)^T$
Learning the tracker $\varphi(I; \theta)$

- $\varphi(I; \theta_1) = (0, 0)^T$
- $\varphi(I; \theta_1) = (-25, 0)^T$
- $\varphi(I; \theta_1) = (25, -15)^T$
- $\varphi(I; \theta_1) = (0, 0)^T$
- $\varphi(I; \theta_2) = (-25, 0)^T$
- $\varphi(I; \theta_2) = (25, -15)^T$
Learning the tracker $\varphi(I; \theta)$

- $\varphi(I; \theta_1) = (0, 0)^T$
- $\varphi(I; \theta_1) = (-25, 0)^T$
- $\varphi(I; \theta_1) = (25, -15)^T$
- $\varphi(I; \theta_2) = (0, 0)^T$
- $\varphi(I; \theta_2) = (-25, 0)^T$
- $\varphi(I; \theta_2) = (25, -15)^T$
Learning the tracker $\varphi(\mathbf{I}; \theta)$

- $\varphi(\mathbf{I}; \theta_1) = (0, 0)^T$
- $\varphi(\mathbf{I}; \theta_1) = (-25, 0)^T$
- $\varphi(\mathbf{I}; \theta_1) = (25, -15)^T$
- $\varphi(\mathbf{I}; \theta_2) = (0, 0)^T$
- $\varphi(\mathbf{I}; \theta_2) = (-25, 0)^T$
- $\varphi(\mathbf{I}; \theta_2) = (25, -15)^T$
Learning the tracker $\varphi(I; \theta)$

$\varphi(I; \theta_1) = (0, 0)^T$

$\varphi(I; \theta_1) = (-25, 0)^T$

$\varphi(I; \theta_1) = (25, -15)^T$

$\varphi(I; \theta_2) = (0, 0)^T$

$\varphi(I; \theta_2) = (-25, 0)^T$

$\varphi(I; \theta_2) = (25, -15)^T$
Learning the tracker $\varphi(I; \theta)$

- $\varphi(I; \theta_1) = (0, 0)^T$
- $\varphi(I; \theta_1) = (-25, 0)^T$
- $\varphi(I; \theta_1) = (25, -15)^T$
- $\varphi(I; \theta_2) = (0, 0)^T$
- $\varphi(I; \theta_2) = (-25, 0)^T$
- $\varphi(I; \theta_2) = (25, -15)^T$
Simultaneous learning of $\varphi$ and $\gamma$

Learning = minimization of the least-squares error

$$(\varphi^*, \gamma^*) = \arg\min_{\varphi, \gamma} \left[ \varphi(\,; \gamma(\,)) - (0, 0)^T \right]^2 +$$

$$\left[ \varphi(\,; \gamma(\,)) - (-25, 0)^T \right]^2 +$$

$$\left[ \varphi(\,; \gamma(\,)) - (25, -15)^T \right]^2 +$$

$$\left[ \varphi(\,; \gamma(\,)) - (0, 0)^T \right]^2 +$$

$$\left[ \varphi(\,; \gamma(\,)) - (-25, 0)^T \right]^2 +$$

$$\left[ \varphi(\,; \gamma(\,)) - (25, -15)^T \right]^2$$
Simultaneous learning of $\varphi$ and $\gamma$

- **Learning** = minimization of the least-squares error

$$(\varphi^*, \gamma^*) = \arg \min_{\varphi, \gamma} \left[ \begin{array}{c} \varphi(\text{Face}; \gamma(\text{Face})) - (0, 0)^T \\ \varphi(\text{Face}; \gamma(\text{Face})) - (-25, 0)^T \\ \varphi(\text{Face}; \gamma(\text{Face})) - (25, -15)^T \\ \varphi(\text{Face}; \gamma(\text{Face})) - (0, 0)^T \\ \varphi(\text{Face}; \gamma(\text{Face})) - (-25, 0)^T \\ \varphi(\text{Face}; \gamma(\text{Face})) - (25, -15)^T \end{array} \right]^2$$
Linear mapping

- $\gamma(J) : \theta = GJ$
- $\varphi(I, \theta) : t = (H_0 + \theta_1H_1 + \cdots + \theta_nH_n)I$
- Criterion is sum of squares of bilinear functions.
Linear mapping

- $\gamma(J) : \theta = GJ$
- $\varphi(I, \theta) : t = (H_0 + \theta_1 H_1 + \cdots + \theta_n H_n) I$
- Criterion is sum of squares of bilinear functions.
Linear mapping

- $\gamma(J) : \theta = GJ$
- $\varphi(I, \theta) : t = (H_0 + \theta_1H_1 + \cdots + \theta_nH_n)I$
- Criterion is sum of squares of bilinear functions.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

- Iterative minimization:
  - initialization $\gamma^0 = \text{rand}$
  - $\varphi^1 = \arg \min \varphi e(\varphi, \gamma^0)$
  - $\gamma^1 = \arg \min \gamma e(\varphi^1, \gamma)$
  - $\varphi^2 = \arg \min \varphi e(\varphi^1, \gamma^1)$
  - $\gamma^2 = \arg \min \gamma e(\varphi^2, \gamma)$
  - $\varphi^3 = \arg \min \varphi e(\varphi^2, \gamma^2)$
  - $\gamma^3 = \arg \min \gamma e(\varphi^3, \gamma)$
  - until convergence reached

- Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion \( e(\varphi, \gamma) \)

\[ \begin{align*}
\varphi &\quad = \text{motion (geometry mapping)}, \\
\gamma &\quad = \text{appearance mapping}
\end{align*} \]

Iterative minimization:

- Initialization: \( \gamma^0 = \text{rand} \)
- \( \varphi^1 = \arg\min_\varphi e(\varphi, \gamma^0) \)
- \( \gamma^1 = \arg\min_\gamma e(\varphi^1, \gamma) \)
- \( \varphi^2 = \arg\min_\varphi e(\varphi, \gamma^1) \)
- \( \gamma^2 = \arg\min_\gamma e(\varphi^2, \gamma) \)
- \( \varphi^3 = \arg\min_\varphi e(\varphi, \gamma^2) \)
- \( \gamma^3 = \arg\min_\gamma e(\varphi^3, \gamma) \)
- Until convergence reached

- Global optimality for linear \( \varphi, \gamma \) experimentally shown.
Algorithm: iterative minimization of criterion \( e(\varphi, \gamma) \)

\( \varphi \) – motion (geometry mapping), \( \gamma \) – appearance mapping

color encodes criterion value \( e(\varphi, \gamma) \)

- **Iterative minimization:**
  - **init**ialization \( \gamma^0 = \text{rand} \)
  - \( \varphi^1 = \arg \min \varphi e(\varphi, \gamma^0) \)
  - \( \gamma^1 = \arg \min \gamma e(\varphi^1, \gamma) \)
  - \( \varphi^2 = \arg \min \varphi e(\varphi, \gamma^1) \)
  - \( \gamma^2 = \arg \min \gamma e(\varphi^2, \gamma) \)
  - \( \varphi^3 = \arg \min \varphi e(\varphi, \gamma^2) \)
  - \( \gamma^3 = \arg \min \gamma e(\varphi^3, \gamma) \)
  - until convergence reached

- Global optimality for linear \( \varphi, \gamma \) experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

- $\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping
- color encodes criterion value $e(\varphi, \gamma)$

▶ Iterative minimization:
  ▶ initialization $\gamma^0 = \text{rand}$
  ▶ $\varphi^1 = \arg\min_{\varphi} e(\varphi, \gamma^0)$
  ▶ $\gamma^1 = \arg\min_{\gamma} e(\varphi^1, \gamma)$
  ▶ $\varphi^2 = \arg\min_{\varphi} e(\varphi, \gamma^1)$
  ▶ $\gamma^2 = \arg\min_{\gamma} e(\varphi^2, \gamma)$
  ▶ $\varphi^3 = \arg\min_{\varphi} e(\varphi, \gamma^2)$
  ▶ $\gamma^3 = \arg\min_{\gamma} e(\varphi^3, \gamma)$
  ▶ until convergence reached

▶ Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

- Iterative minimization:
  - initialization $\gamma^0 = \text{rand}$
  - $\varphi^1 = \arg\min_{\varphi} e(\varphi, \gamma^0)$
  - $\gamma^1 = \arg\min_{\gamma} e(\varphi^1, \gamma)$
  - $\varphi^2 = \arg\min_{\varphi} e(\varphi, \gamma^1)$
  - $\gamma^2 = \arg\min_{\gamma} e(\varphi^2, \gamma)$
  - $\varphi^3 = \arg\min_{\varphi} e(\varphi, \gamma^2)$
  - $\gamma^3 = \arg\min_{\gamma} e(\varphi^3, \gamma)$
  - until convergence reached

- Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

- Iterative minimization:
  - initialization $\gamma^0 = \text{rand}$
  - $\varphi^1 = \arg\min_\varphi e(\varphi, \gamma^0)$
  - $\gamma^1 = \arg\min_\gamma e(\varphi^1, \gamma)$
  - $\varphi^2 = \arg\min_\varphi e(\varphi, \gamma^1)$
  - $\gamma^2 = \arg\min_\gamma e(\varphi^2, \gamma)$
  - $\varphi^3 = \arg\min_\varphi e(\varphi, \gamma^2)$
  - $\gamma^3 = \arg\min_\gamma e(\varphi^3, \gamma)$
  - until convergence reached

- Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

▶ Iterative minimization:
  ▶ initialization $\gamma^0 = \text{rand}$
  ▶ $\varphi^1 = \text{arg min}_\varphi e(\varphi, \gamma^0)$
  ▶ $\gamma^1 = \text{arg min}_\gamma e(\varphi^1, \gamma)$
  ▶ $\varphi^2 = \text{arg min}_\varphi e(\varphi, \gamma^1)$
  ▶ $\gamma^2 = \text{arg min}_\gamma e(\varphi^2, \gamma)$
  ▶ $\varphi^3 = \text{arg min}_\varphi e(\varphi, \gamma^2)$
  ▶ $\gamma^3 = \text{arg min}_\gamma e(\varphi^3, \gamma)$
  ▶ until convergence reached

▶ Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

- Iterative minimization:
  - initialization $\gamma^0 = \text{rand}$
  - $\varphi^1 = \arg\min_\varphi e(\varphi, \gamma^0)$
  - $\gamma^1 = \arg\min_\gamma e(\varphi^1, \gamma)$
  - $\varphi^2 = \arg\min_\varphi e(\varphi, \gamma^1)$
  - $\gamma^2 = \arg\min_\gamma e(\varphi^2, \gamma)$
  - $\varphi^3 = \arg\min_\varphi e(\varphi, \gamma^2)$
  - $\gamma^3 = \arg\min_\gamma e(\varphi^3, \gamma)$
  - until convergence reached

- Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion \( e(\varphi, \gamma) \)

\( \varphi \) – motion (geometry mapping), \( \gamma \) – appearance mapping

color encodes criterion value \( e(\varphi, \gamma) \)

- Iterative minimization:
  - initialization \( \gamma^0 = \text{rand} \)
  - \( \varphi^1 = \arg \min_\varphi e(\varphi, \gamma^0) \)
  - \( \gamma^1 = \arg \min_\gamma e(\varphi^1, \gamma) \)
  - \( \varphi^2 = \arg \min_\varphi e(\varphi, \gamma^1) \)
  - \( \gamma^2 = \arg \min_\gamma e(\varphi^2, \gamma) \)
  - \( \varphi^3 = \arg \min_\varphi e(\varphi, \gamma^2) \)
  - \( \gamma^3 = \arg \min_\gamma e(\varphi^3, \gamma) \)
  - until convergence reached

- Global optimality for linear \( \varphi, \gamma \) experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

- Iterative minimization:
  - initialization $\gamma^0 = \text{rand}$
  - $\varphi^1 = \arg\min_{\varphi} e(\varphi, \gamma^0)$
  - $\gamma^1 = \arg\min_{\gamma} e(\varphi^1, \gamma)$
  - $\varphi^2 = \arg\min_{\varphi} e(\varphi, \gamma^1)$
  - $\gamma^2 = \arg\min_{\gamma} e(\varphi^2, \gamma)$
  - $\varphi^3 = \arg\min_{\varphi} e(\varphi, \gamma^2)$
  - $\gamma^3 = \arg\min_{\gamma} e(\varphi^3, \gamma)$
  - until convergence reached

- Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

Iterative minimization:
- initialization $\gamma^0 = \text{rand}$
- $\varphi^1 = \text{arg min}_\varphi e(\varphi, \gamma^0)$
- $\gamma^1 = \text{arg min}_\gamma e(\varphi^1, \gamma)$
- $\varphi^2 = \text{arg min}_\varphi e(\varphi, \gamma^1)$
- $\gamma^2 = \text{arg min}_\gamma e(\varphi^2, \gamma)$
- $\varphi^3 = \text{arg min}_\varphi e(\varphi, \gamma^2)$
- $\gamma^3 = \text{arg min}_\gamma e(\varphi^3, \gamma)$
- until convergence reached

Global optimality for linear $\varphi, \gamma$ experimentally shown.
Algorithm: iterative minimization of criterion $e(\varphi, \gamma)$

$\varphi$ – motion (geometry mapping), $\gamma$ – appearance mapping

color encodes criterion value $e(\varphi, \gamma)$

Iterative minimization:
- initialization $\gamma^0 = \text{rand}$
- $\varphi^1 = \arg \min_\varphi e(\varphi, \gamma^0)$
- $\gamma^1 = \arg \min_\gamma e(\varphi^1, \gamma)$
- $\varphi^2 = \arg \min_\varphi e(\varphi, \gamma^1)$
- $\gamma^2 = \arg \min_\gamma e(\varphi^2, \gamma)$
- $\varphi^3 = \arg \min_\varphi e(\varphi, \gamma^2)$
- $\gamma^3 = \arg \min_\gamma e(\varphi^3, \gamma)$
- until convergence reached

Global optimality for linear $\varphi$, $\gamma$ experimentally shown.
Experiments - videos I
Experiments - videos II
Comparison of tracking error and computation cost of appearance sensitive and simple tracker.

<table>
<thead>
<tr>
<th>Object</th>
<th>Appearance sensitive</th>
<th>Simple</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error&lt;sub&gt;train&lt;/sub&gt;</td>
<td>Error&lt;sub&gt;test&lt;/sub&gt;</td>
</tr>
<tr>
<td>CUP</td>
<td>5.1</td>
<td>5.3</td>
</tr>
<tr>
<td>BASIL</td>
<td>4.9</td>
<td>5.0</td>
</tr>
<tr>
<td>SIBIL</td>
<td>4.9</td>
<td>7.4</td>
</tr>
<tr>
<td>HEAD 1</td>
<td>2.7</td>
<td>3.4</td>
</tr>
<tr>
<td>HEAD 2</td>
<td>5.0</td>
<td>5.5</td>
</tr>
<tr>
<td>HEAD 2</td>
<td>4.3</td>
<td>4.5</td>
</tr>
<tr>
<td>FLOWER</td>
<td>3.3</td>
<td>3.6</td>
</tr>
</tbody>
</table>
Experiments - Quantitative evaluation

- Comparison of tracking error and computation cost of appearance sensitive and simple tracker.

<table>
<thead>
<tr>
<th>Object</th>
<th>Appearance sensitive</th>
<th>Simple</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\text{Error}_{\text{train}}$</td>
<td>$\text{Error}_{\text{test}}$</td>
</tr>
<tr>
<td>CUP</td>
<td>5.1</td>
<td>5.3</td>
</tr>
<tr>
<td>BASIL</td>
<td>4.9</td>
<td>5.0</td>
</tr>
<tr>
<td>SIBIL</td>
<td>4.9</td>
<td>7.4</td>
</tr>
<tr>
<td>HEAD 1</td>
<td>2.7</td>
<td>3.4</td>
</tr>
<tr>
<td>HEAD 2</td>
<td>5.0</td>
<td>5.5</td>
</tr>
<tr>
<td>HEAD 2</td>
<td>4.3</td>
<td>4.5</td>
</tr>
<tr>
<td>FLOWER</td>
<td>3.3</td>
<td>3.6</td>
</tr>
</tbody>
</table>
Conclusions

- New method for tracking of objects with variable appearance.
- Simultaneous learning motion and appearance.
- Possible extension to sequential estimation as shown in [Zimmermann-PAMI-2008].
- Significant improvement in accuracy, robustness and computational cost w.r.t. appearance insensitive tracker.

Conclusions

- New method for tracking of objects with variable appearance.
- Simultaneous learning motion and appearance.
- Possible extension to sequential estimation as shown in [Zimmermann-PAMI-2008].
- Significant improvement in accuracy, robustness and computational cost w.r.t. appearance insensitive tracker.

Conclusions

- New method for tracking of objects with variable appearance.
- Simultaneous learning motion and appearance.
- Possible extension to sequential estimation as shown in [Zimmermann-PAMI-2008].
- Significant improvement in accuracy, robustness and computational cost w.r.t. appearance insensitive tracker.

Conclusions

- New method for tracking of objects with variable appearance.
- Simultaneous learning motion and appearance.
- Possible extension to sequential estimation as shown in [Zimmermann-PAMI-2008].
- Significant improvement in accuracy, robustness and computational cost w.r.t. appearance insensitive tracker.