DENSE-HOG-BASED 3D FACE TRACKING FOR INFANT PAIN MONITORING

Ronald W.J.J. Saeijs† Walther E. Tjon a Ten† Peter H.N. de With†

† Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands
† Department of Pediatrics, Máxima Medical Center Veldhoven, The Netherlands

ABSTRACT
This paper presents a new algorithm for 3D face tracking intended for clinical infant pain monitoring under challenging conditions. The algorithm uses a cylinder head model and head pose recovery by alignment of dynamically extracted templates based on dense-HOG features. The algorithm is motivated from the application context and compared with a variant based on intensities. The paper reports experimental results on videos of moving infants in hospital who are relaxed or in pain. Results show good short-term tracking behavior for poses up to 50 degrees from upright-frontal, with significantly higher accuracy resulting from the use of dense-HOG features.

Index Terms— Face tracking, pain monitoring, cylinder head model, dense HOG

1. INTRODUCTION

Pain monitoring for infants and toddlers is important in many clinical contexts. Currently, pain monitoring for children is based on repeated pain assessments performed by healthcare professionals. Because assessments are time-consuming, non-continuous and only moderately reliable, there is a need for automatic pain monitoring.

Our goal is to pursue real-time continuous monitoring in a clinical setting. An example of a context in which continuous pain monitoring will be highly beneficial is the diagnosis of gastro-esophageal reflux disease (GERD). Infants suspected of GERD routinely undergo continuous reflux monitoring for 24 hours. If pain can be monitored continuously at the same time, it will be possible to analyze the detailed time relation between pain and reflux to improve diagnosis.

In recent years, pain analysis has become a research subject in computer vision [1][2][3][4]. Most efforts focus on video analysis of facial pain expression, because it is a prime indicator of pain for both adults and children [5]. For adults, promising results are obtained, e.g. for acute shoulder pain [2][3][4]. For infants, the potential of facial expression analysis was demonstrated by recent work on video analysis of discomfort [6] and acute pain [7].

For clinical applications such as for GERD diagnosis with infants, monitoring the face poses extra challenges, as discussed in Section 2. This paper considers the problem of face tracking in the presence of these challenges. We present an approach based on tracking 3D head pose and compare two 3D head tracking algorithms. The first algorithm uses intensities for appearance, as in related work. The new second algorithm uses dense-HOG features, resulting in more accurate short-term tracking.

Section 2 discusses the challenges and motivates our approach in relation to other work. Section 3 describes the algorithm and its variants, while experiments and results are presented in Section 4.

2. CHALLENGES, APPROACH AND RELATED WORK

For infant monitoring, face tracking poses several special challenges. Firstly, facial appearance of infants, and especially very young ones, has special characteristics. They have e.g. hardly any wrinkles or creases and no prominent eyebrows, but sometimes extra furrows around the eyes. Also, their eyes are not visible for long periods of time. Secondly, infants being monitored are not necessarily oriented towards a camera in the way adults are in many applications of facial analysis (e.g. in human-computer interaction or vehicle driver monitoring). As a result, tracking has to handle much larger pose variations than usual and, in general, one static camera may not be sufficient to keep the face in view all the time. For example, in case of monitoring for GERD, infants are in bed where they can freely turn their head and shift their body. Thirdly, in typical clinical settings, parts of the face may be permanently covered or temporarily occluded. For example, in case of monitoring for GERD, there are plasters on the face and a tube into the nose that connects to a pH probe in the esophagus. Also, infants may have a pacifier in their mouth to reduce stress. Since this works best with their familiar pacifier from home, it follows that tracking has to deal with a large variety of shapes and appearances in the region of the mouth. In addition, cuddles, toys or blankets may partially occlude the face.

Our overall approach for face tracking is based on modeling the face as a part of the 3D surface of the head and solving the somewhat more general problem of tracking the 3D head pose. Our main motivation is that this allows maximum use of image information for tracking robustness, because visible features of both face and non-face parts of the head can be used. This also facilitates extension for multiple cameras, in cases where pose variations are too large for a single static camera to keep the infant's face in view all the time. As a second motivation, 3D head tracking immediately informs us about head movements, which may serve as an extra parameter for pain monitoring (as in [4] which reports that acute shoulder pain of adults is detected better by also considering head movements).

Most face tracking approaches, for pain monitoring as well as other applications, are based on fitting a deformable 2D or 3D face model to input images. Such models (as e.g. in [1][2][3][6]) have to be trained to cover all variations of shape and appearance. In our case this is hardly feasible, because of wide pose ranges, unknown appearances of plasters, pacifiers etc. For this reason, we need an approach without a pre-determined model of appearance. Many of such approaches [8] recover head motion, using an assumed 3D head shape. Some recover motion from keypoints, e.g. [9]. For infant heads this is not feasible, as they have few and unstable points. Others recover motion by Lucas-Kanade template alignment [10], e.g. with a cylinder [11] or ellipsoid [12] head model and for multi-camera setups [13]. We adopt the same principle for our algorithm.

The intensity-based methods of [11][12][13] may have problems with less-pronounced texture and less-uniform illumination, as e.g.
in case of a sleeping infant in bed. This motivates our choice of densely sampled histogram-of-oriented-gradient (’dense-HOG’) features [14], which can improve Lucas-Kanade alignment [15].

Our main contribution is the dense-HOG-based algorithm and its evaluation for real-life infant monitoring conditions. The paper offers a comprehensive description of the algorithm for the single-camera case, with the major innovation singled out in Section 3.8 and with minor innovations in Sections 3.5 and 3.6. However, our aim is a multi-camera tracking system.

3. MODELING AND ALGORITHM

3.1. Full-perspective image projection

We represent 2D image locations as \( u = [u, v]^T \in \mathbb{R}^2 \), and 3D points in terms of homogeneous coordinates as \( x = [x, y, z]^T \in \mathbb{R}^3 \times \{1\} \). We assume a camera with arbitrary position and orientation in generic world coordinates. We model full-perspective image projection as a function \( W \) mapping 3D points \( x \) to image locations \( u \):

\[
W(x; C) = [u', v']^T/s', \quad \text{where} \quad [u', v', s']^T = Cx.
\]

(1)

Here, \( C \) is the \( 3 \times 4 \) camera projection matrix that combines the intrinsic and extrinsic characteristics of the camera.

For image projection of a surface, some parts may not be visible. Therefore, we assume a function \( w(x; C) \) that yields positive values for visible surface points and zeros for all other points. For surfaces of convex spatial volumes, the sign of \( w(x; C) \) follows directly from the surface normal at \( x \).

3.2. Rigid body transformation and 3D pose

The 3D head, as a rigid body, can move with 6 degrees of freedom. We model a move as a 3D transformation from the so-called Special Euclidean group \( SE(3) \). For minimal-dimensional parameterization, we associate a vector \( p = [\omega_x, \omega_y, \omega_z, t_x, t_y, t_z]^T \in \mathbb{R}^6 \) with this transformation. This relates to the twist representation [16] of rigid body motion combining rotation and translation, where the \( \omega \)-part is a vector in the direction of the rotation axis having a length equal to the rotation angle and the \( t \)-part is a translation vector. With the exponential map associated with \( SE(3) \), we can take \( p \) as an exponential coordinate to obtain the homogeneous matrix \( G \) in \( SE(3) \) by matrix exponentiation:

\[
G = e^p = I + \tilde{p} = \frac{\tilde{p}^2}{2!} + \frac{\tilde{p}^3}{3!} + \ldots,
\]

(2)

where \( I \) is the \( 4 \times 4 \) identity matrix and operator \( \tilde{\cdot} \) is defined by

\[
\tilde{p} = \begin{bmatrix}
0 & -\omega_z & \omega_y & t_z \\
\omega_z & 0 & -\omega_x & t_y \\
-\omega_y & \omega_x & 0 & t_z \\
0 & 0 & 0 & 0
\end{bmatrix}.
\]

In theory, we can use any of the above representations (vector \( p \), exponential \( e^p \), matrix \( G \)) for denoting 3D pose. In practice, we use the matrix representation for the net outcome of pose tracking, the vector form for computing small pose changes, and the in-between exponential form for our derivations here. This is because composing transformations is easily expressed for matrices (by multiplication), but difficultly for vectors. To see this, note that in general matrix multiplication does not commute, so that \( e^{p_1}e^{p_2} \neq e^{p_1+p_2} = e^{p_1+p_2} \).

Using the above and assuming a 3D reference head, we use \( p \) to define absolute head pose. Pose \( p \) then represents transformation \( e^p \) which transforms reference head points \( x \) to posed-head points \( e^p x \).

3.3. Cylinder head model and template alignment

For initialization we need a 3D reference head and an initial pose. Both typically result from fitting a parametric shape after face detection in the initial image. We present the algorithm for general shape, but employ a cylinder, as it is least sensitive to initial pose error [11].

In a tracking step, we use the known combination of appearance and pose of the head in a previous image to find the new combination in a new image. For this, we extract the known appearance as a template. Let \( p \) and \( a \) be the pose and appearance in the template image, respectively. Then \( p \) induces a one-to-one mapping between a finite set \( U(e^p; C) \) of 2D template pixels \( u \) and a finite set \( X(e^p; C) \) of 3D template points \( x \), such that \( u \leftrightarrow x \equiv [u \text{ is pixel location} \wedge u = W(e^p x; C) \wedge w(e^p x; C) > 0] \).

By extension, \( a \) induces a mapping from template points \( x \) to texture appearance values \( a_t(W(e^p x; C)) \). Our aim for tracking is to find a pose \( p \) that warps the template to align with the new image, i.e., to find \( p \) such that appearance \( a \) in the new image, at projected locations \( W(e^p x; C) \) of template points \( x \), is close to the template texture appearance. It is usual to define this as an alignment problem for minimum sum-of-squared-error cost:

\[
\arg\min_p \sum_{x \in \Omega(e^p; C)} \left\| a(W(e^p x; C)) - a_t(W(e^p x; C)) \right\|^2,
\]

(3)

where summation is performed only over those 3D template points in \( X(e^p; C) \) that are visible in pose \( p \), as defined by \( \Omega \):

\[
\Omega(e^p, e^p; C) = \{ x \in X(e^p; C) \mid w(e^p x; C) > 0 \}.
\]

For brevity, we will omit the \( C \) parameter of functions \( W, w \) and \( \Omega \) in the sequel, when parameter \( C \) remains unchanged.

3.4. Lucas-Kanade gradient descent

To solve the alignment problem, we use the Lucas-Kanade method of gradient descent optimization [10]. For this, we reformulate (3) in terms of a pose estimate \( p \) and a pose update \( \Delta p \) to yield the following update problem:

\[
\arg\min_{\Delta p} \sum_{x \in \Omega(e^p, e^p; C)} \left\| a(W(e^p x; C)) - a_t(W(e^p x; C)) \right\|^2.
\]

(4)

Starting from an initial estimate for \( p \), Lucas-Kanade repeatedly solves (4) for a \( \Delta p \) that is used to update \( p \).

For comparison, we have cast (4) in a form close to the standard Lucas-Kanade formulation in [17]. The difference is in function \( W \) and its arguments, and the \( \Omega \)-restriction on contributing points. Our motion model is a 3D→2D warp function \( \mathcal{W} \) with two parameters \( \Delta p \) and \( p \) defined by \( \mathcal{W}(x; \Delta p, p) = W(e^p e^\Delta p x) \). As a result, updating the warp is also different from [17]: we update \( p \) using \( e^{\Delta p} \triangleleft e^p e^\Delta p \).

Lucas-Kanade solves (4) by approximating for small \( \Delta p \) using a first-order Taylor expansion of \( a(W(e^p e^\Delta p x)) \) to yield

\[
\Delta p = H^{-1} \sum_{x \in \Omega(e^p, e^p)} \left[ \nabla a(W(e^p x)) \nabla \mathcal{W} \right]_{\Delta p} \left[ a(W(e^p x)) - a_t(W(e^p x)) \right],
\]

(5)

Here \( \nabla a \) is the gradient \( \frac{\partial a}{\partial u} \) \( \frac{\partial a}{\partial v} \) of the new image \( \mathcal{W} \), the Jacobian of the warp, and \( H \) the Gauss-Newton approximation of the Hessian:

\[
H = \sum_{x \in \Omega(e^p, e^p)} \left[ \nabla a \mathcal{W} \right]_{\Delta p} \left[ \nabla a \mathcal{W} \right]_{\Delta p}^T.
\]

(6)
The Jacobian $\frac{\partial y}{\partial \Delta p}$ can be computed, for given $x$ and $p$, using the following differential derived from (2) for $\Delta p = 0$ (as a twist [16]):

$$e^{\Delta p} = I + d\Delta p = \begin{bmatrix}
1 & -d\omega_y & d\omega_z & 0

-d\omega_y & 1 & -d\omega_z & 0

-d\omega_z & d\omega_y & 1 & 0

0 & 0 & 0 & 1
\end{bmatrix}.$$ 

### 3.5. Coordinate system adaptation

All of the above is valid for arbitrary world coordinates. In practice, when computing the Lucas-Kanade updates $\Delta p$, we want to ensure that the Hessian $H$ is well-conditioned for inversion. For this, we temporarily change the coordinate system to be centered in the head prior to solving (3). Using matrix $M$ to describe this change of coordinate system, we rewrite the terms with $\Omega$ and $a$ in (3) to yield

$$\text{argmin}_{\hat{p} \in \Omega}(M\hat{p}, M\hat{e}, CM^{-1}) \sum_{p \in \Omega} (a(W(Me\hat{p}x; CM^{-1})) - a(W(e\hat{p}x)))^2.$$ 

From this, we derive a new update problem to replace (4):

$$\text{argmin}_{\Delta p \in \Omega}(e\hat{p}Me\hat{p}x; CN) \sum_{p \in \Omega} (a(W(e\hat{p}Me\hat{p}x; CN)) - a(W(e\hat{p}x)))^2,$$

with $N = M^{-1}$ for the first iteration. In order to avoid adapting $e\hat{p}$ and re-changing the coordinate system for each iteration, we adapt $N$ instead, using $N \leftarrow NCe\hat{p}$, which can be proven to be equivalent. After the last iteration, we adapt $e\hat{p}$ once by $e\hat{p} \leftarrow NMe\hat{p}$. This coordinate adaptation removes the need for the regularization in [11].

### 3.6. Weighted alignment and IRLS

For clarity, we introduced a basic version of the alignment problem in (3) and update problem in (4), where all template points contribute equally. In practice, we use a weighted version in order to handle noise, non-rigid motion and occlusions. In the weighted version, point weights $w$ are used to adjust the influence of individual pixels in the alignment process. For the update problem in (4) this gives:

$$\text{argmin}_{\Delta p \in \Omega}(e\hat{p}Me\hat{p}x; CN) \sum_{p \in \Omega} w(e\hat{p}x) \| a(W(e\hat{p}x)) - a(W(e\hat{p}x)) \|^2.$$ 

Note that we introduced point function $w$ already in Section 3.1, using only its sign to indicate point visibility. We now use the full (non-negative) value of $w$ to represent the point weights. For the weighted version, the closed-form equivalents of (5) and (6) are

$$\Delta p = H^{-1} \sum_{x \in \Omega}(e\hat{p}x) \cdot \left[ a(W(e\hat{p}x)) - a(W(e\hat{p}x)) \right]$$

and

$$H = \sum_{x \in \Omega}(e\hat{p}x) \cdot \left[ a(W(e\hat{p}x)) - a(W(e\hat{p}x)) \right].$$

Partly similar to [11], we set $w$ as the product of a density term $w_D$ and a robustness term $w_R$:

$$w = w_D \cdot w_R.$$ 

The density term relates to the image projection of the head surface. For this, consider an infinitesimal area around a template point $x$ and its projection in the image plane. We use the area ratio of the latter divided by the former (which follows from basic 3D geometry) as our definition of $w_D(x)$. It varies with the direction of the surface normal at $x$ and the distance of $x$ to the image plane. As a result, template points seen from the side contribute less than template points seen from the front. Note that [11] uses a simpler ad-hoc function, which is less amenable for multi-camera setups.

For the robustness term, we use the IRLS technique of [11]. At each iteration, it adapts $w_R$ for the update problem in (7), based on the error resulting from the current pose estimate $p$:

$$w_R(e\hat{p}x) = e^{-\frac{\| a(W(e\hat{p}x)) - a(W(e\hat{p}x)) \|^2}{\sigma_R^2}},$$

where

$$\sigma_R = 1.4826 \cdot \text{median}_{x \in \Omega}(e\hat{p}x) \cdot \| a(W(e\hat{p}x)) - a(W(e\hat{p}x)) \|.$$ 

### 3.7. Template outlier removal and dynamic template updating

In the above, Section 3.3 still leaves a choice in what to extract as a template and when to use it.

The default choice for extraction is to consider the pose output for a given image and extract all pixels that correspond to the head in that pose. For robustness and especially for handling occlusions, we refine this as in [11]. We consider each candidate template pixel with its corresponding 3D point $x$ and compare its appearance $a_i(W(e\hat{p}x))$ with its counterpart appearance $a_{i-1}(W(e\hat{p}x))$ in the previous image (where $p_{i-1}$ is the head pose in the previous image). Using robust statistics on the resulting errors, we consider the pixel for 3D point $x$ as an outlier (and remove it from the template) if

$$\| a_i(W(e\hat{p}x)) - a_{i-1}(W(e\hat{p}x)) \| > c \sigma_{R_{i-1}},$$

where $c$ is a constant (set to 2.5), and where $\sigma_{R_{i-1}}$ is defined as in (9), while replacing $p$ in (9) by $p_{i-1}$.

The default choice for usage is in the next frame, i.e. we update the template continuously while tracking. In practice, we will need to correct for drift by also selectively re-using templates from earlier in the video sequence, e.g. as in the re-registration approach of [11]. Because our experiments below are aimed at comparing the effect of using intensity versus dense-HOG for appearance (which manifests itself as drift) we ignore drift correction in the context of this paper.

### 3.8. Intensity appearance vs. dense-HOG feature appearance

In the above we have used $a$ and the general term ‘appearance’ to denote pixel-based values from an image. We have also used norms $\| \ldots \|$ in our expressions for alignment error, without further specification. In this section we present two variants for appearance.

The first variant uses intensity for appearance. Intensity was used in methods comparable to ours [11][12][13]. In this variant, appearance is 1-dimensional and norm $\| \ldots \|$ is trivial ($= |\ldots|$).

The second variant uses dense-HOG features for appearance. Now, appearance is $N$-dimensional and we use the $L^2$-norm $\| \ldots \|_2$ in (3), (4), (7) and the $L^\infty$-norm $\| \ldots \|_\infty$ in (8), (9), (10). For our experiments, we employ a HOG variant with blocks of 2x2 cells, with cells of 8x8 pixels and 9-bin histograms for signed orientations. Also, we apply trilinear interpolation of location and orientation and $L^2$-Hys normalization [14]. As a result, our dense-HOG appearance has $N = 36$ dimensions.

### 4. EXPERIMENTS

Here we experiment with the dense-HOG-based tracking algorithm, and compare with the intensity-based variant of Section 3.8.

#### 4.1. Dataset and evaluation criteria

For our experiments, we used sequences from a database of videos [7] captured with handheld cameras at the Máximo Medical Center
Veldhoven, with parental consent. They show infants in bed, with various conditions: relaxed, experiencing acute pain during interventions, experiencing post-operative pain, etc. All videos show at least some of the challenging conditions discussed in Section 2.

For quantitative evaluations, we selected 10 videos of different infants with a lot of motion and, for some, also large changes of expression from acute pain. Since this paper concentrates on tracking accuracy in the absence of drift correction, we selected subsequences without extremely non-frontal poses. For this, we use the angle \( \theta(p) \) of rotation from pose \( p \) to an upright-frontal pose (with respect to the camera) as a measure of non-frontality and limit \( \theta < 50 \) degrees (note that one \( \theta \)-value can correspond with many rotation axes and therefore with many combinations of yaw, pitch and roll). Sequences (30 fps) were input as down-scaled gray-level signal with 480×270 pixels for Sequence 1 and 320×180 for others.

For qualitative judgment of our tracking as intermediate for pain analysis, we consider faces in normalized frontal view (NFV). For this, visible head texture is projected into an image that corresponds with a camera positioned upright-frontally before the head, with optical axis through the head cylinder center and same head-to-camera distance as in the first image of the sequence (cf. Figures 1 and 2).

For quantitative judgment, we consider center-of-eye locations in NFVs. As ground truth, we use center-of-eye annotations in input images that are reverse-projected as 3D points on the posed head cylinder and then NFV-projected. We compare NFV-projections of fixed 3D reference points on the cylinder (viz. the reverse-projected annotations of the first image of the sequence) with this ground truth. The comparison metric is the eye location error (ELE), defined for an NFV as the distance between same-eye locations divided by the inter-ocular distance (i.e. between left/right-eye ground-truths in the NFV of the first image). This metric allows to compare results for different sequences, as well as alternative ground-truth annotations.

Note that we annotated eye centers at full input resolution in each 10-th frame (68 sec). Re-annotation trials showed that annotation noise is low: mean ELE below 0.02, and outliers up to 0.08.

### 4.2. Tracking results

Our qualitative impression from experiments on many sequences is that, in general, faces are tracked well for poses with \( \theta < 50 \) degrees, but intensity-based tracking has larger drift and more risk of losing the face for highly non-frontal poses. This especially holds for low-contrast texture, such as in the absence of pacifier and hair, while eyes and mouth are closed. Figure 1 shows results on challenging frames with occlusions, pacifiers, moving shadows, etc.

#### Table 1. Tracking accuracy (intensity vs. dense HOG).

<table>
<thead>
<tr>
<th>nr. length (frames)</th>
<th>( \theta ) (degrees)</th>
<th>intensity( ELE^2 )</th>
<th>dense HOG( ELE^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>max</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>981</td>
<td>8.5</td>
<td>44.9</td>
</tr>
<tr>
<td>2</td>
<td>615</td>
<td>15.5</td>
<td>47.7</td>
</tr>
<tr>
<td>3</td>
<td>430</td>
<td>9.5</td>
<td>31.4</td>
</tr>
<tr>
<td>4</td>
<td>739</td>
<td>7.9</td>
<td>27.9</td>
</tr>
<tr>
<td>5</td>
<td>458</td>
<td>15.7</td>
<td>46.7</td>
</tr>
<tr>
<td>6</td>
<td>1274</td>
<td>2.2</td>
<td>43.6</td>
</tr>
<tr>
<td>7</td>
<td>1007</td>
<td>6.5</td>
<td>48.2</td>
</tr>
<tr>
<td>8</td>
<td>466</td>
<td>20.6</td>
<td>30.5</td>
</tr>
<tr>
<td>9</td>
<td>760</td>
<td>11.3</td>
<td>30.2</td>
</tr>
<tr>
<td>10</td>
<td>480</td>
<td>20.3</td>
<td>33.6</td>
</tr>
</tbody>
</table>

1. Values for \( \theta \) are from the pose outputs of dense-HOG-based tracking.
2. Mean/max ELE are over all left and right eyes with ground truth.
3. Sequence 5 has 20 frames where intensity-based output has 1 invisible eye. Stated mean/max ELE are over remaining frames with ground truth.
4. Sequence 6 has 430 frames where intensity-based output has 1 invisible eye.

As a result, there are no useful ELE statistics for comparison.

#### Table 1. Tracking accuracy (intensity vs. dense HOG).

<table>
<thead>
<tr>
<th>nr. length (frames)</th>
<th>( \theta ) (degrees)</th>
<th>intensity( ELE^2 )</th>
<th>dense HOG( ELE^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>max</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>981</td>
<td>8.5</td>
<td>44.9</td>
</tr>
<tr>
<td>2</td>
<td>615</td>
<td>15.5</td>
<td>47.7</td>
</tr>
<tr>
<td>3</td>
<td>430</td>
<td>9.5</td>
<td>31.4</td>
</tr>
<tr>
<td>4</td>
<td>739</td>
<td>7.9</td>
<td>27.9</td>
</tr>
<tr>
<td>5</td>
<td>458</td>
<td>15.7</td>
<td>46.7</td>
</tr>
<tr>
<td>6</td>
<td>1274</td>
<td>2.2</td>
<td>43.6</td>
</tr>
<tr>
<td>7</td>
<td>1007</td>
<td>6.5</td>
<td>48.2</td>
</tr>
<tr>
<td>8</td>
<td>466</td>
<td>20.6</td>
<td>30.5</td>
</tr>
<tr>
<td>9</td>
<td>760</td>
<td>11.3</td>
<td>30.2</td>
</tr>
<tr>
<td>10</td>
<td>480</td>
<td>20.3</td>
<td>33.6</td>
</tr>
</tbody>
</table>

1. Values for \( \theta \) are from the pose outputs of dense-HOG-based tracking.
2. Mean/max ELE are over all left and right eyes with ground truth.
3. Sequence 5 has 20 frames where intensity-based output has 1 invisible eye. Stated mean/max ELE are over remaining frames with ground truth.
4. Sequence 6 has 430 frames where intensity-based output has 1 invisible eye.

As a result, there are no useful ELE statistics for comparison.

#### 5. CONCLUSIONS

We presented a new algorithm for face tracking under challenging conditions, using a cylinder head model and 3D pose tracking by alignment of dynamically extracted templates based on dense-HOG appearance. We evaluated the algorithm on single-camera videos of moving infants in bed, relaxed or in pain. Tracking was compared with a variant using intensity appearance. Experiments show good short-term tracking for poses up to 50 degrees from upright-frontal, with significantly better accuracy for dense-HOG-based tracking. The algorithm can be extended for multi-camera setups, for use as part of an infant pain monitoring system in a clinical context.
6. REFERENCES


