IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 30, 2020, accepted August 6, 2020, date of publication August 12, 2020, date of current version August 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3016030

Study of Convolutional Neural Networks for Global Parametric Motion Estimation on Log-Polar Imagery

V. JAVIER TRAVER[1] **AND ROBERTO PAREDES**[2]
[1]Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castelló, Spain
[2]Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, 46022 Valencia, Spain

Corresponding author: V. Javier Traver (vtraver@uji.es)

This work was supported in part by the Universitat Jaume I, Castellón, Spain, through the Pla de promoció de la investigació, under Project UJI-B2018-44; and in part by the Spanish Ministerio de Ciencia, Innovación y Universidades through the Research Network under Grant RED2018-102511-T.

ABSTRACT The problem of motion estimation from images has been widely studied in the past. Although many mature solutions exist, there are still open issues and challenges to be addressed. For instance, in spite of the well-known performance of convolutional neural networks (CNNs) in many computer vision problems, only very recent work has started to explore CNNs to *learning* to estimate motion, as an alternative to manually-designed algorithms. These few initial efforts, however, have focused on conventional Cartesian images, while other imaging models have not been studied. This work explores the yet unknown role of CNNs in estimating global parametric motion in log-polar images. Despite its favourable properties, estimating some motion components in this model has proven particularly challenging with past approaches. It is therefore highly important to understand how CNNs behave when their input are log-polar images, since they involve a complex mapping in the motion model, a polar image geometry, and space-variant resolution. To this end, a CNN is considered in this work for regressing the motion parameters. Experiments on existing image datasets using synthetic image deformations reveal that, interestingly, standard CNNs can successfully learn to estimate global parametric motion on log-polar images with accuracies comparable to or better than with Cartesian images.

INDEX TERMS Convolutional neural networks, log-polar images, motion estimation, parametric motion models.

I. INTRODUCTION

Motion estimation from image sequences has been a long-standing problem in computer vision [1]–[5], with many approaches being investigated to deal robustly with the challenging real-world conditions. Parametric global motion are one family of such approaches, and it has also been investigated in log-polar images. Log-polar imaging is a foveal-like spatial sampling of the visual scene where information is acute at the center of the visual field but resolution decreases towards the periphery. This biologically-inspired selection of information brings benefits in some visual problems, most notably in robotics [6], since it offers an interesting trade-off solution between three competing factors: width of the field of view, spatial resolution, and amount of data to process. Beyond significant computational saving, the log-polar and other space-variant resolution strategies allow for storage- and energy-economic solutions, which are particularly important in low-power scenarios such as mobile applications [7]. Higher noise tolerance has also been reported from algorithms based on log-polar images [8], [9]. However, these advantages also come with some challenges. For instance, for motion estimation, which is the problem addressed in this work, the non-conventional space-variant resolution, and the polar-like geometry preclude many existing methods developed for Cartesian images from being (directly) applicable.

One of the main and well-known benefits of the log-polar sampling for motion estimation tasks is that rotation and scaling can map to a simple space-invariant translation, a property often known as edge invariance. However, a simple translation in the Cartesian domain maps to a complex non-linear deformation in the cortical domain [10]. Actively

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani.

tracking objects can facilitate the problem by keeping retinal shift small [11], but this concept is not easy to exploit for motion models that are more complex than the similarity motion model [12]. Furthermore, motion estimation for models with components like shear, has proven to be hard, at least following some algorithms [9]. Some other more general approaches [13] benefit from the computational saving and the implicit focus of attention, but may not naturally exploit properties of the log-polar images that are more motion specific. Hybrid solutions [8] that combine Cartesian and log-polar images are interesting, but cannot be applied in some practical scenarios such as when log-polar images come directly from a foveal sensor.

Besides the direct and global use of log-polar images as input to algorithms, another interesting application of log-polar sampling is in the scope of local image detectors and descriptors. Although the concept has been used in the past (e.g. [14], [15]), it has recently been revisited and shown to provide suitable representations for learning descriptors [16], resulting in more robust matching of local descriptors across larger scale ranges. By leveraging on the scale and rotation invariance of log-polar images applied to local patches, and including a mechanism to mimic biological visual saccades, an object recognition system has recently been suggested [17].

Other space-variant image models [18]–[20] or sampling techniques [21], [22] have been proposed to more efficiently or effectively address specific visual problems or data. Therefore, although this work focuses on the log-polar imaging model, because of its biological motivation, its popularity, and its suitability in egomotion estimation and several other problems in robotics, the issue explored in this work may also be relevant to other imaging and visual sampling approaches in a variety of application domains. It is important to note that, in addition to its natural application for robot navigation, egomotion has shown to play a role as a supervisory signal [23], [24], for representational learning, which is a topic of much recent interest (e.g. [25], [26]).

With the advent of deep learning and convolutional neural networks (CNNs), the possibility of avoiding manually engineered solutions and learn the motion estimation task from training examples in a unified way that is both problem- and geometry-agnostic, is certainly intriguing and interesting. This is particularly true for the most challenging problems, like the unconventional geometry of log-polar images.

In the recent few years, significant work has been done on the *local* motion estimation in terms of learning optical flow estimation. Besides the initial approach and its derivations [27], [28], others proposed constraining weights to facilitate learning [29], or imposed strong application-based priors [30]. Some alternatives include pyramidal solutions [31], [32], and learning *while* predicting [33]. Lately, more powerful and innovative solutions have emerged, following the trend of unsupervised solutions [34] or self-supervision [35], [36], and jointly estimating flow, depth, camera motion, and motion segmentation [37].

Comparatively to optical flow, the *global* motion estimation (i.e. estimating a motion affecting the whole image) with CNNs has been much less explored, either using optical flow as input [38], exploring unsupervised [39] or hierarchical approaches [40], or combining motion and depth estimation [41]. After these first initial efforts, there are still many open issues. For instance, given that CNNs have essentially been designed for uniformly sampled Cartesian images, it is not obvious whether they can also be used to estimate motion on images with other geometric layouts such as the log-polar one. Work on CNNs tangentially related to the one presented in this article is limited, and explores specific (non-standard) CNN architectures, and focuses on spatial invariances [42], learning task-specific sampling [43], or recognition tasks [44], and thus have no straightforward connection to motion estimation on log-polar images.

Therefore, towards filling this knowledge gap, the main contribution of this work is a first exploration of whether a simple standard CNN can deal with image deformations on these particular and less conventional log-polar images, using their cortical representation. To the best of our knowledge, this has not been studied before, despite the interest of log-polar imaging for robotics and for computational and biological vision sciences at large. Additionally, this work compares the performance of CNNs with log-polar and Cartesian images, and analyses a few other issues of practical interest. Note that it is not a goal of this work to propose a novel approach or get superior performance with log-polar images, but to gain some understanding of the possibilities and limitations of CNNs with log-polar imaging.

II. METHODOLOGY

Our study considers a parametric motion model (Sec. II-A) to produce and estimate global motion, a log-polar mapping (Sec. II-B) to generate log-polar images from conventional Cartesian images, and a training procedure (Sec. II-C) where synthetic deformations are produced to train a convolutional neural network (Sec. II-D) whose prediction performance is evaluated with geometric error metrics (Sec. II-E).

A. MOTION MODEL

The following motion parameters are considered in this work: translational components (horizontal, t_x, and vertical, t_y), rotation (θ), change of scale factor (α), and shear (β). The following simplified affine model is used,

$$\mathbf{f} \equiv \begin{cases} x' = \alpha \cdot \cos\theta \cdot x - \alpha \cdot \sin(\theta + \beta) \cdot y + t_x \\ y' = \alpha \cdot \sin\theta \cdot x + \alpha \cdot \cos(\theta + \beta) \cdot y + t_y, \end{cases} \tag{1}$$

which maps any point $\mathbf{p}(x, y)$ in one image I_1 to the corresponding point $\mathbf{p}'(x', y')$ in another related image I_2. In our case, the images come from a sequence, and therefore points and images are related through the motion model $\mathbf{f}$ with a motion parameter vector $\mathbf{m} = [t_x, t_y, \theta, \alpha, \beta]$ such that $I_2 = \mathbf{f}(I_1; \mathbf{m})$ and $\mathbf{p}' = \mathbf{f}(\mathbf{p}; \mathbf{m})$. Note that the zero-motion values for these parameters are $\mathbf{m}_0 = [0, 0, 0, 1, 0]$. From

this general model we will instantiate and test with different motion models, defined as the combination of a subset of motion parameters and particular ranges for them.

B. LOG-POLAR MAPPING

In this work, log-polar images are generated from input Cartesian images of size $M \times N$, with $M = N = 128$, following the log-polar transform [6] from Cartesian coordinates (x, y) to log-polar coordinates (u, v),

$$\begin{cases} u = \left\lfloor \log_a \left(\dfrac{\rho}{\rho_0} \right) \right\rfloor \\ v = \left\lfloor \dfrac{S}{2\pi} \phi \right\rfloor, \end{cases} \tag{2}$$

where u and v correspond, respectively, to the eccentricity and angular axes in the retinal domain. The polar coordinates (ρ, ϕ) are used as intermediate variables to simplify (2),

$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \phi = \arctan \dfrac{y}{x}. \end{cases}$$

The remaining parameters are ρ_0, the size of a central blind spot that is left unmapped; a, the radial growth factor of the receptive fields, and $\rho_{\max}$, the maximum radius considered. The size of the resulting log-polar image (also known as cortical representation) is $R \times S$, with R and S being the number of concentric rings (axis u) and angular sectors (axis v) of the resulting log-polar images. The number of sectors S directly relates to the angular resolution, as in (2). The number of rings R relates to ρ_0, $\rho_{\max}$ and a by substituting u and ρ in (2) by their largest values (i.e. for the outermost ring), $u = R$ and $\rho = \rho_{\max}$.

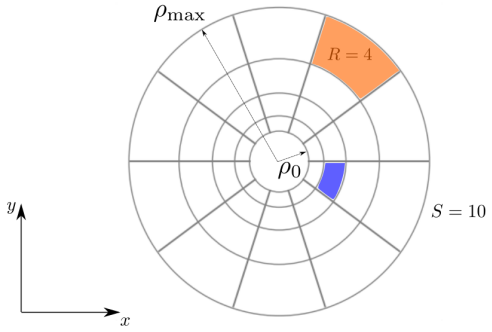
In addition to the geometric transformation from Cartesian coordinates (x, y) to log-polar coordinates (u, v), the log-polar mapping involves an image sampling where photometric values inside the corresponding regions in the original space, the so-called receptive fields, are averaged to define the values in the corresponding log-polar pixel. These concepts are illustrated in Fig. 1 for an example with $R = 4$ rings and $S = 10$ sectors.

We set $\rho_{\max} = \min(M, N)/2$, the size of the blind spot as $\rho_0 = 5$, and a maximum oversampling of 4, and then follow the design criteria of having receptive fields of unit aspect ratio [45]. Solving for these constraints results in $R = 27$ and $S = 64$, which we approximate to $R = 30$ and $S = 60$. In this article, Cartesian 128×128 images will be referred to as **C** images, and the 30×60 log-polar images as **LP** images. An example of log-polar mapping from an actual input Cartesian image is shown in Fig. 2.

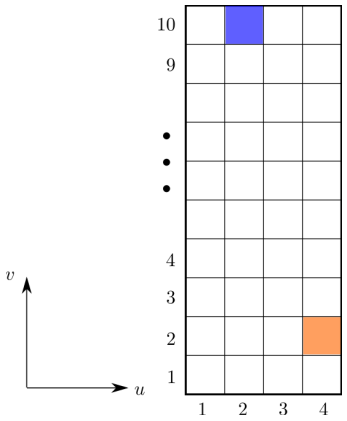
C. TRAINING PROCEDURE

To generate synthetic motion, motion parameters $\mathbf{m} \in \mathbb{R}^m$ are sampled at random uniformly in a given range $\left[p_i^{\min}, p_i^{\max}\right]$ for each parameter $i \in \{1, \ldots, n\}$ are applied to training images. A central window of $M \times N$ pixels is cropped from the larger $M' \times N'$ original and the deformed images. In the case of using log-polar images, the log-polar mapping is also performed, and the original Cartesian images are then ignored. After that, the gray levels of input images are scaled to the range [0, 1]. Each ground-truth motion parameter is linearly normalized from their tested range to $[-1, 1]$. Figure 3 formalises these steps.

A CNN (Sec. II-D) is trained on E epochs (iterations) of batches of B image pairs $(I_1, \mathbf{f}(I_1; \mathbf{m})) = (I_1, I_2)$ as input, using $\mathbf{m}$ as the ground-truth vector of motion parameters. A set of m_{tr} training images is used as a pool from where images are picked at random. Therefore, a total of $m_i = E \cdot B$ training instances are used, which imply that, on average, $r = m_i/m_{tr}$ image deformations per training image are applied. We used $E = 9,600$ and $B = 64$, and thus $m_i = 614,400$.



(a) Receptive fields in Cartesian space (x, y)

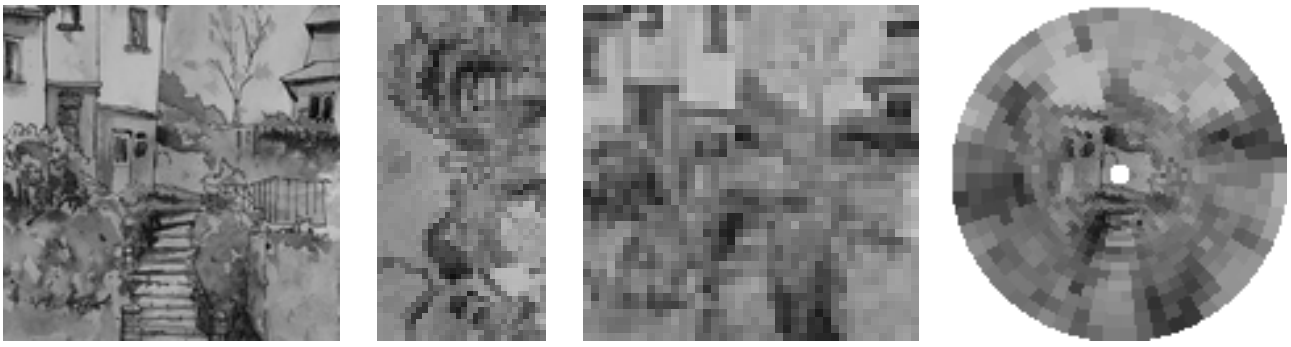


(b) Log-polar pixels in cortical domain (u, v)

FIGURE 1. Main elements that define the log-polar transform. Two example receptive fields are marked in orange and blue, which correspond, respectively, to the pixels in the log-polar image at ($u = R = 4$, $v = 2$), i.e. at the outermost ring, and at ($u = 2$, $v = S = 10$), i.e. at the last sector.

D. NETWORK CONFIGURATION

A CNN with 4 convolutional (conv) layers and 2 fully connected (FC) layers is used (Table 1). The convolutional filters are of size 3×3 in all convolutional layers, with stride of 1 pixel in both directions. For the first convolutional layer,



(a) C (128×128) (b) LP (30×60) (c) C_s (42×42) (d) $\tilde{\mathsf{C}}$ (128×128)

FIGURE 2. In the experiments, we use (a) original Cartesian images (C), (b) their conversion to log-polar image in the cortical representation (LP), and (c) the resizing of C to the smaller Cartesian C_s to (approximately) match the number of pixels in LP. Image (d) $\tilde{\mathsf{C}}$ is the LP image converted back to Cartesian ("retinal") space, but it is not used in the experiments; it is often useful for visualisation purposes. Images (b) LP and (c) C_s are about 9 times smaller than C, and are shown enlarged in this figure for visualization purposes only.

$I_1 \leftarrow \text{SampleRandomImage}(dataset)$
$\mathbf{m} \leftarrow \text{SampleRandomMotion}\left(\left[p_i^{\min}, p_i^{\max}\right]_{i=1}^{m}\right)$
$I_2 \leftarrow \text{TransformImage}(I_1; \mathbf{m})$
$\mathsf{C}_1 \leftarrow \text{ImageCrop}(I_1; [M, N])$
$\mathsf{C}_2 \leftarrow \text{ImageCrop}(I_2; [M, N])$
$\mathbf{m}_n \leftarrow \text{ScaleParameters}(\mathbf{m}; [-1, 1])$
if using log-polar images **then**
  $\mathsf{LP}_1 \leftarrow \text{LogPolarTransform}(\mathsf{C}_1)$
  $\mathsf{LP}_2 \leftarrow \text{LogPolarTransform}(\mathsf{C}_2)$
  $\mathsf{LP}_1 \leftarrow \text{ScaleGrayLevels}(\mathsf{LP}_1, [0, 1])$
  $\mathsf{LP}_2 \leftarrow \text{ScaleGrayLevels}(\mathsf{LP}_2, [0, 1])$
  return $(\mathsf{LP}_1, \mathsf{LP}_2; \mathbf{m}_n)$
else
  $\mathsf{C}_1 \leftarrow \text{ScaleGrayLevels}(\mathsf{C}_1, [0, 1])$
  $\mathsf{C}_2 \leftarrow \text{ScaleGrayLevels}(\mathsf{C}_2, [0, 1])$
  return $(\mathsf{C}_1, \mathsf{C}_2; \mathbf{m}_n)$
end if

FIGURE 3. Algorithm for the generation of one training instance, which consists of an image pair, either the Cartesian images (C_1, C_2) or the corresponding log-polar images (LP_1, LP_2), along with the corresponding (scaled) ground-truth motion $\mathbf{m}_n$ that geometrically relates the images.

padding is used to get the output of the same size as the input. No padding is applied to the other convolutional layers. After the convolutional layers, a max pooling of 2×2 and then a dropout layer (with a dropout rate of 0.25) is applied. As for the activation functions, the ReLU is used in the convolutional layers, and an hyperbolic tangent for the first FC layer. Since the intended goal of the network is to predict the values of the motion components corresponding to the input images, a regression task is considered, and the loss function $\mathscr{L}$ used is the mean squared error between the true $\mathbf{m} = [t_x, t_y, \theta, \alpha, \beta]$ and predicted motion parameters $\hat{\mathbf{m}} = [\hat{t_x}, \hat{t_y}, \hat{\theta}, \hat{\alpha}, \hat{\beta}]$ in their normalised ranges $[-1, 1]$,

$$\mathscr{L}(\mathbf{m}, \hat{\mathbf{m}}) = \frac{1}{n} ||\mathbf{m} - \hat{\mathbf{m}}||_2^2.$$

TABLE 1. Network topology. The number of units refers to the number of filters in convolutional (conv) layers and to the number of neurons in fully-connected (FC) layers. The size of the last FC layer matches the number of motion parameters n considered. The total number of weights differ in C and LP images because their input sizes are different (Sec. II-B).

Layer	Type	No. units	No. weights	
			C	LP
1	conv	32	608	
2	conv	32	9,248	
3	conv	32	9,248	
4	conv	16	4,624	
5	max-pool	–	–	
6	dropout	–	–	
7	FC	128	7,620,736	663,680
8	FC	n	$128 \cdot n + 1$	
	Total	$n = 1$	7,644,593	687,537
		$n = 5$	7,645,105	688,049

All the networks weights are initialised randomly with a normal distribution. For weight optimisation, the ADADELTA [46] method, which adopts an adaptive learning rate, is used.

E. EVALUATION

The estimation accuracy for each tested motion parameter $p \in \mathbf{m}$ for a sampling set of m_{te} testing instances is evaluated with the mean absolute error (MAE),

$$\text{MAE} = \frac{1}{m_{te}} \sum_{i=1}^{m_{te}} |p^{(i)} - \hat{p}^{(i)}|.$$

To account for the error in relation with the magnitude of the true value of the parameter, the mean relative error (MRE) is also defined,

$$\text{MRE} = \frac{1}{m_{te}} \sum_{i=1}^{m_{te}} \frac{|p^{(i)} - \hat{p}^{(i)}|}{|p^{(i)}|}.$$

We used $m_{te} = 10,000$.

To quantify the estimation performance with a single measure even if several motion parameters are involved, a unified geometric measure, the end-point error (EPE), is defined over a set of n_p "canonical" points $\{\mathbf{p}_j\}_{j=1}^{n_p}$ as the deviation between the true target points $\mathbf{p}_j^{(i)} = \mathbf{f}(\mathbf{p}_j; \mathbf{m}_i)$ and the estimated ones $\hat{\mathbf{p}}_j^{(i)} = \mathbf{f}(\mathbf{p}_j; \hat{\mathbf{m}}_i)$ for a given test instance i, $i \in \{1, \ldots, m_{te}\}$, as

$$\mathsf{EPE} = \frac{1}{m_{te}} \sum_{i=1}^{m_{te}} \mathsf{EPE}^{(i)},$$

with

$$\mathsf{EPE}^{(i)} = \frac{1}{n_p} \sum_{j=1}^{n_p} ||\mathbf{p}_j^{(i)} - \hat{\mathbf{p}}_j^{(i)}||_2.$$

We used the four vertices (i.e. $n_p = 4$) of a unit square centered at $(0, 0)$ as an arbitrary choice for these canonical points. The canonical points as used in the computation of EPE are illustrated in Fig. 4.

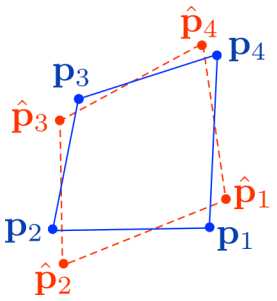


FIGURE 4. How the end-point error (EPE) is computed. The blue points $\mathbf{p}_j$ correspond to the canonical points after the ground truth motion $\mathbf{m}$, whereas the red points $\hat{\mathbf{p}}_j$ are the canonical points transformed with the estimated motion $\hat{\mathbf{m}}$. The measure EPE is given by the average pairwise distance of the four points $(\mathbf{p}_j, \hat{\mathbf{p}}_j), j \in \{1, 2, 3, 4\}$. For simplicity, the superscript (i) that denotes the test instance has been dropped from $\mathbf{p}_j^{(i)}$.

For any of these three measures (MAE, MRE, EPE), the lower their values, the better the performance.

III. EXPERIMENTS

A. DATASETS AND SOFTWARE

Existing datasets for visual recognition tasks can also be used as a source of images for motion-estimation learning with synthetic deformations. We split the Pascal Visual Object Classes (VOC) dataset [47] into training and test sets using the prefix of the image file names so that we get about 80%-20% ratio (Table 2). In particular, images whose file names begin with `2010` are used for the test set and all the others for training. Similarly, we use Caltech256 [48], with 205 classes for training and 52 for test, and images with any side length lower than 200 pixels are resized to this minimum side length to facilitate the synthetic data generation procedure (Sec. II-C). Python packages for image and numerical computing (Numpy, Scipy, PIL), and the Keras framework [49] for CNNs, were used.

TABLE 2. Datasets used in the experiments.

Dataset	No. images	Training size	Test size
VOC	17,125	13,622	3,503
Caltech256	30,607	24,564	6,043

B. INDIVIDUAL MOTION COMPONENTS

The estimation performances for C and LP are first compared under single-parameter motions, with a discussion of the motion component in these imaging formats.

1) TRANSLATION (t_x)

Although translations are the simplest motion model in Cartesian domain (a constant, space-invariant shift), they map to complex space-variant, non-linear model in the log-polar domain [10]. Despite this complication, LP images result in smaller estimation error than C images (Table 3, first row), which provides some evidence that CNN can naturally and inherently cope with geometries other than Cartesian images *without* any particular design or modification.

2) ROTATION (θ) AND CHANGE OF SCALE (α)

For log-polar images following the model (2), centered rotations and changes of scale map to simple shifts δ_v and δ_u in log-polar images, respectively, due to their aforementioned edge-invariance property. From (2), and given a rotation angle θ and a change of scale α, the values for these shifts, $\delta_v = \frac{S}{2\pi}\theta$, and $\delta_u = \log_a \alpha$, can be derived [12]. For the range of rotations and scale factors considered, this means that we can expect the CNN to deal with *location-invariant* translations of up to $\delta_u = 5$ pixels along u and about $\delta_v = 8$ pixels along v, unlike the Cartesian case. Higher performance is also observed (Table 3, rows 2–3) with LP images than with C images.

TABLE 3. Estimation performance for per-parameter learning.

parameter	range	image	MAE	MRE	EPE
t_x	$[-10, 10]$	C	1.42	1.04	= MAE
		LP	0.36	0.25	= MAE
θ	$[-45°, 45°]$	C	2.92	0.91	0.85
		LP	1.31	0.22	0.68
α	$[0.7, 1.3]$	C	0.0132	0.0139	0.0093
		LP	0.0069	0.0072	0.0049
β	$[-20°, 20°]$	C	0.69	0.32	0.29
		LP	0.66	0.28	0.29

3) SHEAR (β)

Shear can be understood as a combination of rotation and an isotropic scaling, and unlike rotation and scaling, its effect in the cortical plane in LP is not simple, and not straightforward to model. However, the estimation performance with LP is competitively similar to C (Table 3, bottom row).

This per-parameter study and comparison is completed below (Sec. III-D) by performing a per-range analysis, and by including smaller Cartesian images, of lower resolution, into the comparison.

C. INCREASING MOTION MODEL COMPLEXITY

When using more than one parameter, the set of samplings of all the possible ranges of arbitrary motion possibilities increases exponentially with the number of parameters. Therefore, to have estimation accuracies similar to those in the single-parameter cases, a more complex architecture (e.g. a multi-level one), and/or significantly more training effort in terms of amount of data and/or iterations, might be required. Another practical choice would be to learn only a very reduced subset of all possible combinations of parameter ranges, for instance by learning only from expected deformations, e.g. from real or realistic sequences, as in [38], [41]. In our synthetic-deformation scenario, we translate this latter choice into a reduced range of motion values for each involved parameter. When using n motion parameters, the sampling effort grows at the rate w^n, where w measures the range of values sampled for a single motion parameter. This computational burden is dealt with by reducing w.

Results (Table 4) suggest that by reducing the range of values in the parameters when combining them, the estimation performance remains comparable to the single-parameter cases, and LP images still outperform C ones. These results also indicate that the difficulty in the estimation may come more from the *range* of the expected deformations (say, the size of the *search space*) than the fact of having several motion components being combined, and that the CNN can deal with this combinations both with C and LP images. This analysis is completed below by including a variation of the CNN considered (Sec. III-F).

TABLE 4. Estimation performance (EPE) for different motion models.

t_x	θ	image	EPE
$[-10, 10]$	0	C	1.42
		LP	0.36
0	$[-45°, 45°]$	C	0.85
		LP	0.68
$[-5, 5]$	$[-10°, 10°]$	C	0.80
		LP	0.55

D. SMALLER CARTESIAN IMAGES

It has been observed that the estimation performance with C images is inferior to that with LP images, which means that learning is being less effective with C images. The fact that the corresponding CNN has more weights than the log-polar CNN may be a reason, and heavier training or an alternative network might help. Alternatively, one way to reduce the number of weights while keeping the network topology and, at the same time, exploring the impact of different spatial sampling in motion estimation, is to use smaller Cartesian images matching the size of log-polar images, i.e. $M \cdot N \approx R \cdot S$ by setting $M = N = \left\lfloor \sqrt{R \cdot S} \right\rfloor$. For $R = 30, S = 60$, this implies $M = N = 42$. To that end, we keep the field-of-view (FOV) and resize the image from 128×128 to 42×42. Another choice would be cropping a central area of 42×42 from the 128×128 image, thus keeping the original resolution but at the expense of a narrower FOV. With this smaller-sized Cartesian image inputs, the resulting network has the same number of weights as the network with LP images as input. We refer to these smaller Cartesian images as C_s (Fig. 2)c.

We repeated the tests reported before (Sec. III-B, Table 3), for C_s images, and a more detailed analysis was performed in terms of the deformation range of the true motion parameters (roughly, ''small'', ''medium'' and ''large'' deformations). For each parameter p, the distribution of the estimation error is visualized with kernel density estimation of $e = p - \hat{p}$, so the more peaked the distribution at $e = 0$, the better the performance (i.e. lower errors happen more often).

Results for t_x (Fig. 5) indicate that C_s images outperform both C and LP images for the medium and large translations. This can be explained because of C_s's smaller resolution. Additionally, and somehow surprisingly, for the smaller translations, the performance of C_s is similar to that of LP, which can be explained by a remarkable ability of the network to estimate sub-pixel image shifts: a translation of $t_x = 1$ pixel in the original C represents a motion of only about a third of a pixel in C_s ($42/128 \approx 0.33$). These plots also reveal the difficulty of C images, particularly for the larger translations.

For θ, C_s images have a performance in between C and LP images for the medium and large ranges of the tested rotations, and similar to C for the smaller rotations. This behaviour can be explained by taking the lower resolution of C_s in mind: it can cope better with larger rotations than with more subtle ones.

Regarding α, performance with C_s is similar or better than that with C for the larger scale changes, but worse than C for smaller scale changes. Both in θ and α, LP outperforms both C and C_s for all ranges of the tested ranges, which is in agreement of the expected benefit of the edge invariance property of LP images.

Since C_s images have been shown to outperform C ones, LP images are compared only to C_s ones from now on.

E. CONVOLUTIONAL FILTERS

It is interesting to inspect the activation maps corresponding to different motion components for different image samplings. These maps are a form of visualising what convolutional filters have learned to represent the underlying motion, and it can be noted they are motion- and image-sampling-specific (Fig. 6). In same cases it can be appreciated that the image areas that are mostly activated correspond to structures roughly orthogonal to the relevant motion. For instance, for the horizontal translation, vertical-like blobs are activated, whereas under change of scale (zoom), radially-distributed

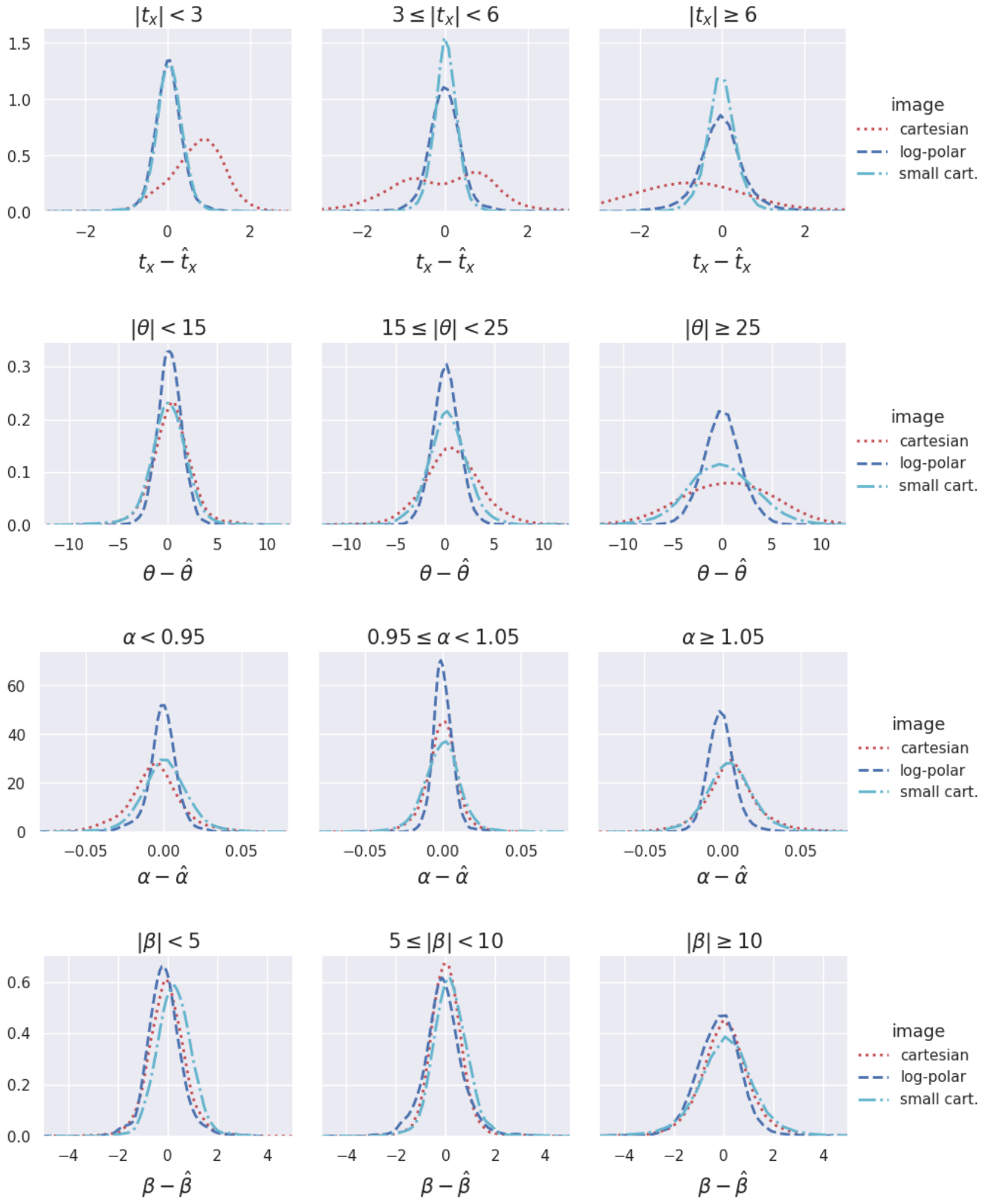


FIGURE 5. Per-parameter, per-range estimation error.

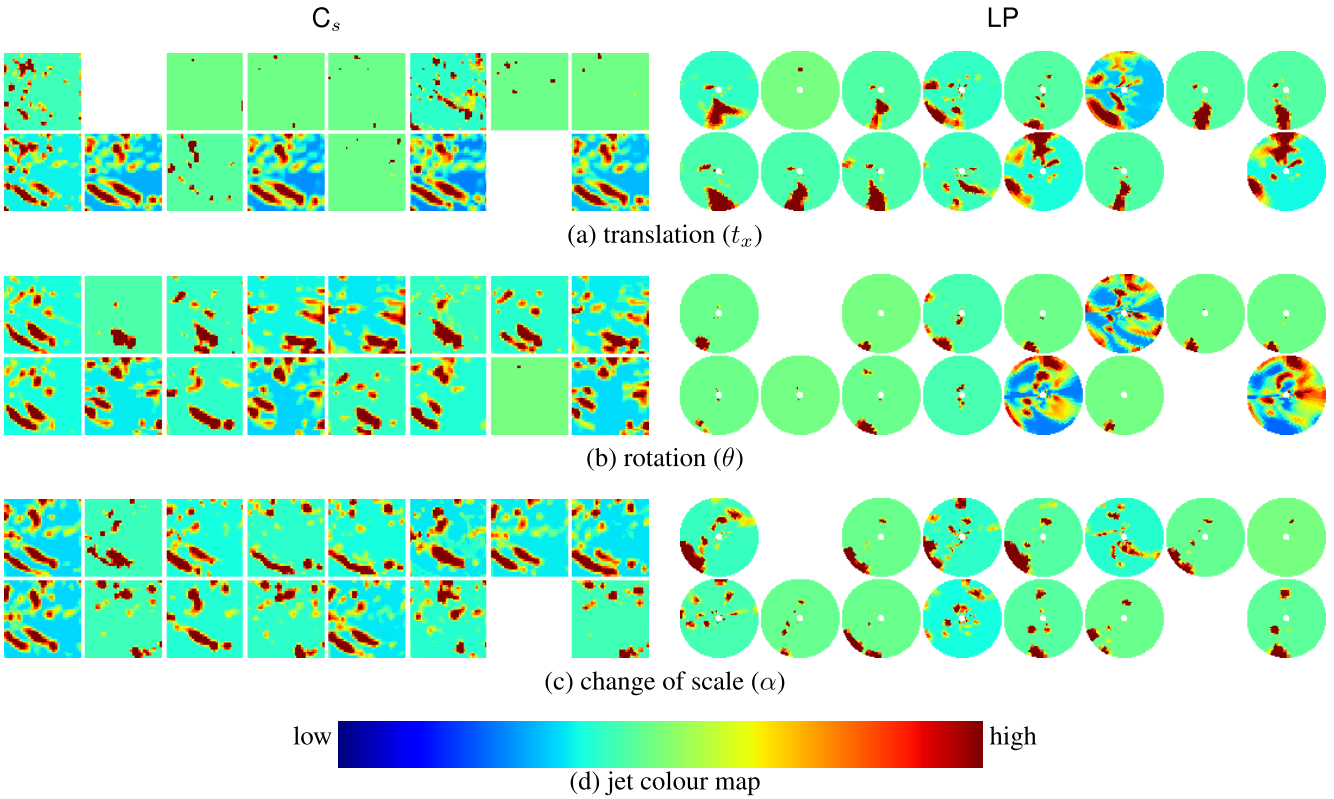


FIGURE 6. Activation maps after the fourth convolutional layer of networks trained for different motion models (a,b,c) and image samplings (left: C_s, right: LP) using as input the images corresponding to the same source image pair and motion parameter values. For the LP case, the actual maps are in the cortical domain, but the inverse log-polar mapping (Fig. 2d) is performed for visualisation. For visualisation purposes the activation maps have been normalised and the jet colour map (d) used.

activations may emerge. For this particular input image, these observations are somehow more noticeable in the log-polar case.

Another important aspect has to do with how apt each image geometry is for each motion component. As discussed earlier, Cartesian translations are more naturally estimated with Cartesian images than with log-polar images. As a consequence, one or more of the following three facts tend to happen: (1) there are fewer activation maps with significant responses, (2) lower activations within some more maps, or (3) more similar maps, in the Cartesian case than in the log-polar case. Understandably, due to the space-variant and polar-like nature of log-polar images, more diversity can be appreciated (Fig. 6a). For the rotation and change of scale (Fig. 6b–c), the situation is somehow the opposite, since these motion components can be more easily characterised in the log-polar representation.

To study this hypothesis that more filters are generally required to capture how translations map to different regions of a log-polar image, the number of convolutional filters were halved in all the convolutional layers, and the resulting network retrained. It was found (Table 5) that the estimation performance degraded significantly more (error increased by 25%) for log-polar images than for Cartesian images (about 4%). Therefore, the convolutional filters are arguably one of the ingredients for the CNNs to implicitly cope with space-variant image geometries.

TABLE 5. Change in the estimation performance (EPE) for motion $t_x \in [-10, 10]$ on C_s and LP images after halving the default number of convolutional filters at each layer.

No. of filters	C_s	LP
default	0.27	0.36
half	0.28	0.45
error increase (%)	3.7	25.0

F. EARLY VS MIDDLE MERGING

The network architecture used in previous tests (Sec. III-B–III-E) consists of having the two motion-related gray-level images as two channels of the input to the network (Sec. II-D); we call this choice the "early" merge, since input images are merged (concatenated) early along the network's depth. The rationale behind this is that low-level features relating the two images as a two-channel input can be captured with the learnable convolutional filters from the very beginning, and this can be beneficial. Alternatively, it can be argued that it can be better to relate higher-level features at later layers and let parallel siamese branches

of the network learn these features for each input image before merging (concatenating) the corresponding feature channels. To test this hypothesis, we tried keeping the input images separated and merging their branches after the second convolutional layer. We refer to this choice as middle (''mid'') merge. We experienced convergence issues in our tests when merging at later layers (after the first FC layer), and therefore only the early and mid merges are compared.

Results (Table 6) suggest that the mid merge does not generally bring a performance benefit over the early merge: some performance decay can actually be observed, with some exception (C_s and the model with parameters $\{t_x, \theta, \alpha\}$). The increase in the complexity of the motion models affects similarly the C_s and LP images, and the merge choice does not help in this sense, although it seems that the trend is that with more complex motions and C_s, the mid merge might represent a competitive or better choice over early merge. Despite that LP exhibits better estimation performance for particular ranges of individual motion components (Fig. 5), C_s images offer generally lower mean EPE than LP do, although results with LP can be *on par* or even superior in some case. This may relate to the fact of conventional CNNs being generally more effective for Cartesian images since the former were essentially designed for the latter. This implies that some adaptation of the CNN design might be required to make the most of the particular characteristics of the log-polar images, an hypothesis left as further work.

TABLE 6. Estimation performance (EPE) at early and mid merge for C_s and LP images, and different motion models.

t_x	θ	α	merge	C_s	LP
$[-10, 10]$	0	1	early	**0.27**	0.36
			mid	0.38	0.49
$[-5, 5]$	$[-10°, 10°]$	1	early	0.59	**0.55**
			mid	0.61	0.78
$[-5, 5]$	$[-10°, 10°]$	[0.95,1.05]	early	1.11	0.87
			mid	**0.81**	0.92

G. CROSS-DATASET PERFORMANCE

Previous tests have used images from the VOC dataset. Although the training and test sets used are disjoint, images within the same dataset may have some bias [50]. Thus, to better evaluate the impact of the image contents and the generalization ability, we include results when training and testing with Caltech256 as well. With respect with training and testing on the same dataset A, when training on dataset A and testing on dataset B, results (Table 7) turn out to be poorer for $A =$ VOC and $B =$ Caltech256, but better for $A =$ Caltech256 and $B =$ VOC. This suggests that the generalization ability in this case depends more on the dataset than on the network itself. For the purpose and procedure of our work, images in Caltech256 might be more varied or richer, which poses a challenge not only when training on the VOC dataset but also on other classes of Caltech256 itself. Consequently, after training with these more helpful Caltech256 images, the VOC test images are found comparatively ''easier''. This observation essentially holds for both C_s and LP. However, when comparing C_s and LP in this cross-dataset scenario, the decay in performance is more severe with LP images than with C_s images in the ($A =$ VOC, $B =$ Caltech256) case, and similar in the other case ($A =$ Caltech256, $B =$ VOC). The quality of the training images is therefore very important but may affect differently distinct image sampling choices.

TABLE 7. Cross-dataset (VOC and Caltech256) estimation performance (EPE) for early merge, and motion model [$t_x \in [-5, 5]$, $\theta \in [-10°, 10°]$], using the same number of training and test images for Caltech256 as used in VOC (Table 2).

training	image	test	
		VOC	Caltech256
VOC	C_s	0.59	0.66
	LP	0.55	1.05
Caltech256	C_s	0.60	0.63
	LP	0.67	0.71

TABLE 8. Estimation performance (EPE) for varying ratio r of training instances m_i to training images m_{tr} (Sec. II-C), by fixing m_i and varying m_{tr} for Caltech256, using early merge, and the same motion model used in the cross-dataset test (Table 7).

m_{tr}	r	C_s	LP
2,000	307.2	0.72	0.80
6,000	102.4	0.92	0.73
13,622	45.1	0.63	0.71
24,000	25.6	0.62	0.68

H. VARIETY OF IMAGES VS DEFORMATIONS

For learning purposes, it can be expected that both the number of training images m_{tr} and the number of deformations m_i are important, so that a variety of both image contents and image deformations are observed. However, it is unclear which of these two ingredients affects more the estimation performance. For instance, a reasonable hypothesis might be that sampling many deformations of a few images can be better than sampling fewer deformations of more images. To gain some insight in this respect, we varied the number of training images for the same number of training instances, and results (Table 8) suggest that although a variety of images is important, there seems to be a point of diminishing returns, where more images do not bring a significant performance improvement. In our case, this happens at $m_{tr} \approx 13,000$ in C_s and at $m_{tr} \approx 6,000$ in LP.

IV. CONCLUSION

Convolutional neural networks (CNNs) have essentially been designed having Cartesian images in mind, and it is still largely unknown how they behave with other imaging models. Experiments with existing image datasets and synthetic deformations reveal that CNNs with log-polar images as input

perform reasonably well in learning to estimate parametric global motion, despite the polar geometry of these images, their space-variant resolution, and the non-linear mapping of motion in the cortical domain. This can partially be explained by the flexibility offered by multiple learnable convolutional filters, which allows CNNs to cope with space-variant motion effects. In comparison with Cartesian images, and using exactly the same CNN topology, the estimation performance has been found to be similar, log-polar images outperforming the Cartesian ones, or vice versa, depending on the particular experimental conditions or motion ranges considered. The relevance of this work goes beyond the particular problem of motion estimation, since it provides evidence that CNNs and foveal imaging can work in tandem, thus encouraging further investigation. Future work may address whether specific CNN architectures can be more suitable for log-polar images, and whether fusing different imaging models might bring some advantages by leveraging the benefits of each.

REFERENCES

[1] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992.

[2] C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Process. Mag.*, vol. 16, no. 4, pp. 70–91, Jul. 1999.

[3] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[4] H. Steiner, H. Sommerhoff, D. Bulczak, N. Jung, M. Lambers, and A. Kolb, "Fast motion estimation for field sequential imaging: Survey and benchmark," *Image Vis. Comput.*, vol. 89, pp. 170–182, Sep. 2019.

[5] S.-H. Park and J.-W. Kang, "Fast affine motion estimation for versatile video coding (VVC) encoding," *IEEE Access*, vol. 7, pp. 158075–158084, 2019.

[6] V. J. Traver and A. Bernardino, "A review of log-polar imaging for visual perception in robotics," *Robot. Auton. Syst.*, vol. 58, no. 4, pp. 378–398, Apr. 2010.

[7] E. Silva, R. da S. Torres, A. Pinto, L. Tzy Li, J. E. S. Vianna, R. Azevedo, and S. Goldenstein, "Application-oriented retinal image models for computer vision," *Sensors*, vol. 20, no. 13, p. 3746, Jul. 2020.

[8] S. Zokai and G. Wolberg, "Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1422–1434, Oct. 2005.

[9] V. J. Traver and F. Pla, "Motion analysis with the radon transform on log-polar images," *J. Math. Imag. Vis.*, vol. 30, no. 2, pp. 147–165, Feb. 2008.

[10] V. J. Traver and F. Pla, "Dealing with 2D translation estimation in log-polar imagery," *Image Vis. Comput.*, vol. 21, no. 2, pp. 145–160, Feb. 2003.

[11] H. Tunley and D. Young, "Dynamic fixation of a moving surface using log polar sampling," in *Proc. Brit. Mach. Vis. Conf.*, 1994, pp. 57.1–57.10.

[12] V. J. Traver and F. Pla, "Similarity motion estimation and active tracking through spatial-domain projections on log-polar images," *Comput. Vis. Image Understand.*, vol. 97, no. 2, pp. 209–241, Feb. 2005.

[13] A. Bernardino, J. Santos-Victor, and G. Sandini, "Foveated active tracking with redundant 2D motion parameters," *Robot. Auton. Syst.*, vol. 39, nos. 3–4, pp. 205–221, Jun. 2002.

[14] J. S. Belongie, J. Malik, and J. Puzicha, "Shape context: A new descriptor for shape matching and object recognition," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2000, pp. 831–837.

[15] G. Mori and J. Malik, "Recovering 3D human body configurations using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1052–1062, Jul. 2006.

[16] P. Ebel, E. Trulls, K. M. Yi, P. Fua, and A. Mishchuk, "Beyond Cartesian representations for local descriptors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 253–262.

[17] T. Kurbiel and S. Khaleghian, "RetinotopicNet: An iterative attention mechanism using local descriptors with global context," 2020, *arXiv:2005.05701*. [Online]. Available: http://arxiv.org/abs/2005.05701

[18] F. Tong and Z.-N. Li, "Reciprocal-wedge transform for space-variant sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 5, pp. 500–511, May 1995.

[19] M. W. Peters and A. Sowmya, "A real-time variable sampling technique: DIEM," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1, 1998, pp. 316–321.

[20] M. González, A. Sánchez-Pedraza, R. Marfil, J. Rodríguez, and A. Bandera, "Data-driven multiresolution camera using the foveal adaptive pyramid," *Sensors*, vol. 16, no. 12, p. 2003, Nov. 2016.

[21] M. Abdel-Nasser, A. Moreno, and D. Puig, "Temporal mammogram image registration using optimized curvilinear coordinates," *Comput. Methods Programs Biomed.*, vol. 127, pp. 1–14, Apr. 2016.

[22] Q. Dai, H. Chopp, E. Pouyet, O. Cossairt, M. Walton, and A. Katsaggelos, "Adaptive image sampling using deep learning and its application on X-ray fluorescence image reconstruction," *IEEE Trans. Multimedia*, early access, Dec. 9, 2019, doi: 10.1109/TMM.2019.2958760.

[23] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 37–45.

[24] D. Jayaraman and K. Grauman, "Learning image representations tied to ego-motion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1413–1421.

[25] D. Kim, D. Cho, and I. S. Kweon, "Self-supervised video representation learning with space-time cubic puzzles," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8545–8552.

[26] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6707–6717.

[27] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[28] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. CVPR*, Jul. 2017, pp. 1647–1655.

[29] D. Teney and M. Hebert, "Learning to extract motion from videos in convolutional neural networks," in *Proc. ACCV*, 2016, pp. 412–428.

[30] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, and R. Urtasun, "Deep rigid instance scene flow," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3614–3622.

[31] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2720–2729.

[32] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[33] D. Maurer and A. Bruhn, "ProFlow: Learning to predict optical flow," in *Proc. BMVC*, 2018, pp. 1–13.

[34] A. Ahmadi, I. Marras, and I. Patras, "LikeNet: A siamese motion estimation network trained in an unsupervised way," in *Proc. BMVC*, 2018, p. 296.

[35] P. Liu, M. Lyu, I. King, and J. Xu, "SelFlow: Self-supervised learning of optical flow," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4571–4580.

[36] J. Hur and S. Roth, "Self-supervised monocular scene flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7396–7405.

[37] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12240–12249.

[38] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with CNNs for frame-to-frame ego-motion estimation," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 18–25, Jan. 2016.

[39] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar, "Unsupervised deep homography: A fast and robust homography estimation model," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2346–2353, Jul. 2018.

[40] F. E. Nowruzi, R. Laganiere, and N. Japkowicz, "Homography estimation from image pairs with hierarchical convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 904–911.

[41] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6612–6619.

[42] F. J. Henriques and A. Vedaldi, "Warped convolutions: Efficient invariance to spatial transformations," in *Proc. 34th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 70, D. Precup and Y. W. Teh, Eds., Aug. 2017, pp. 1461–1469.
[43] A. Recasens, P. Kellnhofer, S. Stent, W. Matusik, and A. Torralba, "Learning to zoom: A saliency-based sampling layer for neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 52–67.
[44] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, "Polar transformer networks," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–14.
[45] V. J. Traver and F. Pla, "Log-polar mapping template design: From task-level requirements to geometry parameters," *Image Vis. Comput.*, vol. 26, no. 10, pp. 1354–1370, Oct. 2008.
[46] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701 and https://dblp.uni-trier.de/rec/bibtex/journals/corr/abs-1212-5701
[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[48] G. Griffin, A. D. Holub, and P. Perona, "The Caltech 256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2007-001, 2006. [Online]. Available: https://authors.library.caltech.edu/7694/ and https://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001
[49] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io/getting_started/faq/#how-should-i-cite-keras
[50] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1521–1528.

V. JAVIER TRAVER received the B.Sc. degree in computer science from the Universitat Politècnica de València and the Ph.D. degree in computer engineering from Jaume-I University, Castellón, Spain. He is currently an Associate Professor with Jaume-I University. He regularly lectures at undergraduate levels in two degrees (computer engineering, and videogames design and development), and in graduate levels in a master on intelligent systems. His research interests, at the Institute of New Imaging Technologies, include foveal imaging, video analysis, human action and gesture recognition, and egocentric vision.

ROBERTO PAREDES received the Ph.D. degree *(cum laude)* in computer science from the Universidad Politècnica de València (UPV), Spain, in 2003.

In 2000, he joined the Department of Computer Science, UPV. From 2012 to 2016, he was the President of the Spanish Association for Pattern Recognition and Image Analysis (AERFAI). He is currently an Associate Professor with UPV, the Head of the Pattern Recognition and Human Languages Technologies (PRHLT) Research Center, the CTO and the Co-Founder of Solver Machine Learning (a spin-off of the UPV), a Lead Developer of the European Distributed Deep Learning Library (EDDL), and a Valencia AI Ambassador. His current research interests include statistical pattern recognition, machine learning, deep learning, biometrics, large-scale problems, multimedia retrieval, and relevance feedback.