Biologically-Inspired Motion Encoding for Robust Global Motion Estimation

Evangelos Sariyanidi, Hatice Gunes, and Andrea Cavallaro

***Abstract*—The growing use of cameras embedded in autonomous robotic platforms and worn by people is increasing the importance of accurate global motion estimation (GME). However, existing methods may degrade considerably under illumination variations. In this paper, we address this problem by proposing a biologically-inspired GME method that achieves high estimation accuracy in the presence of illumination variations. We mimic the early layers of the human visual cortex with the spatio-temporal Gabor motion energy by adopting the pioneering model of Adelson and Bergen and we provide the closed-form expressions that enable the study and adaptation of this model to different application needs. Moreover, we propose a normalisation scheme for motion energy to tackle temporal illumination variations. Finally, we provide an overall GME scheme which, to the best of our knowledge, achieves the highest accuracy on the Pose, Illumination, and Expression (PIE) database.**

***Index Terms*—bio-inspired motion encoding, illumination normalisation, global motion estimation**

I. Introduction

GLOBAL motion estimation (GME) is an increasingly important task because of the growing use of cameras embedded in autonomous platforms or worn by people. Estimating accurately the rigid motion between images is desirable for measuring egomotion for mobile robotics [1], [2], [3] and wearable cameras [4], image stabilisation [5], [6], [7], object segmentation [8] and motion magnification [9].

The main challenges for GME [10] are outliers, illumination variations and textureless regions. *Outliers* are produced by local motions or lens distortions that cannot be represented with rigid transformations. Lens distortions are particularly prominent in wide-angle cameras. *Illumination variations* change pixel intensities and therefore the apparent motion in the images. Finally, *textureless* regions provide limited or noisy information about the real motion [11]. Existing GME methods cannot deal with all these challenges concurrently while also achieving high accuracy, as we will discuss in Section II.

Motion perception, an active research field in biology, aims at understanding how mammals perceive the speed and orientation of visual elements in dynamic scenes. The human visual cortex has a very efficient motion perception ability that discerns subtle motions [12] while also being adaptive to changes in lighting conditions [13]. The pioneering work of Adelson and Bergen [14] showed that the early layers of the visual cortex perceive motion through complex cells that behave like speed- and orientation-sensitive Gabor filters.

In this paper, we encode motion by emulating the behaviour of the lower layers of the visual cortex [13] (see Fig. 1). We provide closed-form expressions that can be used to study the Adelson and Bergen's model [14] and we also propose an illumination normalisation scheme that renders the output of Gabor filtering robust against the brightness value of dynamic elements as well as temporal illumination variations. Furthermore, we propose an overall GME scheme, which, to the best of our knowledge, is the first biologically-inspired GME approach that is validated on real sequences with challenging illumination variations. We show that this scheme achieves high accuracy even in the presence of challenging illumination variations, outperforming state-of-the-art GME methods. The major contributions of this paper are summarized below.

- We develop the closed-form mathematical expressions that can be used to study the motion perception model of Adelson and Bergen for 2D motion, as the original model is developed for 1D motion only. Specifically, we provide the formulation of *Gabor motion energy* for a *moving line* and show how to tune a spatio-temporal Gabor filter pair to a specific type of motion.
- We propose an illumination normalisation scheme that reduces the sensitivity of Gabor motion energy to temporal illumination variations.
- We show that a statistical approach can efficiently model the non-linear relationship between local Gabor features and the corresponding local motion vectors.

This paper is organized as follows. Section II discusses existing work. Section III summarises the proposed GME scheme. In Section IV we discuss how Gabor motion energy encodes motion and we provide the closed form expressions of Gabor motion energy. Section V describes the proposed illumination normalisation scheme, and Section VI describes the statistical modelling we employ while producing local motion vector estimates from motion energy. Section VII provides the experimental validation of our work, and finally Section VIII concludes the paper and discusses future work.

II. Related Work

GME approaches can be categorised in five groups, namely keypoint matching, estimation from a coarse motion vector (MV) field, estimation from dense MV field, global transformation and direct minimisation/maximisation.

Keypoint matching methods estimate motion using a number of sparsely located image points that are centred on visually salient regions with rich texture [15]. These methods prevailed mainly due to their tolerance to large outlier motions,

E. Sariyanidi and A. Cavallaro are with the Centre for Intelligent Sensing in Queen Mary University of London, UK. H. Gunes is with the Computer Laboratory, University of Cambridge, UK.
E-mail: {e.sariyanidi, a.cavallaro}@qmul.ac.uk, hatice.gunes@cl.cam.ac.uk

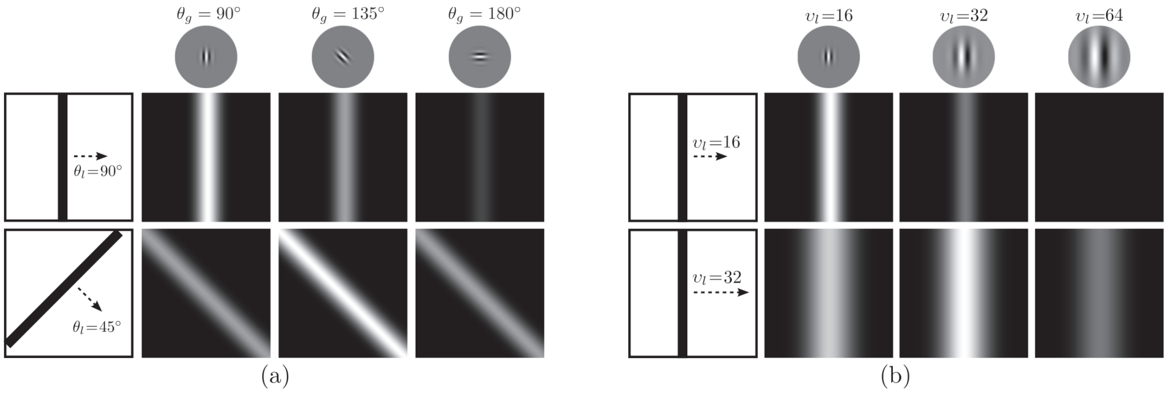

Fig. 1. Two exemplar cases that illustrate how Gabor motion energies computed from multiple pairs of spatio-temporal Gabor filters enable the identification of motion speed and orientation. (a) Two lines that are moving with the same speed but in different orientations (90° and 45°): The maximal energy for each line is produced with the filter pair that is tuned to the lines' orientation. (b) Two lines that are moving in towards the same direction but with different speeds (16 and 32): The maximal motion energy is produced with the filter that is tuned to the lines' speed.

which is achieved by robust estimators such as RANSAC [16]. Keypoint matching methods tend to fail when regions of (outlier) local motions contain rich texture while regions of global motion are relatively flat, and with illumination variations that severely reduce the number of matched features [6].

Methods based on a coarse MV field divide the input space into non-overlapping blocks (*e.g.* 4×4, 8×8) and compute an MV for each block. Then, global motion can be estimated from the motion model that best approximates the set of given MVs. The influence of outliers can be reduced by discarding outliers based on an error histogram, or by employing robust estimators such as RANSAC or M-estimator [23], [21]. In the presence of a foreground object, robustness can be increased further by first detecting and then excluding foreground motion from GME [37]. Methods based on a coarse MV field are typically employed for video coding where computational cost is critical [38]. For this reason, MVs are computed with simple methods (*e.g.* block matching [38]). However, such methods are sensitive to illumination variations (see Section VII). Moreover, the performance of GME based on a coarse MV field can degrade if the input images contain low-texture areas.

The techniques discussed in the rest of this section are expected to be more resilient to lack of texture as they use information from all the pixels during GME. Methods based on a dense MV field compute an MV for each pixel rather than a block and then compute global motion with a (robust) estimator. A dense MV field can be computed with optical flow techniques, which can achieve higher accuracy than coarse MV computation techniques. The optical flow formulation of Horn and Schunck [24], which computes MVs by minimising an energy function with global constraints to tackle the aperture problem, was extended to improve robustness against illumination variations by relaxing these constraints [25] and/or employing robust estimators [26]. One of the best-performing optical flow estimators [39] is the 2010 winner of Middlebury evaluation [27], which is an extension of the Horn and Schunck formulation with a non-linear penalty function and median filtering for intermediate flow fields. However, this method provides a limited accuracy for GME with challenging illumination variations (see Section VII).

Methods based on global transformation exploit the properties of Fourier [29], [30], Fourier-Mellin [32] or Radon transformations [31], [15]. These methods cannot model generic motions such as perspective motions with 8 degrees of freedom (DoF). In fact, the Fourier transformation can model at most Euclidean motions (4 DoF [29], [30]), and the Radon transformation can model affine motions with 5 DoF [31] or 6 DoF [15]. Furthermore, methods based on global transformation are sensitive to outlier local motions and illumination variations [29]. Although a robust version of fast Fourier transform (FFT) [29] proves successful against these challenges, its accuracy in simpler conditions without illumination variations can be lower than the accuracy of feature-based methods [6].

A typical approach based on *direct minimisation* is the Lucas-Kanade (LK) method [34] which estimates the motion between two frames by optimising an energy function, such as the sum of squared difference between frames. The optimisation algorithm (*e.g.* gradient descent) relies on local function approximation theories (*e.g.* Taylor expansion [34]). These theories are typically built on a smoothness assumption (*i.e.* once/twice differentiability) [34], which may be violated in the presence of sudden illumination variations, shadows or partial occlusions [26]. LK-based approaches tolerate outlier motions, as the contribution of each local image patch to the optimised energy function is independent from the rest of the image. However, if outlier local motions occur in a large portion of the image, their contribution to the energy function may become significant and undermine the accuracy of GME [40]. Numerous extensions of LK have been proposed, which differ in the energy function that is optimised, the optimisation algorithm that is employed or the domain where the optimisation is performed [34], [36], [41], [35], [42], [43]. The LK methods that operate on the pixel domain are particularly sensitive to illumination variations. Pre-processing with Gabor filters [35] improves robustness of LK methods against illumination variations, but only to a degree [36]. Another similar method, which is one of the most robust methods against illumination variations, is based on the direct maximisation of gradient correlation coefficient (MGCC) [36].

TABLE I

REPRESENTATIVE WORKS FROM DIFFERENT GME PARADIGMS. †DENSE OPTICAL FLOW ESTIMATION METHODS RATHER THAN GME METHODS, HOWEVER THEY CAN BE USED FOR GME AFTER ELIMINATING OUTLIER MOTION VECTORS. *THE TOLERANCE OF THESE METHODS TO LOW TEXTURE MAY VARY BASED ON THE DENSITY OF THE MV FIELD.

	Reference	Approach to Motion Estimation	Texture	Tolerance to Outliers	Tolerance to Illumination Variations	Motion Model	Tested Against Illumination?
Feature-based	Lowe '04 [17]	SIFT feature matching	Strong	RANSAC	—	8 DoF	—
	Bay '06 [18]	SURF feature matching	Strong	RANSAC	—	8 DoF	—
	Okade '14 [19]	MSER feature matching	Strong	RANSAC	—	8 DoF	—
	Ryu '12 [20]	KLT feature matching	Strong	RANSAC	—	8 DoF	—
Coarse MV-based	Smolić '00 [21]	M-Estimator on MVs	Strong*	M-Estimator	—	8 DoF	—
	Chen '10 [22]	Least squares on MVs	Strong*	Cascaded outlier elim.	—	8 DoF	—
	Su '05 [23]	Least squares on MVs	Strong*	Histogram-based elim.	—	8 DoF	—
Dense MV-based	Horn† '81 [24]	Error min. with global constr.	Mild	Outlier elim.†	—	8 DoF	—
	Gennert† '87 [25]	Error min. with global constr.	Mild	Outlier elim.†	Relaxed constr.	8 DoF	Synthetic
	Kim '05 [26]	Error min. with global constr.	Mild	Outlier elim.†	Relaxed constr.&M-estim.	8 DoF	Synthetic
	Sun '10 [27]	Error min. with global constr.	Mild	Outlier elim.†	Texture/structure dec.	8 DoF	—
Global Transform	Tzimirop. '11[28]	Fourier transform	Mild	—	Gradient correlation	2 DoF	—
	Tzimirop. '10[29]	Fourier transform (log-polar)	Mild	Cosine kernel	Gradient correlation	4 DoF	—
	Pan '09 [30]	Multi-layer Fourier transform	Mild	—	—	4 DoF	—
	Traver '08 [31]	Radon transform	Mild	—	—	5 DoF	—
	Xiong '14 [15]	Radon transform	Mild	—	—	6 DoF	—
	Kumar '11 [32]	Fourier transform (block-based)	Mild	RANSAC	—	6 DoF	—
Direct minimisation/ maximisation	Dufaux '00 [33]	SSD error min.	Mild	Histogram-based elim.	—	6 DoF	—
	Baker '04 [34]	Lucas-Kanade error min.	Mild	Local op.	—	6 DoF	—
	Ashraf '10 [35]	Lucas-Kanade error min.	Mild	Local op.	(2D) Gabor filtering	6 DoF	Real (N/A)
	Tzimirop. '11[36]	Gradient correlation max.	Mild	Cosine kernel	Gradient correlation	6 DoF	Real (182 pairs)
Dense MV-based	This work	Bio-inspired motion encoding, statistical MV estimation	Mild	RANSAC	(3D) Gabor filtering, Illumination Norm.	8 DoF	Real (400 pairs)

MGCC employs a cosine kernel, which improves robustness against outliers and illumination by eliminating local mismatches. However, this approach removes some information, most notably the gradient magnitude, which may cause MGCC to underperform in simpler conditions illumination variations. The methods discussed in this section are summarized in Table I and compared with the proposed method.

III. Overview of the Proposed Approach

Fig. 2 illustrates our overall GME scheme on an exemplar pair of images collected with a wearable (head-mounted) camera, where the aim is to estimate the global motion due to camera (*i.e.* head) movement. Ours is a dense and local method, as it computes local motion vectors for each pixel, and has three principal processing layers: low-level motion encoding, local motion estimation and global motion estimation. *Low-level motion encoding* emulates the behaviour of the human visual system by employing spatio-temporal Gabor filter pairs to compute *motion energy*, which provides implicit information about the speed and orientation of each pixel in the input image pair. Gabor filters provide some tolerance against salt-and-pepper and Gaussian (*e.g.* white) noise [44]. Also, they are partly robust to illumination variations as they are localised in space, and we improve their robustness further by applying illumination normalisation over time.

In *local motion estimation* we model local motion *statistically* with a function that takes as input the local motion energy within a $P \times P$-sized region around a pixel, and outputs a local motion vector estimation for the pixel — we produce a local motion vector for each pixel by applying this function to all the pixels. Specifically, we use a single-hidden-layer neural network, which is one of the best established techniques used for (non-linear) statistical modelling [45]. A statistical approach has the advantage of modelling this relationship using existing data that link local features to motion vectors in a bottom-up fashion, instead of making assumptions on the relationship between features and motion vectors.

In the third layer, *global motion estimation*, we employ RANSAC to eliminate outlier motion vectors that may be caused by local motion estimation errors, by occlusions or by motions that are not congruent with the global motion. We finally estimate global motion from inlier motion vectors as a projective transformation with 8 DoF.

IV. Biologically-inspired Motion Encoding

Motion perception at the lower layers of the visual cortex is typically analysed based on the response of a visual cell to a moving line [14]. In this section, we first obtain the closed-form expression of Gabor motion energy for a moving line, and then show how to tune a Gabor filter to a particular speed and direction.

A. Motion Energy for a Moving Line

Let $\mathbf{I} \triangleq \mathbf{I}(x, y, t)$ denote a sequence of a moving line as:

$$\mathbf{I}(x, y, t) \triangleq c\delta\left(x \cos\theta_l - y \sin\theta_l - tv_l\right), \qquad (1)$$

where δ is Dirac's delta, x, y, t denote spatial coordinates and time; θ_l defines the orientation of $\mathbf{I}$; $v_l \geqslant 0$ defines the speed and $c > 0$ is the luminance value of $\mathbf{I}$. A $3D$ Gabor filter can be represented as [46]:

$$g(x, y, t) \triangleq \frac{\gamma}{2\pi\sqrt{2\pi\sigma^2\tau}} \cos\left(\frac{2\pi}{\lambda}(\bar{x} + v_g t + \alpha)\right) e^{-\frac{\bar{x}^2+\gamma\bar{y}^2}{2\sigma^2} - \frac{t^2}{2\tau^2}}, \qquad (2)$$

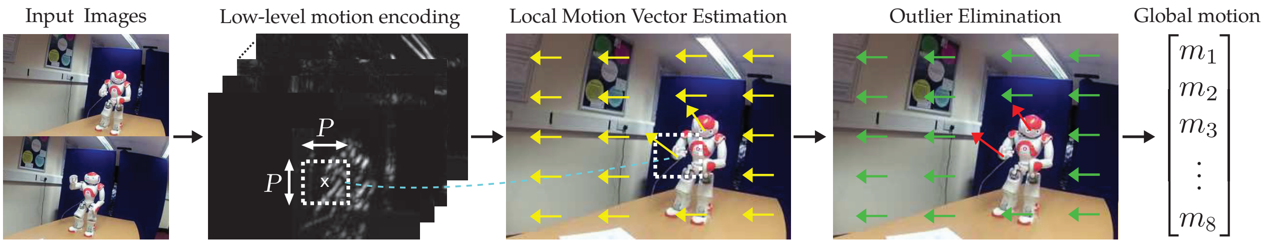

Fig. 2. Illustration of our GME approach. We estimate local motion vectors across the image, then, we eliminate outlier motion vectors (shown in red) through RANSAC, and finally estimate global motion from inlier vectors (shown in green). While estimating the motion vector around a pixel, we use the energy values across a $P \times P$-sized region centred on the pixel. We compute the energy with multiple filter pairs, each tuned to a different orientation.

where $\bar{x} = x\cos(\theta_g) + y\sin(\theta_g)$ and $\bar{y} = -x\sin(\theta_g) + y\cos(\theta_g)$. To avoid cluttering, we define the following parameters as $\gamma = 1, \lambda = 2\pi, \tau = 1/\sqrt{2}$ and $\sigma = 1/\sqrt{2}$ (see [46] for a detailed discussion on these parameters). The parameters θ_g and $v_g \geqslant 0$ define the orientation and speed of motion that the filter is tuned for, and α is the phase offset which can be set to $\alpha = 0$ to obtain an even-phased (cosine) filter, g^e, and $\alpha = \frac{\pi}{2}$ to obtain an odd-phased (sine) filter, g^o. The two filters form a quadrature pair (g^e, g^o).

Adelson and Bergen define the *motion energy* for a sequence $\mathbf{I}$ through a quadrature filter pair as [14]:

$$E_{\mathbf{I}}(x, y, t) \triangleq (\mathbf{I} * g^e)^2 + (\mathbf{I} * g^o)^2. \tag{3}$$

In one (spatial) dimension, the energy gets maximal when the speed of the line is equal to the speed of the filter [14]. Energy gets monotonically smaller as the speed of the filter becomes larger or smaller than the speed of the line. This well-defined relationship between speed and magnitude of energy enables the identification of the speed of the line.

For the two-dimensional line, energy depends not only on line and filter speeds v_l, v_g but also orientations (or directions) θ_l and θ_g. To interpret energy correctly, we must know how to tune the filter parameters θ_g and v_g so as to yield maximal energy $E_{\mathbf{I}}$ for a given line $\mathbf{I}$. In the remainder of this section, we first obtain the closed-form expression of $E_{\mathbf{I}}$, and then show how $E_{\mathbf{I}}$ can be maximised.

To compute $E_{\mathbf{I}}$, we must compute the convolutions $\mathbf{I}*g^e$ and $\mathbf{I}*g^o$. The convolution $\mathbf{I}*g^e$ requires the computation a triple integral that can be challenging even for a computer algebra system. Therefore we make use of the Convolution Theorem, which states that, under suitable conditions, the convolution of two functions is equivalent to the pointwise product of their Fourier transforms, *i.e.* $\mathcal{F}\{\mathbf{I} * g\} = \mathcal{F}\{\mathbf{I}\}\mathcal{F}\{g\}$. The Fourier transforms of $\mathbf{I}$, g^e are denoted respectively with $\hat{\mathbf{I}}$, $\hat{g}^e$ and are computed using Mathematica as[1]:

$$\hat{\mathbf{I}}(\xi_1, \xi_2, \xi_3) = c\frac{\sqrt{2\pi}}{\cos\theta_l}\,\delta\left(\xi_1 v_l \sec\theta_l + \xi_3\right) \\ \delta(\xi_1 \tan\theta_l + \xi_2), \tag{4}$$

$$\hat{g^e}(\xi_1, \xi_2, \xi_3) = \frac{1}{4\sqrt{2}\pi\sqrt{\pi}}\left(1 + e^{\xi_3 v_g + \xi_1\cos\theta_g + \xi_2\sin\theta_g}\right) \\ e^{-\frac{2\xi_2\sin\theta_g + 1 + \xi_1^2 + \xi_2^2 + (\xi_3 + v_g)^2 + 2\xi_1\cos\theta_g}{4}}. \tag{5}$$

$\mathbf{I}*g^e$ is obtained with an inverse transform, $\mathbf{I}*g^e = \mathcal{F}^{-1}\{\hat{\mathbf{I}}\hat{g^e}\}$:

$$\mathbf{I} * g^e = \frac{c\,\mathrm{sgn}(\sec\theta_l)}{2\sqrt{2}\pi^2\sqrt{1+v_l^2}} \\ \cos\frac{(v_g v_l - \cos\theta_{gl})(t v_l - x\cos\theta_l + y\sin\theta_l)}{1+v_l^2} \\ e^{-\frac{v_g v_l\cos\theta_{gl} + 4tyv_l\sin\theta_l - 4x\cos\theta_l(tv_l + \sin\theta_l)}{2(1+v_l^2)}} \\ e^{-\frac{1+4x^2+4y^2+2v_g^2+(2+8t^2)v_l^2-\cos 2\theta_{gl}+4(x^2-y^2)\cos 2\theta_l}{8(1+v_l^2)}}, \tag{6}$$

where $\theta_{gl} \triangleq \theta_g + \theta_l$. The convolution with the odd-phased filter, $\mathbf{I} * g^o$, produces a similar output — the only difference is that the first cos function is replaced with $-\sin$. Finally, using $\mathbf{I} * g^e$ and $\mathbf{I} * g^o$, we can compute the energy for the moving line $E_{\mathbf{I}} = (\mathbf{I} * g^e)^2 + (\mathbf{I} * g^o)^2$ as:

$$E_{\mathbf{I}} = \frac{\bar{c}^2}{1+v_l^2} e^{-\frac{v_g v_l\cos\theta_{gl} + 4tyv_l\sin\theta_l - 4x\cos\theta_l(tv_l + y\sin\theta_l)}{1+v_l^2}} \\ e^{-\frac{1+4x^2+4y^2+2v_g^2+(2+8t^2)v_l^2-\cos 2\theta_{gl}+4(x^2-y^2)\cos 2\theta_l}{4(1+v_l^2)}}, \tag{7}$$

where $\bar{c} \triangleq \frac{c}{2\sqrt{2}\pi^2}$. An interactive plot that shows how $E_{\mathbf{I}}$ varies with filter parameters θ_g, v_g and line parameters θ_l, v_l is provided as supplementary material[1].

B. Tuning a Gabor Filter Pair

In order to tune a Gabor filter pair to a particular speed v_l and spatial orientation θ_l, we should find the v_g and θ_g values that maximise $E_{\mathbf{I}}$. To this end, we first find all extrema of $E_{\mathbf{I}}$, and then find which of these are the maxima.

To find extrema, we compute the first-order partial derivatives of $E_{\mathbf{I}}$ with respect to v_g and θ_g:

$$\frac{\partial E_{\mathbf{I}}}{\partial v_g} = -\frac{1}{1+v_l^2}(v_g + v_l\cos\theta_{gl})E_{\mathbf{I}}, \tag{8}$$

$$\frac{\partial E_{\mathbf{I}}}{\partial \theta_g} = -\frac{1}{1+v_l^2}(\cos\theta_{gl} - v_g v_l)\sin\theta_{gl}E_{\mathbf{I}}. \tag{9}$$

The solutions that make both partial derivatives zero can be considered as four sets, S_1, S_2, S_3, S_4, that are defined as:

$$S_1 \triangleq \{(v_g, \theta_g) : (0, -\pi/2 - \theta_l + 2\pi\,k), k \in \mathbb{Z}\}, \tag{10}$$

$$S_2 \triangleq \{(v_g, \theta_g) : (0, \pi/2 - \theta_l + 2\pi\,k), k \in \mathbb{Z}\}, \tag{11}$$

$$S_3 \triangleq \{(v_g, \theta_g) : (v_l, \pi - \theta_l + 2\pi\,k), k \in \mathbb{Z}\}, \tag{12}$$

$$S_4 \triangleq \{(v_g, \theta_g) : (-v_l, -\theta_l + 2\pi\,k), k \in \mathbb{Z}\}. \tag{13}$$

[1] The Mathematica files that are used to obtain the expressions throughout this section are on http://www.eecs.qmul.ac.uk/~es302/gme

We eliminate S_4 as we assume $v_l, v_g \geqslant 0$, so the only solution to satisfy S_4 is $v_l = v_g = 0$, which implies that the line is not moving. To determine whether there is a maximum among the remaining solutions, S_1, S_2, S_3, we use the second derivative test. The second partial derivatives of $E_\mathbf{I}$ are:

$$\begin{aligned}\frac{\partial E_\mathbf{I}^2}{\partial^2 v_g} =& \big[v_l \cos\theta_{gl}(2v_g + v_l\cos\theta_{gl}) \\ & - 1 + v_g^2 - v_l^2\big]\frac{E_\mathbf{I}}{(1+v_l^2)^2}, \quad (14)\\ \frac{\partial E_\mathbf{I}^2}{\partial^2 \theta_g} =& \big[(1+v_l^2)(v_g v_l\cos\theta_{gl} - \cos 2\theta_g l) \\ & + (\cos\theta_{gl} - v_g v_l)^2 \sin^2\theta_{gl}\big]\frac{E_\mathbf{I}}{(1+v_l^2)^2}, \quad (15)\\ \frac{\partial E_\mathbf{I}^2}{\partial\theta_g \partial v_g} =& \big[v_l(3 - 2v_g^2 + 2v_l^2 + \cos 2\theta_{gl}) \\ & - 2v_g\cos\theta_{gl}(v_l^2 - 1)\big]\frac{E_\mathbf{I}\sin\theta_{gl}}{2(1+v_l^2)^2}. \quad (16)\end{aligned}$$

To perform the second partial derivative test, we construct the Hessian matrix H and compute its determinant as a function $D(v_g, \theta_g)$ as follows:

$$H = \begin{bmatrix} \frac{\partial E_\mathbf{I}^2}{\partial^2 v_g} & \frac{\partial E_\mathbf{I}^2}{\partial v_g \partial\theta_g} \\ \frac{\partial E_\mathbf{I}^2}{\partial\theta_g \partial v_g} & \frac{\partial E_\mathbf{I}^2}{\partial^2 \theta_g} \end{bmatrix}, \quad (17)$$

$$D(v_g, \theta_g) \triangleq \det(H) = \frac{\partial E_\mathbf{I}^2}{\partial^2 v_g}\frac{\partial E_\mathbf{I}^2}{\partial^2 \theta_g} - \left(\frac{\partial E_\mathbf{I}^2}{\partial\theta_g \partial v_g}\right)^2. \quad (18)$$

To determine whether the solutions S_1, S_2 or S_3 are extrema, we denote the determinants of those solutions respectively as $D_{S_1}, D_{S_2}, D_{S_3}$ and compute them as[1]:

$$\begin{aligned}D_{S_1} = D_{S_2} =& -Ke^{\frac{2(y^2-x^2)\cos 2\theta_l - 8tyv_l\sin\theta_l + 4xy\sin 2\theta_l}{1+v_l^2}} \\ & e^{-\frac{8txv_l\cos\theta_l - 1 - 2x^2 - 2y^2 - v_l^2 - 4t^2v_l^2}{1+v_l^2}}, \quad (19)\\ D_{S_3} =& Ke^{-\frac{4}{1+v_l^2}(tv_l - x\cos\theta_l + y\sin\theta_l)^2}, \quad (20)\end{aligned}$$

where $K = \frac{c^2}{(2\sqrt{2}\pi^2)^4(1+v_l^2)^3} > 0$. Since the outcome of the exp function is always positive, D_{S_1}, D_{S_2} are always negative; therefore, S_1, S_2 contain saddle points and not extrema. On the other hand, S_3 contains extrema as $D_{S_3} > 0$. To check whether S_3 contains maxima or minima, we check the partial derivative $\frac{\partial E_\mathbf{I}^2}{\partial^2 v_g}$ for the solutions of S_3:

$$\left.\frac{\partial E_\mathbf{I}^2}{\partial^2 v_g}\right|_{(v_g,\theta_g)\in S_3} = \frac{-c\,e^{-\frac{2(y\sin\theta_l + tv_l - x\cos\theta_l)^2}{1+v_l^2}}}{8\pi^4(1+v_l^2)^2}. \quad (21)$$

This expression is always negative, therefore $(v_g, \theta_g) \in S_3$ are maxima. In conclusion, to tune the filters g^e, g^o to a line moving with spatial orientation θ_l and speed v_l, the filter parameters v_g and θ_g must be defined as follows:

$$\begin{aligned} v_g &= v_l, & (22)\\ \theta_g &= \pi - \theta_l + 2\pi k. & (23)\end{aligned}$$

Once we know how to tune one filter, we can obtain a complete motion representation by computing multiple energy functions, each involving a different filter pair tuned to a different speed and orientation [14]. Such a representation enables the identification of the speed and direction of an *unknown* line: the Gabor filters that are tuned to a motion similar to that of the unknown line would produce a higher energy than other filters (Fig. 1). In Fig. 1a we show how the motion energy computed from multiple Gabor filters enables us to identify the orientations of two different lines (one at each row) that move with the same speed. The line at the top of Fig. 1a moves with an orientation of $\frac{\pi}{2}$ and the maximal energy is produced with the spatio-temporal Gabor filter that is tuned to the speed of the line, that is, according to (23), $\theta_g = \frac{\pi}{2}$. Similarly, the maximal energy for the line at the bottom of Fig. 1a is produced by the filter with $\theta_g = \frac{3\pi}{4}$ as this is the filter that is tuned to the orientation of the line (*i.e.* $\frac{\pi}{4}$). A similar discussion applies for the examples in Fig. 1b, where we show how energy can be used to discover the speed of two lines: The maximal motion energies are produced by the filters that are tuned through (22) to the speed of each line.

V. Illumination Normalisation

Motion energy is sensitive not only to the brightness of moving elements but also to temporal illumination variations. In this section we propose a normalisation scheme to reduce illumination sensitivity. We show how this scheme eliminates the dependence on the initial brightness of a moving line and we extend it to tackle temporal illumination variations for generic sequences.

A. Normalisation of Line Brightness

As it can be seen in (7), the motion energy of a moving line is sensitive to illumination conditions even if there are no temporal illumination variations, due to the illumination-dependent constant c^2. This section aims at obtaining a *normalised sequence* $\tilde{\mathbf{I}} \triangleq \tilde{\mathbf{I}}(x, y, t)$ such that the energy of this sequence, $E_{\tilde{\mathbf{I}}}$, is illumination-independent and yet its functional form is still equal to that of $E_\mathbf{I}$. Such a sequence can be obtained by dividing the frames of $\mathbf{I}$ with a coefficient that is proportional to c^2; this is akin to the *contrast normalisation* that is arguably employed by the mammal visual cortex [47], [13]. We now show how such a coefficient can be obtained.

Assume that we have a sequence of a *static* line whose luminance value is c. Then, according to (7), the energy of the static line will be proportional to c^2 and, because the line is not moving, the energy will be constant over time. The energy of this static line provides us with the coefficient that we need for normalisation: A coefficient that is constant over time and proportional to c^2.

In fact, we *do* have a way of obtaining such a sequence: We can take a frame from $\mathbf{I}$ at any time t_k, and obtain a static sequence, $\mathbf{I}^{t_k} \in \mathbb{R}^3$, by replicating this frame over time. We illustrate this for the exemplar horizontally moving line in Fig. 3a by creating two static sequences, $\mathbf{I}^{t_0}, \mathbf{I}^{t_1}$ (see Fig. 3b,c). Such static sequences obtained from a sequence $\mathbf{I}$ can be defined as $\mathbf{I}^{t_k}(x, y, t) \triangleq \mathbf{I}(x, y, t_k)$.

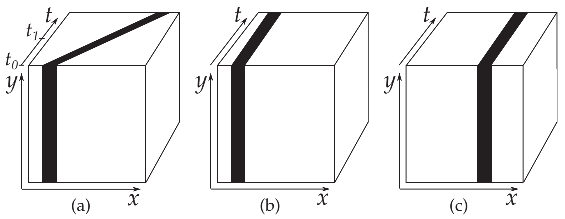

Fig. 3. Illustration of how we create static sequences. (a) $\mathbf{I}$: sequence of a line that moves horizontally. (b) $\mathbf{I}^{t_0}$: static sequence created from $\mathbf{I}$ using the frame at time t_0; (c) $\mathbf{I}^{t_1}$: static sequence created from $\mathbf{I}$ with the frame at t_1.

Let us obtain our normalisation coefficient using the static line at time $t_k = 0$. The speed of this static line, v_l, is zero, and therefore its energy is:

$$E_{\mathbf{I}^0} = \bar{c}^2 e^{-\frac{1+4x^2+4y^2+2v_g^2+4(x^2-y^2)\cos 2\theta_l - \cos 2\theta_{gl} - 8xy\sin 2\theta_l}{4}}. \tag{24}$$

As expected, the energy of this static line is constant over time and proportional to $\bar{c}^2$ (and c^2). To complete our normalisation, we need to extract a single coefficient from the function $E_{\mathbf{I}^0}$. This can be achieved by integrating $E_{\mathbf{I}^0}$ over the entire sequence domain, $\mathbf{\Omega} = X \times Y \times T$. Let $Z_{\mathbf{I}^t}$ be a function that computes the *normalisation coefficient* as:

$$Z_{\mathbf{I}^t} \triangleq \int\limits_{\mathbf{\Omega}} E_{\mathbf{I}^t}(\mathbf{x}')\mathrm{d}\mathbf{x}', \tag{25}$$

where $\mathbf{x}' = (x', y', t')$. Then, the normalisation coefficient of $E_{\mathbf{I}^0}$ can be computed as:

$$Z_{\mathbf{I}^0} = \int\limits_{\mathbf{\Omega}} E_{\mathbf{I}^0}(\mathbf{x}')\mathrm{d}\mathbf{x}' = \bar{c}^2 \int\limits_{\mathbf{\Omega}} \frac{E_{\mathbf{I}^0}(\mathbf{x}')}{\bar{c}^2}\mathrm{d}\mathbf{x}' = \frac{c^2}{8\pi^4}S, \tag{26}$$

where S denotes the output of a definite integral that is another constant but one that does not depend on the illumination of the line. Finally, we obtain the normalised sequence as:

$$\tilde{\mathbf{I}} = \frac{1}{\sqrt{Z_{\mathbf{I}^0}}}\mathbf{I} = \frac{2\sqrt{2}\pi^2}{\sqrt{S}}\delta(x\cos\theta_l - y\sin\theta_l - tv_l). \tag{27}$$

As desired, the energy of this line, $E_{\tilde{\mathbf{I}}}$, will be independent of c, and its functional form would be equal to that of $E_{\mathbf{I}}$.

B. Normalisation of Temporal Illumination Variations

The normalised sequence in (27) was obtained by dividing the sequence with a single coefficient $Z_{\mathbf{I}^0}$. To tackle temporal variations, we divide each frame of the sequence with a separate (time-dependent) coefficient $Z_{\mathbf{I}^t}$:

$$\tilde{\mathbf{I}}(x, y, t) = \frac{1}{\sqrt{Z_{\mathbf{I}^t}}}\mathbf{I}(x, y, t), \tag{28}$$

where $Z_{\mathbf{I}^t}$ is computed as in (25) but, because now it is computed from a generic sequence without a closed-form expression, we can represent it only as a definite integral.

To show that this extension is able to tackle temporal illumination variations for sequences without closed-form expressions, we recast the problem of illumination normalisation as follows. Consider a sequence $\mathbf{I}_p \triangleq \mathbf{I}_p(x, y, t)$ where there are no illumination variations, and another sequence $\mathbf{I}_q$ that contains the same motion as $\mathbf{I}_p$ but is affected by a temporal variation such as $\mathbf{I}_q(x, y, t) \triangleq (\alpha t + \beta)\mathbf{I}_p(x, y, t)$. Our goal with illumination normalisation is to have normalised versions of these sequences that have equal energies, that is, $E_{\tilde{\mathbf{I}}_p} = E_{\tilde{\mathbf{I}}_q}$.

The energies of normalised sequences can be written as:

$$E_{\tilde{\mathbf{I}}_p} = \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2, \tag{29}$$

$$E_{\tilde{\mathbf{I}}_q} = \left[\int \frac{\mathbf{I}_q(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_q^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}_q(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_q^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2, \tag{30}$$

where $\mathbf{u} = (u, v, w)$ and $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{u}$. Note that $\mathbf{I}_q^w(x, y, t) = \mathbf{I}_q(x, y, w) = (\alpha w + \beta)\mathbf{I}_p(x, y, w) = (\alpha w + \beta)\mathbf{I}_p^w(x, y, t)$, and because the convolution involved in energy computation is a linear operator, we can compute $Z_{\mathbf{I}_q^w}$ as:

$$\begin{aligned} Z_{\mathbf{I}_q^w} &= \int\limits_{\mathbf{\Omega}} E_{\mathbf{I}_q^w}(\mathbf{x}')\mathrm{d}\mathbf{x}' = \int\limits_{\mathbf{\Omega}} (\alpha w + \beta)^2 E_{\mathbf{I}_p^w}(\mathbf{x}')\mathrm{d}\mathbf{x}' \\ &= (\alpha w + \beta)^2 \int\limits_{\mathbf{\Omega}} E_{\mathbf{I}_p^w}(\mathbf{x}')\mathrm{d}\mathbf{x}'. \end{aligned} \tag{31}$$

Therefore, we can rewrite (30) as:

$$\begin{aligned} E_{\tilde{\mathbf{I}}_q} &= \left[\int \frac{(\alpha w + \beta)\mathbf{I}_p(\mathbf{u})}{(\alpha w + \beta)\sqrt{Z_{\mathbf{I}_p^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \\ &\quad \left[\int \frac{(\alpha w + \beta)\mathbf{I}_p(\mathbf{u})}{(\alpha w + \beta)\sqrt{Z_{\mathbf{I}_p^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 \\ &= \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2, \end{aligned} \tag{32}$$

As desired, the energies of the sequences $E_{\tilde{\mathbf{I}}_p}$ and $E_{\tilde{\mathbf{I}}_q}$ are equal, which can be seen by comparing (29) and (32).

The $Z_{\mathbf{I}_p^w}$ in (32) that remains after cancelling out the temporal illumination variations depends on time w. This may cause the trend of the energy function to change during normalisation, which is prohibitive, as the tuning of the filter parameters v_g, θ_g was based on the energy function to follow a specific trend. However, $Z_{\mathbf{I}_p^w}$ shows little sensitivity to time w (see Appendix A), and therefore the trends of the normalised and un-normalised energy functions are similar.

It must be noted that our normalisation scheme is most applicable when local slices of a sequence are processed and normalised independently from one another (similarly to our pipeline in Section VI-A), particularly in the presence of non-uniform illumination variations. Local processing and illumination normalisation are also biologically plausible [48] and are employed by state-of-the-art spatial [49] and spatio-temporal [50] image processing pipelines.

VI. Statistical Motion Modelling

Our goal is to obtain an explicit (rigid) motion transformation, but Gabor motion energy encodes motion implicitly. For this reason, we estimate local MVs for each pixel and produce a global motion estimation from all local MVs.

A. Statistical Local Motion Estimation

We aim to estimate a vector $\mathbf{u}^{ij} \triangleq (u^{ij}, v^{ij})$ that represents the motion of the pixel located at (i, j) between two discrete images $I_0 \triangleq I_0[x, y], I_1 \triangleq I_1[x, y]$, where u^{ij} and v^{ij} describe horizontal and vertical translation, respectively. To obtain the estimation, $\hat{\mathbf{u}}^{ij} \triangleq (\hat{u}^{ij}, \hat{v}^{ij})$, we use the Gabor motion energy around the pixel; specifically, the energy values across a $P \times P$-sized area centred on the pixel. One can compute those energy values after cropping the input images based on (i, j). Let $\mathbf{I}_{ij}$ be $\mathbf{I}_{ij} \triangleq (I_{ij,0}, I_{ij,1})$ where each $I_{ij,t}$ is a square image patch with edge size $P+2\delta P$, cropped as $I_{ij,t}[x, y] \triangleq I_t[x+i, y+j]$, and δP is the spatial padding size required for convolution. Energy is computed from a two-frame sequence, therefore we set the temporal length of the Gabor filters also as two frames, *i.e.* $T_g = 2$. Then, the even- and odd-phased (discrete) Gabor filters can also be represented as $g^e = (g^e_0, g^e_1)$ and $g^o = (g^o_0, g^o_1)$, where each g^e_t, g^o_t is a 2D array. The discrete energy can then be represented in terms of $2D$ convolutions as:

$$E_{\tilde{\mathbf{I}}_{ij}}[x, y] = \Big(\sum_{t=0}^{T_g-1} \frac{1}{\sqrt{Z_{\mathbf{I}^t_{ij}}}} (I_{ij,t} * g^e_{T_g-1-t})[x, y] \Big)^2 + \Big(\sum_{t=0}^{T_g-1} \frac{1}{\sqrt{Z_{\mathbf{I}^t_{ij}}}} (I_{ij,t} * g^o_{T_g-1-t})[x, y] \Big)^2 . \quad (33)$$

Note that here energy is a $2D$ instead of a $3D$ array, as we perform "valid" convolution (*i.e.* apply no zero-padding) [51], and since both the images and filters are of temporal length 2, the temporal length of convolution outputs is $2-2+1 = 1$ [51].

Since one Gabor filter pair (g^e, g^o) is tuned to a single orientation and speed, the energy computed only from one filter pair would not be sufficient to identify arbitrary motion [14]. Therefore, we construct and use a filter bank that comprises K filter pairs, each tuned to a different orientation. We reduce the dimensionality of energy output via mean pooling, which is a biologically plausible [48] and computationally efficient [52] way of reducing the dimensionality of Gabor filtering output. Overall, we describe the motion of the pixel at (i, j) with one feature vector $\mathbf{\Phi}^{ij} = (\phi^{ij}_1, \phi^{ij}_2, \ldots, \phi^{ij}_K)$, where ϕ^{ij}_k denotes the pooling output for the energy values computed with the k^{th} filter pair, $E^k_{\tilde{\mathbf{I}}_{ij}}[x, y]$, as

$$\phi^{ij}_k = \frac{1}{P^2} \sum_{x,y=\delta P+1}^{P+\delta P} E^k_{\tilde{\mathbf{I}}_{ij}}[x, y]. \quad (34)$$

In Section VII-B we discuss an efficient computation of ϕ^{ij}_k.

The critical question is how to estimate the motion vector $\mathbf{u}^{ij}$ from a given $\mathbf{\Phi}^{ij}$, *i.e.* how to model their relationship. We use a model that learns the relationship from data. Specifically, we use a neural network with one hidden layer that contains H hidden nodes. At local level, we model motion with a Euclidean transformation, which can be represented with four parameters as $\boldsymbol{\theta}^{ij} = (\theta^{ij}_1, \theta^{ij}_2, \theta^{ij}_3, \theta^{ij}_4)$, and then compute $\hat{\mathbf{u}}^{ij}$ by applying this Euclidean transformation to the central pixel of the patch. The neural network equation can be denoted as:

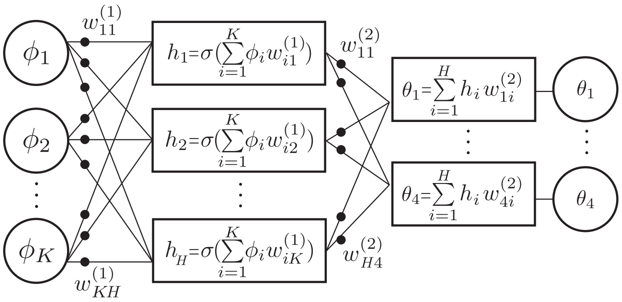
$w^{(1)}_{11}$
ϕ_1
ϕ_2
ϕ_K
$w^{(1)}_{KH}$
$h_1 = \sigma(\sum_{i=1}^{K} \phi_i w^{(1)}_{i1})$
$h_2 = \sigma(\sum_{i=1}^{K} \phi_i w^{(1)}_{i2})$
$h_H = \sigma(\sum_{i=1}^{K} \phi_i w^{(1)}_{iK})$
$w^{(2)}_{11}$
$w^{(2)}_{H4}$
$\theta_1 = \sum_{i=1}^{H} h_i w^{(2)}_{1i}$
$\theta_4 = \sum_{i=1}^{H} h_i w^{(2)}_{4i}$
θ_1
θ_4

Fig. 4. The structure of the neural network that is used for predicting local motion $\boldsymbol{\theta}$ from a feature vector $\mathbf{\Phi}$. The coefficients $w^{(1)}_{pq}, w^{(2)}_{pq}$ represent weights contained in $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$, respectively.

$$\boldsymbol{\theta}^{ij} = y(\mathbf{\Phi}^{ij}; \mathbf{w}^{(1)}, \mathbf{w}^{(2)}), \quad (35)$$

where $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ are respectively the weights of the input layer and the hidden layer, and the activation function of the hidden layers, $\sigma(\cdot)$, is a $\tanh(\cdot)$ function (see Fig. 4). Obtaining training data to train this network is straightforward. To create one training sample $(\mathbf{\Phi}_n, \boldsymbol{\theta}_n)$, we crop a square image patch with edge size $P+2\delta P$ from any image dataset, and then perturb the patch by applying a random transformation $\boldsymbol{\theta}_n$. We then obtain the features $\mathbf{\Phi}_n$ from the pair that comprises the original and perturbed image, using (34). In this process we set $\delta P = P/2$. Note that the perturbed image we created may have blank regions near the image edges, due to the transformation $\boldsymbol{\theta}_n$. The features computed through (34) ignore the energy values across a margin of δP pixels, and therefore are usually not affected by such blank regions.

B. Global Motion Estimation

To estimate the global motion between a *template* frame I_0 and a *target* frame I_1, we use the iterative scheme illustrated in Algorithm 1. At each iteration k, we estimate local motion vectors $\hat{\mathbf{u}}^{ij}$ for all pixels (i, j) except those that lie on a margin, and store them in a set $\mathcal{U}_k$. Then, we predict a projective transformation $\mathbf{m}_k = (m_1, m_2, \ldots, m_8)$ by applying least mean squares regression based on the local motion vectors in $\mathcal{U}_k$ after eliminating outliers motion vectors via RANSAC [16]. Next, we convert the transformation $\mathbf{m}_k$ into a homography matrix simply as:

$$\hat{\mathbf{H}}_k = h(\mathbf{m}_k) = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix}. \quad (36)$$

We continue iterations until the element-wise L_2 matrix norm $||\hat{\mathbf{H}}_k - \mathbf{I}_3||_2$ gets smaller than a *convergence threshold* ϵ, or a maximal number of iterations $K_{\max}$ is reached. To estimate large-scale global motion efficiently, we apply Algorithm 1 in a coarse-to-fine manner through a pyramid representation [40].

Algorithm 1: Global Motion Estimation

```
Input:       $I_0, I_1$: template and target frames
Output:      $\hat{\mathbf{H}}^{-1}$: predicted projective transformation
Definitions: $\mathbf{I}_3$: $3 \times 3$ identity matrix
             $M, N$: Width, height of input frames
             $r$: Margin for discarded pixels, $r = \delta P + [P/2]$
             $\odot$: Operator that warps $I_1$ based on $\hat{\mathbf{H}}$

$\hat{\mathbf{H}} \leftarrow \mathbf{I}_3$
for $k \leftarrow 1, K_{\max}$ do
    $\mathcal{U}_k \leftarrow \left\{\hat{\mathbf{u}}_k^{ij} = y(\mathbf{\Phi}^{ij}) \mid (i,j) \in \left(\mathbb{N}_{[r,M-r]} \times \mathbb{N}_{[r,N-r]}\right)\right\}$
    $\mathbf{m}_k \leftarrow \text{RANSAC}(\mathcal{U}_k)$
    $\hat{\mathbf{H}}_k \leftarrow h(\mathbf{m}_k)$
    if $||\hat{\mathbf{H}}_k - \mathbf{I}_3||_2 < \epsilon$ then
        return
    end if
    $I_1 \leftarrow \hat{\mathbf{H}}_k^{-1} \odot I_1$
    $\hat{\mathbf{H}} \leftarrow \hat{\mathbf{H}}_k \hat{\mathbf{H}}$
end for
```

If $K_{\max}$ is large enough, a sufficient condition for convergence is that the error between the estimated and the actual homography transformation, $||\hat{\mathbf{H}} - \mathbf{H}||_2$, is reduced at each iteration k, which implies that the MVs must be estimated more accurately as k increases. In fact, unlike early iterations that can accept larger errors in the estimation of MVs as long as $||\hat{\mathbf{H}} - \mathbf{H}||_2$ decreases, only small errors are allowed near convergence when the magnitude of MVs is small. In the next section we analyse whether our MV estimation adheres to this error profile by reporting variations in estimation performance with respect to the magnitude of MVs.

VII. Experimental Validation

In this section we evaluate the sensitivity of local MV estimation to a number of parameters and compare our method with other methods both in constant and changing illumination conditions. We also evaluate the GME performance in the presence of outlier motions (including barrel distortions and local motions incongruent with global motion) and in the presence of illumination variations by comparing our method with various state-of-the-art GME methods. The source code and data needed to reproduce experimental results are available at http://www.eecs.qmul.ac.uk/~es302/gme.

A. Methods Under Comparison

We compare the local MV estimation performance of our method with a block matching method (BM) [38], which is typically used for coarse MV field estimation, and with an optical flow method, which is used for dense MV (D-MV) estimation [27]. We also aim to quantify if there is any benefit to replacing the neural network with another regressor, and compare the performance of the neural network (NN) with a well-established regressor, namely, elastic net [53] (EN). For BM we used the built-in block matching function in MATLAB. This function originally outputs values at 1 pixel resolution; we increased its resolution to 0.25 pixels by resizing the blocks 4 times prior to estimation. We implemented D-MV through its original code [39].

We compare our overall GME performance with two feature-based methods, one based on SURF [18] and one based on MSER [54] features; two direct minimisation/maximisation methods, the standard LK with inverse compositional algorithm (IC-LK) [34] and a method based on maximising gradient correlation coefficient (MGCC) [36]; and one dense MV-based method [27]. When testing against illumination variations, we additionally compare with two more methods that are more robust against illumination variations: a LK method that uses (spatial) Gabor filters (G-LK) [35] and a robust FFT method (R-FFT) [29]. We implement SURF and MSER using the functions provided in the Image Processing toolbox of MATLAB. For IC-LK, MGCC and D-MV we use the implementations provided by the authors. For RANSAC estimation, which is needed by SURF, MSER and D-MV, we use the geometric transformation estimation function of MATLAB. For G-LK, we used the implementation provided for comparison in [36], and for R-FFT we use the implementation that was kindly provided by the authors of the technique.

B. Implementation and Parameters

Experiments are conducted on a workstation with an Intel Xeon CPU (2.40GHz). We report results for two different RANSAC implementations: our implementation and the default OpenCV implementation. The latter is faster, however, based on our experimental observations, requires a higher rate of inliers. We used our RANSAC implementation during PIE ([55]) experiments where inlier rates are lower due to illumination variations. We trained the neural network using NETLAB [56] and performed optimisation with conjugate gradients optimisation. We implemented Algorithm 1 on C++, and used summed-area tables [57] to compute $\mathcal{U}_k$ efficiently. Cropping the input images for each MV from scratch and then computing energy (*i.e.* as described in Section VI-A), may be computationally costly. Instead, we can first compute the convolutions on the entire images and then use the necessary values for an MV at (i, j) as follows. For a filter g, the equality $(I_{ij,t} * g)[x, y] = (I_t * g)[x+i, y+j]$ holds due to our definition of $I_{ij,t}$ (Section VI-A) and the fact that translation commutes with convolution. Let A_t^{xy}, B_t^{xy} be $A_t^{xy} \triangleq (I_t * g^e_{T_g-1-t})[x, y]$ and $B_t^{xy} \triangleq (I_t * g^o_{T_g-1-t})[x, y]$. Using the afore-mentioned equality, we can perform the pooling in (34) as:

$$\begin{aligned}\phi^{ij} =& \frac{1}{P^2}\sum_{x,y}\left[\left(\sum_{t=0}^{T_g-1}\frac{1}{\sqrt{Z_{\mathbf{I}^t_{ij}}}}A_t^{xy}\right)^2 + \left(\sum_{t=0}^{T_g-1}\frac{1}{\sqrt{Z_{\mathbf{I}^t_{ij}}}}B_t^{xy}\right)^2\right] \\ =& \frac{\sum\limits_{x,y}(A_0^{xy})^2}{P^2 Z_{\mathbf{I}^0_{ij}}} + \frac{\sum\limits_{x,y}(A_1^{xy})^2}{P^2 Z_{\mathbf{I}^1_{ij}}} + \frac{\sum\limits_{x,y}(B_0^{xy})^2}{P^2 Z_{\mathbf{I}^0_{ij}}} + \frac{\sum\limits_{x,y}(B_1^{xy})^2}{P^2 Z_{\mathbf{I}^1_{ij}}} \\ &+ \frac{2}{P^2\sqrt{Z_{\mathbf{I}^0_{ij}} Z_{\mathbf{I}^1_{ij}}}}\left(\sum_{x,y}A_0^{xy}A_1^{xy} + \sum_{x,y}B_0^{xy}B_1^{xy}\right), \quad (37)\end{aligned}$$

where we dropped the dependence of ϕ_k^{ij} to k for clarity. The the sums in the right-hand-side run over $(x, y) \in \mathbb{N}_{[i+1,i+P]} \times$

Fig. 5. Top: exemplar frames from the ego-centric camera videos. Bottom: exemplar frames from the PIE dataset with illumination variations.

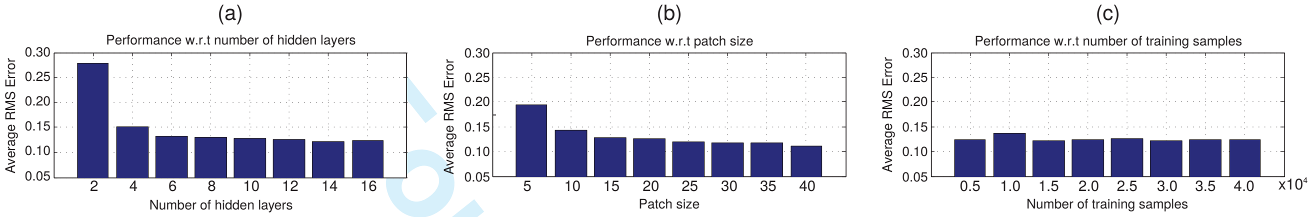

Fig. 6. Sensitivity of local motion estimation (a) against the number of hidden layers H, (b) patch size P, (c) and number of training samples N_{tra}.

$\mathbb{N}_{[j+1,j+P]}$. After writing the sums as in (37), we can employ summed-area tables, which enable the computation of each sum with four instead of P^2 operations [57]. The integrals (*i.e.* sums) required for $Z_{\mathbf{I}^0_{ij}}, Z_{\mathbf{I}^1_{ij}}$ can also be computed in a similar manner, once the summed-area table of the static energies, $E_{\mathbf{I}^0}, E_{\mathbf{I}^1}$, where $\mathbf{I} \triangleq (I_0, I_1)$, are pre-computed.

We use the same parameters in all experiments. We use $K = 8$ filter pairs tuned to orientations of $0°, 45°, \ldots, 315°$ with filter speeds $v_g = 1$ and we estimate large-scale global motion hierarchically at scales of 1/6, 1/4, 1/3, 1/2, 2/3, and 1/1. We set the number of maximal iterations as $K_{\text{max}} = 5$ per scale, and the convergence threshold to $\epsilon = 10^{-5}$. Based on the analysis in Section VII-C, we set N_{tra}, H and P as $N_{\text{tra}} = 20000, H = 8$ and $P = 20$.

C. Local Motion Estimation Performance

We evaluate local MV estimation performance on synthesised MVs. For this purpose, we create sets of test samples $\mathcal{I}$ that comprise N_{tes} pairs of $2P \times 2P$-sized randomly cropped patches $\mathbf{I}_i = (I_i, I'_i)$; that is, $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{N_{\text{tes}}}\}$. The patches I'_i and I_i differ by a random (Euclidean) motion $\boldsymbol{\theta}_i$, which causes the centre of the first patch to move by a translation $\mathbf{u}_i = (u_i, v_i)$. The goal is to obtain accurate predictions $\hat{\mathbf{u}}_i = (\hat{u}_i, \hat{v}_i)$ of the true motions $\mathbf{u}_i$. We evaluate accuracy as the *average root mean square* (RMS) error $\bar{e}_{\text{RMS}}$ between the predicted and the true motion vectors:

$$\bar{e}_{\text{RMS}} = \sum_{i=1}^{N_{\text{tes}}} \sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2}. \qquad (38)$$

To evaluate on constant illumination conditions, we crop the patches from the INRIA holidays dataset [58], which contains mostly natural images. For evaluation on changing illumination conditions, we use the PIE dataset [55], which contains facial sequences. The subjects of the dataset are sitting in front of a camera, while the illumination conditions are changed rapidly in a controlled manner (Fig. 5, bottom).

There are three parameters that have influence on the performance of our local MV estimation method: the number of hidden layers, H, the local patch size, P, and the number of training samples, N_{tra}. We measure the sensitivity against these parameters over a test set of $N_{\text{tes}} = 2000$ samples. We perform tests for each parameter separately, and keep the other two parameters that are not tested fixed. When not tested, these parameters are set to $N_{\text{tra}} = 20000, H = 8$ and $P = 20$.

The three graphs in Fig. 6 illustrate how performance varies with the variation in H, P and N_{tra}. For the number of hidden layers, H, we see that the performance does not vary much for values between 6 and 16, therefore we can pick a low value such as 6 or 8 to keep the neural network complexity low. In the rightmost graph we see that the number of training samples N_{tra} does not have a significant effect in performance within the range of 5,000-40,000. For the patch size P we notice that there is a considerable improvement when we increase patch size from 5 to 15, and a slight improvement as we keep increasing P further. However, increasing P too much reduces the image area where local motion vectors can be extracted (see r in Algorithm 1). Therefore, we limit the patch size to $P = 20$.

Fig. 7a compares the performance of our method with BM and D-MV on MVs synthesised from the INRIA dataset. For our method, we provide results for two different illumination normalisation schemes: one where normalisation coefficients are computed from one frame only (*i.e.* first frame) as in (27), and another time-dependent one where normalisation coefficients are computed from both frames as in (28). As discussed in Section VI-B, our GME algorithm requires very accurate MV estimation when the magnitude of an MV, $\rho_i \triangleq ||\mathbf{u}_i||$, is small, whereas errors may be tolerated when ρ_i is large. Therefore, in Fig. 7a we show performance for

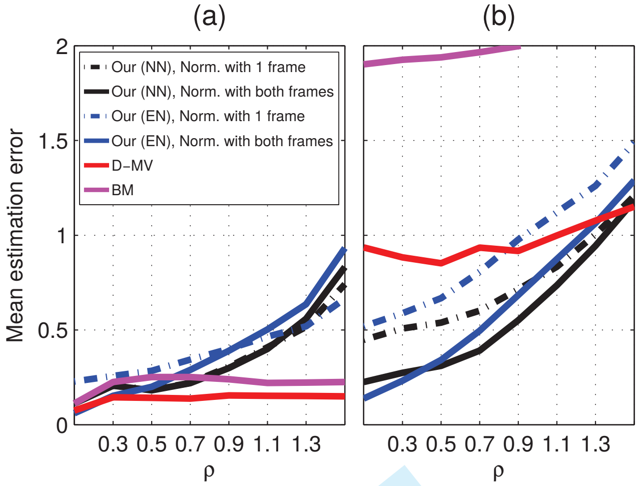

Fig. 7. Local MV estimation performance. (a) on the INRIA dataset, which contains no illumination variations; (b) on the PIE dataset, which does contain illumination variations.

various magnitude ranges $[\rho{-}\delta\rho, \rho{+}\delta\rho]$ separately by creating $N_{tes} = 500$ samples for each range. The x axis shows the center of the range ρ, and we set the range radius as $\delta\rho = 0.1$.

Fig. 7a shows that our method (both with NN and EN) yields higher errors than BM and D-MV for MVs with high magnitude (*e.g.* when $\rho > 0.7$). However, those errors pose little problem for GME, as the error profile adheres to that described in Section VI-B: Errors become lower as the motion magnitude ρ becomes lower, and particularly low for ρ values as small as 0.3 or 0.5. Normalising illumination with one frame or with both frames makes little difference on the INRIA dataset, which does not contain temporal illumination variations. Overall, the D-MV method achieves the best performance, and its accuracy depends little on the motion magnitude ρ. D-MV is an optical flow method; MVs are the final output of the algorithm, therefore they are expected to be accurate independently of the magnitude of the MVs. For our method, MVs are intermediate quantities that serve for the final GME output.

Fig. 7b shows the results of the same type of experiment for the PIE dataset with illumination variations. As expected, the simple BM method cannot perform reliably. However, D-MV cannot perform very reliably either, even though this method is computationally more complex and is designed to be partly robust to illumination variations. Our method outperforms other methods, particularly for small ρ values. Also, Fig. 7 suggests that, in the presence of illumination variations, using both frames for contrast normalisation (*i.e.* time-dependent normalisation) is consistently better than using a single frame. Overall, the error trend of our method is similar to that in Fig. 7a, *i.e.* errors get lower as ρ decreases. The decrease at lower rate suggests that accurate GME would require more iterations.

D. Global Motion Estimation Experiments

To evaluate GME performance in the presence of outlier motions, we adopt a popular validation scheme [40] and use test sequences that contain mostly camera motion in addition to outlier local motions. The goal is to compensate the global motion between pairs of consecutive images by warping the second (target) frame in each pair onto the first (template) frame. To evaluate performance in the presence of barrel distortions, we use 3 sequences acquired during a study where two human participants are involved in a structured conversation driven by a small humanoid robot and each participant wears an ego-centric camera placed on their forehead [4]. We refer to these sequences as ego-centric1-3. In addition to outlier local motions (*e.g.* moving robot, people), these sequences have the challenge of high barrel (lens) distortion, which is not possible to compensate with homographic transformations. Unlike local motions, barrel distortions produce systematic outliers, where all straight lines throughout the image are curved outward, and even more so near the edges of the image — this effect gets worse as the global motion gets larger and edges look more different. To test against local outlier motions, we use widely-known MPEG-4 test sequences [8], [22], [32], [22], namely, City, Coastguard, Flower, Foreman, Mobile, Stefan, Tempete, Waterfall. The true global motion is not known for any of those sequences, therefore we use an indirect metric, namely the *peak-signal-to-noise ratio* (PSNR) between the warped target and template frame, which gets higher as the global motion is compensated with higher accuracy. PSNR results on some sequences can be misleadingly low due to large outlier motions [22], therefore, for some sequences we provide additional PSNR results by ignoring the foreground object/person — this metric is also referred to as background PSNR (BPSNR) [59]. We provide BPSNR results for the egocentric1–3 sequences and for the Stefan, Coastguard and Foreman. Manually annotated foreground masks are provided in supplementary material.

TABLE II
GLOBAL MOTION COMPENSATION PERFORMANCE IN TERMS OF PSNR (THE HIGHER THE BETTER). BEST VALUES FOR EACH SEQUENCE ARE TYPED IN BOLD. †BACKGROUND PSNR (BPSNR) VALUES.

	SURF [18]	MSER [54]	IC-LK [34]	MGCC [28]	D-MV [27]	Our Work
Ego-centric1	30.57	30.22	30.59	30.26	30.09	**30.61**
Ego-centric1†	35.91	35.12	35.38	35.63	35.99	**36.47**
Ego-centric2	26.30	25.60	**26.42**	25.93	26.04	26.15
Ego-centric2†	30.04	29.19	29.58	29.83	30.43	**30.66**
Ego-centric3	28.67	27.89	28.72	28.22	28.31	**28.74**
Ego-centric3†	31.96	30.90	31.77	31.53	31.90	**32.33**
City	30.23	29.87	29.94	29.78	30.26	**30.47**
Coastguard	26.68	27.25	**28.07**	27.54	27.56	27.54
Coastguard†	27.66	28.51	29.70	29.94	**30.06**	30.03
Flower	**27.01**	26.53	26.72	26.32	25.67	25.85
Foreman	27.69	28.69	28.93	**30.06**	29.94	29.97
Foreman†	29.90	33.40	34.06	35.93	**36.43**	35.91
Mobile	25.01	25.41	25.71	25.71	**25.85**	**25.85**
Stefan	**27.32**	27.11	26.79	26.86	27.22	27.16
Stefan†	**33.37**	32.99	31.71	32.45	33.03	33.23
Tempete	27.80	27.41	27.85	**27.86**	**27.86**	**27.86**
Waterfall	38.25	36.78	37.94	38.43	38.43	**38.45**

Table II shows the performance of all methods on the egocentric1-3 and MPEG-4 sequences, and Fig. 8 shows exemplar difference images for several sequences — the difference images show less variation when global motion is compensated

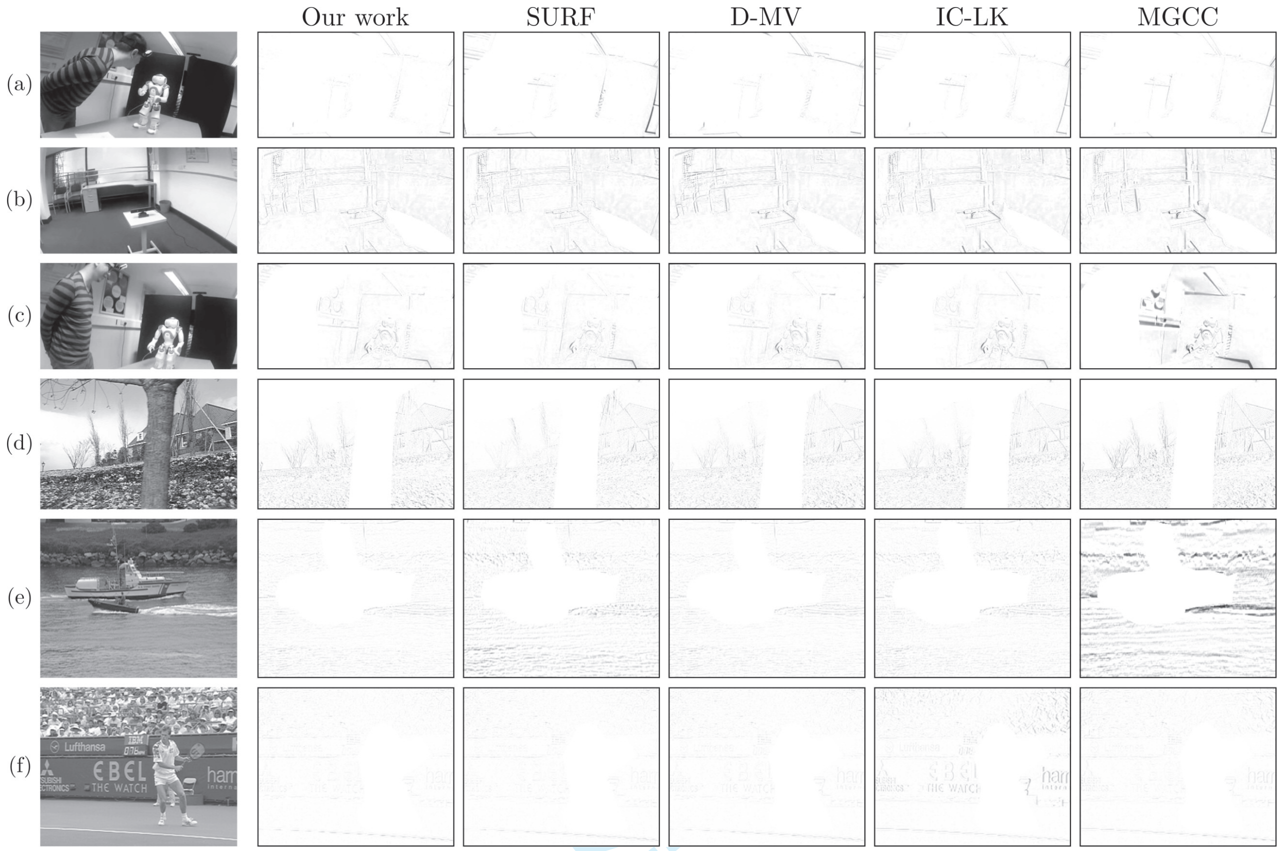

Fig. 8. Exemplar frames from test sequences and the corresponding difference images obtained after global motion was compensated with each of the methods. To enhance the interpretation of the difference images, we segmented out the largest moving foreground object from difference images manually and also inverted the difference images in color. (a) Ego-centric1, the SURF method in this frame was affected by barrel distortions which caused higher noise near the edges of the difference; (b) Ego-centric2, a scene with a large number of lines curved by barrel distortion; (c) Ego-centric3, a scene with barrel distortions and large outlier motions; (d) Flower, a scene with elements of varying depth; (e) Coastguard, two (visually salient) ships are moving towards each other; (f) Stefan, a sequence where the tennis player causes large outlier motions.

with higher accuracy. For the ego-centric sequences, we note that the local and dense methods IC-LK, MGCC perform generally well, but they are prone to be misled in the presence of large barrel distortions or large outlier motions (Fig. 8b,c). The feature-based SURF method performs also well and is generally not affected by local motions (caused by the robot or by people). However, it degrades when the RANSAC estimator is misled by salient features that are extracted from highly curved edges (see Fig. 8a). The performance on ego-centric sequences is better interpreted through BPSNR, which suggest that our method performs best, followed by the D-MV method. These methods have in common a RANSAC estimation that is performed on a dense input, which is less likely to be misled than the (sparse) feature-based methods.

On the MPEG-4 test sequences, feature-based methods, and particularly SURF, perform well in sequences where there are visually salient regions to estimate the global motion from (*e.g.* Waterfall, Flower), even in the presence of large outlier local motions (Mobile, Stefan). However, they may degrade considerably when salient regions are concentrated on regions of outlier motions (Coastguard, Foreman). IC-LK and MGCC are less dependent on visual saliency. However, they are occasionally affected by the amount of outlier motions, particularly when they involve elements with rich texture such as in Stefan. Moreover, these methods estimate only affine transformations, and their motion compensation is limited in sequences that undergo a projective transformation, such as City. Overall, our method and D-MV have the highest number of best or next-to-best PSNR results on MPEG-4 test sequences — the difference between the two methods is marginal in most cases. The absence of visual saliency affects less our method and D-MV than feature-based methods. Moreover, our method and D-MV are less sensitive to outliers and to perspective transformations than direct methods such as LK and MGCC.

To evaluate GME performance in the presence of illumination variations, we use the PIE dataset. The local and global illumination variations in the PIE dataset affect the background and foreground (*i.e.* face) of the images differently, creating challenging shadows and spurious motions (see Fig. 5). The advantage of evaluation on PIE is that we can evaluate using a direct metric. Because the subjects are sitting stably and there is no camera motion, we can apply the global motion ourselves in a controlled manner and then try to estimate this known motion — this approach was used with (other datasets) while evaluating LK methods [36], [55], [42]. For each template

TABLE III
GME ACCURACY ON PIE DATASET MEASURED AS AVERAGE MAE (IN PIXELS) OVER ALL TEST SAMPLES (THE LOWER THE BETTER) AND THE PERCENTAGE OF TEST SAMPLES WITH LESS THAN θ PIXELS (THE HIGHER THE BETTER)

	SURF	MSER	FFT	IC-LK	G-LK	MGCC	D-MV	Our Work
MAE	32.21	108.86	6.20	3.79	2.36	0.59	2.13	**0.47**
$\theta = 0.25$	0.00	0.00	5.25	2.50	4.25	25.75	3.75	**47.25**
$\theta = 0.50$	0.25	0.00	20.00	29.00	39.00	65.50	20.25	**89.75**
$\theta = 1.00$	1.75	1.25	59.75	61.00	74.00	91.00	57.00	**94.25**
$\theta = 2.00$	15.25	10.75	87.00	80.25	87.25	95.75	80.00	**96.00**
$\theta = 3.00$	28.25	22.25	95.00	86.75	90.00	**98.00**	86.50	96.75
$\theta = 5.00$	48.75	43.50	95.00	89.00	92.75	**99.00**	91.00	98.75

frame we define two *canonical points*[2] [55], the leftmost and rightmost point that sit in the vertical middle of the frame. We perturb these points with a Gaussian noise with standard deviation $\sigma = 3$. This is a value where LK methods [34], [35] and MGCC [36] generally converge, yet to ensure their convergence, we run those methods with 60 iterations (*i.e.* the double of what is used in the original papers). The canonical versus perturbed points define a Euclidean transformation, which is then applied to the target frame. The first evaluation metric we use is the *mean absolute error* (MAE) in the predicted locations of the two perturbed points. However, MAE can be misleading if there are large errors in a small number of samples. Therefore we use a second metric: The ratio of samples whose MAE is smaller than a certain number of pixels [60]. We test on the 400 pairs of frames, which are obtained from all the frames of the first 20 subjects' sequences. We cropped the faces in these datasets to 200 × 200 pixels based on the facial landmark points provided with the dataset.

Table III shows the experimental results on PIE dataset both in terms of average MAE in all 400 pairs and in terms of ratio of samples whose MAE is smaller than a certain number of pixels. Feature-based methods degrade notably as the number of matched features drops severely with the illumination variations. R-FFT offers some robustness against illumination variations by achieving less than 3 pixels MAE in 95.00% of images, however, the ratio of estimations with small errors (*e.g.* 1 pixel) is quite low. IC-LK and D-MV are affected negatively by illumination variations, and the usage of spatial Gabor filters for LK (G-LK) provides only a limited improvement over IC-LK. The two methods that emerge as robust and accurate in the presence of illumination variations are our method and MGCC. While MGCC achieves a slightly higher precision by registering 98.00% of the samples with less than 2 pixels error, our method stands out with a notably high accuracy, registering 89.75% of the images with less than 0.5 pixels error.

E. Computational Cost

In Fig. 9 we show the average computation time per iteration for several processing components, for GME on an image sized 352×288. One iteration takes 3.45 seconds on average with OpenCV's RANSAC implementation and 4.60 secs with our RANSAC implementation. If we exclude RANSAC, the most time-consuming task computing the convolutions for Gabor motion energy (1.75 seconds), which is followed by the neural network-based MV prediction (for all pixels combined together, 0.94 seconds). The overall GME for all iterations and scales (see Section VII-B) is 29.9 seconds on average with OpenCV's RANSAC implementation and 60.1 secs with our RANSAC implementation.

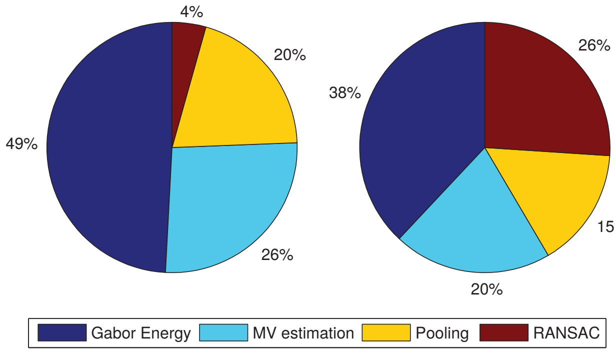

Fig. 9. Computational cost per iteration in terms of percentage. Left: OpenCV's RANSAC implementation; right: our RANSAC implementation.

VIII. CONCLUSION

We presented a global motion estimation (GME) scheme whose key components are the biologically-inspired low-level motion features and the usage of a statistical approach to model the relationship between low-level motion features and the corresponding motion vectors. We derived analytical expressions of the spatio-temporal Gabor motion energy for a moving line and we showed how to tune a spatio-temporal Gabor filter to a line moving with a particular speed and direction. Moreover, we proposed a normalisation scheme to render the output of Gabor convolution robust to temporal illumination variations.

By encoding motion with Gabor energy and estimating motion vectors through statistical learning, we achieve state-of-the-art GME accuracy in the presence of illumination variations, which are an open challenge for most GME approaches.

To reduce the computational cost of spatio-temporal Gabor filters, the proposed scheme can be applied to sparse parts of images only, or efficient hardware solutions can be employed [61]. Another solution can be searching for efficient approximations of Gabor filters [62], in which case the (illumination-normalised) Gabor motion energy equations we provide in this paper can serve as a baseline to compare the response of the approximated filters.

APPENDIX A

We show that the $Z_{\mathbf{I}_p^w}$ coefficient that appears in (32) after illumination is cancelled out changes slowly with time w, and therefore causes little variation in the trend of the signal we aim to measure (*i.e.* motion energy). This is important, as it ensures that during normalisation we are not altering the characteristic behaviour of motion energy, which was

[2] The number of canonical points in other papers is typically 3, defining an affine transformation. We pick only 2 points to be able to compare with the Robust FFT method [29], which can model only Euclidean transformations.

discussed throughout Section IV. We analyse the sensitivity of $Z_{\mathbf{I}^t}$ to t on a sequence $\mathbf{I}$ where there is a global translation and no illumination variations — a sequence that adheres to the definition of $\mathbf{I}_p$ in Section V-B. We may show that $Z_{\mathbf{I}^t}$ varies slowly with time by showing that the ℓ_1 distance between the coefficients of two frames, $|Z_{\mathbf{I}^{t_m}} - Z_{\mathbf{I}^{t_n}}|$, is small. To compute $Z_{\mathbf{I}^{t_m}}, Z_{\mathbf{I}^{t_n}}$, we first need to obtain the static sequences $\mathbf{I}^{t_m}, \mathbf{I}^{t_n}$. Since the only difference between the frames of $\mathbf{I}$ is a global translation, the static sequences are translated versions of each other (similarly to Fig. 3b,c), that is, $\mathbf{I}^{t_m}(\mathbf{x}) = \mathbf{I}^{t_n}(\mathbf{x}+\boldsymbol{\tau})$ for some $\boldsymbol{\tau} = (\tau_x, \tau_y, 0)$.

To compute $Z_{\mathbf{I}^{t_m}}$ as in (25), we first need to compute the energy, which is based on convolution. Let $h_n(\mathbf{x}, g) \triangleq (\mathbf{I}^{t_n} * g)(\mathbf{x})$ and $h_m(\mathbf{x}, g) \triangleq (\mathbf{I}^{t_m} * g)(\mathbf{x})$. Since translation commutes with convolution, h_m can be rewritten as:

$$h_m(\mathbf{x}, g) = h_n(\mathbf{x}+\boldsymbol{\tau}, g). \tag{39}$$

The energies $E_{\mathbf{I}^{t_n}}(\mathbf{x})$ and $E_{\mathbf{I}^{t_m}}(\mathbf{x})$ can then be computed as:

$$E_{\mathbf{I}^{t_n}}(\mathbf{x}) = (h_n(\mathbf{x}, g^e))^2 + (h_n(\mathbf{x}, g^o))^2, \tag{40}$$

$$\begin{aligned} E_{\mathbf{I}^{t_m}}(\mathbf{x}) &= (h_m(\mathbf{x}, g^e))^2 + (h_m(\mathbf{x}, g^o))^2 \\ &= (h_n(\mathbf{x}+\boldsymbol{\tau}, g^e))^2 + (h_n(\mathbf{x}+\boldsymbol{\tau}, g^o))^2 \\ &= E_{\mathbf{I}^{t_n}}(\mathbf{x}+\boldsymbol{\tau}). \end{aligned} \tag{41}$$

Then, for a volume $\boldsymbol{\Omega} \triangleq X \times Y \times T \triangleq (x_0, x_f) \times (y_0, y_f) \times (t_0, t_f)$, the coefficients $Z_{\mathbf{I}^{t_n}}$, $Z_{\mathbf{I}^{t_m}}$ can be computed as:

$$Z_{\mathbf{I}^{t_n}} = \int_{\boldsymbol{\Omega}} E_{\mathbf{I}^{t_n}}(\mathbf{x}')\mathrm{d}\mathbf{x}', \tag{42}$$

$$Z_{\mathbf{I}^{t_m}} = \int_{\boldsymbol{\Omega}} E_{\mathbf{I}^{t_m}}(\mathbf{x}')\mathrm{d}\mathbf{x}' = \int_{\boldsymbol{\Omega}} E_{\mathbf{I}^{t_n}}(\mathbf{x}'+\tau)\mathrm{d}\mathbf{x}'. \tag{43}$$

The distance $|Z_{\mathbf{I}^{t_n}} - Z_{\mathbf{I}^{t_m}}|$ can rewritten with a change of variable in the integral of $Z_{\mathbf{I}^{t_m}}$. Let $X' \triangleq (x_0+\tau_x, x_f+\tau_x)$, $Y' \triangleq (y_0+\tau_y, y_f+\tau_y)$ and $\boldsymbol{\Omega}' \triangleq X' \times Y' \times T$. Then, it can be shown that:

$$|Z_{\mathbf{I}^{t_n}} - Z_{\mathbf{I}^{t_m}}| = \left| \int_{\boldsymbol{\Omega}} E_{\mathbf{I}^{t_n}}(\mathbf{x}')\mathrm{d}\mathbf{x}' - \int_{\boldsymbol{\Omega}'} E_{\mathbf{I}^{t_n}}(\mathbf{x}')\mathrm{d}\mathbf{x}' \right|. \tag{44}$$

We can interpret (44) better by excluding the region of intersection, $\boldsymbol{\Omega} \cap \boldsymbol{\Omega}'$: This region will have no contribution to the distance in (44), as the integrands of the two integrals in (44) are equal, and their difference would yield zero when the integration is done over the same region. The non-zero contribution to (44) can only come from the non-intersecting regions: $\boldsymbol{\Omega}\backslash\boldsymbol{\Omega}'$ and $\boldsymbol{\Omega}'\backslash\boldsymbol{\Omega}$. These regions depend on the amount of translation: if translation is small, then $\boldsymbol{\Omega}\backslash\boldsymbol{\Omega}', \boldsymbol{\Omega}'\backslash\boldsymbol{\Omega}$ become small, and therefore $|Z_{\mathbf{I}^{t_n}} - Z_{\mathbf{I}^{t_m}}|$ is likely to be small.

In Fig. 10 we show quantitatively how two successive coefficients $Z_{\mathbf{I}^0}$, $Z_{\mathbf{I}^1}$ change with respect to the amount of translation. To this end, we crop $N = 1000$ image samples, $\{I_n\}_{n=1}^N$, each of size $2P \times 2P$, from randomly picked regions in the first frames of the MPEG-4 sequences (Section VII-D). We synthesize two-frame sequences such as $\mathbf{I}_{n,\tau} = (I_n, I_n')$, where I_n denotes a sample and I_n' denotes the same sample after being translated horizontally by τ pixels. We synthesise 10 sequences per sample, $\mathbf{I}_{n,\tau_0}, \mathbf{I}_{n,\tau_1}, \ldots, \mathbf{I}_{n,\tau_9}$, such as $\tau_i = i/2$.

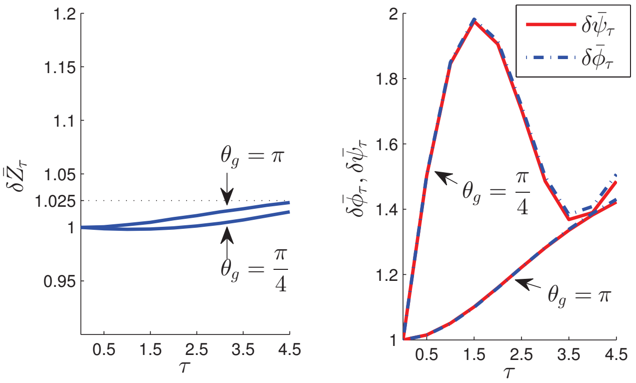

Fig. 10. Illustration which depicts that the $Z_{\mathbf{I}^t}$ coefficient shows small variation over time (left), and that this variation causes a negligible change in the trend of the motion energy function. Results are obtained with two pairs of filters tuned to different orientations θ_g but to a common speed $v_g = 1$.

We can measure how $Z_{\mathbf{I}^t_{n,\tau}}$ varies between the two frames of $\mathbf{I}_{n,\tau}$ through the ratio $Z_{\mathbf{I}^1_{n,\tau}}/Z_{\mathbf{I}^0_{n,\tau}}$. Specifically, we use the average of this ratio over all sequences,

$$\delta\bar{Z}_\tau \triangleq \sum_{n=1}^{N} \frac{Z_{\mathbf{I}^1_{n,\tau}}}{Z_{\mathbf{I}^0_{n,\tau}}}. \tag{45}$$

In Fig. 10 (left) we show how $\delta\bar{Z}_\tau$ changes with τ, for coefficients computed with two filter pairs tuned to different orientations. We note that $\delta\bar{Z}_\tau$ generally deviates from 1 proportionally to τ, which is in accordance with the conclusion we reached after (44). However, this increase is relatively small; the deviation in $\delta\bar{Z}_\tau$ never exceeds 2.5%, which shows that $Z_{\mathbf{I}^t_{n,\tau}}$ has little sensitivity to amount of translation, τ.

We now analyse whether this increase is significant: We illustrate how normalisation with the time-dependent $Z_{\mathbf{I}^t}$ coefficients changes the trend of the signal that we aim to measure — the Gabor motion energy. For this purpose, we compute how the pooling output of the sequences varies with τ:

$$\delta\bar{\phi}_\tau \triangleq \sum_{n=1}^{N} \frac{\phi_{n,\tau}}{\phi_{n,\tau_0}}, \tag{46}$$

where $\phi_{n,\tau}$ is the output of mean pooling of the normalised energy of $\mathbf{I}_{n,\tau}$. We compare $\delta\bar{\phi}_\tau$ with the original (*i.e.* unnormalised) energy, by computing the following ratio:

$$\delta\bar{\psi}_\tau \triangleq \sum_{n=1}^{N} \frac{\psi_{n,\tau}}{\psi_{n,\tau_0}}, \tag{47}$$

where $\psi_{n,\tau}$ denotes a pooling output computed from the unnormalised energy. Note that $\delta\bar{\phi}_\tau$ and $\delta\bar{\psi}_\tau$ can be compared fairly, because both are divided with the pooling output of the non-moving sequence. Ideally, we would like $\delta\bar{\psi}_\tau$ and $\delta\bar{\phi}_\tau$ to be the same for any τ value.

Finally, in Fig. 10 (right) we compare $\delta\bar{\phi}_\tau$ with $\delta\bar{\psi}_\tau$: The difference between $\delta\bar{\phi}_\tau$ and $\delta\bar{\psi}_\tau$ is small even for the largest τ value. It is therefore reasonable to assume that the $Z_{\mathbf{I}^t}$ coefficients cause a negligible change in the trend of the energy, given that the sequence they are computed from contains no illumination variations.

References

[1] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual slam," *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.

[2] D. Floreano, J.-C. Zufferey, M. V. Srinivasan, and C. Ellington, *Flying insects and robots*. Springer, 2009.

[3] A. Eliazar and R. Parr, "Dp-slam: Fast, robust simultaneous localization and mapping without predetermined landmarks," in *IJCAI*, vol. 3, 2003, pp. 1135–1142.

[4] O. Celiktutan and H. Gunes, "Computational analysis of human-robot interactions through first-person vision: Personality and interaction experience," in *Proc. IEEE Int'l Symp. on Robot and Human interactive Communication*, 2015.

[5] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition." in *Proc. British Machine Vision Conf.*, 2008, pp. 1–10.

[6] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Probabilistic temporal subpixel registration for facial expression analysis," in *Proc. Asian Conf. Computer Vision*, 2014.

[7] D. Perperidis, R. H. Mohiaddin, and D. Rueckert, "Spatio-temporal free-form registration of cardiac MR image sequences," *Medical Image Analysis*, vol. 9, no. 5, pp. 441 – 456, 2005.

[8] B. Qi, M. Ghazal, and A. Amer, "Robust global motion estimation oriented to video object segmentation," *IEEE Trans. on Image Processing*, vol. 17, no. 6, pp. 958–967, 2008.

[9] H.-Y. Wu, M. Rubinstein, E. Shih, J. V. Guttag, F. Durand, and W. T. Freeman, "Eulerian video magnification for revealing subtle changes in the world." *ACM Trans. Graph.*, vol. 31, no. 4, p. 65, 2012.

[10] D. Sun, S. Roth, and M. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int'l J. of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2014.

[11] Y. Furukawa, A. Sethi, J. Ponce, and D. Kriegman, "Structure and motion from images of smooth textureless objects," in *Proc. European Conf. on Computer Vision*. Springer, 2004, pp. 287–298.

[12] Z. Ambadar, J. W. Schooler, and J. Cohn, "Deciphering the enigmatic face: The importance of facial dynamics in interpreting subtle facial expressions," *Psychological Science*, vol. 16, no. 5, pp. 403–410, 2005.

[13] B. A. Wandell, *Foundations of vision*. Sinauer Associates, 1995.

[14] E. H. Adelson and J. R. Bergen, "Spatio-temporal energy models for the perception of motion," *The J. of the Optical Society of America*, vol. 2, no. 2, pp. 284–299, 1985.

[15] X. Xiong and K. Qin, "Linearly estimating all parameters of affine motion using Radon transform," *IEEE Trans. on Image Processing*, vol. 23, no. 10, pp. 4311–4321, Oct 2014.

[16] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. European Conf. Computer Vision*, 2006, pp. 404–417.

[19] M. Okade and P. K. Biswas, "Video stabilization using maximally stable extremal region features," *Multimedia Tools and Applications*, vol. 68, no. 3, pp. 947–968, 2014.

[20] Y. G. Ryu and M. J. Chung, "Robust online digital image stabilization based on point-feature trajectory without accumulative global motion estimation," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 223–226, 2012.

[21] A. Smolić, M. Hoeynck, and J.-R. Ohm, "Low-complexity global motion estimation from p-frame motion vectors for MPEG-7 applications," in *Proc. of Int'l Conf. on Image Processing*, vol. 2, 2000, pp. 271–274.

[22] Y.-M. Chen and I. Bajic, "Motion vector outlier rejection cascade for global motion estimation," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 197–200, 2010.

[23] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 232–242, 2005.

[24] B. K. Horn and B. G. Schunck, "Determining optical flow," in *1981 Technical symposium east*. Int'l Society for Optics and Photonics, 1981, pp. 319–331.

[25] M. A. Gennert and S. Negahdaripour, "Relaxing the brightness constancy assumption in computing optical flow," *M.I.T AI Memos*, June 1987.

[26] Y.-H. Kim, A. M. Martinez, and A. C. Kak, "Robust motion estimation under varying illumination," *Image and Vision Computing*, vol. 23, no. 4, pp. 365 – 375, 2005.

[27] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2432–2439.

[28] G. Tzimiropoulos, V. Argyriou, and T. Stathaki, "Subpixel registration with gradient correlation," *IEEE Trans. on Image Processing*, vol. 20, no. 6, pp. 1761–1767, 2011.

[29] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki, "Robust FFT-based scale-invariant image registration with image gradients," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1899–1906, 2010.

[30] W. Pan, K. Qin, and Y. Chen, "An adaptable-multilayer fractional fourier transform approach for image registration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 400–414, March 2009.

[31] V. J. Traver and F. Pla, "Motion analysis with the radon transform on log-polar images," *J. of Mathematical Imaging and Vision*, vol. 30, no. 2, pp. 147–165, 2008.

[32] S. Kumar, H. Azartash, M. Biswas, and T. Nguyen, "Real-time affine global motion estimation using phase correlation and its application for digital image stabilization," *IEEE Trans. Image Processing,*, vol. 20, no. 12, pp. 3406–3418, Dec 2011.

[33] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Trans. on Image Processing*, vol. 9, no. 3, pp. 497–501, 2000.

[34] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int'l J. Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[35] A. Ashraf, S. Lucey, and T. Chen, "Fast image alignment in the fourier domain," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2480–2487.

[36] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Robust and efficient parametric face alignment," in *IEEE Int'l Conf. on Computer Vision*. IEEE, 2011, pp. 1847–1854.

[37] Y.-M. Chen and I. V. Bajic, "A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1316–1328, 2011.

[38] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. on Communications*, vol. 41, no. 4, pp. 604–612, 1993.

[39] M. J. Black, "Optical flow software," last accessed on Jul 01, 2015. [Online]. Available: http://cs.brown.edu/~black/code.html

[40] X. Qian, *Global Motion Estimation and Its Applications*. INTECH Open Access Publisher, 2012.

[41] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.

[42] S. Lucey, R. Navarathna, A. B. Ashraf, and S. Sridharan, "Fourier lucas-kanade algorithm," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1383–1396, 2013.

[43] N. Dowson and R. Bowden, "Mutual information for Lucas-Kanade tracking (MILK): An inverse compositional formulation," *IEEE Trans. on Pattern Analysis & Machine Intelligence*, no. 1, pp. 180–185, 2007.

[44] J.-K. Kamarainen, V. Kyrki, and H. Kalviainen, "Noise tolerant object recognition using Gabor filtering," in *Int'l Conf. Digital Signal Processing*, vol. 2, 2002, pp. 1349–1352.

[45] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[46] N. Petkov and E. Subramanian, "Motion detection, noise reduction, texture suppression, and contour enhancement by spatiotemporal Gabor filters with surround inhibition," *Biological Cybernetics*, vol. 97, no. 5-6, pp. 423–439, 2007.

[47] D. J. Heeger, "Normalization of cell responses in cat striate cortex," *Visual neuroscience*, vol. 9, no. 02, pp. 181–197, 1992.

[48] N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard?" *PLoS computational biology*, vol. 4, no. 1, p. e27, 2008.

[49] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int'l Symposium on Circuits and Systems*, 2010, pp. 253–256.

[50] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. European Conf. Computer Vision*, 2010, pp. 140–153.

[51] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief net-

works," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[52] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. Int'l Conf. Machine Learning*, 2010, pp. 111–118.

[53] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. of Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[54] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[55] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.

[56] I. Nabney, *NETLAB: algorithms for pattern recognition*. Springer Science & Business Media, 2002.

[57] F. C. Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 207–212, 1984.

[58] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometry consistency for large scale image search-extended version," in *Proc. European Conf. Computer Vision*, 2008.

[59] M. Tok, A. Glantz, A. Krutz, and T. Sikora, "Monte-carlo-based parametric motion estimation using a hybrid model approach," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 607–620, 2013.

[60] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic, "Generic active appearance models revisited," in *Proc. Asian Conf. on Computer Vision*, 2013, pp. 650–663.

[61] A. Rahman, D. Houzet, D. Pellerin, S. Marat, and N. Guyader, "Parallel implementation of a spatio-temporal visual saliency model," *Journal of Real-Time Image Processing*, vol. 6, no. 1, pp. 3–14, 2011.

[62] M. Hansard and R. Horaud, "A differential model of the complex cell," *Neural computation*, vol. 23, no. 9, pp. 2324–2357, 2011.