Efficiency Analysis of POC-Derived Bases for Combinatorial Motion Estimation

Alejandro Reyes*, Alfonso Alba**, and Edgar Arce-Santana

Facultad de Ciencias, Universidad Autónoma de San Luis Potosí,
Av. Salvador Nava Mtz. S/N, Zona Universitaria, 78290,
San Luis Potosí, SLP, México
rey_es@ymail.com, fac@fc.uaslp.mx, arce@fciencias.uaslp.mx

Abstract. Motion estimation is a fundamental problem in many computer vision applications. One solution to this problem consists in defining a large enough set of candidate motion vectors, and using a combinatorial optimization algorithm to find, for each point of interest, the candidate which best represents the motion at the point of interest. The choice of the candidate set has a direct impact in the accuracy and computational complexity of the optimization method. In this work, we show that a set containing the most representative maxima of the phase-correlation function between the two input images, computed for different overlapping regions, provides better accuracy and contains less spurious candidates than other choices in the literature. Moreover, a preselection stage, based in a local motion estimation algorithm, can be used to further reduce the cardinality of the candidate set, without affecting the accuracy of the results.

1 Introduction

The problem of optical flow estimation consists in finding the apparent (projected) motion of each object in a sequence of images [1]. This motion does not always correspond to the true displacement of the object in the tri-dimensional scene. As an example, consider an object that moves towards the viewer along the camera axis. As the object approaches, it will appear larger in size and thus the optical flow field will resemble a radial pattern where each pixel seems to be moving outwards from the center of the object. Nevertheless, the estimation of optical flow has numerous applications in computer vision, robot vision, robot and vehicle navigation, non-parametric (smooth) image registration and morphing, video encoding and video stabilization. It is also often used, along with stereo disparity estimation, as the grounds for 3D motion estimation, which is also called *scene flow* [2].

Formally, a given point in a moving object is projected to a point $\boldsymbol{x} = (x(t), y(t))$ in the 2D frame, where t denotes time. The optical flow is thus defined as $\boldsymbol{v}(\boldsymbol{x}, t) = d\boldsymbol{x}/dt$. Since t is usually discrete, one can alternatively write

* A. Reyes was supported by CONACyT scholarship No. 327373.
** This work was partially supported by CONACyT grant 154623.

R. Klette, M. Rivera, and S. Satoh (Eds.): PSIVT 2013, LNCS 8333, pp. 124–135, 2014.

$\boldsymbol{v}(\boldsymbol{x},t) \approx \boldsymbol{x}(t+1) - \boldsymbol{x}(t)$. Therefore, the problem consists in estimating $\boldsymbol{v}(\boldsymbol{x},t)$ from a given sequence of video frames $I(\boldsymbol{x},t)$. In some cases, one is interested in estimating the optical flow only at specific points, which typically correspond to recognizable image features such as edges or corners. In general, one may want to estimate the flow field densely, that is, estimate $\boldsymbol{v}(\boldsymbol{x},t)$ for each pixel $\boldsymbol{x}$.

Many algorithms for the estimation of optical flow have been proposed in the literature. For a comprehensive review of the most relevant methods, please refer to [1, 3]. Most state of the art methods for the estimation of dense optical flow are based on the variational approach proposed by Horn and Schunck [4]. The approach followed by these methods is based on the *optical flow constraint* which states that the color intensity of a given point $\boldsymbol{x}(t)$ does not vary (or varies very smoothly) across time, and the solution usually requires minimizing a energy function $U(\boldsymbol{v}) = M(\boldsymbol{v}) + R(\boldsymbol{v})$ where $M(\boldsymbol{v})$ is a matching term which penalizes the differences between $I(\boldsymbol{x}(t),t)$ and $I(\boldsymbol{x}(t+1),t+1)$ (thus enforcing the optical flow constraint), and $R(\boldsymbol{v})$ is a regularization term which penalizes abrupt spatial changes in $\boldsymbol{v}$ (thus reducing the effects of noise). The energy function originally proposed by Horn and Schunck is quadratic, resulting in a convex optimization problem whose solution can be efficiently found using linear algebra iterative methods, such as Gauss-Seidel. However, this method usually leads to oversmooth solutions where the edges and details of the objects are not well preserved. More recent proposals rely on more complex energy functions which provide accurate estimations, but require non-linear optimization techniques whose implementation is more difficult and computationally expensive.

Other methods follow a combinatorial approach, where a large enough set of plausible motion vectors $\mathcal{D} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_N}\}$ is defined, and then a discrete optimization algorithm is used to assign the most adequate base vector to each pixel. We call such a set $\mathcal{D}$ a *basis*, although it is not a basis in the linear algebra sense, since the base vectors are not linearly independent. The optimization process which assigns a candidate vector to each pixel can be performed locally or globally. In a local approach, one usually estimates a cost function $C_k(\boldsymbol{x})$ which measures how adequately the base vector $\boldsymbol{v}_k$ describes the motion at pixel $\boldsymbol{x}$. For example, the classic block matching (BM) method uses the sum of absolute differences (SAE) or the sum of squared differences (SSD) of intensities between a window centered at $I(\boldsymbol{x},t)$ and an equally-sized window centered at $I(\boldsymbol{x}+\boldsymbol{v}_k, t+1)$. Since both SAE and SSD can be computed very efficiently, for example, by using the integral images [5] or aggregate cost techniques, the local approach adapts very well to realtime applications, as long as the size of the basis N is relatively small. On the other hand, global combinatorial approaches are similar to variational methods in the sense that one usually has to minimize an energy function $U(e)$, where $e(\boldsymbol{x})$ is a field of labels (often modeled as a Markov Random Field) which indicate the base vector assigned to each pixel (i.e., the vector assigned to pixel $\boldsymbol{x}$ is $\boldsymbol{v}_{e(\boldsymbol{x})}$); however, the minimization of $U(e)$ is a combinatorial problem which usually requires computationally expensive methods such as graph-cuts [6], or belief propagation [7]. In both approaches, local and global, computational complexity increases directly with respect to the

number N of vectors in the basis. Moreover, the presence of spurious vectors in the basis; that is, vectors which do not correspond to the motion of any of the objects in the scene and should not be assigned to any pixel, may increase the uncertainty and have a negative impact in the accuracy of the estimations. For these reasons, it is desirable to choose a basis that is small yet represents well the true motions of the objects, with the least possible amount of spurious vectors.

Various algorithms have been proposed to obtain a good basis for a given input scene. As a first approach, one could define the basis to contain all vectors with integer coordinates within a certain range, for example, from $-D$ to D, forming a regular grid of size $(2D+1) \times (2D+1)$. Clearly, even for small values of D the size of the basis will be relatively large, and will likely contain many spurious candidates. Another choice is to use a polar grid, where the basis vectors are uniformly spaced in angle and magnitude, and with magnitude up to D [8]. For example, for a displacement range of $D = 4$ pixels, the rectangular grid approach would define a basis with 81 vectors, whereas the polar grid approach with 8 different angles (at intervals of $\pi/4$ radians) would yield 33 candidates, including the vector $(0, 0)$. Both of these approaches typically yield relatively large bases and are severely limited by the displacement range D. Another solution consists in performing, in a first stage, a crude estimation of the optical flow with a very large set of candidates, for instance, a rectangular grid with a large enough D, using a fast local method such as block matching; then, an heuristic is used to form a reduced basis for each pixel, containing the most likely candidates which resulted from the previous stage, and the reduced bases are used in a slower global optimization method [9]. One disadvantage of this method is that any spurious candidates that may have been wrongly chosen for a given pixel during the local matching stage will also form part of the reduced basis; another disadvantage is that this reduction technique has limited use in realtime implementations.

Recently, another method was proposed to obtain a reduced basis with very little computational overhead [10]. This method is based on the estimation of the phase-only correlation (POC) function, which for two 1D discrete-time signals $f(x)$ and $g(x)$ is defined as

$$r(x) = \mathcal{F}^{-1} \left\{ \frac{F(k)G^*(k)}{|F(k)G^*(k)|} \right\}, \tag{1}$$

where F and G are the Fourier transforms of f and g, respectively, G^* denotes the complex conjugate of G, and $\mathcal{F}^{-1}$ is the inverse Fourier transform. It is easy to show that if g is a shifted version of f, i.e., $g(x) = f(x-d)$, then $r(x) = \delta(x+d)$, where $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise; therefore, one can estimate the displacement between f and g as $\hat{d} = \arg\max_x\{r(x)\}$. A more general case is when $g(x)$ can be modeled as a version of $f(x)$ where different segments suffer different displacements; in other words, when $g(x) = f(x - d(x))$ where $d(x)$ is a piece-wise constant function, then the POC function is a sum of distorted delta functions, centered at the locations which corresponds to the displacement values that the function $d(x)$ can take (see [10] for details). This result can be

extended to 2D signals (images) and higher-dimensional signals. In the context of combinatorial motion estimation, the locations of the maxima of the 2D POC between the input images form an adequate basis for computationally efficient implementations.

This work focuses exclusively on combinatorial motion estimation methods which rely on a finite candidate vector set (a basis) from which pixel velocities can be selected, and presents an in-depth evaluation of the bases that can be obtained by finding multiple maxima in the POC function, in terms of (1) the accuracy with which they can represent the true motions in the scene, and (2) their computational cost (mainly given by the cardinality of the basis). Using these criteria, we also present a comparison against the bases obtained using typical rectangular and polar grids, and applying a heuristic to reduce the size of the bases using a block matching stage. Finally, we assess the accuracy of an actual estimation of the optical flow field with each of the bases under study using state-of-the-art Quadratic Markov Measure Field models [11]. The results from these analyses demonstrate that the POC-derived bases have a significant positive impact on the computational efficiency, without sacrificing the accuracy of the estimated motions.

2 Methodology

2.1 Basis Estimation

Let $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ be the input images (e.g., two consecutive frames in a video sequence), where $\boldsymbol{x}$ denotes the position of a pixel. Let $N_x \times N_y$ be the size of the images (in pixels) and let L be the lattice where the image is defined; i.e., $L = [0, \ldots, N_x - 1] \times [0, \ldots, N_y - 1]$.

To prevent interference between too many peaks in the POC function, the input images are divided in overlapping square regions of size $W \times W$, where W is chosen as a power of two so that FFTs can be computed efficiently. In our tests, we have obtained the best results using $W = 128$. The overlap between adjacent regions can be as small as the largest displacement $D_{\max}$ one wants to find. With this consideration, the image is divided horizontally in M_x and vertically in M_y regions, where

$$M_x = \left\lceil \frac{N_x - W}{W - D_{\max}} \right\rceil + 1, \quad M_y = \left\lceil \frac{N_y - W}{W - D_{\max}} \right\rceil + 1, \tag{2}$$

and the horizontal and vertical spacing between regions is thus given by $s_x = (N_x - W)/(M_x - 1)$ and $s_y = (N_y - W)/(M_y - 1)$, respectively. Therefore, region (i, j) is defined by the sub-lattice $L_{ij} = [is_x, \ldots, is_x + W - 1] \times [js_y, \ldots, js_y + W - 1]$ for all $i = 0, \ldots, M_x - 1$ and $j = 0, \ldots, M_y - 1$.

For each region (i, j), the POC function $r_{ij}(\boldsymbol{d})$ is computed between the sub-images of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ defined at $\boldsymbol{x} \in L_{ij}$. Then, the set $\mathcal{D}_{ij} = \{\boldsymbol{d}_{ij}^1, \ldots, \boldsymbol{d}_{ij}^P\}$, which contains the locations of the P most significant maxima of r_{ij}, is found. We have obtained good results with P between 5 and 8. The vectors $\boldsymbol{d}_{ij}^q$ represent

likely displacements for the objects observed in region (i, j). One can also define the set of likely displacements for the full image as $\mathcal{D} = \cup_{i,j}\mathcal{D}_{ij} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K\}$, where K is the cardinality of this set.

It is also useful to estimate a candidate subset $\mathcal{D}_x$ for each pixel $\boldsymbol{x}$. To do this, one can build, for each candidate $\boldsymbol{d}_k \in \mathcal{D}$, a mask image $m_k(\boldsymbol{x})$ which indicates if $\boldsymbol{d}_k$ is an adequate candidate for pixel $\boldsymbol{x}$ as follows:

$$m_k(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \exists i, j \ : \ \boldsymbol{x} \in L_{ij}, \ \boldsymbol{d}_k \in D_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Then, the subset $\mathcal{D}_{\boldsymbol{x}}$ of candidates for pixel $\boldsymbol{x}$ is given by $\mathcal{D}_{\boldsymbol{x}} = \{\boldsymbol{d}_k \in \mathcal{D} \ : \ m_k(\boldsymbol{x}) = 1\}$.

2.2 Basis Reduction

Following the idea in [9], given a basis $\mathcal{D}$, one can obtain a reduced basis $\hat{\mathcal{D}}$ by performing a fast local optical flow estimation (such as block-matching), and eliminating any motion vectors that were not assigned to any pixel during this process. Formally, the estimated motion $\boldsymbol{v}(\boldsymbol{x})$ for a pixel $\boldsymbol{x}$ is obtained by minimizing the sum of a given cost function c over a window centered at $\boldsymbol{x}$, over the set of candidate motion vectors:

$$\boldsymbol{v}(\boldsymbol{x}) = \arg\min_{\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{x}}} \left\{ \sum_{\boldsymbol{r} \in \mathcal{W}} c(f(\boldsymbol{x} + \boldsymbol{r}) - g(\boldsymbol{x} + \boldsymbol{r} + \boldsymbol{d})) \right\}, \tag{4}$$

where $\mathcal{W}$ is a moving window of size $(2w + 1) \times (2w + 1)$, defined by $\mathcal{W} = [-w, \ldots, w] \times [-w, \ldots, w]$. The size w of the window represents a trade-off between noise reduction and detail preservation; however, in this stage we are not interested in the preservation of borders and details, so it is recommendable to use a large window to avoid noisy estimations that could let spurious candidates to be included in the reduced basis. In our tests, we have obtained the best results with $w = 14$ (a 29×29 window). The cost function c is usually chosen to be the absolute value $c(x) = |x|$, or the square $c(x) = x^2$. For color images, the cost can be defined in terms of the L_2 norm. In our tests, we have chosen a truncated absolute value, given by $c(x) = \min\{|x|, \epsilon\}$, which is slightly more robust to outliers. The value of ϵ is chosen as a proportion of the full dynamic range of the non-truncated cost function; for instance, $\epsilon = \kappa R$, where R is the dynamic range of the images (i.e., the difference between the maximum and minimum intensity values). In our tests, we have obtained the best average results with $\kappa = 0.03$.

Once the optical flow $\boldsymbol{v}(\boldsymbol{x})$ has been estimated, one can define the reduced basis $\hat{\mathcal{D}}$ as the set of motion vectors used in $\boldsymbol{v}$; that is, $\hat{\mathcal{D}} = \{\boldsymbol{v}(\boldsymbol{x}) \ : \ \boldsymbol{x} \in L\}$. It is also possible to find a reduced candidate set $\hat{\mathcal{D}}_{\boldsymbol{x}}$ for each pixel $\boldsymbol{x}$ as $\hat{\mathcal{D}}_{\boldsymbol{x}} = \hat{\mathcal{D}} \cap \mathcal{D}_{\boldsymbol{x}}$.

2.3 Optimal Ground Truth Reconstruction

To evaluate the quality of a given basis, one could use any combinatorial method for the estimation of the optical flow for a scene with a known ground truth

$\boldsymbol{w}(\boldsymbol{x})$, and compare the estimated flow field against the ground truth. However, the quality of the results will strongly depend on the accuracy of the optical flow method and may not reflect the adequateness of the basis itself. One alternative consists in measuring how well the basis vectors can represent the ground truth, regardless of the method used for the actual estimation of the optical flow. To do this, one can obtain an optimal reconstruction of the ground truth, for a given basis $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k\}$, by taking, for each pixel $\boldsymbol{x}$, the candidate which is closer to the true motion vector at $\boldsymbol{x}$. In other words, the optimal ground truth reconstruction $\boldsymbol{w}_{\mathcal{D}}$ for a given basis $\mathcal{D}$ is computed as $\boldsymbol{w}_{\mathcal{D}}(\boldsymbol{x}) = \arg\min_{d \in \mathcal{D}} \{||d - \boldsymbol{w}(\boldsymbol{x})||\}$.

From this reconstruction, one can compute the average end-point error (AEE) and average angular error (AAE) with respect to the ground truth $\boldsymbol{w}$, as defined in [3]. Note that, for a given basis $\mathcal{D}$, the optimal reconstruction $\boldsymbol{w}_{\mathcal{D}}$ is designed to minimize the AEE, and in this sense represents the best possible estimation one can obtain.

It is also interesting to find which, and how many, of the candidates are actually useful for reconstructing the ground truth. These are given by $\mathcal{D}_{GT} = \{\boldsymbol{w}_{\mathcal{D}}(\boldsymbol{x}) \ : \ \boldsymbol{x} \in L\}$. We define the *efficiency* $E_{\mathcal{D}}$ of a given basis $\mathcal{D}$ as the percentage of basis vectors which are used in the optimal reconstruction; that is, $E_{\mathcal{D}} = (|\mathcal{D}_{GT}|/|\mathcal{D}|) \times 100\%$.

2.4 Optical Flow Estimation with a QMMF Model

We have also implemented a state-of-the-art global optimization algorithm based on an Entropy-Controlled Quadratic Markov Measure Field model (EC-QMMF) [11]. Under this model, one estimates the probabilities $b_k(\boldsymbol{x}) = P(\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{d}_k \mid f, g, \mathcal{D})$, where f and g are the input images and $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K\}$ is a given basis. Note that, in order for $\boldsymbol{b}(\boldsymbol{x}) = (b_1(\boldsymbol{k}), \ldots, b_K(\boldsymbol{x}))$ to be a proper probability measure, it is necessary that $b_k(\boldsymbol{x}) \geq 0$ and $\sum_k b_k(\boldsymbol{x}) = 1$. Once the measure field $\boldsymbol{b}$ is known, once can compute an estimator for $\boldsymbol{v}(\boldsymbol{x})$ as the central tendency of $\boldsymbol{b}(\boldsymbol{x})$. For instance, using the mode as central tendency measure, the flow field could be estimated by $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{d}_{e(\boldsymbol{x})}$, $e(\boldsymbol{x}) = \arg\max_k\{b_k(\boldsymbol{x})\}$, while using the mean as central tendency measure, the optical flow estimation is given by

$$\boldsymbol{v}(\boldsymbol{x}) = \sum_{k=1}^{K} b_k \boldsymbol{d}_k. \tag{5}$$

Note that the mean estimator given by (5) can take values outside of the candidate set $\mathcal{D}$ and is therefore able to produce smoother optical flow fields. However, if the measures $b(\boldsymbol{x})$ are highly entropic (i.e., approximately uniform), then the mean estimator given by (5) will approach a constant vector (the average of the candidates) for all $\boldsymbol{x}$. The EC-QMMF models attempt to avoid this problem by imposing both smoothness and entropy constraints to the measure field b. Under this model, the optimal b is obtained by minimizing the energy

function $U(b)$ given by

$$U(b) = \sum_{\boldsymbol{x}\in L}\sum_{k=1}^{K} b_k^2(\boldsymbol{x})[-\log\hat{b}(\boldsymbol{x}) - \mu] + \lambda \sum_{x\in L}\sum_{y\in\mathcal{N}_{\boldsymbol{x}}} \beta_{x,y}||b(\boldsymbol{x}) - b(\boldsymbol{y})||^2, \quad (6)$$

subject to

$$b_k(\boldsymbol{x}) \geq 0,\ \forall x \in L, k \in \{1,\ldots,K\} \quad (7)$$

$$\sum_{k=1}^{K} b_k(\boldsymbol{x}) = 1,\ \forall x \in L. \quad (8)$$

Here, $\hat{b}(\boldsymbol{x}) = \left(\hat{b}_1(\boldsymbol{x}),\ldots,\hat{b}_K(\boldsymbol{x})\right)$ is a normalized likelihood measure, which in our case is given by

$$\hat{b}_k(\boldsymbol{x}) = \exp\left\{-c(f(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{d}_k))\right\}, \quad (9)$$

where $c(x)$ is the truncated absolute cost function used in Section 2.2, the set $\mathcal{N}_{\boldsymbol{x}}$ is a neighborhood of $\boldsymbol{x}$, which in our case consists of all pixels whose distance to $\boldsymbol{x}$ is less or equal than 1, and the variables λ and μ are hyperparameters that control, respectively, the degree of smoothing and the entropy penalization (for details on the probabilistic framework from which the energy function $U(b)$ is derived, please see [11]). Finally, the coefficients $\beta_{x,y}$ measure the likelihood of pixels $\boldsymbol{x}$ and $\boldsymbol{y}$ belonging to the same object, in order to preserve detail at the edges of the objects in the reference image. These coefficients are given by $\beta_{x,y} = \exp\left\{-\frac{\gamma}{R}||f(x) - f(y)||\right\}$, where the hyperparameter γ controls the awareness of the algorithm to edges and R is the dynamic range of the image [12].

Since $U(b)$ is quadratic, it can be minimized very efficiently by solving a linear system with constraints. Constraint (8) can be handled by introducing the corresponding Lagrange multipliers so that the system remains linear, while constraint (7) is handled by a simple projection method: each time a measure $b(\boldsymbol{x})$ is updated, any negative components $b_k(\boldsymbol{x})$ are set to zero, and $b(\boldsymbol{x})$ is renormalized. Our implementation uses the Gauss-Seidel method to solve the unconstrained linear system, and the proposed projection method.

3 Results and Discussion

Several tests were performed to assess the quality of the bases obtained using the proposed method and compare them with other methods in the literature. The test scenes were obtained from the Middlebury optical flow database; there are eight training scenes with known ground truth: Dimetrodon, Grove2, Grove3, Hydrangea, RubberWhale, Urban2, Urban3 and Venus. A technical description of each scene and the challenges they present can be found in [3].

All tests were performed in a computer with a quad-core 3.1 GHz Intel Core i5 CPU and 8 Gb of RAM; however, our implementations have not been thoroughly

Table 1. Number of candidates in each of the bases under study, for each test scene

	Dimetrodon	Grove2	Grove3	Hydrangea	Rubberwhale	Urban2	Urban3	Venus
$\mathcal{D}_{\text{POC5}}$	21	20	65	40	19	52	55	29
$\mathcal{D}_{\text{POC5-R}}$	14	13	61	37	13	46	52	22
$\mathcal{D}_{\text{POC8}}$	39	35	81	65	34	77	86	50
$\mathcal{D}_{\text{POC8-R}}$	21	23	74	53	18	55	75	29
$\mathcal{D}_{\text{RGRID}}$	625	625	625	625	625	625	625	625
$\mathcal{D}_{\text{RGRID-R}}$	34	77	222	65	40	378	315	105
$\mathcal{D}_{\text{PGRID}}$	385	385	385	385	385	385	385	385
$\mathcal{D}_{\text{PGRID-R}}$	31	89	167	77	52	131	227	114

optimized and do not run in multiple cores. The reported results are meant to serve only as an additional characterization of the different bases under study and do not necessarily demonstrate the full potential of the proposed methods in terms of accuracy and computational efficiency.

3.1 Test Bases

For each scene, we construct four different bases:

- $\mathcal{D}_{\text{POC5}}$ - A basis obtained using the proposed POC method with $P = 5$ candidates per region, and regions of size 128×128.
- $\mathcal{D}_{\text{POC8}}$ - A basis obtained using the proposed POC method with $P = 8$ candidates per region, and regions of size 128×128.
- $\mathcal{D}_{\text{RGRID}}$ - A rectangular grid with a displacement range $D = 12$; that is, $\mathcal{D}_{\text{RG}} = [-12, \ldots, 12] \times [-12, \ldots, 12]$. This basis contains 625 candidate vectors.
- $\mathcal{D}_{\text{PGRID}}$ - A polar grid with range $D = 24$ and 16 different angles (at intervals of $\pi/8$). That is, $\mathcal{D}_{\text{RG}} = \{(d\cos(a\pi/8), d\sin(a\pi/8)) \ : d \in \{0, \ldots, 24\}, a \in \{0, \ldots, 15\}\}$. This basis contains 385 vectors.

For each of these bases, we also perform the reduction stage described in Section 2.2 to obtain the reduced bases $\mathcal{D}_{\text{POC5-R}}$, $\mathcal{D}_{\text{POC8-R}}$, $\mathcal{D}_{\text{RGRID-R}}$ and $\mathcal{D}_{\text{PGRID-R}}$, respectively. The exact number of candidates contained in each base, for each one of the test scenes, is shown in Table 1. Note that the cardinality of the POC-derived bases is considerably smaller than the cardinality of the grid bases.

3.2 Optimal Ground Truth Reconstruction Error

Table 2 shows the average and standard deviation (over the eight Middlebury training scenes) of the AEE and AAE between the ground truth and the optimal reconstruction with each of the bases under study. The efficiency (percentage of base vectors used in the reconstruction) is also shown. Note that the lowest AEE is achieved with the POC-derived bases. This is probably due to the fact that the displacement range in the grid bases is limited, particularly in the case of

Table 2. Evaluation of the different bases under study with respect to the optimal ground truth reconstruction that can be obtained with each basis. The columns show the mean and standard deviation (SD) of the average end-point error (AEE, in pixels), average angular error (AAE, in degrees) and Efficiency percentage, measured over the eight test scenes. Values shown in bold face correspond to the best case.

	Mean AEE	**SD AEE**	**Mean AAE**	**SD AAE**	**Mean Eff**	**SD Eff**
$\mathcal{D}_{\text{POC5}}$	0.372	0.092	4.859	2.401	73.620	18.180
$\mathcal{D}_{\text{POC5-R}}$	0.373	0.091	4.861	2.401	**86.553**	**13.215**
$\mathcal{D}_{\text{POC8}}$	0.360	0.091	4.724	2.432	55.738	23.356
$\mathcal{D}_{\text{POC8-R}}$	**0.360**	**0.090**	4.723	2.431	73.778	22.308
$\mathcal{D}_{\text{RGRID}}$	0.641	0.773	4.699	2.781	6.920	5.509
$\mathcal{D}_{\text{RGRID-R}}$	0.642	0.773	4.701	2.781	39.064	34.023
$\mathcal{D}_{\text{PGRID}}$	0.454	0.254	**4.309**	**2.177**	10.584	6.754
$\mathcal{D}_{\text{PGRID-R}}$	0.455	0.254	**4.309**	**2.177**	40.462	27.321

the rectangular grid. One could increase this range, however, it would result in a very large number of candidates that would be impractical in many cases. On the other hand, the lowest AAE is achieved with the polar grid approach. This is possibly because the other methods (POC and rectangular grid) only produce candidates with integer coordinates, and thus fail to accurately represent the direction of short movements.

It is also worth noting that the reduction stage using block matching considerably reduces the cardinality of the basis, and increases its efficiency, but has a neglible impact in the reconstruction accuracy. This suggests that the proposed reduction stage does eliminate a fair amount of spurious candidates.

One could question if *any* of these bases is adequate for the estimation of the optical flow between the two input images. One way to answer this question is to determine if a given basis $\mathcal{D}$ is better than a random basis $\mathcal{D}_{\text{rand}}$, in terms of the accuracy and efficiency with which the ground truth can be reconstructed. To do this, one can generate a R random bases (where R is large enough), and for each case, compute the ground truth reconstruction and estimate the accuracy of the reconstruction (by means of the AEE and AAE) and its efficiency. With this data, one can construct the empirical distributions of the accuracy and efficiency for random bases and test if the results for a given basis $\mathcal{D}$ could be drawn from these distributions. The empirical distributions can be obtained, for instance, using kernel density estimation with a bandwidth given by Silverman's rule of thumb [13]. For a given scene, the random bases are constructed by choosing vectors with a uniformly random direction, and an uniformly random magnitude between 0 and a displacement range $D = 24$. For a given scene, the cardinality of all the random bases is chosen to be equal to $|\mathcal{D}_{\text{POC8}}|$, which is the largest cardinality of the POC-derived bases. Figure 1 shows the distributions (both as histograms and kernel density estimators) for AAE, AEE and efficiency for two different scenes: Rubberwhale and Urban3, and the positions where the bases under study lie within these distributions. Note that, in most cases, the scores obtained with the test bases lie near or completely outside the tails of

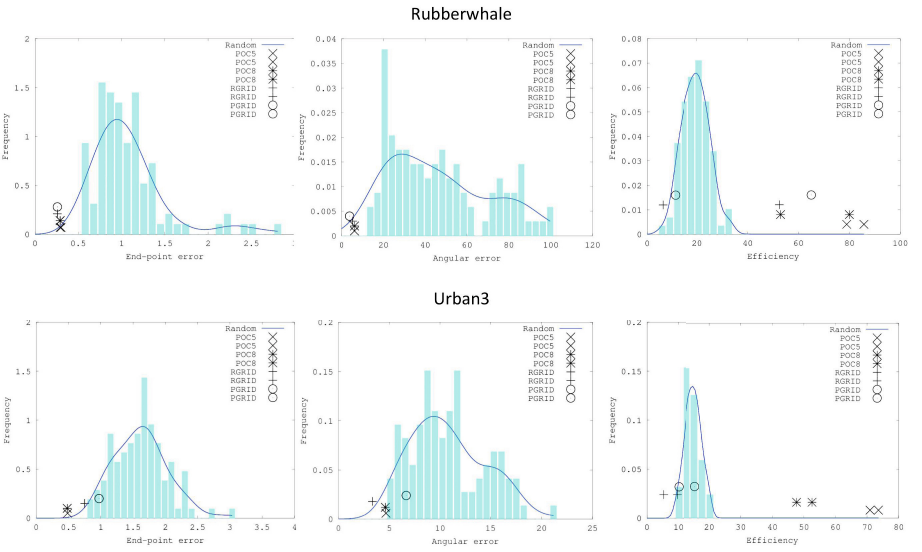

Fig. 1. Distributions of AEE, AAE and Efficiency between the ground truth and the optimal reconstruction using a random basis, for two of the Middlebury scenes: Rubberwhale and Urban3. The black marks indicate where the results obtained with non-random bases lie within the corresponding distributions.

the distributions (left tail for AEE and AAE, right tail for efficiency), which supports the notion that these bases are adequate for the estimation of the optical flow.

3.3 Optical Flow Estimation Error

For the final test presented in this paper, we estimated the optical flow for six of the eight Middlebury training scenes using the EC-QMMF model. For this test, we used only the reduced bases, since the reduction stage does not seem to have a significant impact in the quality of the results. We also decided to remove the results corresponding to the Urban2 and Urban3 scenes due to the fact that our current implementation can handle a maximum of 255 candidates, but the RGRID-R bases have more than 255 candidates for the Urban scenes. This limitation will be avoided in future implementations. The parameters used for all tests were $\lambda = 100$, $\mu = 1$, $\gamma = 20$, and 50 Gauss-Seidel iterations; these values were obtained empirically and might not be the optimal parameters for every scene.

The average results are presented in Table 3. The POC-derived bases show a slight improvement with respect to the reduced grid bases, both in accuracy (AEE and AAE), and a considerable improvement in efficiency. The increased accuracy may be explained by the fact that the reduced grid bases contain a large number of spurious candidates which increase the uncertainty in the QMMF

Table 3. Evaluation of the different bases under study with respect to the EC-QMMF optical flow estimation that can be obtained with each basis. The columns show the mean and standard deviation (SD) of the average end-point error (AEE, in pixels), average angular error (AAE, in degrees) and Efficiency percentage, measured over six test scenes. The B. Time column indicates the average time required for the basis estimation (including the block-matching reduction stage), whereas E. Time represents the average time required for the optical flow estimation. Total computation time is the sum of B. Time and E. Time. Values shown in bold face correspond to the best case.

	Mean AEE	SD AEE	Mean AAE	SD AAE	Mean Eff	SD Eff	B. Time	E. Time
$\mathcal{D}_{\text{POC5-R}}$	0.493	0.223	7.529	2.967	**89.136**	**14.376**	**0.266**	**1.233**
$\mathcal{D}_{\text{POC8-R}}$	**0.480**	**0.212**	7.412	2.931	75.320	24.523	0.368	1.520
$\mathcal{D}_{\text{RGRID-R}}$	0.495	0.225	**7.478**	**2.433**	48.318	34.092	3.532	3.305
$\mathcal{D}_{\text{PGRID-R}}$	0.588	0.310	8.198	2.445	46.760	29.102	2.087	3.164

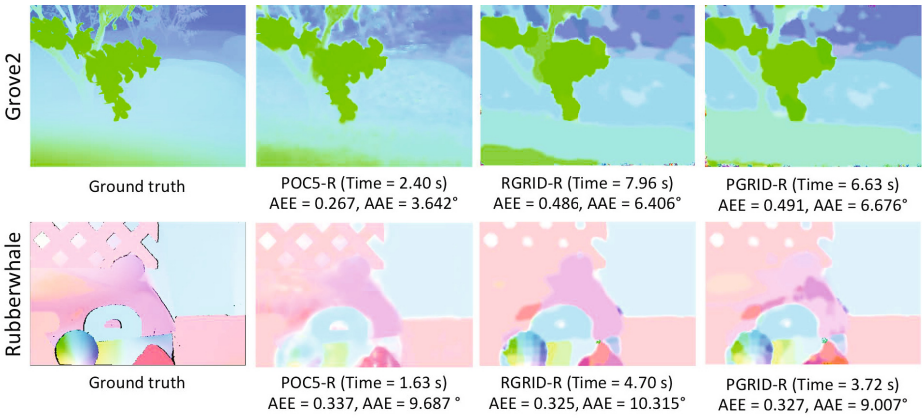

Fig. 2. Optical flow estimation results for the Grove2 scene (upper row) and the Rubberwhale scene (lower row) using the POC5-R, RGRID-R, and PGRID-R bases. The AAE, AEE, and total computation time are shown for each case. The parameters used for the EC-QMMF model were: $\lambda = 1000$, $\mu = 1$, $\gamma = 20$, and 50 Gauss-Seidel iterations.

optimization algorithm. Note also that the computation times are considerably smaller for the POC bases, in particular the basis reduction time which includes the block-matching stage. This is because, in this stage, all candidates must be tested for every pixel, and the initial grid bases are highly inefficient. Finally, Figure 2 presents the results for the Grove2 and Rubberwhale scenes.

4 Conclusions

A methodology for the estimation of efficient vector bases for the combinatorial estimation of optical flow was presented. These bases are obtained as the locations of the maxima of the phase-correlation function estimated for overlapping

regions of the images, and can be further reduced by a candidate pre-selection stage based on a block-matching optical flow estimation with low computational cost. Our tests show that the resulting bases, when used in a global optical flow estimation, perform better than bases obtained from regular grids, in terms of both computational complexity and estimation accuracy. This is likely due to the fact that the POC-derived bases contain a higher percentage of vectors which are similar to the true motions, and a lower number of spurious vectors which may confound the optimization algorithms.

References

1. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. International Journal of Computer Vision 12, 43–77 (1994)
2. Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., Cremers, D.: Stereoscopic scene flow computation for 3d motion understanding. International Journal of Computer Vision 95, 29–51 (2011)
3. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision 92, 1–31 (2011)
4. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. Artificial Intelligence 17, 185–203 (1981)
5. Viola, P., Jones, M.: Robust Real-time Object Detection. International Journal of Computer Vision 57, 137–154 (2002)
6. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 1222–1239 (2001)
7. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 787–800 (2003)
8. Ramirez-Manzanares, A., Rivera, M., Kornprobst, P., Lauze, F.: Variational multivalued velocity field estimation for transparent sequences. Journal of Mathematical Imaging and Vision 40, 285–304 (2011)
9. Veksler, O.: Reducing search space for stereo correspondence with graph cuts. In: British Machine Vision Conference, vol. 2, pp. 709–719 (2006)
10. Alba, A., Arce-Santana, E., Aguilar Ponce, R.M., Campos-Delgado, D.U.: Phase-correlation guided area matching for realtime vision and video encoding. Journal of Real-Time Image Processing (2012) (in press)
11. Rivera, M., Ocegueda, O., Marroquin, J.: Entropy-controlled quadratic markov measure field models for efficient image segmentation. IEEE Transactions on Image Processing 16, 3047–3057 (2007)
12. Reyes, A., Alba, A., Arce-Santana, E.R.: Optical flow estimation using phase only-correlation. Procedia Technology 7, 103–110 (2013)
13. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)