Local Jet Feature Space Framework for Image Processing and Representation

Antoine Manzanera
Laboratoire d'Electronique et Informatique
ENSTA-ParisTech, Paris, France
http://www.ensta.fr/~manzaner

***Abstract*—We present a unified framework for processing and representing images using a feature space related to local similarity. The visual data is represented by the multiscale and versatile local jet feature space, which can be reduced by vector quantisation and/or represented by data structures enabling efficient nearest neighbours search (*e.g.* kd-trees). We demonstrate the interest of the local jet feature space processing through three fundamental low level tasks: noise reduction, motion estimation and background modelling/subtraction. We also show the potential of the framework in terms of visual representation for higher level (*e.g.* object modelling and recognition) tasks.**

***Keywords*-vision framework; multiscale local jets; similarity space; nearest neighbours; optical flow; non local means; background modelling;**

I. Introduction

Many problems in image processing and vision relate to visual similarity. Since the earliest processes of denoising or perceptual grouping, to the higher level tasks of object recognition, measuring the resemblance, or matching two objects according to the visual appearance are fundamental functions. In the traditional space × time representations of video sequences, the canonical distance is not related to visual similarity, which induces a major computational drawback. Indeed, similar objects from the video data are expected to interact in the processing, and then should be contiguous in the representation. These general remarks do not only apply to the current data, image or recent frames history, but also to the global visual knowledge that the vision system is constructing during its operating lifetime.

The purpose of this work is to design a global, generic and computationally tractable framework for the representation and the processing of the visual data, based on: (1) the projection of the space × time image data within a transformed space whose metrics correspond to visual similarity, (2) a set of functions operating in the transformed and/or the image domain, for extracting relevant information from the video, updating the transformed domain structure accordingly, and/or modifying the video in the image domain according to some specific task (filtering, detecting, predicting), and (3) dedicated data structures for making such framework computationally feasible, in terms of memory and processing time. In our philosophy, such unified framework should be usable for the whole vision process, from the lowest level of regularisation and enhancement to the levels of higher semantics related to recognition and understanding. The framework should also be compliant with real-time video processing, which implies both dynamical and efficient construction of the visual representation.

The inspiring and related works are presented in Section II. In our work the preferred similarity space is made of the collection of spatial derivatives estimated at different scales (the local jet). This feature space and its data structure are presented in Section III. The following sections present the applications of the framework for different low level visual processing tasks: non-local means image denoising (Sec. IV), optical flow estimation (Sec. V) and background subtraction based motion detection (Sec. VI). Section VII presents some visual models that can be extracted from the feature space data structure, for higher level representations.

II. Related works

Our work is related to Peyré's manifold model [1]. In this theoretical framework, the image data is projected within a higher dimensional feature space, forming a manifold. Many inverse problems in low level computer vision can be expressed by regularising this manifold and then back-projecting the transformed manifold within the image space. In this sense, the different low level algorithms proposed in this paper can be seen as instances of the manifold model. Conversely, our work is also an extension of this model, with the aim of extracting higher level representations from the manifold structure.

Our framework exploits many ideas from previous works on textured objects modelling, segmentation and recognition. Filter banks have been used for a long time as a way to extract meaningful local information on direction, scale, and frequency [2]. Quantising such information is also a commonplace in textons [3] or bag of features [4] approaches. Compared with those methods, one fundamental property of our framework is that the feature is intrinsically dense in the image space, making the corresponding information available at any location. Another particularity of our work is that reducing the information support is done by finding the isolated or clustered points in the feature space, thus avoiding the common separation between detection and description of the salient structures [5], [6].

The importance of the local jet in image representation has been identified a few decades ago. Koenderink and

Van Doorn [7] pointed out the fundamental role of the first three orders of derivatives in the human visual system. They also noticed that some Euclidean distance on the local jet vectors could be used to approximate the sum of squared differences between image patches. To our knowledge this has not been really used in the literature, maybe because the approximation is crude for complicated patches. But, as we will see later, distances based on the local jet are actually significant to distinguish similar pixels.

Anyway, the local jet has been much used for the construction of invariants, particularly in image retrieval [8]. It has also been used more recently for the classification of pixels according to their local geometry, see for example [9]. As shown later, such classification can be exploited to reduce the dimension of the local jet descriptor.

III. Multiscale Local Jet Feature Space

A. Similarity space

Using the partial derivatives to measure the local similarity is a natural choice [10] since the local behaviour of any differentiable function f can be predicted from its derivatives (Taylor expansion at order r):

$$f(\mathbf{x}+\mathbf{c}) = \sum_{k=0}^{r}\sum_{i=0}^{k} \frac{\binom{k}{i}}{k!} c_1^{k-i} c_2^{i} \frac{\partial^k f}{\partial \mathbf{x}_1^{k-i} \partial \mathbf{x}_2^{i}}(\mathbf{x}) + o(||\mathbf{c}||^r) \ , \tag{1}$$

with $\mathbf{x}_1$ and $\mathbf{x}_2$ a basis of $\mathbb{R}^2$, in which the components of the residual $\mathbf{c}$ are $c_1 = \mathbf{c}\cdot\mathbf{x}_1$ and $c_2 = \mathbf{c}\cdot\mathbf{x}_2$. To simplify we denote $f_{ij} = \frac{\partial^{i+j} f}{\partial \mathbf{x}_1^i \partial \mathbf{x}_2^j}$. The relevance of the local jet as a description vector is confirmed by the first singular (or eigen) vectors that arise in SVD or PCA based decomposition of natural image patches, that look much like the first derivatives of a 2d Gaussian function (see for example [11]). In digital images, the derivative only makes sense up to a level of regularity corresponding to the scale of estimation [12]:

$$f_{ij}^{\sigma} = f \star \frac{\partial^{i+j} G_\sigma}{\partial \mathbf{x}_1^i \partial \mathbf{x}_2^j} \ , \tag{2}$$

where G_σ is the 2d Gaussian function of standard deviation σ. The multiscale local jet is then the collection $\{f_{ij}^{\sigma}; i+j \leq r, \sigma \in S\}$, where r is the order of derivation, $S = \{\sigma_1, \ldots \sigma_q\}$ the selected scales. Figure 1 illustrates the induced representation for a few points taken from a natural image, at one scale $\sigma = 1.0$. The image is split into 15×15 patches, and the reconstruction is performed by Taylor expansion on patches of the same size, using only the local jet computed at the patch centre.

Our representation does not use patches but a multiscale local jet vector in every pixel, with normalised components combining the scale normalisation from scale space theory [12], and the number of $(i+j)$-order derivatives:

$$F_{ij}^{\sigma} = \frac{\sigma^{i+j}}{i+j+1} f_{ij}^{\sigma} \ . \tag{3}$$

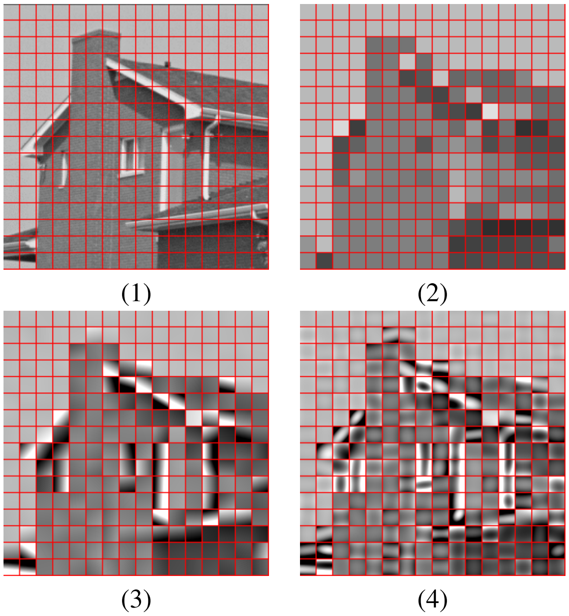

Figure 1. Local representation by the local jet at one scale ($\sigma = 1.0$): (1) Original patches (2) Order 0 (1d feature) (3) Order 1 (3d feature), (4) Order 2 (6d feature)

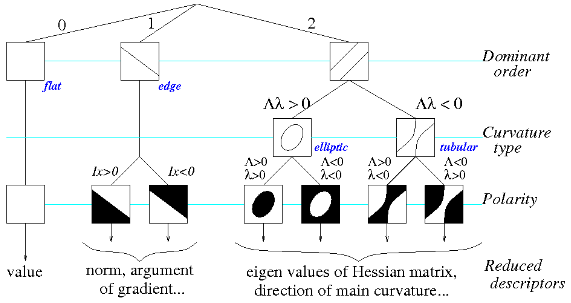

Figure 2. Pixel categorisation for the reduction of the local jet descriptors: at order 2, pixels can be categorised into 4 categories (7 if the polarity is considered).

If required, rotation invariant derivatives can be obtained by expressing the derivatives within the local basis of coordinates of the gradient and isophote components. The local jet also provides contrast invariant measures, *e.g.* the direction of the gradient or isophote at order 1, or the direction of main curvatures (eigen vectors of the Hessian matrix) at order 2. Finally, following [9], the local behaviour of every pixel can be categorised at every scale according to the dominant order of derivation: flat zone for order 0, straight contours for order 1, and elliptic or tubular curvatures for order 2 (according to the signs of Λ and λ, the eigen values of the Hessian matrix). Once categorised, the dimensions of the local jet descriptor can be reduced to significant derivatives (see Figure 2).

B. Metrics

To measure similarity, we typically consider three types of distance in the applications. Let F be the full local jet vector $F = (F_{ij}^{\sigma})_{i+j\leq r, \sigma\in S}$, we denote $\hat{\mathbf{x}} = F(\mathbf{x})$ the feature vector associated to pixel $\mathbf{x}$. Let $F^{\sigma} = (F_{ij}^{\sigma})_{i+j\leq r}$ be the local jet at scale σ, and $||.||$ the Euclidean norm. First, the *single scale* distance, checking whether $\mathbf{x}$ at scale σ_1 is similar to $\mathbf{y}$ at scale σ_2:

$$d_F^{(\sigma_1,\sigma_2)}(\hat{\mathbf{x}},\hat{\mathbf{y}}) = ||F^{\sigma_1}(\mathbf{x}) - F^{\sigma_2}(\mathbf{y})|| \ . \qquad (4)$$

Second, the *pan-scalic* distance, checking whether $\mathbf{x}$ and $\mathbf{y}$ are similar for all the scales $\sigma \in S$:

$$D_F^S(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \max_{\sigma\in S} d_F^{(\sigma,\sigma)}(\hat{\mathbf{x}},\hat{\mathbf{y}}) \ . \qquad (5)$$

However for representation purposes (next subsection) the use of Euclidean distance $||\hat{\mathbf{x}} - \hat{\mathbf{y}}||$ can be more convenient.

Third, the *trans-scalic* pseudo-distance, checking whether there exists a couple of scales for which $\mathbf{x}$ and $\mathbf{y}$ are similar:

$$\delta_F^S(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \min_{(\sigma_n,\sigma_m)\in S^2} \max_{(\sigma_{n+p},\sigma_{m+p})\in S^2} d_F^{(\sigma_{n+p},\sigma_{m+p})}(\hat{\mathbf{x}},\hat{\mathbf{y}}) \ . \qquad (6)$$

Figure 3 shows different examples of similarity maps using those distances. For a pixel $\mathbf{x_0}$ the similarity map is defined as $M_{\mathbf{x_0}}^{d_F}(\mathbf{x}) = \Phi(d_F(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}))$, with d_F the local jet distance, and Φ some real increasing function. In this figure $\Phi(z) = 1 - e^{-\frac{z^2}{C^2}}$, with $C = 10$. Those maps show the properties of the different distances and local jet components in terms of rotation and scale invariance.

C. Data structures

The first step of the representation then consists in projecting the image data into the chosen similarity space. For every pixel, a feature vector is computed, and the collection of features is kept in adequate data structure for further processing. If the feature space dimensionality is low, the data structure may be a simple array, whose coordinates are indexed by each component of the feature space, which must then be quantised properly. The data structure is then a hash table whose hash function is the quotient of the quantisation. As the memory cost of such structure grows exponentially with the dimension, other solutions must be used for higher dimension, like the classical kd-tree [13], which is optimal in terms of memory occupation.

The kd-tree is a useful tool for performing nearest neighbours (NN) search in the feature space. It will be extensively used in the following to perform efficiently operations based on visual similarity, that are intrinsically non local in the image space. However, there are many operations where NN search will be needed very intensively (*e.g.* for every pixel / feature vector). In that case, the computational cost will remain too important for real-time video. Two important optimisations are employed: (1) Approximate Nearest Neighbour (ANN) search techniques [14] that reduces both worst case and average search complexity, and (2) Quantisation of the feature space, that reduces the size of the kd-tree. We have used in our experiments the ANN library developed by Arya and Mount [14], and a simple approximation of the K-means clustering method for vector quantisation. Depending on the used metrics, one kd-tree per scale or one single kd-tree has to be calculated. The figure 4 illustrates in dimension 2 the projection in the feature space and the construction of the kd-tree, without and with quantisation.

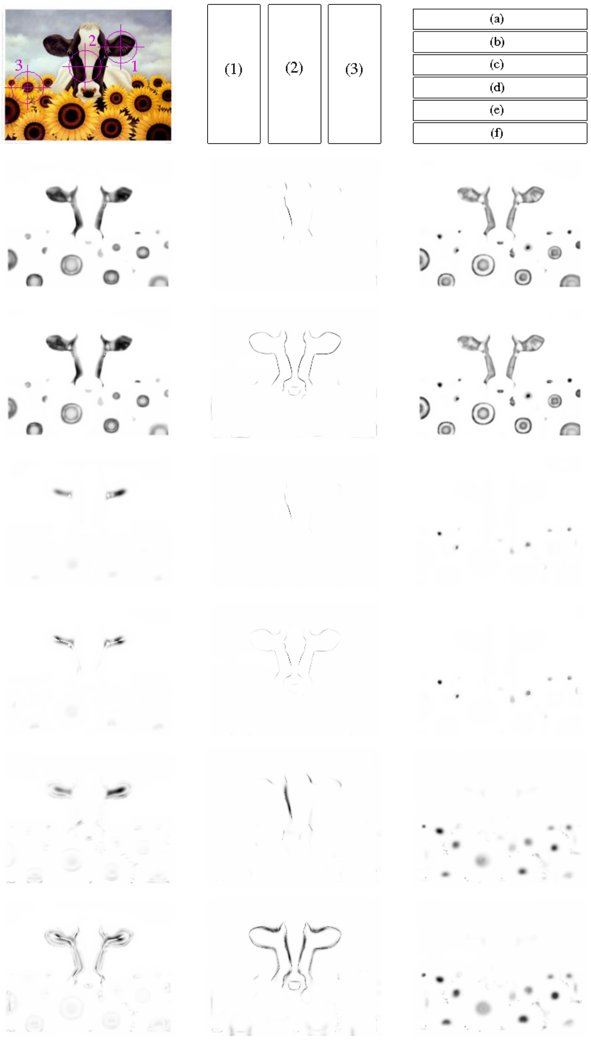

Figure 3. Similarity maps based on 2-order local jet metrics, for 3 different pixels: (1), (2) and (3), and 6 different distances: Single (same) scale: (a) canonical components (CC), (b) rotation invariant components (RIC), Pan-scalic (4 scales): (c) CC, (d) RIC, and Trans-scalic (4 scales): (e) CC, (f) RIC. (Painting by Lowell Herrero)

D. Useful notations

Let $\mathbf{x}$ be a pixel from the image space. We denote $\hat{\mathbf{x}}_f$ the projection of $\mathbf{x}$ in the feature space of image f. Let $\mathcal{F}_f$ be the set of features of image f. If $\mathbf{u}$ is a feature vector, let $\nu_k^{\mathcal{F}_f}(\mathbf{u})$ be its k-th nearest neighbour in the feature space of

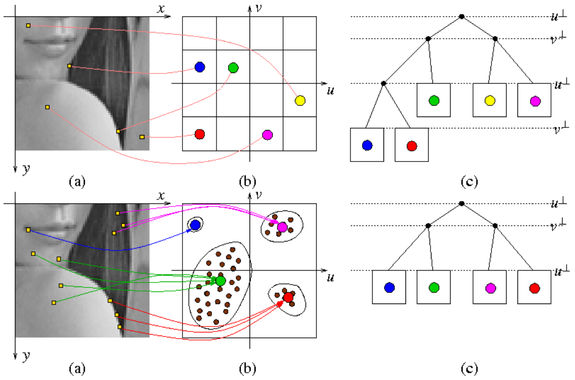

Figure 4. (a) Image data (b) projected into the feature space, and (c) collected into a kd-tree structure, without (top) and with (bottom) vector quantisation.

f. We denote $\mathcal{F}_f^{-1}(\mathbf{u})$ the set of pixels which are assigned to the codeword $\mathbf{u}$ in the quantised feature space (codebook) of f. If there is no quantisation, the notation remains valid as $\mathcal{F}_f^{-1}(\mathbf{u}) = \{\mathbf{x}\}$ such that $\hat{\mathbf{x}}_f = \mathbf{u}$.

IV. Non-local means video filtering

The non-local (NL) means filter, originally proposed by Buades *et al* [15] is a powerful image denoising technique, in which every pixel value is replaced by a weighted average of the other pixels, the weights depending on pixel similarity, not on pixel distance in the image space (hence the "non local" property). In our framework, the NL-means is simply expressed by calculating the weights using a distance in the feature space. Let $\mathbf{u}$ and $\mathbf{v}$ be two feature vectors. $\omega(\mathbf{u}, \mathbf{v})$ the relative (symmetric) weight of $\mathbf{u}$ with respect to $\mathbf{v}$, is defined as follows:

$$\omega(\mathbf{u},\mathbf{v}) = e^{-\frac{d_F(\mathbf{u},\mathbf{v})^2}{h^2}} \quad , \tag{7}$$

where h is a decay parameter, related to the amount of noise to be removed. Now two variants of the NL means can be considered:

1) Limited range (LR) method

$$f^{NL}_{LR}(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} f(\mathbf{y})\,\omega(\hat{\mathbf{x}}_f,\hat{\mathbf{y}}_f)}{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \omega(\hat{\mathbf{x}}_f,\hat{\mathbf{y}}_f)} \tag{8}$$

2) Unlimited range (UR) method

$$f^{NL}_{UR}(\mathbf{x}) = \frac{\sum_{\mathbf{u}\in\mathcal{W}(\hat{\mathbf{x}}_f)} \check{f}(\mathbf{u})\,\omega(\hat{\mathbf{x}}_f,\mathbf{u})}{\sum_{\mathbf{u}\in\mathcal{W}(\hat{\mathbf{x}}_f)} \omega(\hat{\mathbf{x}}_f,\mathbf{u})} \tag{9}$$

where $\mathcal{N}(\mathbf{x})$ (resp. $\mathcal{W}(\mathbf{v})$) is a neighbourhood of $\mathbf{x}$ (resp. $\mathbf{v}$), corresponding to the k nearest neighbours of $\mathbf{x}$ (resp. $\mathbf{v}$) in the image (resp. feature) space. See figure 5.

$\check{f}(\mathbf{u})$ is defined as:

$$\check{f}(\mathbf{u}) = \frac{1}{|\mathcal{F}_f^{-1}(\mathbf{u})|} \sum_{\mathbf{x}\in\mathcal{F}_f^{-1}(\mathbf{u})} f(\mathbf{x}) \ , \tag{10}$$

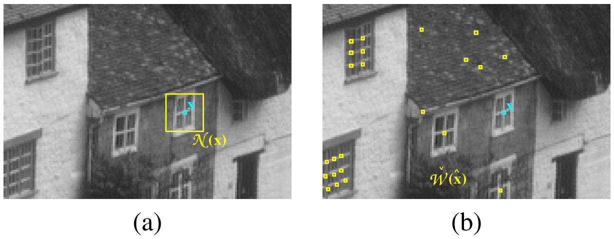
$\mathcal{N}(\mathbf{x})$
$\check{\mathcal{W}}(\hat{\mathbf{x}})$
(a)
(b)

Figure 5. Limited (a) *vs* Unlimited (b) range approaches in the computation of the NL-means. $\mathcal{N}(\mathbf{x})$: nearest neighbours of $\mathbf{x}$ in the image space. $\check{\mathcal{W}}(\hat{\mathbf{x}})$: nearest neighbours of $\hat{\mathbf{x}}$ in the feature space back-projected in the image space.

i.e. the average value of f on the pixels corresponding to feature $\mathbf{u}$ (recursively calculated during the quantisation).

In our experiments, the decay parameter h is automatically adjusted, using a fast estimation of the noise variance (see [16] for more details).

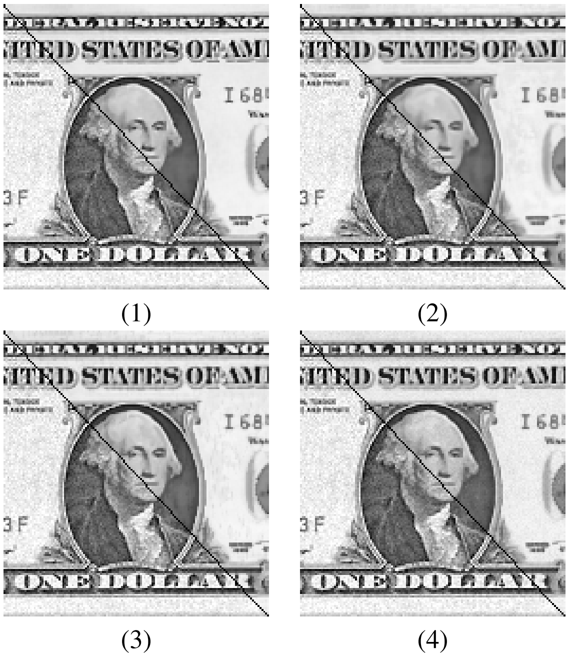

Figure 6. NL-means filtering in the local jet feature space. (1) LR ($\mathcal{N}(\mathbf{x})$: 17 × 17 neighbourhood of $\mathbf{x}$ in the image domain). (2) UR, exact search ($\mathcal{W}(\hat{\mathbf{x}}_f)$: 30 NN in the local jet domain), (3) UR, approximate search ($\varepsilon = 10.0$), (4) UR, approx. search, with quantised feature space (1938 words in the dictionary).

It can be said that the local jet based NL-means, by changing the order of derivation and number of scales, form a *continuum* between tone space (or bilateral) filtering and patch based NL-means. However, even at one single scale, the order 2 local jet based NL-means results are very close of patch based ones. See figure 6 for some results on the same noisy image (only the top half diagonal is processed). It is somewhat surprising that the denoising quality looks better for the LR (Fig. 6(1)) than for the UR (Fig. 6(2)).

But on the one hand, the edge and corner pixels are more affected by the UR methods, the relative weights of their neighbours being much higher in the feature than in the image space. One the other hand, for large noisy homogeneous regions, the UR method is able to find patterns that tends to exaggerate the texturing of these regions. Using kd-trees, the UR method is generally faster than the LR one since the cardinality of $\mathcal{W}(\hat{\mathbf{x}}_f)$ is usually much smaller than for $\mathcal{N}(\mathbf{x})$. Furthermore, using approximate search (Fig. 6(3)), and quantising the local jet space (Fig. 6(4)) significantly lowers the computation time, while partially compensating the drawbacks of the UR method evoked above, but more quantitative evaluation is needed.

V. Optical flow estimation

The apparent motion, or optical flow estimation turns out to be - from a conceptual point of view at least - one of the most straightforward applications of the feature space based similarity. At frame t, for image f_t, and for every pixel $\mathbf{x}$, we compute $\mathbf{u}(f_{t-1}, f_t, \mathbf{x})$, the nearest neighbour of the feature vector associated to $\mathbf{x}$, in the feature space of f_{t-1}:

$$\mathbf{u}(f_{t-1}, f_t, \mathbf{x}) = \arg \min_{\mathbf{v} \in \mathcal{F}_{f_{t-1}}} d_F(\hat{\mathbf{x}}_{f_t}, \mathbf{v}) = \nu_1^{\mathcal{F}_{f_{t-1}}}(\hat{\mathbf{x}}_{f_t}) \ . \tag{11}$$

Then we can compute $\mathbf{y}(f_{t-1}, f_t, \mathbf{x})$, the pixel from f_{t-1} which is the most similar to $\mathbf{x}$ from f_t:

$$\mathbf{y}(f_{t-1}, f_t, \mathbf{x}) = \arg \min_{\mathbf{z} \in \mathcal{F}^{-1}_{f_{t-1}}(\mathbf{u}(f_{t-1}, f_t, \mathbf{x}))} d_I(\mathbf{x}, \mathbf{z}) \ , \tag{12}$$

with d_I the distance in the image space. Without quantisation, this is simply the pixel corresponding to feature $\mathbf{u}$ in f_{t-1}, otherwise it is the pixel from the set of pixels associated to codeword $\mathbf{u}$ which is the closest from $\mathbf{x}$ in the image space: see Figure 7.

Finally, the velocity vector is computed as the difference:

$$\mathbf{c}(f_{t-1}, f_t, \mathbf{x}) = \mathbf{x} - \mathbf{y}(f_{t-1}, f_t, \mathbf{x}) \ . \tag{13}$$

At one single scale the result is hardly usable, but using several scales, the method provides a dense estimation without explicit regularisation of the vector field, that allows a fair estimation of the motion at a global level: See Figure 8 for examples taken from classical test sequences. Here the number of NN is 1, the local jet is at order 2, and 5 different scales, without quantisation.

VI. Background subtraction

Background modelling and subtraction is a popular approach of motion detection. It consists in calculating locally (say for every pixel or block), a set of temporal statistics measures of the background, and comparing every new value with those measures, to decide whether this value is typical or not. This problem is challenging in many cases where the background is not completely static. The precision of modelling has strong influence on the computational cost, in terms of memory and time. One good trade-off is obtained by the sample and consensus methods [17], [18], which consist in keeping in memory a limited set of sampled values, and then comparing the current value to those samples, to decide whether the pixel is background or not. Vector quantisation has also been used for background modelling [19] in colour/brightness space. The algorithm we propose here is a combination of sample/consensus and vector quantisation in the local jet feature space.

In this application, we use for the whole sequence one single codebook $\mathcal{F}_f$ of quantised features, but that may evolve over time. The principle is the following: In every pixel $\mathbf{x}$, the temporal activity is modelled by a set of M

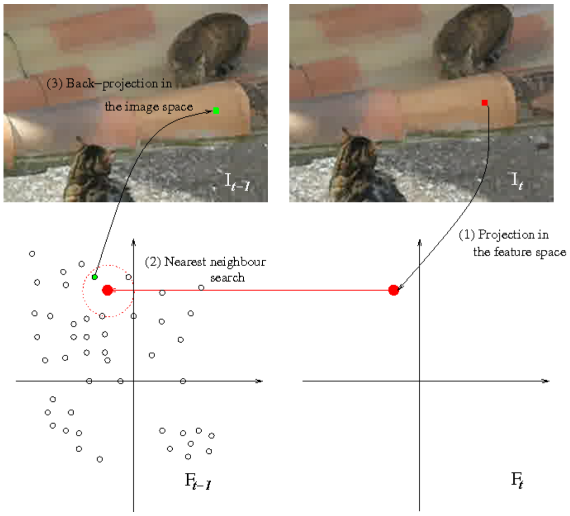

Figure 7. Optical Flow estimation by Nearest Neighbour Search in the Local Jet Feature Space.

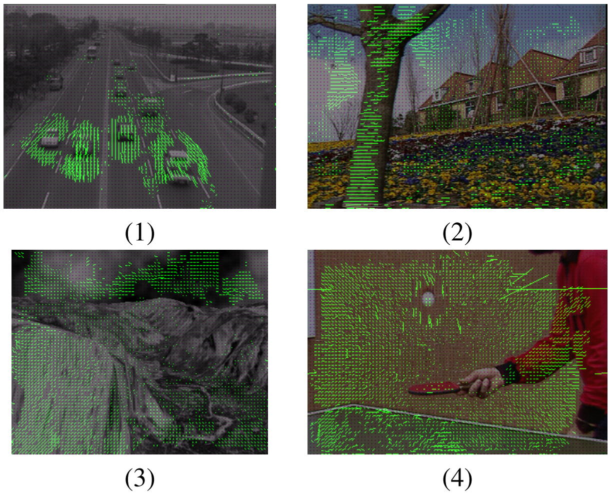

Figure 8. Optical flow fields estimated by NN search in the feature space. (1) Stationary camera, (2) Horizontal travelling, (3) Forward zooming, (4) Backward zooming and moving objects.

prototypes $\Pi(\mathbf{x}) = \{\mathbf{m}_j(f,\mathbf{x})\}_{j\in\{1,M\}} \subset \mathcal{F}_f$, that represent a sample of its past values in the feature space, and M is a temporal depth parameter.

Let ρ be a positive number; τ an integer such that $1 < \tau < M$; let $\mathcal{B}^{d_F}(\mathbf{u}, r)$ be the ball of centre $\mathbf{u}$ and radius r for the distance d_F. The foreground label $e(f,t,\mathbf{x})$, indicating whether $\mathbf{x}$ in f_t belongs to a moving object or not is calculated as follows:

$$\begin{aligned} e(f,t,\mathbf{x}) &= 1 \text{ if } |\Pi(\mathbf{x}) \cap \mathcal{B}^{d_F}(\hat{\mathbf{x}}_{f_t}, \rho)| < \tau \ , && (14)\\ &= 0 \text{ otherwise.} && (15)\end{aligned}$$

Then a pixel whose feature vector is at a distance smaller than ρ for less than τ of its M prototypes is considered foreground, elsewhere it is classified as background. The advantage of using a complex feature space instead of the mere colour is that we are able to capture more sophisticated image structure and then make the background modelling more robust. On the other hand, the vector quantisation dramatically reduces the memory cost, because only the index of the word from the codebook is used instead of a high dimensional vector. It is typically observed that a large majority of pixels only have one or two different indexes within their M background prototypes, whereas some more complicated background pixels (*e.g.* waving trees) can have much more indexes.

Our practical implementation for coding and updating the prototypes is a simple adaptation of the state-of-the-art *ViBe* algorithm [18]: The pixel prototypes are represented by a list of codebook indexes and weights (frequencies) such that the sum of weights is M. At time t the index of $\hat{\mathbf{x}}_{f_t}$ replaces one of the prototypes randomly selected, by decrementing the weight of one prototype, then incrementing the weight of another one or creating a new prototype index (See Figure 9).

For the creation of the codebook, we use, as in the NL-mean case a basic incremental version of the K-means algorithm for real-time video purposes. It is worth mentioning that the codebook does not need to be updated for every frame, nor everywhere, for example, it can be updated every 5 frames for the foreground pixels, and every 100 frames for the whole image. See Figure 10 for an example of foreground labelling in an outdoor colour sequence, using a 2 order, 1 scale, and 3 colour local jet feature space (*i.e.* 18D vector features), with a codebook of 3,000 words, a temporal depth $M = 20$, distance threshold $\rho = 0.08 d_{\max}$, and consensus threshold $\tau = M/2$. Note that, unlike [18], no spatial diffusion is performed, and the update is not strictly conservative, *i.e.* the update is made every 4 frames for background pixels, and every 16 frames for foreground pixels.

VII. Image and object characteristics

In our framework, the feature space should be used not only for image processing, but also for extracting relevant visual representation, usable at a higher level. The first descriptor we can consider is the quantised local jet itself (parented to the classical texton approaches), whose statistics provide information on the visual appearance of objects (like in the classical bag of features methods). The histogram, or weight vector of the codebook is computed recursively during the quantisation, or the updating of the codebook. Figure 11 shows an example of local jet quantisation back-projected in the image space. The detail image (right) illustrates one advantage of the dense representation, with the possible use in terms of higher order statistics (*i.e.* co-occurrence) of visual words from the codebook.

The nearest neighbour framework also provides an interesting new conception of salient points. Whereas the classi-

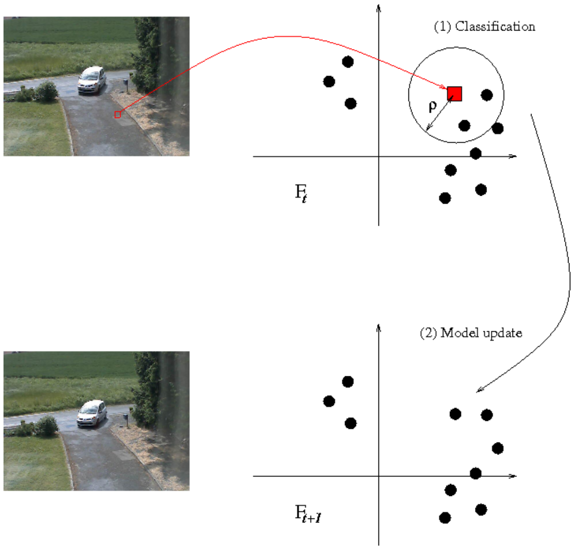

Figure 9. Adaptation of the ViBe algorithm to the local jet feature dictionary. Top, classification step: The pixel $\mathbf{x}$ is classified foreground if the number of prototypes at distance less than ρ from $\hat{\mathbf{x}}_{f_t}$ is inferior to a certain threshold. Bottom, update step: $\hat{\mathbf{x}}_{f_t}$ replaces one of the prototypes, randomly selected.

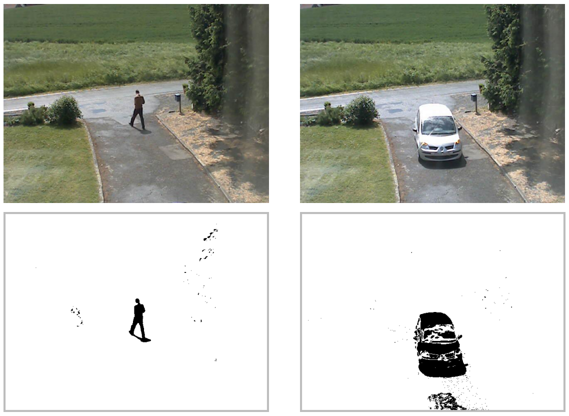

Figure 10. Background subtraction based on sample and consensus using a codebook of colour local jet features.

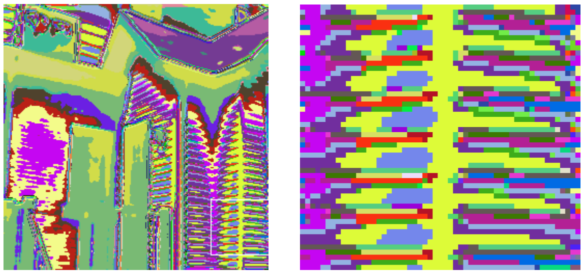

Figure 11. Quantisation of the local jet space (506 vectors). The right image is a detail of the white rectangle in the left image.

cal characterisation of interest points is purely geometrical and relatively independent of the image content, the NN feature based salience is entirely statistical and content-dependent: The salient points correspond to the isolated points in the feature space. This has been done before in the space of patches by Kervrann and Boulanger [20]. More formally, the rarest pixels are defined as:

$$R_1^{\mathcal{F}_f} = \mathcal{F}_f^{-1}(\arg\max_{\mathbf{u}\in\mathcal{F}_f} \frac{1}{m}\sum_{k=1}^{m} d_F(\mathbf{u},\nu_k^{\mathcal{F}_f}(\mathbf{u}))\ . \tag{16}$$

The rarest pixels are those assigned to the word with maximal average distance to its m nearest neighbours. Without quantisation, there is only one such pixel. The second rarest pixels $R_2^{\mathcal{F}_f}$ are defined similarly by excluding the word with maximal distance and so on. The only parameter m merely acts as a filtering value and is of moderate practical importance. Figure 12 shows examples of NN based salient points in a single scale local jet feature space. The difference with the geometric approach is clearly visible on the left image.

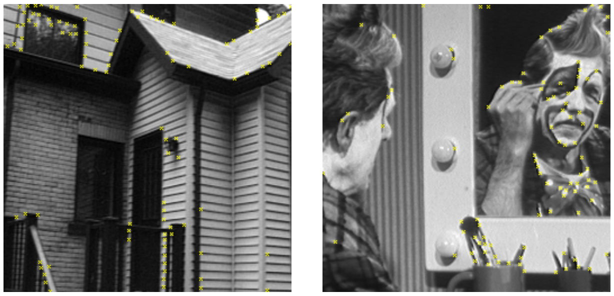

Figure 12. Salient points (isolated points in the feature space back-projected in the image): 100 rarest pixels ($m = 10$, Local jet of order 2, one single scale $\sigma = 1.5$, no quantisation); a minimal exclusion distance of 5 pixels is used to avoid clustering of the salient pixels.

Finally we propose another descriptor whose purpose is to provide an intermediate representation between the global codebook histogram and local salient point. It is based on the statistical modes of the feature space. Mode selection in multidimensional data is a difficult problem which has received relatively few attention. We use an adaptation of the method proposed by Burman and Polonik [21], implemented through the framework of geodesic reconstruction in the feature space.

Suppose defined a topology in the feature space, and let the centre of the main cluster $\kappa_1^{\mathcal{F}_f}$ be defined as the feature vector with minimal average distance to its m NN. The main cluster $K_1^{\mathcal{F}_f}$ is then defined as the connected component of $\mathcal{F}_f$ that contains $\kappa_1^{\mathcal{F}_f}$, or equivalently the geodesic reconstruction of $\kappa_1^{\mathcal{F}_f}$ within $\mathcal{F}_f$. The second main cluster $K_2^{\mathcal{F}_f}$ is defined the same way on $\mathcal{F}_f \setminus K_1^{\mathcal{F}_f}$, and so on. Now we get a topology which dynamically adapts to the data by using a distance threshold defined as the geometric mean between $\mu_m^{\mathcal{F}_f}$ and $\tau_m^{\mathcal{F}_f}$, respectively the average and minimal mean distance of a feature vector to its m NN. Then two feature vector $\mathbf{u}$ and $\mathbf{v}$ are connected if and only if: $d_F(\mathbf{u},\mathbf{v}) < \sqrt{\mu_m^{\mathcal{F}_f}\tau_m^{\mathcal{F}_f}}$. Figure 13 shows the result for 2 images. The modes appear as a complementary information of singularities (Figure 12). They represent homogeneous zone, simple regular textures, or long straight contours.

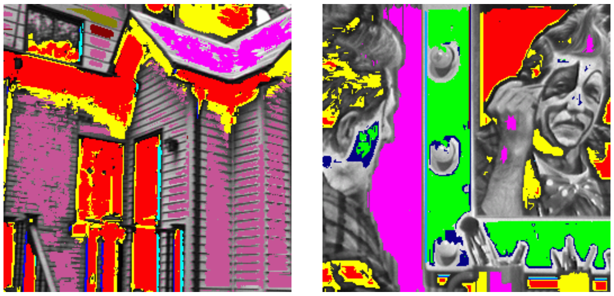

Figure 13. 12 first modes of the local jet representation : clusters of the feature space back-projected in the image space. ($m = 20$, local Jet of order 2, with 2 scales $\{\sigma_1 = 1.0, \sigma_2 = 2.0\}$, no quantisation).

VIII. Conclusion and discussion

We have proposed a unified framework to address: (1) a large variety of video processing applications with the same formalism, by projection, distance based calculation in the feature space, and back-projection in the image space, and (2) a higher level visual characterization obtained by searching significant structures in the feature space (singularities and modes).

Regarding the choice of the feature space, the same framework can probably be used with other features, like wavelets, steerable or Gabor filters. However, the local jet space is easier to justify because of the Taylor expansion. It is also one of the most general because it implicitly contains many other features.

We have shown the relevance of the approach for several low level vision tasks. This representation also naturally provides image reduction and description tools that can be used at a higher processing level. We particularly think to object modelling and recognition, which is part of our ongoing work.

The presented work also contains more specific contributions, that we recall hereunder:

- The definition of distances in the local jet space, which, although proposed earlier, had not been used in practice to our knowledge.
- The local jet based NL-Mean filters, which can be seen as a continuum between tone space filtering and patch based NL-Means by increasing the order of derivation of the local jet.
- The optical flow solution as a nearest neighbour search in a similarity space.
- The singularities (isolated points) of the feature space, as way to fuse the detection and the characterization of interest points, classically addressed independently.
- The mode detection in the feature space, as a complementary information to salient (singular) features.

Because the aim of this work is to find a vision framework as universal as possible, we do not expect every application to compete with state-of-the-art dedicated algorithms. Experimental results were shown in this paper to convince that the framework makes sense, but obviously further evaluation is needed in every single case. The same applies for some parameters which were chosen either according to similar algorithm from the literature or empirically.

Some of the proposed algorithms, for example local jet based NL-means and background subtraction based on sample and consensus in the local jet space are particularly efficient and can be easily adapted to real-time. However, the computational cost remains an issue for different implementations: the optical flow by nearest neighbour search in the local jet space and the computation of the mode of the local jet distribution are two important examples. We are then investigating new ways to compute the nearest neighbours in the feature space using parallel implementations.

REFERENCES

[1] G. Peyré, “Manifold models for signals and images,” *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.

[2] W. Freeman and E. Adelson, “The design and use of steerable filters,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[3] Y. Rubner and C. Tomasi, “Texture-based image retrieval without segmentation,” in *Proc. ICCV*, Kerkyra, Greece, 1999, pp. 1018–1024.

[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in *Workshop on Statistical Learning in Computer Vision (ECCV’04)*, 2004, pp. 1–22.

[5] K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors,” *Int. J. of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[6] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] J. Koenderink and A. Van Doorn, “Representation of local geometry in the visual system,” *Biological Cybernetics*, vol. 55, pp. 367–375, 1987.

[8] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–534, 1997.

[9] M. Crosier and L. Griffin, “Using basic image features for texture classification,” *Int. J. of Computer Vision*, vol. 88, no. 3, pp. 447–460, 2010.

[10] L. Florack, B. Ter Haar Romeny, M. Viergever, and J. Koenderink, “The Gaussian scale-space paradigm and the multiscale local jet,” *Int. J. of Computer Vision*, vol. 18, no. 1, pp. 61–75, January 1996.

[11] J. Orchard, M. Ebrahimi, and A. Wong, “Efficient non-local means denoising using the SVD,” in *Proc. ICIP*, 2008, pp. 1732–1735.

[12] T. Lindeberg, “Feature detection with automatic scale selection,” *Int. J. of Computer Vision*, vol. 30, no. 2, pp. 77–116, 1998.

[13] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Com. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[14] D. Mount and S. Arya, “ANN: A library for approximate nearest neighbor searching,” in *CGC Workshop on Computational Geometry*, 1997, http://www.cs.umd.edu/~mount/ANN/.

[15] A. Buades, B. Coll, and J. Morel, “A non-local algorithm for image denoising,” in *Proc. CVPR*, vol. 2, 2005, pp. 60–65.

[16] A. Manzanera, “Local jet based similarity for NL-means filtering,” in *Proc. ICPR*, Istambul, Turkey, 2010, pp. 2668–2671.

[17] H. Wang and D. Suter, “A consensus-based method for tracking: Modelling background scenario and foreground appearance,” *Pattern Recognition*, vol. 40, no. 3, pp. 1091–1105, 2007.

[18] O. Barnich and M. Van Droogenbroeck, “ViBe: a powerful random technique to estimate the background in video sequences,” in *Proc. ICASSP*. IEEE, April 2009, pp. 945–948.

[19] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Background modeling and subtraction by codebook construction,” in *Proc. ICIP*, vol. 5. IEEE, 2004, pp. 3061–3064.

[20] C. Kervrann and J. Boulanger, “Local adaptivity to variable smoothness for exemplar-based image denoising and representation,” *Int. J. of Computer Vision*, vol. 79, no. 1, pp. 45–69, August 2008.

[21] P. Burman and W. Polonik, “Multivariate mode hunting: Data analytic tools with measures of significance,” *J. Multivar. Anal.*, vol. 100, no. 6, pp. 1198–1218, 2009.