Probabilistic Subpixel Temporal Registration for Facial Expression Analysis

Evangels Sariyanidi, Hatice Gunes and Andrea Cavallaro

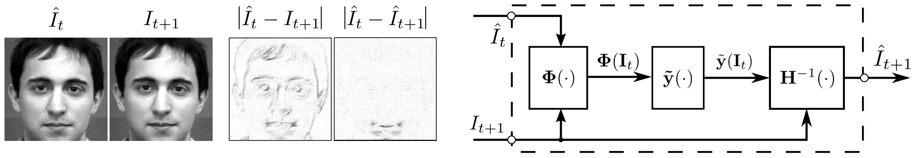
Queen Mary University of London,
Centre for Intelligent Sensing
{e.sariyanidi, h.gunes, a.cavallaro}@qmul.ac.uk

Abstract. Face images in a video sequence should be registered accurately before any analysis, otherwise registration errors may be interpreted as facial activity. Subpixel accuracy is crucial for the analysis of subtle actions. In this paper we present PSTR (Probabilistic Subpixel Temporal Registration), a framework that achieves high registration accuracy. Inspired by the human vision system, we develop a motion representation that measures registration errors among subsequent frames, a probabilistic model that learns the registration errors from the proposed motion representation, and an iterative registration scheme that identifies registration failures thus making PSTR aware of its errors. We evaluate PSTR's temporal registration accuracy on facial action and expression datasets, and demonstrate its ability to generalise to naturalistic data even when trained with controlled data.

1 Introduction

The automatic recognition of facial actions, activity and expressions is a fundamental building block for intelligent and assistive technologies for various domains including healthcare (*e.g.* pain analysis), driving (*e.g.* drowsiness detection), lip reading, animation (*e.g.* facial action synthesis) and social robotics [1, 2]. Inaccurate temporal registration of face images is detrimental to facial action and expression analysis as local intensity and texture variations introduced by registration errors can be interpreted as facial activity [3]. Even small errors of 0.5 pixels can cause a larger variation than the one caused by facial actions (see Fig.1a,b). Registration errors have an adverse effect on other components of the systems that analyse facial activity in various contexts such as AU detection [4] and basic emotion recognition ([5] vs. [6]).

Facial expression recognisers [7, 8, 5, 9, 3, 10, 11] rely on *spatial* registration techniques, which ignore the consistency among subsequent video frames as they register each frame independently. The common approach is to register faces based on a set of facial landmarks. However, state-of-the-art landmark detectors cannot achieve subpixel accuracy [12, 13] and therefore subsequent frames cannot be registered with respect to each other. One deviation from the literature is the work of Jiang *et al.* [14], which crops the first frame based on landmarks



(a) (b) (c)

Fig. 1. (a) Two consecutive unregistered images $(\hat{I}_t, I_{t+1})$ with registration error $t_x = t_y = 0.5$ pixels. (b) Difference between the images of the pair in (a) before $(|\hat{I}_t, I_{t+1}|)$ and after $(|\hat{I}_t, \hat{I}_{t+1}|)$ registration. (c) Illustration of how the proposed framework registers two consecutive frames.

and registers subsequent frames to the first frame using Robust FFT [15]. Although Robust FFT can maintain high registration accuracy, particularly for large registration errors, it does not achieve the desired subpixel accuracy.

We aim at providing accurate registration for spatio-temporal facial expression analysis. In particular, we consider registration via *homographic* transformation for suppressing the registration errors that occur due to rigid head or body movements, or the errors induced when cropping a face after face detection. We attribute local non-rigid registration errors to facial actions, and therefore leave these errors intact in order to enable their analysis in subsequent system layers (*e.g.* facial representation and classification). Specifically, we are interested in Euclidean registration as more general homographic transformations such as projective or affine transformation, do not necessarily preserve the shape of the face and can introduce distortions that alter the facial display.

In this paper we propose a Probabilistic Subpixel Temporal Registration (PSTR) framework that achieves high registration accuracy for Euclidean face registration. Influenced by the studies on motion perception [16], we propose a *motion representation* to implicitly encode the registration errors in a sequence. We then develop a *supervised probabilistic model* that takes the motion representation and estimates the registration errors in a sequence using the information encoded in the representation. We finally develop an *iterative registration framework* that has the supervised probabilistic model in its core. This framework formulates registration as an optimisation problem, and relies on the probabilistic nature of the supervised model to achieve convergence and terminate the optimisation. The framework benefits further from the probabilistic nature of the model and identifies its own errors.

The contribution of this work is three-fold: the development of (i) a motion representation that is robust to illumination variations (Section 3), (ii) a probabilistic model that learns the relationships between the features of motion representation and the corresponding registration errors (Section 4), and (iii) a registration error estimator which enables PSTR to detect its own errors (Section 5).

2 Formulation

Let $\mathbf{S} = (I_1, I_2, \ldots, I_T)$ be a video sequence where $T \in \mathbb{N}^+$ and $I_1, I_2, \ldots, I_T$ are the consecutive frames. Our goal is to obtain a registered sequence $\hat{\mathbf{S}} = (\hat{I}_1, \hat{I}_2, \ldots, \hat{I}_T)$, *i.e.* a sequence where any two images $\hat{I}_i, \hat{I}_j$ are registered with respect to each other. To achieve this, we aim to perform pairwise registration among all consecutive image pairs in $\mathbf{S}$ starting from the first pair. We consider the first image I_1 as the reference image, denote it with $\hat{I}_1$ and register I_2 to $\hat{I}_1$. In general, $\mathbf{I}_t$ denotes a pair of images consisting of a reference (registered) image and an unregistered image as $\mathbf{I}_t = (\hat{I}_t, I_{t+1})$. The registration is performed for all pairs $\mathbf{I}_t$ for $t = 1, \ldots, T-1$.

The registration of a pair $\mathbf{I}_t$ is illustrated in Fig. 1c. Firstly, the motion representation $\mathbf{\Phi}(\cdot)$ is extracted from the images in $\mathbf{I}_t$. Then, the features $\mathbf{\Phi}(\mathbf{I}_t)$ are fed into the registration error estimator $\tilde{\mathbf{y}}(\cdot)$. Finally, the estimated errors $\tilde{\mathbf{y}}(\mathbf{I}_t)$ and the unregistered image I_{t+1} are passed to a homographic back-transformation $\mathbf{H}^{-1}(\cdot)$, which outputs the registered image $\hat{I}_{t+1}$.

3 Motion Representation

Our work is influenced by the biology literature that studies motion perception [17, 16], that is, the ability of inferring the speed and direction of objects in a dynamic scene. The main idea is to consider the registration errors among subsequent frames as a source of *motion*, and to discover this motion using motion perception models. Many motion perception models are developed by analysing the motion of a moving line [16, 17]. Adelson and Bergen [16] showed that convolution with an appropriately designed spatio-temporal Gabor filter pair can be used to discover the speed and orientation of a moving line.

We first discuss how a Gabor filter pair can be used to identify the speed and orientation of a moving pattern. We then describe how to extract Gabor features that are robust to illumination variations. We finally develop a *motion representation* that extracts features using multiple Gabor filter pairs.

3.1 Gabor Motion Energy

Let us denote a $2D$ moving line with $f_l(x, y, t)$:

$$f_l(x, y, t) = c\delta\left(x \cos\theta_l - y \sin\theta_l - t\upsilon_l\right), \tag{1}$$

where θ_l defines the spatial orientation of f_l as well as the direction of motion; υ_l defines the speed and c controls the luminance value of the line.

A $3D$ Gabor filter can be represented as in [18] (see the reference for a detailed discussion on parameters):

$$g_\phi(x, y, t) = \frac{\gamma}{2\pi\sqrt{2\pi}\sigma^2\tau} e^{\left(-\frac{\bar{x}^2+\gamma\bar{y}^2}{2\sigma^2} - \frac{(t-\mu_t)^2}{2\tau^2}\right)} \cos\left(\frac{2\pi}{\lambda}(\bar{x} + \upsilon_g t + \phi)\right) \tag{2}$$

where $\bar{x} = x\cos(\theta_g) + y\sin(\theta_g)$ and $\bar{y} = -x\sin(\theta_g) + y\cos(\theta_g)$. The parameters θ_g and υ_g define the orientation and speed of motion that the filter is tuned for. The parameter ϕ is the phase offset of the filter. It can be set to $\phi = 0$ to obtain an even-phased (cosine) filter g^e and $\phi = \frac{\pi}{2}$ to obtain an odd-phased (sine) filter g^o — the two filters together form a quadrature pair (g^e, g^o).

The convolution $f_l * g_\phi$ provides useful information towards understanding the motion of the line [16]. This can be illustrated for the $2D$ line f_l as follows. When a vertical bar $(\theta_l \approx \pi/2)$ moves with a speed $\upsilon_l = \upsilon_g > 0$, the convolution response gets maximal for $\theta_g = \theta_l$ and strictly smaller as $\theta_g \to -\pi/2$. The response is almost flat when $\theta_g = -\theta_l$. This behaviour is useful as it provides information about the speed and orientation of the motion, and can discriminate between forward and backward motion, *i.e.* it is selective in terms of direction as it yields no output for motion in opposite direction.

Although the convolution $f_l * g_\phi$ helps identifying the speed and orientation of the line, it also poses some difficulties [16]. Firstly, the convolution $f_l * g_\phi$ yields an oscillating output due to the trigonometric $\cos(\cdot)$ function, therefore it is hard to derive a meaningful conclusion by looking at a particular part of the response. Secondly, the convolution output is sensitive to luminance polarity, *i.e.* the response would change if we would invert the luminance of the bar [16]. To deal with these shortcomings, Adelson and Bergen [16] suggested to use *motion energy*, which is defined as:

$$E_{f,\upsilon_g,\theta_g}(x,y,t) = (f * g^e)^2 + (f * g^o)^2. \tag{3}$$

Instead of oscillating, the energy $E_f = E_{f,\upsilon_g,\theta_g}$ generates a uniform peak at the points where the line sits at any given time t. Furthermore, E_f is insensitive to luminance polarity, *i.e.* the response is not affected if we were to invert the luminance of the line with the background [16].

3.2 Pooling

The $3D$ convolution involved in the computation of E_f can yield a high dimensional output. This dimensionality must be reduced to improve computational performance and avoid the curse of dimensionality [19]. To this end, we perform pooling, which proved to be a biologically plausible [20–22] and computationally efficient [23] approach. We use two types of pooling, namely mean and maximum (max) pooling, denoted respectively with $\phi_f^\mu = \phi^\mu(\mathbf{E}_f)$ and $\phi_f^\cap = \phi^\cap(\mathbf{E}_f)$, where $\mathbf{E}_f$ is the volume of energy obtained by computing E_f for all $(x,y,t) \in \mathbf{\Omega}$ where $\mathbf{\Omega} = X \times Y \times T$ is the domain of the sequence f. We add another statistical descriptor, the standard deviation $\phi_f^\sigma = \phi^\sigma(\mathbf{E}_f)$. The three features can be computed as follows:

$$\phi_f^\mu = \frac{1}{|\mathbf{\Omega}|}\int\limits_{\mathbf{\Omega}} E_f(\mathbf{x})\mathrm{d}\mathbf{x}, \quad \phi_f^\cap = \max_{\mathbf{x}\in\mathbf{\Omega}} E_f(\mathbf{x}), \quad \phi_f^\sigma = \sqrt{var(\mathbf{E}_f)} \tag{4}$$

where $|\mathbf{\Omega}|$ denotes the volume of $\mathbf{\Omega}$ and $\mathbf{x} = (x,y,t)$ is a point in space-time.

3.3 Contrast Normalisation

The energy E_f is sensitive to the average intensity value of f as Gabor filters are not zero mean [24]. Therefore, a contrast normalisation is essential for increasing the generalisation ability of the Gabor features.

Let $\mathbf{I} = \mathbf{I}(\mathbf{x}) = \mathbf{I}(x, y, t)$ be a sequence of a moving pattern. Consider two sequences $\mathbf{I}_i(\mathbf{x}), \mathbf{I}_j(\mathbf{x})$ which contain the same moving pattern as in $\mathbf{I}(\mathbf{x})$ but differ from $\mathbf{I}(\mathbf{x})$ with a linear illumination variation such as $\mathbf{I}_i(\mathbf{x}) = (\alpha_i t + \beta_i)\mathbf{I}(\mathbf{x})$, and $\mathbf{I}_j(\mathbf{x}) = (\alpha_j t + \beta_j)\mathbf{I}(\mathbf{x})$. Ideally, we would desire the features extracted for both patterns to be identical, *i.e.* $\phi(\mathbf{I}_i) = \phi(\mathbf{I}_j)$. To map the features of $\phi(\mathbf{I}_i)$ and $\phi(\mathbf{I}_j)$ close together, we perform normalisation. On the one hand, if normalisation is performed on individual images (*e.g.* z-normalisation, contrast-stretching, histogram equalisation) apparent motion along the sequence can be generated. On the other hand, a normalisation performed on the entire sequence may not necessarily map the features $\phi(\mathbf{I}_i)$ and $\phi(\mathbf{I}_j)$ close to one another. To overcome such problems, we define a new energy function, the *normalised energy* $\tilde{E}$. Normalisation is achieved by dividing each frame in an input sequence with a coefficient that is proportional to the illumination coefficient in the frame.

Let $I_i^{t_k}$ be an image from the sequence $\mathbf{I}_i$ at any fixed time t_k. We use the image $I_i^{t_k}$ to synthesise a static sequence $\mathbf{I}_i^{t_k}$ of length $(t_f - t_0)$ by repeating the same image throughout the time interval, *i.e.* $\mathbf{I}_i^{t_k}(\mathbf{x}) \equiv \mathbf{I}_i(x, y, t_k) = (\alpha_i t_k + \beta_i)\mathbf{I}(x, y, t_k) = (\alpha_i t_k + \beta_i)\mathbf{I}^{t_k}(\mathbf{x})$. We can compute the energy of $\mathbf{I}_i^{t_k}$ as follows:

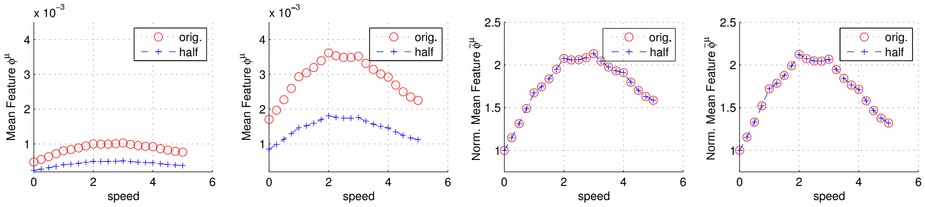
$$\begin{aligned}
E_{\mathbf{I}_i^{t_k}}(\mathbf{x}) &= \left[\int (\alpha_i t_k + \beta_i)\mathbf{I}^{t_k}(\mathbf{x}-\mathbf{u})g^e(\mathbf{u})\mathrm{d}\mathbf{u}\right]^2 + \left[\int (\alpha_i t_k + \beta_i)\mathbf{I}^{t_k}(\mathbf{x}-\mathbf{u})g^o(\mathbf{u})\mathrm{d}\mathbf{u}\right]^2 \\
&= (\alpha_i t_k + \beta_i)^2 \left\{\left[\int \mathbf{I}^{t_k}(\mathbf{x}-\mathbf{u})g^e(\mathbf{u})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \mathbf{I}^{t_k}(\mathbf{x}-\mathbf{u})g^o(\mathbf{u})\mathrm{d}\mathbf{u}\right]^2\right\} \\
&= (\alpha_i t_k + \beta_i)^2 E_{\mathbf{I}^{t_k}}(\mathbf{x}) \qquad (5)
\end{aligned}$$

where $\mathbf{u} = \begin{bmatrix} u \; v \; w \end{bmatrix}$ is the convolution variable. Since for fixed t_k the term $\alpha_i t_k + \beta_i$ is constant, a feature $\phi_{\mathbf{I}_i^{t_k}}$ for $\phi \in \{\phi^{\cap}, \phi^{\mu}, \phi^{\sigma}\}$ can be computed through (4) as:

$$\phi_{\mathbf{I}_i^{t_k}} = \phi(\mathbf{E}_{\mathbf{I}_i^{t_k}}) = (\alpha_i t_k + \beta_i)^2 \phi(\mathbf{E}_{\mathbf{I}^{t_k}}) = (\alpha_i t_k + \beta_i)^2 \phi_{\mathbf{I}^{t_k}}. \qquad (6)$$

Note that (6) includes $(\alpha_i t_k + \beta_i)$, which will cancel out the illumination term in the input sequence. Let us define the normalised energy $\tilde{E}$ for $\mathbf{I}_i$ as:

$$\begin{aligned}
\tilde{E}_{\mathbf{I}_i}(\mathbf{x}) &= \left[\frac{\mathbf{I}_i}{(\phi_{\mathbf{I}_i^t})^{\frac{1}{2}}} * g^e\right]^2 + \left[\frac{\mathbf{I}_i}{(\phi_{\mathbf{I}_i^t})^{\frac{1}{2}}} * g^o\right]^2 \\
&= \left[\int \frac{(\alpha_i w + \beta_i)\mathbf{I}(\mathbf{u})}{(\alpha_i w + \beta_i)(\phi_{\mathbf{I}^w})^{\frac{1}{2}}} g^e(\mathbf{x}-\mathbf{u})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{(\alpha_i w + \beta_i)\mathbf{I}(\mathbf{u})}{(\alpha_i w + \beta_i)(\phi_{\mathbf{I}^w})^{\frac{1}{2}}} g^o(\mathbf{x}-\mathbf{u})\mathrm{d}\mathbf{u}\right]^2 \\
&= \left[\int \frac{\mathbf{I}(\mathbf{u})}{(\phi_{\mathbf{I}^w})^{\frac{1}{2}}} g^e(\mathbf{x}-\mathbf{u})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}(\mathbf{u})}{(\phi_{\mathbf{I}^w})^{\frac{1}{2}}} g^o(\mathbf{x}-\mathbf{u})\mathrm{d}\mathbf{u}\right]^2. \qquad (7)
\end{aligned}$$



(a) S1, un-normal. (b) S2, un-normal. (c) S1, normalised (d) S2, normalised

Fig. 2. The un-normalised (ϕ^{μ}) and normalised (ϕ^{μ}) mean features extracted from moving face sequences of two subjects (S1, S2) for varying speeds.

The illumination coefficients are cancelled out by dividing each frame I_i^t in $\mathbf{I}_i$ with the feature of the synthesised sequence $\mathbf{I}_i^t$, *i.e.* $\phi_{\mathbf{I}_i^t}$. Based on the normalised energy, we define the normalised features $\tilde{\phi}^{\mu}, \tilde{\phi}^{\frown}, \tilde{\phi}^{\sigma}$ as:

$$\tilde{\phi}_f^{\mu} = \phi^{\mu}(\tilde{\mathbf{E}}_f), \quad \tilde{\phi}_f^{\frown} = \phi^{\frown}(\tilde{\mathbf{E}}_f), \quad \tilde{\phi}_f^{\sigma} = \phi^{\sigma}(\tilde{\mathbf{E}}_f) \tag{8}$$

where $\tilde{\mathbf{E}}_f$ is the normalised energy volume $\tilde{E}$ computed for a $\mathbf{\Omega} = X \times Y \times T$.

A prominent illumination issue is gray-scale shift (*e.g.* due to imaging conditions or skin color differences), *i.e.* $\alpha_i = 0, \beta_i \neq 0$. The effect of normalisation against gray-scale shift is shown in Fig. 2. Moving sequences of various speeds are synthesised from the faces of two subjects (S1, S2). Each plot displays the variation of features with respect to the motion speed. The features of each sequence are computed for two cases: (1) original intensities (orig.) and (2) intensities multiplied with 0.5 (half). The un-normalised features $\phi(\cdot)$ are affected by both gray-scale variation (Fig. 2a and 2b) and inter-personal variation (Fig. 2a vs. 2b), whereas normalised features $\tilde{\phi}(\cdot)$ not only suppress gray scale shift completely (Fig. 2a and 2b), but also map the features of different subjects closer (Fig. 2c vs. 2d).

3.4 Motion in Various Speeds and Orientations

Although $E_{\upsilon,\theta}$ (or $\tilde{E}_{\upsilon,\theta}$) identifies the motion that it is tuned for, it cannot directly identify various speeds and orientations. For this reason, we construct a Gabor filter bank with filter pairs tuned to various speeds and orientations: $\mathbf{G} = \{(g^e_{\upsilon_i,\theta_j}, g^o_{\upsilon_i,\theta_j}) : \upsilon_i \in \{\upsilon_1, \ldots, \upsilon_{K_\upsilon}\}, \theta_j \in \{\theta_1, \ldots, \theta_{K_\theta}\}\}$.

The feature vector for a pair of consecutive images $\mathbf{I}$ is computed as follows. Firstly, $\tilde{\mathbf{E}}_{\upsilon_i,\theta_j}$ is computed for each pair g_{υ_i,θ_j} in $\mathbf{G}$. Secondly, each $\tilde{\mathbf{E}}_{\upsilon_i,\theta_j}$ is partitioned into spatio-temporal slices $\tilde{\mathbf{E}}^{m,n}_{\upsilon_i,\theta_j}$. Next, the normalised feature of each slice $\tilde{\phi} = \phi(\tilde{\mathbf{E}}^{m,n}_{\upsilon_i,\theta_j})$ is computed for a single $\phi \in \{\phi^{\mu}, \phi^{\frown}, \phi^{\sigma}\}$ (*i.e.* a motion representations consists of only one feature type). Finally, the feature vector is obtained by concatenating all features $\tilde{\phi}$ computed for positive integers i, j, m, n such that $i \leqslant K_\upsilon, j \leqslant K_\theta, m \leqslant M, n \leqslant N$.

In the following sections, we will denote each feature with $\phi_k = \phi(\tilde{\mathbf{E}}^{m,n}_{\upsilon_i,\theta_j})$ and the final feature vector with $\mathbf{\Phi}(\mathbf{I}) = \left[\phi_1 \ldots \phi_k \ldots \phi_K\right]$ where K is the size of the feature vector and k the feature index that can be computed as $k = (m-1)M + (n-1)N + (i-1)\,K_\upsilon + (j-1)\,K_\theta + 1$.

4 Estimating Registration Errors

To model the relations between the features $\mathbf{\Phi}(\cdot)$ and corresponding registration errors, we use a *discrete* probabilistic model. A continuous model would require an assumption over the distribution of the features (*e.g.* Poisson, Gaussian), whereas the discrete model is trained straight from data without any assumption. Also, the proposed model can be trained in a single iteration and does not require the optimisation of parameters that would risk overfitting to a dataset.

4.1 Labeling

Since the probabilistic model we use is discrete, we define our labels to be also discrete. The misalignment between the images of a pair $\mathbf{y}(\mathbf{I})$ is defined as $\mathbf{y}(\mathbf{I}) = (\delta t_x, \delta t_y, \delta s, \delta\theta)$ where $\delta t_x, \delta t_y, \delta s$ and $\delta\theta$ are respectively the horizontal translation, vertical translation, scaling and rotation difference between the images of $\mathbf{I}$. We define $\mathit{\Delta} t_x, \mathit{\Delta} t_y, \mathit{\Delta} s$ and $\mathit{\Delta}\theta$, the sets that represent the range of each variation as $\mathit{\Delta} t_x = \{\delta t_x^-, \delta t_x^- + dt_x, \ldots, \delta t_x^+\}$, $\mathit{\Delta} t_y = \{\delta t_y^-, \delta t_y^- + dt_y, \ldots, \delta t_y^+\}$, $\mathit{\Delta} s = \{\delta s^-, \delta s^- + ds, \ldots, \delta s^+\}$ and $\mathit{\Delta}\theta = \{\delta\theta^-, \delta\theta^- + d\theta, \ldots, \delta\theta^+\}$. The first (*e.g.* δt_x^-) and last elements (*e.g.* δt_x^+) in each set represent the minimum and maximum value for each variation, and the increment values dt_x, dt_y, ds and $d\theta$ the difference between successive labels (*i.e.* the resolution of our labels). The set of all registration errors that our framework will deal with is referred to as the *label space* $\mathcal{L}$ and is defined as $\mathcal{L} = \mathit{\Delta} t_x \times \mathit{\Delta} t_y \times \mathit{\Delta} s \times \mathit{\Delta}\theta$.

Since we use a supervised model, we need training samples (pairs $\mathbf{I}^j$) and labels (registration errors $\mathbf{y}^j = \mathbf{y}(\mathbf{I}_j)$). Let $\mathcal{X}$ be a set containing N samples $\mathcal{X} = \{\mathbf{I}^1, \ldots, \mathbf{I}^N\}$, $\mathit{\Phi}$ be the set of features $\mathit{\Phi} = \{\mathbf{\Phi}^1, \ldots, \mathbf{\Phi}^N\}$ where $\mathbf{\Phi}^j = \mathbf{\Phi}(\mathbf{I}^j)$ and $\mathcal{Y}$ the set that contains the labels $\mathcal{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^N\}$ where $\mathbf{y}^j \in \mathcal{L}$. A practical issue that needs to be addressed is how the samples $\mathbf{I}^j$ and their labels $\mathbf{y}^j$ will be obtained. Suppose that we have face sequences where we know that the subject does not display any head or body motion. Then, if we define a fixed face rectangle and crop the entire sequence based on this rectangle, the cropped sequence will contain only facial activity and no registration errors. To obtain one training sample $\mathbf{I}^j$, we firstly pick any two consecutive frames from the cropped sequence. Next, we apply a random Euclidean transformation to both frames. The label $\mathbf{y}^j$ can be easily computed from the random transformation. By picking frames that are temporally farther (rather than consecutive pairs), we can obtain pairs that involve larger facial activity and train a system that is more robust to large facial activity. Thus, using a number of face sequences, we can automatically synthesize as many training samples as we need.

4.2 Modeling

To model the relationships between the features extracted from the pairs $\mathbf{\Phi}^j$ and the corresponding registration errors $\mathbf{y}^j$, we define two discrete random variables $\mathbf{X}$ (for $\mathbf{\Phi}^j$) and $\mathbf{Y}$ (for $\mathbf{y}^j$).

Since $\mathbf{X}$ is discrete, we need to discretise the continuous feature vectors $\mathbf{\Phi}^j$. To this end we perform uniform quantisation over all features $\phi_k^j \in \mathbf{\Phi}^j$. We divide the space $[0, 1]$ into k bins and map each ϕ_k^j to an integer q such as $q = 1, 2, \ldots, Q$. Before this mapping, we normalise ϕ_k^j to map onto $[0, 1]$. The normalisation is based on the training dataset, specifically to the maximum and minimum values of each feature k. Let $\min(\phi_k)$ and $\max(\phi_k)$ be defined as $\min(\phi_k) = \min\{\phi_k^p \in \mathbf{\Phi}^p : \mathbf{\Phi}^p \in \mathit{\Phi}\}$ and $\max(\phi_k) = \max\{\phi_k^p \in \mathbf{\Phi}^p : \mathbf{\Phi}^p \in \mathit{\Phi}\}$. We denote the bin index of each feature ϕ_k^j with q_k^j and compute it as follows:

$$q_k^j = \arg_q \min \left\{ \left| \frac{\phi_k^j - \min(\phi_k)}{\max(\phi_k) - \min(\phi_k)} - \left(\frac{3q}{2} - 1\right) \right| : q = 1, \ldots, Q \right\}, \qquad (9)$$

where $\frac{3q}{2} - 1$ is the center of the bin with index q and $|\cdot|$ is the L_1 metric. We shall denote the quantised vector of all the features in $\mathbf{\Phi}^j$ with $\mathbf{q}^j = \mathbf{q}(\mathbf{I}^j) = (q_1^j, q_2^j, \ldots, q_K^j)$, and the set that contains the quantised vectors extracted from all of the training samples in $\mathcal{X}$ with $\mathcal{Q} = \{\mathbf{q}^1, \ldots, \mathbf{q}^N\}$.

The random variable $\mathbf{X} = (X_1, \ldots, X_K)$ takes on values $\mathbf{q} = (q_1, \ldots, q_K)$ and $\mathbf{Y}$ takes on values $\mathbf{y} \in \mathcal{L}$. The registration errors and the Gabor features of image pairs are modelled jointly by computing the joint distribution $\mathbf{P}(\mathbf{X} = \mathbf{q}, \mathbf{Y} = \mathbf{y})$. For computational simplicity, we rely on the naive Bayes assumption and compute the joint distribution as follows:

$$\begin{aligned} \mathbf{P}(\mathbf{X} = \mathbf{q}, \mathbf{Y} = \mathbf{y}) &= \mathbf{P}(X_1 = q_1, \ldots, X_K = q_K \mid \mathbf{Y} = \mathbf{y})\mathbf{P}(\mathbf{Y} = \mathbf{y}) \\ &\approx \mathbf{P}(\mathbf{Y} = \mathbf{y}) \prod_{i=1}^{K} \mathbf{P}(X_k = q_k \mid \mathbf{Y} = \mathbf{y}). \end{aligned} \qquad (10)$$

To compute this distribution, we must compute the individual likelihood functions $\mathbf{P}(\mathbf{Y} = \mathbf{y} \mid X_k = q_k)$ for each $\mathbf{y} \in \mathcal{L}$. To this end, we adopt the frequency interpretation of probability and learn each likelihood function from the training samples. Let $\mathcal{U}$ and $\mathcal{V}$ be two sets defined respectively as $\mathcal{U} = \{q_k^j, \mathbf{y}^j : \mathbf{y} = \mathbf{y}^j \wedge q_k = q_k^j, \mathbf{y}^j \in \mathcal{Y}, q_k^j \in \mathbf{q}^j \in \mathcal{Q}\}$ and $\mathcal{V} = \{\mathbf{y}^j : \mathbf{y} = \mathbf{y}^j, \mathbf{y}^j \in \mathcal{Y}\}$. The likelihood can be computed as :

$$\mathbf{P}(X_k = q_k \mid \mathbf{Y} = \mathbf{y}) = \frac{|\mathcal{U}|}{|\mathcal{V}|}, \qquad (11)$$

where $|\cdot|$ is the cardinality of the set. We assume the priors to be uniform $\mathbf{P}(\mathbf{Y} = \mathbf{y}) = 1/|\{\mathcal{L}\}|$ for each $\mathbf{y} \in \mathcal{L}$.

4.3 Estimation

Once we learn the model $\mathbf{P}(\mathbf{X}, \mathbf{Y})$, the task of estimating the misalignment in a given image pair $\mathbf{I}$ is fairly straightforward. We rely on Bayesian inference and

Algorithm 1 Estimating registration errors between two images

```
Input Unregistered pair $\mathbf{I} = (\hat{I}, I')$
Output Registration error estimation $\tilde{\mathbf{y}}^*$

$\tilde{\mathbf{y}}_1 \leftarrow \tilde{\mathbf{y}}(\mathbf{I})$; $\tilde{\mathbf{y}}^* \leftarrow \tilde{\mathbf{y}}_1$; $\tilde{I}_1 \leftarrow \mathbf{H}^{-1}(\tilde{\mathbf{y}}^*)I'$        ▷ Estimate, Update
for $i \leftarrow 1, T$ do
    if $\tilde{\mathbf{y}}_i = \underline{0}$ then
        return $\tilde{\mathbf{y}}^*$        ▷ Converged
    end if
    $\tilde{\mathbf{y}}_{i+1} \leftarrow \tilde{\mathbf{y}}((\hat{I}, \tilde{I}_i)) \oplus \tilde{\mathbf{y}}^*$; $\tilde{\mathbf{y}}^* \leftarrow \tilde{\mathbf{y}}_{i+1}$; $\tilde{I}_{i+1} \leftarrow \mathbf{H}^{-1}(\tilde{\mathbf{y}}_{i+1})I'$        ▷ Estimate, Update
end for
$i^* \leftarrow \arg_{i \in \{1,\dots,T\}} \max \mathbf{P}_{\underline{0}}((I, \tilde{I}_i))$
return $\tilde{\mathbf{y}}_{i*}$        ▷ Best iteration
```

find the label $\mathbf{y} \in \mathcal{L}$ that maximises the posterior probability:

$$\mathbf{P}(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{q}) = \frac{\mathbf{P}(\mathbf{Y} = \mathbf{y})\mathbf{P}(\mathbf{X} = \mathbf{q} \mid \mathbf{Y} = \mathbf{y})}{\sum\limits_{\mathbf{y}_l \in \mathcal{L}} \mathbf{P}(\mathbf{Y} = \mathbf{y}_l)\mathbf{P}(\mathbf{X} = \mathbf{q} \mid \mathbf{Y} = \mathbf{y}_l)}. \tag{12}$$

The posterior probability is computed for all $\mathbf{y} \in \mathcal{L}$, and the registration error between the images of a pair $\mathbf{I}$ is finally estimated by selecting the label $\mathbf{y} \in \mathcal{L}$ that maximises the above posterior probability as follows:

$$\tilde{\mathbf{y}}(\mathbf{I}) = \arg_{\mathbf{y} \in \mathcal{L}} \max \mathbf{P}(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{q}(\mathbf{I})). \tag{13}$$

5 Registration

Ideally, a single estimation $\tilde{\mathbf{y}}(\cdot)$ of the model $\mathbf{P}(\cdot)$ would be sufficient for registering two images. However, in practice $\tilde{\mathbf{y}}(\cdot)$ may not approximate the actual errors $\mathbf{y}(\cdot)$ with high accuracy in a single estimation, especially for large registration errors. Therefore, we deal with this as an optimisation problem where the output is estimation of the registration error denoted with $\tilde{\mathbf{y}}^*$. Once we compute $\tilde{\mathbf{y}}^*$, we obtain the registered image $\hat{I}'$ through $\hat{I}' = \mathbf{H}^{-1}(\tilde{\mathbf{y}}^*)I'$ where $\mathbf{H}^{-1}$ is a Euclidean back-transformation. To compute $\tilde{\mathbf{y}}^*$, we perform estimation and back-transformation iteratively.

The overall procedure for registering a pair of images is summarised in Algorithm 1 — the $\oplus$ operator is defined for $\mathbf{y}_1, \mathbf{y}_2$ as $\mathbf{y}_1 \oplus \mathbf{y}_2 = (\delta_{x1} + \delta_{x2}, \delta_{y1} + \delta_{y2}, \delta_{s1}\delta_{s2}, \delta_{\theta 1} + \delta_{\theta 2})$. The optimisation terminates either by converging within the allowed number of iterations, or by reaching the maximum number of iterations and returning the error that is the 'closest' to convergence according to the *convergence probability* $\mathbf{P}_{\underline{0}}(\mathbf{I}) = \mathbf{P}(\mathbf{Y} = \underline{0} \mid \mathbf{X} = \mathbf{q}(\mathbf{I}))$. As was illustrated in Fig. 1c, the registration of the entire sequence is performed by registering the pairs $\mathbf{I}_t$ consecutively for all $t = 1, \dots, T-1$.

5.1 Coarse-to-fine Estimation

To achieve high accuracy, we keep the resolution of our label space $\mathcal{L}$ high by selecting small $dt_x, dt_y, ds, d\theta$ values. However, this increases the size of the space $|\mathcal{L}|$. Therefore, we adopt a coarse-to-fine approach that allows us to simultaneously achieve high registration accuracy and keep the label space dimensionality low. We train multiple models $\mathbf{P}^i(\cdot)$ with label spaces $\mathcal{L}_i$, *i.e.* $i = 1, \ldots, K_{\mathcal{L}}$. The spaces are defined from coarse to fine — $\mathcal{L}_1$ is the coarsest and $\mathcal{L}_2, \mathcal{L}_3, \ldots$ are increasingly finer spaces. We cascade the models $\mathbf{P}(\cdot)^i$ and apply Algorithm 1 to each model $\mathbf{P}(\cdot)^i$ sequentially. We obtain the final estimation by accumulating the error estimations of all models $\mathbf{P}(\cdot)^i$.

5.2 Identifying Failure

The convergence probability $\mathbf{P}_{\underline{0}}(\mathbf{I})$ provides the confidence needed to verify whether the two images in $\mathbf{I}$ are registered correctly. To complete the verification, we compare $\mathbf{P}_{\underline{0}}(\mathbf{I})$ with a threshold probability P_θ.

Consider that we have *positive* and *negative* sample pairs — a positive sample is a pair of two correctly registered images and a negative sample is a pair of two unregistered images. The task is to find a threshold probability P_θ that will enable separation with a high true positive rate and a low false positive rate. To this end, we compute the convergence probability $\mathbf{P}_{\underline{\mathbf{0}}}(\cdot)$ for all positive and negative samples.

We then compute a ROC curve by setting the threshold P_θ to various values by incrementing it with a small step size. We set the final threshold P_θ to a value that yields a false positive rate as low as 0.5%. Then the registration of an image pair is verified if $\mathbf{P}_{\underline{0}}(\mathbf{I}) > P_\theta$ or otherwise it is assumed that the images of $\mathbf{I}$ are not registered correctly.

6 Experiments

6.1 Setup and Evaluation Measures

We evaluate PSTR for pair and sequence registration. We test the performance of each feature type in $\{\tilde{\phi}^{\mu}, \tilde{\phi}^{\cap}, \tilde{\phi}^{\sigma}\}$ for parameters $N, M = 2, 3$ (Section 3.4). The Gabor filter bank $\mathbf{G}$ is obtained with filters of 8 orientations and 5 speeds such that $\upsilon_i \in \{1, 2, \ldots, 5\}, \theta_j \in \{0°, 45°, \ldots, 360°\}$. All images are resized to 200×200. The bin number for quantisation Q (Section 4.2) is set to 8 after experimenting with the values $4, 6, 8, \ldots, 20$ and not observing performance gain for more than 8 bins. As shown in Table 1, we train four probabilistic models for different label spaces $\mathcal{L}_i$ (see Section 5). To show that we can increase accuracy through finer labels, we report two results: one obtained by excluding $\mathcal{L}_4$ (*i.e.* selecting $\mathcal{L}_3$ as the finest label space) and one by including $\mathcal{L}_4$.

For *pair registration*, we measure performance using the mean absolute error (MAE) ε^p computed separately for translation ($\varepsilon^p_{t_x}$, $\varepsilon^p_{t_y}$ in pixels), scaling (ε^p_s as a percentage %) and rotation (ε^p_θ in degrees) as follows. Let $\mathbf{I}_i$ be one of the

	$\delta t_x^{-\dagger}$	$\delta t_x^{+\dagger}$	$\delta t_y^{-\dagger}$	$\delta t_y^{+\dagger}$	δs^{-*}	δs^{+*}	$\delta \theta^{-\ddagger}$	$\delta \theta^{+\ddagger}$	$dt_x{}^{\dagger}$	$dt_y{}^{\dagger}$	ds^*	$d\theta^{\ddagger}$
$\mathcal{L}_1$	-12	12	-12	12	0.85	1.15	-15	15	3	3	0.03	3
$\mathcal{L}_2$	-4	4	-4	4	0.94	1.06	-3	3	1	1	0.01	1
$\mathcal{L}_3$	-1.5	1.5	-1.5	1.5	0.99	1.01	-1	1	0.5	0.5	0.002	0.2
$\mathcal{L}_4$	-0.5	0.5	-0.5	0.5	0.998	1.002	-0.2	0.2	0.125	0.125	0.001	0.1

Table 1. Parameters that describe the label spaces $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ and $\mathcal{L}_4$. †Pixels, *percentage ratio, ‡degrees.

pairs, *i.e.* $i = 1, \ldots, N_p$, and $\bar{\delta}^i_{t_x}$ be the horizontal translation error for i^{th} pair. The MAE $\varepsilon^p_{t_x}$ is computed as $\varepsilon^p_{t_x} = \sum_i^{N_p} \bar{\delta}^i_{t_x}/N_p$. The MAEs $\varepsilon^p_{t_y}, \varepsilon^p_s$ and ε^p_θ are computed similarly. We additionally compare PSTR with (Robust) FFT [15] as it is a state-of-the-art registration technique already used for facial expression recognition [14]. PSTR cannot be compared with the registration methods of most facial action analysis systems as they crop faces across an ad-hoc rectangle defined through a number of fiducial points [7]. Similarly to Robust FFT [15], we compare PSTR with RANSAC registration using SURF [25] and MSER [26].

For *sequence registration*, we measure the average MAE over sequences (ε^s) separately for translation ($\varepsilon^s_{t_x}$, $\varepsilon^s_{t_y}$) scaling (ε^s_s) and rotation (ε^s_θ) computed as follows. Let $\mathbf{S}_i$ denote one of the N_s sequences where the length of each sequence is equivalently T. Let $\bar{\delta}^{i,j}_{t_x}$ denote the horizontal translation error of j^{th} pair in i^{th} sequence. The average MAE for horizontal translation $\varepsilon^s_{t_x}$ is computed as $\varepsilon^s_{t_x} = \sum_i^{N_s}(\sum_j^{T-1} \bar{\delta}^i_{t_x}/(T-1))/N_s$. The MAEs $\varepsilon^s_{t_y}, \varepsilon^s_s$ and ε^s_θ are computed similarly.

We use standard datasets for evaluation, namely the CK+, PIE [27] and SEMAINE [28] datasets. The training *for all* the experiments is performed on CK+ dataset. In the CK+ and PIE datasets there exist sequences with almost no head pose variation and body movement. We select 129 such sequences from CK+ dataset, and we use 112 of them for training and the remaining 17 for testing. The 112 training sequences include 1814 consecutive pairs, which are randomly transformed to synthesise as many pairs as needed (as described in Section 4.1). The 17 testing sequences include 244 consecutive pairs of images — random homographic transformations are applied to them to obtain the unregistered pairs and sequences.

To evaluate both the robustness against illumination variation and the usefulness of the failure identification ability of PSTR, we perform experiments on the PIE dataset, which contains rapid illumination variations. We demonstrate performance on 200 pairs obtained from 10 sequences of 10 subjects.

We also test PSTR for naturalistic expressions on the SEMAINE dataset. However, since naturalistic expressions include head/body motion, we are not able to obtain a ground truth for this dataset and therefore provide only qualitative results through a video (Section 6.4).

ϕ	N	$\varepsilon^{p\,\dagger}_{t_x}$		$\varepsilon^{p\,\dagger}_{t_y}$		ε^{p*}_{s}		$\varepsilon^{p\ddagger}_{\theta}$		$\varepsilon^{s\,\dagger}_{t_x}$		$\varepsilon^{s\,\dagger}_{t_y}$		ε^{s*}_{s}		$\varepsilon^{s\ddagger}_{\theta}$	
ϕ	N	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$	$\mathcal{L}_{1\text{-}3}$	$\mathcal{L}_{1\text{-}4}$
ϕ^{μ}	2	.07	.08	.07	.08	.08	.05	.07	.03	$\underline{.31}$	$\underline{.23}$	$\underline{.38}$	.26	.28	.19	$\underline{.18}$	$\underline{.06}$
$\phi^{\cap}$	2	.11	.08	.15	.09	.11	.06	.16	.03	.34	$\underline{.23}$	.46	.26	.31	$\underline{.18}$	.24	.08
ϕ^{σ}	2	$\underline{.05}$	.08	$\underline{.06}$	$\underline{.07}$	.08	.05	.06	$\underline{.02}$	.34	$\underline{.23}$	.43	$\underline{.24}$	$\underline{.23}$	$\underline{.18}$	$\underline{.18}$	$\underline{.06}$
ϕ^{μ}	3	.07	$\underline{.06}$	.08	$\underline{.07}$	.07	$\underline{.04}$	.06	$\underline{.02}$	.50	.60	.56	.79	.28	.78	.24	.25
$\phi^{\cap}$	3	.07	$\underline{.06}$	$\underline{.06}$	$\underline{.07}$	.07	$\underline{.04}$	$\underline{.05}$	$\underline{.02}$	.65	.53	.81	.60	.64	.40	.34	.27
ϕ^{σ}	3	$\underline{.05}$	$\underline{.06}$	$\underline{.06}$	$\underline{.07}$	$\underline{.06}$	$\underline{.04}$	.06	$\underline{.02}$	.55	.54	.59	.56	.36	.41	.22	.28
FFT	–	.18		.26		.57		.17		–	–	–	–	–	–	–	–
SURF	–	.24		.29		.10		.05		–	–	–	–	–	–	–	–
MSER	–	.38		.37		.17		.09		–	–	–	–	–	–	–	–

Table 2. Pair (left of double lines) and sequence (right of double lines) registration performance on CK+ dataset.

Method	$\varepsilon^{p\,\dagger}_{t_x}$	$\varepsilon^{p\,\dagger}_{t_y}$	ε^{p*}_{s}	$\varepsilon^{p\ddagger}_{\theta}$	# Eliminated Pairs
PSTR	0.13	0.11	0.07	0.05	11 (automatically)
FFT	0.29	0.25	0.55	0.16	10 (manually)
SURF	0.75	0.80	0.52	0.29	44 (manually)
MSER	1.78	2.55	1.43	0.95	73 (manually)

Table 3. Pair registration performance with illumination variation (PIE dataset).

6.2 Pair Registration

The translation output of FFT is an integer with 1 pixel resolution. To evaluate subpixel registration performance, we perform registration with FFT at double the image size (400 × 400) and reduce the estimated translation to half, *i.e.* increase the translation resolution of the FFT method to 0.5 pixels. The translation resolution of PSTR is also limited at 0.5 pixels for $\mathcal{L}_3$ (Table 1).

Table 2 shows the pair registration errors of PSTR and the FFT method. PSTR outperforms FFT as well as RANSAC-based registration with SURF or MSER features. The mean (ϕ^{μ}) and standard deviation features (ϕ^{σ}) perform slightly better than max ($\phi^{\cap}$). Increasing the number of pooling regions N does not provide a major performance improvement for ϕ^{μ} and $\phi^{\cap}$, and therefore N can be set to 2 to keep the dimensionality low. Note that we are able to reduce errors, particularly for scaling and rotation, by including the model trained with the finest label space $\mathcal{L}_4$. The average computation time for PSTR is approximately 5 seconds (on a conventional desktop computer with Intel$^{\text{TM}}$i5 processor), which is larger compared to Robust FFT, RANSAC-SURF and RANSAC-MSER methods whose average computation time is respectively 0.25, 0.33 and 0.46 seconds. The bottleneck for PSTR is convolution with $3D$ Gabor filters. The speed of PSTR can be increased if the Gabor representation can be replaced with a motion representation that is computationally more efficient.

In Fig. 3a,b we show examples from the SEMAINE dataset. Fig. 3a shows the difference between the images of a pair with mouth expression obtained

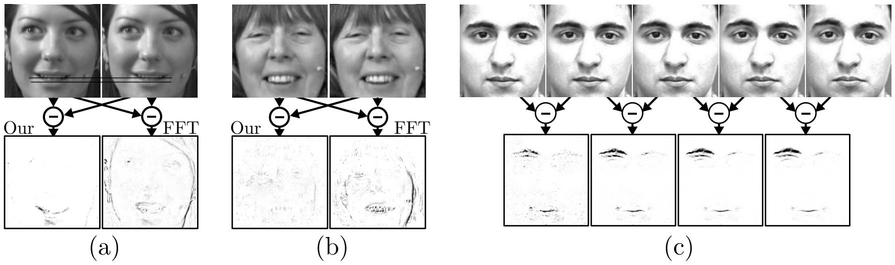


Fig. 3. (a) Difference between pair of images with a subtle mouth expression, after registering the images with PSTR and FFT; (b) Difference between images without expression; (c) Difference of each pair in a sequence after registration.

after applying the mean errors of PSTR (for $\phi = \phi^\sigma$ and $N = 3$) and FFT to the second image in each pair (Fig. 3a). While the differences provided by FFT hardly help identifying the location of the expression, PSTR clearly shows where the expression occurs. Identifying the *absence* of facial activity is as important as detecting facial activity. We applied a similar test but for a pair with no facial activity (Fig. 3b). The differences provided by FFT generate spurious activity. Instead, the difference image of PSTR shows no signs of facial activity except from minor artifacts introduced by interpolation.

6.3 Identifying Failure

In the PIE dataset, the transition from 16^{th} to 17^{th} frame in all sequences involves a very sudden illumination variation, and causes PSTR, FFT and RANSAC-based methods to fail. PSTR identifies failures automatically.

Table 3 provides the MAE performance on the PIE dataset — the PSTR results are obtained with the parameters $\phi = \phi^\sigma, N = 2$ and label spaces $\mathcal{L}_{1-4}$. The typical symptom of failure in PIE experiments is large estimation error, in which case the mean error MAE gets very high even when only a single failure occurs. We therefore compute the MAE only over the pairs where registration did not fail. For our method, failure is identified using the threshold probability P_θ as described in Section 5.2 — the threshold is computed as $P_\theta = e^{-34}$ using samples synthesised from the CK+ dataset. For FFT and RANSAC, we manually eliminated the pairs with a translation error larger than 5 pixels. The rightmost column in Table 3 lists the number of pairs eliminated when computing the results.

Table 3 suggests that PSTR and FFT are robust against illumination variations as the performance of both methods on the PIE dataset is similar to their performance on CK+ dataset. The number of pairs where failure is expected (pairs obtained from the 16^{th} and 17^{th} frame) is 10. RANSAC-based methods failed in more than 10 pairs, whereas FFT failed only on the 10 pairs. PSTR also failed on these 10 pairs and identified these failures successfully. PSTR produced only 1 false negative by eliminating a correctly registered pair.

6.4 Sequence Registration

Sequence registration performance on CK+ dataset is given in Table 2 (right). Similarly to pair registration, we give two values at each cell — one obtained by including $\mathcal{L}_4$ and one by excluding $\mathcal{L}_4$. Expectedly, errors are slightly higher than in pair registration. The ground truth is common for all images in a sequence $\mathbf{S}_i$ (essentially all frames are mapped to the first frame), and since facial expressions display larger variation in a sequence than in a pair, errors are more likely to occur. Also, the exactness of ground truth cannot be guaranteed. Although we selected sequences with almost no head/body motion and limited sequence length to $T = 7$, minor motions might have been displayed by the subjects.

In Fig. 3c we show an example of a registered sequence from the CK+ dataset. The images on top are obtained after registration, and the ones on bottom are obtained by taking the difference between consecutive image pairs. The sequence contained a slowly evolving mouth expression and (right) eyebrow movement. The resulting difference images clearly illustrate the usefulness of PSTR — no matter how slowly the expression evolves, the difference images capture face actions and *only* face actions.

We provide a demo video that depicts the sequences after registration — the video is available as supplementary material and also on an online channel[1]. Although we perform training only with the controlled CK+ dataset, PSTR is able to perform accurate registration for naturalistic expressions with head/body and background motion (SEMAINE dataset) as well as sequences with rapid illumination variations (PIE dataset).

7 Conclusions

We presented a probabilistic framework for temporal face registration (PSTR) that achieves subpixel registration accuracy. The framework is based on a *motion representation* that measures registration errors between subsequent frames, a supervised *probabilistic model* that learns the registration errors from the proposed representation, and an iterative *registration error estimator*. We demonstrated on three publicly available datasets that the proposed framework not only achieves high registration accuracy but can also generalise to naturalistic data even when trained only with controlled data. Although as a proof of concept we evaluated the framework on facial action and expression data, the proposed method can be used for multiple application domains which require facial activity analysis. The source code of PSTR is available to the research community via `http://cis.eecs.qmul.ac.uk/software.html`.

Acknowledgement. The work of E. Sariyanidi and H. Gunes is partially supported by the EPSRC MAPTRAITS Project (Grant Ref: EP/K017500/1).

[1] The demo video is available on `http://www.youtube.com/user/AffectQMUL`

References

1. Vinciarelli, A., Pantic, M., Bourlard, H.: Social signal processing: Survey of an emerging domain. Image and Vision Computing **27** (2009) 1743 – 1759
2. Gunes, H., Schuller, B.: Categorical and dimensional affect analysis in continuous input: Current trends and future directions. Image and Vision Computing **31** (2013) 120 – 136
3. Valstar, M., Schuller, B., Smith, K., Eyben, F., Jiang, B., Bilakhia, S., Schnieder, S., Cowie, R., Pantic, M.: AVEC 2013 – the continuous audio/visual emotion and depression recognition challenge. In: Proc. ACM Int'l Workshop on Audio/Visual Emotion Challenge. (2013) 3–10
4. Almaev, T., Valstar, M.: Local Gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In: Proc. Int'l Conf. Affective Computing and Intelligent Interaction. (2013) 356–361
5. Zhao, G., Pietikäinen, M.: Boosted multi-resolution spatiotemporal descriptors for facial expression recognition. Pattern recognition letters **30** (2009) 1117–1127
6. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Analysis and Machine Intelligence **29** (2007) 915 –928
7. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Trans. Pattern Analysis and Machine Intelligence **31** (2009) 39–58
8. Jiang, B., Valstar, M., Martinez, B., Pantic, M.: Dynamic appearance descriptor approach to facial actions temporal modelling. IEEE Trans. Systems, Man and Cybernetics – Part B **44** (2014) 161–174
9. Huang, X., Zhao, G., Zheng, W., Pietikäinen, M.: Towards a dynamic expression recognition system under facial occlusion. Pattern Recognition Letters **33** (2012) 2181 – 2191
10. Valstar, M.F., Pantic, M.: Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In: HumanComputer Interaction. Volume 4796. (2007) 118–127
11. Valstar, M., Jiang, B., Mehu, M., Pantic, M., Scherer, K.: The first facial expression recognition and analysis challenge. In: Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition. (2011) 921–926
12. Çeliktutan, O., Ulukaya, S., Sankur, B.: A comparative study of face landmarking techniques. EURASIP J. Image and Video Processing **2013** (2013) 13
13. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2012) 2879–2886
14. Jiang, B., Valstar, M., Pantic, M.: Action unit detection using sparse appearance descriptors in space-time video volumes. In: Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition. (2011) 314–321
15. Tzimiropoulos, G., Argyriou, V., Zafeiriou, S., Stathaki, T.: Robust FFT-based scale-invariant image registration with image gradients. IEEE Trans. Pattern Analysis and Machine Intelligence **32** (2010) 1899–1906
16. Adelson, E.H., Bergen, J.R.: Spatio-temporal energy models for the perception of motion. J. of the Optical Society of America **2** (1985) 284–299
17. Kolers, P.A.: Aspects of motion perception. Pergamon Press Oxford (1972)
18. Petkov, N., Subramanian, E.: Motion detection, noise reduction, texture suppression, and contour enhancement by spatiotemporal Gabor filters with surround inhibition. Biological Cybernetics **97** (2007) 423–439

19. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton, NJ, USA (1957)
20. Amano, K., Edwards, M., Badcock, D.R., Nishida, S.: Adaptive pooling of visual motion signals by the human visual system revealed with a novel multi-element stimulus. Journal of Vision **9** (2009)
21. Pinto, N., Cox, D.D., DiCarlo, J.J.: Why is real-world visual object recognition hard? PLoS computational biology **4** (2008) e27
22. Webb, B.S., Ledgeway, T., Rocchi, F.: Neural computations governing spatiotemporal pooling of visual motion signals in humans. The Journal of Neuroscience **31** (2011) 4917–4925
23. Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: Int'l Conf. Machine Learning. (2010) 111–118
24. Fischer, S., Šroubek, F., Perrinet, L., Redondo, R., Cristóbal, G.: Self-invertible 2d log-Gabor wavelets. Int'l J. Computer Vision **75** (2007) 231–246
25. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Proc. European Conf. Computer Vision. (2006) 404–417
26. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22** (2004) 761–767
27. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression database. IEEE Trans. Pattern Analysis and Machine Intelligence **25** (2003) 1615–1618
28. McKeown, G., Valstar, M., Cowie, R., Pantic, M., Schroder, M.: The SEMAINE database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. IEEE Trans. Affective Computing **3** (2012) 5–17