Automatic Estimation of Self-Reported Pain by Interpretable Representations of Motion Dynamics

Benjamin Szczapa[1,3], Mohamed Daoudi[2], Stefano Berretti[3], Pietro Pala[3], Alberto Del Bimbo[3] and Zakia Hammal[4]

[1] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France*
[2] *IMT Lille Douai, Univ. Lille, CNRS, UMR 9189 CRIStAL, F-59000 Lille, France*
[3] *Department of Information Engineering, University of Florence, Italy*
[4] *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA*

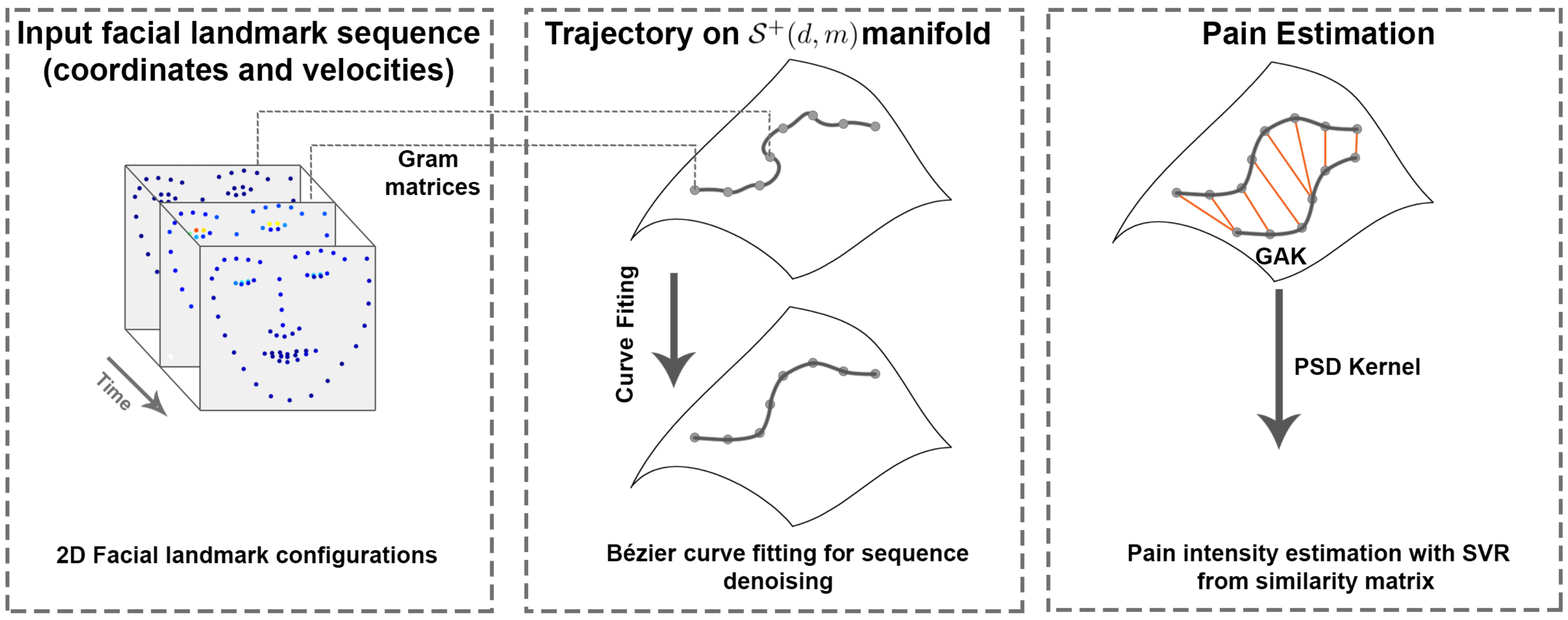
Motivations

- Usage of a shape representation invariant to undesirable transformations
- New temporal modeling of landmark sequences
- Distance between arbitrary sequences
- Solution for self-reported pain estimation

Contributions

- Geometry based approach to estimate self-reported pain
- Facial dynamics based on Gram matrix computation and trajectory modeling
- The manifold $\mathcal{S}^+(d,n)$ of positive-semidefinite matrices is endowed with an optimised metric for 2D
- A recent curve fitting method is used to smooth trajectories on the manifold
- The use of Global Alignment Kernel for temporal alignment, instead of DTW

Approach overview



Overview of the proposed approach - After facial landmarks extraction for each frame of a sequence, velocities are computer as the displacement of the coordinates between two consecutive frames. The Gram matrices are computed from the combination of the landmark coordinates and velocities to build the trajectory on the $\mathcal{S}^+(d,n)$ manifold. We apply a curve fitting algorithm on the trajectory for denoising and smoothing. Global Alignment Kernel (GAK) is then used to align the trajectories on the manifold. Finally, we use the kernel generated from GAK with SVR to estimate the self-reported pain level.

Shape Representation and Trajectory Modeling

Shape Representation

Each sequence is characterized by a set of landmarks $\{Z_0,\ldots,Z_\tau\}$, where τ is the number of frames of the sequence, and each configuration matrix Z_i $(1 \leq i \leq \tau) \in \mathrm{R}^{n\times d}$ encodes the position of the n landmarks in d dimensions. Velocities are computed as the magnitude of the displacement between two consecutive landmark configurations Z_i and Z_{i+1}. The final facial representation A is the concatenation of the landmark coordinates and velocities. To measure the dynamic changes of the curves, while remaining invariant to rotation and translation, we use Gram matrices that are $2n \times 2n$ positive semi-definite matrices, of rank equal to d. It is computed as $G = AA^T$.

Metric

We consider the manifold $\mathcal{S}^+(d,2n)$ as the quotient manifold $\mathbb{R}_*^{2n\times d}/\mathcal{O}_d$, where $\mathbb{R}_*^{2n\times d}$ is the set of full-rank $2n \times d$ matrices. The mapping $\pi : \mathbb{R}_*^{2n\times d} \to \mathbb{R}_*^{2n\times d}/\mathcal{O}_d$, that transforms points to their equivalence class, induces a Riemannian metric on the quotient manifold from the Euclidean metric in $\mathbb{R}_*^{2n\times d}$. The distance between two elements is expressed in terms of facial configurations $A_i, A_j \in \mathbb{R}_*^{2n\times d}$ as:

$$d(G_i,G_j) = \min_{Q\in\mathcal{O}_d} \|A_jQ - A_i\|_F \ ,$$

The optimal solution is $Q^* := VU^T$, where $A_i^TA_j = U\Sigma V^T$ is the singular value decomposition. When $d = 2$, the distance can be formulated as follows:

$$d(G_i,G_j) = \mathrm{tr}(G_i) + \mathrm{tr}(G_j) - 2\sqrt{(a+d)^2 + (c-b)^2} \ ,$$

where $A_i^TA_j = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Alignment and Classification

Global Alignment Kernel (GAK) is used to align sequences on the manifold. Let us consider $G^1 = \{G_0^1,\cdots,G_{\tau_1}^1\}$ and $G^2 = \{G_0^2,\cdots,G_{\tau_2}^2\}$, two trajectories of Gram matrices. Given a metric to compute the distance between two elements of each sequence, we propose to compute the matrix D of size $\tau_1 \times \tau_2$, where each $D(i,j)$ is the distance between two elements of the sequences, with $1 \leq i \leq \tau_1$ and $1 \leq j \leq \tau_2$. The kernel $\tilde{k}$ can now be computed using the halved Gaussian Kernel on this same matrix D, as well as the kernel k. They are defined as:

$$\tilde{k}(i,j) = \frac{1}{2} * exp\left(-\frac{D(i,j)}{\sigma^2}\right) \qquad k(i,j) = \frac{\tilde{k}(i,j)}{(1-\tilde{k}(i,j))}$$

To compute the similarity score between the two trajectories G^1 and G^2, we define a zeros matrix M of size $(\tau_1+1)\times(\tau_2+1)$ with $M_{0,0} = 1$ that will contain the path to the similarity between our two sequences. The terms of M are computed as:

$$M_{i,j} = (M_{i,j-1} + M_{i-1,j-1} + M_{i-1,j}) * k(i,j).$$

The similarity score is the value at $M_{(\tau_1+1),(\tau_2+1)}$. Finally, we build a new matrix K of size $n_{seq}\times n_{seq}$, where n_{seq} is the number of sequences in the dataset that contains all the similarity score between all the sequences and is used directly with SVR for self-reported pain estimation.

Experimental results

Experimental results on the UNBC-McMaster Shoulder Pain Archive are presented below. To test our method, we used 3 different protocols: Leave-One-Sequence-Out, Leave-One-Subject-Out cross validation and 5-folds cross validation

UNBC-McMaster

Protocol	% of frames	MAE	RMSE
Leave-One-Sequence-Out	25%	**2.3166**	3.1459
	100%	2.5291	3.3263
Leave-One-Subject-Out cross validation	25%	**2.523**	3.2692
	100%	2.9176	3.5133
5-fold cross validation	25%	**2.4365**	3.147
	100%	2.7944	3.5088

Table 1: Results of our method with the 3 different protocols on the UNBC-McMaster Shoulder Pain Archive

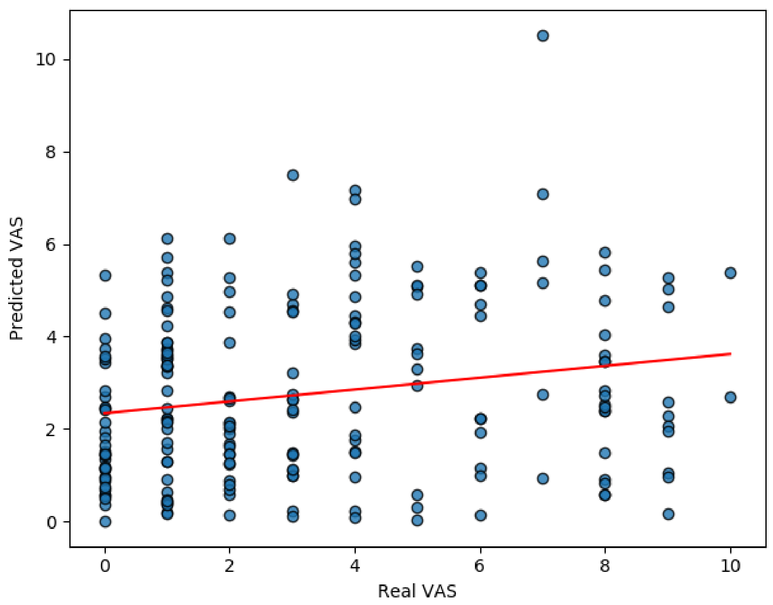


Figure 2: Distribution of the predicted VAS values compared with the real VAS using the 5-fold cross validation protocol

Comparison with State-of-the-Art

Method	Protocol	Labels for training	MAE
DeepFaceLift (2017)*	5-fold cross validation	VAS	2.30
RNN-HCRF (2017)*	random split	VAS & PSPI	2.46
Ours	5-fold cross validation	VAS	2.4365

Table 2: Comparison of our approach with state-of-the-art results.

*: Deep learning based approaches