Deformable block-based motion estimation in omnidirectional image sequences

Francesca De Simone, Pascal Frossard
Signal Processing Laboratory (LTS4)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Neil Birkbeck, Balu Adsumilli
Google Inc
Mountain View, California, USA

***Abstract*—This paper presents an extension of block-based motion estimation for omnidirectional videos, based on a translational object motion model that accounts for the spherical geometry of the imaging system. We use this model to design a new algorithm to perform block matching in sequences of panoramic frames that are the result of the equirectangular projection. Experimental results demonstrate that significant gains can be achieved with respect to the classical exhaustive block matching algorithm in terms of accuracy of motion prediction. In particular, average quality improvements up to approximately 6 dB in terms of Peak Signal to Noise Ratio (PSNR), 0.043 in terms of Structural SIMilarity index (SSIM), and 2 dB in terms of spherical PSNR, can be achieved on the predicted frames.**

***Index Terms*—omnidirectional video, motion estimation, equirectangular panorama, block matching**

I. Introduction

The videos captured by fully omnidirectional cameras are usually stored as rectangular *panoramic videos*, resulting from the projection of the spherical surface to a plane [1]. Panoramic videos can be encoded using classical block-based transform encoders [2]. However, they significantly differ from cartesian videos captured by perspective cameras, for which the encoders have been optimized. Particularly, since the sphere that typically supports omnidirectional signals is not a developable surface, warping distortions and discontinuities may appear in panoramic videos. These inevitably modify the statistics of the signal. An adaptation of the encoding tools used by block-based encoders to account for the characteristics of panoramic videos is thus expected to improve the compression efficiency as well as the image quality.

Motion estimation is a key step in video compression. Its goal is to predict the motion occurring in a sequence of frames so that the temporal redundancy in the video can be reduced and compact video representations can be achieved. In block-based encoders, motion estimation relies on *block matching algorithms* [3], which identify for each block (i.e., a squared region non-overlapping with any other region in the frame) in a frame at time t_1 (*anchor frame*), the matching block in a frame at time t_2 (*target frame*) among a set of candidate blocks within a *search window*. The best matching block is the one that minimises the reconstruction error: the block in the anchor frame can be represented by translating the matching block in the target frame by a displacement vector (*motion vector*). The anchor frame can then be predicted by using the target frame and the motion vectors. A low energy error signal is eventually encoded to compensate for potential prediction errors.

Classical block matching algorithms assume that the video has been captured by a perspective camera and that any motion of an object in space can be modelled by block translations in the imaging plane, i.e., the video frame. According to this *translational motion model*, a constant displacement on the motion plane corresponds to a constant displacement on the imaging plane. Additionally, since the camera field of view is limited, the object may disappear from the image, when moving outside the field of view. This model is however not correct for panoramic videos. The omnidirectional imaging surface can be modelled as a sphere, thus, a constant displacement on the motion plane corresponds to a non constant angular displacement on the spherical imaging surface (Fig. 1). Also, the camera field of view is unlimited, so a projection of the object onto the imaging surface always exists. Finally, the projection used to map the spherical surface into a panoramic frame (*map projection*) introduces warping distortions, modifying further the motion vectors and the area of the projected object.

To solve this problem, we introduce an object motion model that accounts for the omnidirectional camera model and the projection used to create the panoramic video. Second, we design an adaptation of the block matching algorithm for panoramic videos, based on the proposed motion model. We consider panoramic videos resulting from the equirectangular projection (defined in Section III-A), since this is one of the most commonly used projections nowadays. The proposed approach could however be adapted to any projection. Experimental results show that our method outperforms the *exhaustive block matching algorithm* (EBMA) [3] in terms of accuracy of the motion estimation. Furthermore, our solution is compatible with the classical block-based encoding flow and only requires minimal modifications of the decoder behaviour. As such, it could easily be implemented in existing compression engines.

In Section II, we review related works on models of complex motions in perspective videos as well as on motion estimation for omnidirectional videos. Our motion model and block matching algorithm are described in Sections III and IV, respectively. The test conditions considered to evaluate the performance of our method and the experimental results are discussed in Section V. Conclusions are drawn in Section VI.

978-1-5090-3649-3/17/$31.00

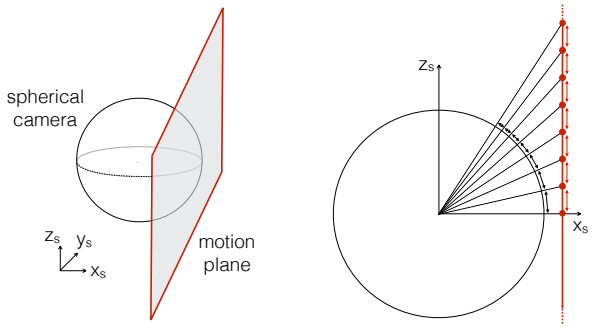


Fig. 1. Omnidirectional camera and example of motion plane (a). View for $y_s = 0$ (b): a constant displacement on the motion plane (red) corresponds to a nonconstant angular displacement on the spherical imaging surface (black).

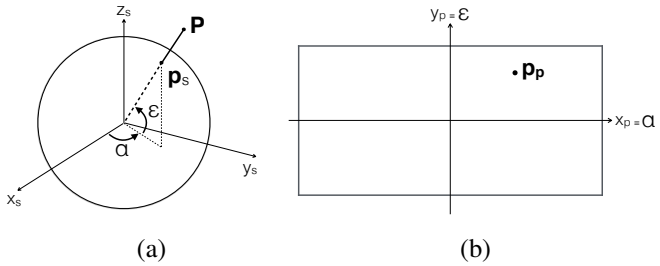


Fig. 2. The omnidirectional camera model (a) and the result of the equirectangular map projection (b).

II. Related works

The translational motion model used by traditional video encoders cannot accurately describe complex motions in perspective videos. Partitioning the frame in blocks of variable sizes [4] or using high-order motion models [5], [6] have been proposed as solutions to overcome the limitations of this model, at the price of an overhead to signal the partitions and increased complexity. Such solutions may also improve motion estimation in omnidirectional videos, as demonstrated in [7], where a rate-distortion optimal selection of translational, affine or quadratic motion model has been investigated for panoramic videos resulting from cylindrical or equirectangular map projection. Nevertheless, they do not take into account the spherical camera geometry and the map projection, thus, for example, they are unable to handle discontinuities.

A solution to perform motion estimation directly in spherical domain has been proposed in [8]. It is based on a multiresolution decomposition of the spherical images, in order to improve the consistency of the motion estimation, and determines pairs of similar solid angles, instead of blocks of pixels. Due to the fact that the matching is performed on solid angles, this solution is not compatible with the block-based coding flow that requires an estimation of motion for each block of the panoramic video input to the encoder.

Finally, block matching algorithms adapted to the geometry of fish-eye and catadioptric images have been proposed in [9], [10], [11], [12]. Due to the specificity of the considered acquisition systems, these algorithms are however not directly applicable to video sequences acquired with fully omnidirectional multi-dioptric camera systems available nowadays.

III. Camera and motion model

A. Omnidirectional camera model

An omnidirectional central camera can be modelled as an ideal spherical sensor [13]: the camera, located at the origin of the right-handed world coordinate system, projects the point $\mathbf{P} = (X, Y, Z)^T$ in the 3D space to the point $\mathbf{p_s} = (x_s, y_s, z_s)^T$ on the spherical imaging surface of radius r, i.e., the viewing sphere, with $\mathbf{p_s} = r\mathbf{P}/||\mathbf{P}||$ (Fig. 2 (a)). Each point on the sphere surface can be identified by its elevation $\epsilon = \pi/2 - \cos^{-1}(z_s/r)$ ($|\epsilon| \leq \pi/2$) and azimuth $\alpha = \arctan2(y_s, x_s)$ ($-\pi \leq \alpha < \pi$). For simplicity, r can be set to 1.

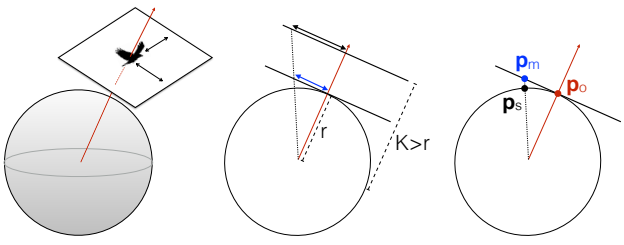
In practice, the omnidirectional image defined on the sphere is represented as a planar rectangular image, by applying a *map projection*, such as the equirectangular, cube, or dodechaedron one [14]. When the *equirectangular projection* is applied, each point on the sphere surface is projected onto a plane by using its elevation and azimuth as coordinates on the plane, i.e., $\mathbf{p_p} = (x_p, y_p) = (\alpha, \epsilon)$ (Fig. 2 (b)). This projection results in a horizontal stretching of the area elements defined on the spherical surface, by a factor $1/\cos(\epsilon)$ [15]. Despite the strong distortions, this projection is still the most common nowadays, due to its simplicity. Thus, this is the map projection that we consider in this paper.

B. Motion model

Traditional motion models that are successfully used in video coding approximate any motion in space by translations of blocks on a motion plane parallel to the imaging plane. This is usually a good approximation for natural scenes that are far enough from a static camera. We build on the same assumptions and extend the model to omnidirectional cameras.

Any motion of an object in space is approximated by translational movements on a *motion plane* whose normal is perpendicular to the imaging surface (Fig. 3 (a)). Any displacement on a plane at distance $K > r$ from the camera center can be represented as motion on the corresponding tangent plane ($K = r$) (Fig. 3 (b)). Therefore, any motion of an object in space is approximated by translational movements on a motion plane tangent to the imaging surface at any elevation and azimuth. In more details, the projection on the sphere of a point $\mathbf{p_m} = (x, y)$ on a motion plane tangent to the sphere at $\mathbf{p_o} = (\alpha_o, \epsilon_o)$ is the point $\mathbf{p_s} = (\alpha, \epsilon)$ resulting from the *inverse oblique gnomonic projection* [1] (Fig. 3 (c)), with:

$$\begin{aligned} \alpha &= g_\alpha(x, y) = \alpha_o + \tan^{-1}\left(\frac{x \sin\eta}{\gamma}\right) \\ \epsilon &= g_\epsilon(x, y) = \sin^{-1}\left(\cos\eta \sin\epsilon_o + \frac{y \sin\eta \cos\epsilon_o}{\rho}\right) \end{aligned} \tag{1}$$



(a) (b) (c)

Fig. 3. Proposed object motion model. The camera is static and the object is translating on a *motion plane* whose normal is perpendicular to the imaging surface (a). Any displacement on a plane at distance $K > r$ from the camera center can be represented as motion on a tangent plane ($K = r$) (vertical section shown in (b)). The projection of a point $\mathbf{p_m}$ from the motion plane tangent to the sphere at $\mathbf{p_o}$ to the sphere is $\mathbf{p_s}$ (vertical section shown in (c)).

where $\gamma = \rho \cos\epsilon_o \cos\eta - y \sin\epsilon_o \sin\eta$, $\rho = \sqrt{x^2 + y^2}$ and $\eta = \tan^{-1}\rho$. Alternatively, the forward projection, from the sphere surface to the motion plane, permits to compute $\mathbf{p_m}$ from $\mathbf{p_s}$ as follows:

$$x = f_x(\alpha, \epsilon) = \frac{\sin(\Delta\alpha)\cos(\epsilon)}{\cos\psi}$$
$$y = f_y(\alpha, \epsilon) = \frac{\sin(\epsilon)\cos(\epsilon_o) - \sin(\epsilon_o)\cos\epsilon\cos(\Delta\alpha)}{\cos\psi} \quad (2)$$

where $\Delta\alpha = \alpha - \alpha_o$, and ψ is the angular distance of $\mathbf{p_s}$ from $\mathbf{p_o}$, such that $\cos\psi = \sin\epsilon_o \sin\epsilon + \cos\epsilon_o \cos\epsilon \cos\Delta\alpha$.

It is evident that a displacement $\mathbf{d_m} = (\delta_x, \delta_y)$ of a point $\mathbf{p_m}$ on the motion plane corresponds to an angular displacement $\mathbf{d_s} = (\delta_\alpha, \delta_\epsilon)$ of its projection $\mathbf{p_s}$ on the sphere surface that depends non linearly on the distance between $\mathbf{p_m}$ and $\mathbf{p_o}$. Additionally, $\mathbf{d_s}$ corresponds to a displacement $\mathbf{d_p}$ on the equirectangular plane that depends non linearly on ϵ_o.

Accordingly, if we consider a compact set of points on the motion plane, its projection on the sphere surface results in a change of the shape of the set of points. The distance between the points will be further modified when these are projected from the sphere to the equirectangular plane. Knowing the coordinates of each point on the sphere corresponds to knowing the position of its projection on the equirectangular image: thus, the projection of a moving set of points on the motion plane onto the equirectangular image, that is the imaging plane, can be analytically determined.

IV. Modified block matching

We propose a new motion estimation algorithm adapted to the spherical geometry. The proposed solution is based on an adaptation of the classical exhaustive block-matching algorithm (EBMA) that accounts for the change of shape of an object translating in space when projected on the equirectangular imaging plane. The goal of the proposed adaptation is to minimally modify a classical block-based hybrid transform encoding pipeline when performing motion estimation on equirectangular videos to account for the camera and motion model detailed in Section III.

The equirectangular anchor and target frames, A and T, respectively, are regular lattices of pixels defined on the equirectangular plane, i.e., $A = A(\mathbf{p}_i)$ and $T = T(\mathbf{p}_i)$, where $\mathbf{p}_i = (\alpha_i, \epsilon_i)$ is the position of the i-th pixel in the equirectangular plane (i.e., $\mathbf{p_p}$ in Section III-A), with $i = [1, W \times H]$, W and H being the equirectangular image width and height in pixels. The anchor frame A is commonly partitioned into N non-overlapping squared (or rectangular) blocks such that:

$$\bigcup_{j \in \mathcal{N}} B_j = A \text{ and } B_j \bigcap B_k = \emptyset \quad (3)$$

where B_j represents the j-th image block, $\mathcal{N} = \{1, 2, ..., N\}$ and $j \neq k$. Since the motion estimation process is carried out independently for each block, we omit the subscripts j and i for simplicity. Given the set of pixels in block B in the anchor frame A, the classical EBMA finds the corresponding set of pixels in the target frame T, which minimizes the prediction error:

$$\arg\min_{\mathbf{d} \in \Omega} \sum_{\mathbf{p} \in B} |A(\mathbf{p}) - T(\mathbf{p} + \mathbf{d})|^\mu \quad (4)$$

where $\mathbf{d}$ is the spatial displacement vector between pixels in the anchor and target frames, i.e. the motion vector of B. The search window Ω represents the set of candidate motion vectors, which is typically defined as all the motion vectors that have a norm smaller than a predefined threshold. When $\mu = 1$, the error measure is called sum of absolute differences (SAD). According to the classical translational motion model, every $\mathbf{p} \in B$ in A is moved to $\mathbf{p} + \mathbf{d}$ in T. Thus, if B is a squared (or rectangular) block, i.e., a set of $L = L_w \times L_h$ pixels arranged as a regular lattice on the digital image, the corresponding set of pixels in T is also a squared (or rectangular) block of L pixels.

The classical EBMA described above does not account for the spherical geometry and the effect of the map projection. According to the camera and motion model described in Section III, a block B on the equirectangular anchor frame, corresponds to a portion of spherical surface whose shape and area actually vary with the elevation of the block on the sphere (Fig. 4). This portion of spherical surface can be interpreted as the projection on the imaging spherical surface of an object in space. According to our translational motion model, an object moves in space by translating on the motion plane (Fig. 5). By varying the displacement of the object on the motion plane and projecting its replica onto the spherical surface, we obtain a set of candidate matching projected set of pixels on the target frame (Fig. 6). Both the search window size and the shape of the candidate set of pixels vary depending on the elevation at which the block in the anchor frame is located. The best matching set of pixels on the target frame is that whose pixels are the most similar to the pixels in the anchor block.

Fixing the distance of the motion plane from the camera and the amplitude of the displacement on the motion plane, is equivalent to consider a motion plane tangent to the sphere and a displacement on this plane that depends on the distance of the object from the camera center, as discussed in Section

III-B. The closed form expression of the gnomonic projection and its inverse can then be exploited to define the candidate set of matching pixels and adapt the search range used by the motion estimation algorithm, as follows:

1) First, the polar coordinates of the centroid of the block B on the anchor equirectangular image, $\mathbf{p_o} = (\alpha_o, \epsilon_o)$, are associated to the origin of the motion plane tangent to the sphere at $\mathbf{p_o}$. Each pixel of the block B has coordinates on the sphere $\mathbf{p} = (\alpha, \epsilon)$, where $\alpha = [\alpha_o - (L_w - 1)\Delta\alpha/2; \alpha_o + (L_w - 1)\Delta\alpha/2]$ and $\epsilon = [\epsilon_o - (L_h - 1)\Delta\epsilon/2; \epsilon_o + (L_h - 1)\Delta\epsilon/2]$, with $\Delta\alpha = 2\pi/W$, $\Delta\epsilon = \pi/H$.
When projected on the motion plane, these become:
$$\mathbf{q} = (x, y) = (f_x(\alpha, \epsilon), f_y(\alpha, \epsilon)) \tag{5}$$
where f_x and f_y are defined in Eq. (2). These points define "the object" M on the motion plane, whose image on the equirectangular imaging surface (i.e. the anchor frame) is the block B (Fig. 4).
2) On the motion plane, M is assumed to undergo a rigid translation, described by the displacement $\mathbf{d_m}$. The candidate motion vectors on the motion plane form the search window $\Omega_m = \{\mathbf{d_m} = (\delta_x, \delta_y) = (n\Delta x, m\Delta y)\}$ with $n \in [-\Omega_x, \Omega_x]$ and $m \in [-\Omega_y, \Omega_y]$, where Δx and Δy are the minimum horizontal and vertical displacement on the motion plane, respectively, and Ω_x and Ω_y define the horizontal and vertical size of the search window (Fig. 5).
When the object M translates by $\mathbf{d_m}$, with $\mathbf{d_m} \in \Omega_m$, its pixels have coordinates on the motion plane:
$$\mathbf{q}^{\mathbf{d_m}} = \mathbf{q} + \mathbf{d_m} = (x + \delta_x, y + \delta_y). \tag{6}$$
The projection of these points back to the spherical, thus equirectangular, domain (Fig. 6) yields to the coordinates of each pixel in a candidate set of pixels in the equirectangular target frame:
$$\mathbf{p}^{\mathbf{d_m}} = g(\mathbf{q}^{\mathbf{d_m}}) = (g_\alpha(x + \delta_x, y + \delta_y), g_\epsilon(x + \delta_x, y + \delta_y)) \tag{7}$$
where g_α and g_ϵ are defined in Eq. (1). The shape, area and displacement of the set of corresponding pixels in the target frame change depending on ϵ_o, δ_x and δ_y. Since the position $\mathbf{p}^{\mathbf{d_m}}$ of the candidate pixel might be not on the regular lattice that defines the target frame, bilinear interpolation on the equirectangular target frame is used to determine the pixel value at the corresponding position.

When an exhaustive search is performed on the omnidirectional images, the block matching algorithm selects among all motion vectors on the motion plane in the search window Ω_m, the one that corresponds to a set of pixels in the target frame that minimises an error measure with respect to the block B in the anchor frame:
$$\mathbf{d}^* = \arg\min_{\mathbf{d_m} \in \Omega_m} \sum_{\mathbf{p} \in B} |A(\mathbf{p}) - T(\mathbf{p}^{\mathbf{d_m}})|^\mu. \tag{8}$$

A prediction error is further computed as $E(\mathbf{p}) = A(\mathbf{p}) - T(\mathbf{p}^{\mathbf{d}^*})$ and coded along with the motion vector information. At the decoder side, since the projections are analytically known, the process can be inverted and the reconstructed anchor block can be generated from the target block, the prediction error and the motion vector on the motion plane.

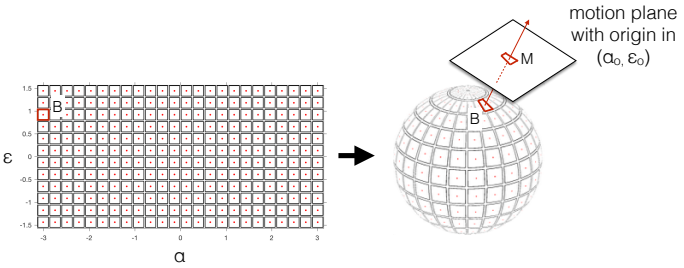


Fig. 4. A block B in the anchor equirectangular frame corresponds to a portion of spherical surface, which can be interpreted as the image of an object M in space, defined on a plane at a certain distance from the omnidirectional camera. The spherical coordinates of the block centroid (α_o, ϵ_o) identify the origin of the plane. In the example $\alpha_o = -23\pi/24$ and $\epsilon_o = 7\pi/24$.

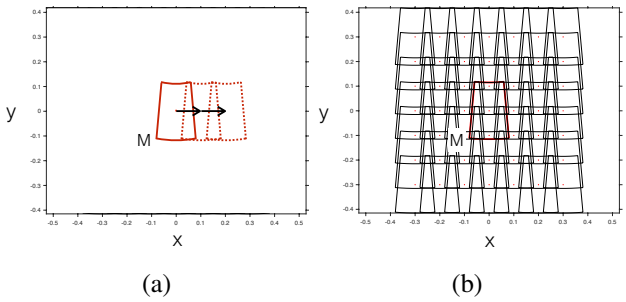


Fig. 5. According to the translational motion model the object M undergoes rigid translations on the motion plane, each associated to a motion vector $\mathbf{d_m}$ on the motion plane. Examples of two object replica associated to two motion vectors (a) and set of replica corresponding to motion vectors in Ω_m (b).

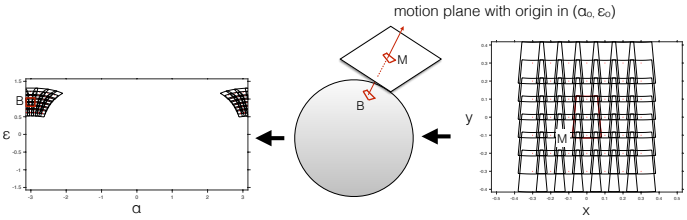


Fig. 6. By varying the displacement of the object M on the motion plane and projecting its replica onto the spherical surface, we obtain the set of candidate sets of pixels on the target equirectangular frame. In the example the block B with $\alpha_o = -23\pi/24$ and $\epsilon_o = 7\pi/24$.

V. Results

We compared the performance of the proposed block matching algorithm (360-EBMA) with a fixed accuracy to that of classical EBMA with one-pixel and half-pixel accuracy, as done in state of the art works on motion estimation.

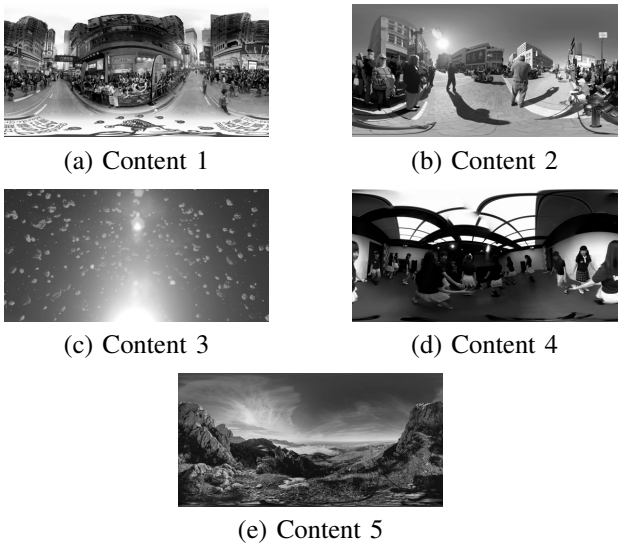
We considered five YouTube 360-degree videos as test sequences: the first selected frame for each content is depicted in Fig. 7. In all sequences the camera is static but different kinds of motion are present, some deviating from our translational motion model. To reduce the computational complexity of the simulations, we resized the original 4K resolution frames to 256×512 pixels. We extracted 10 consecutive frames from each video sequence and applied the motion estimation considering two different scenarios: in the first scenario, all anchor frames are estimated based on one target frame, which is the first frame in the set (configuration $TAAA$); in the second scenario, each anchor frame is estimated based on its previous frame (configuration $TATA$).

We considered a block size of 8×8 pixels, symmetric search windows, a search field range of 8 pixels, i.e. each block was compared to up to $(8 * 2 + 1)^2 = 289$ possible candidate matching blocks, and $\mu = 1$. For 360-EBMA, we fixed the object displacement on the motion plane to $\Delta x = \Delta y = 0.01$. These values correspond to a vertical or horizontal angular displacement of one pixel on equirectangular content at 256×512 pixel resolution for a displacement from the origin of the motion plane tangent to the sphere at the equator. This means that the comparison with EBMA with half pixel accuracy is unfair to our method: nevertheless, the results discussed hereafter show that, on some content, our method can achieve a better prediction even in this case.

Tables I and II report the average PSNR, SSIM [16] and spherical PNSR (S-PSNR) [17] improvement achieved on the predicted anchor frame when the prediction is done using 360-EBMA versus classical EBMA with one-pixel accuracy and half-pixel accuracy. This is computed as mean difference between the quality of each predicted anchor frame, with respect to its original version, when the prediction is using 360-EBMA versus EBMA, across all anchor frames of each content. The frame quality is assessed in equirectangular domain by PSNR and SSIM and in spherical domain by S-PSNR. For the S-PSNR computation, we considered spiral sampling with a total of $(W \times H)/4$ samples.

Overall, it can be observed that the quality of the predicted frames improves by using the proposed method, independently from the metric, the anchor-target configuration, and the accuracy of the EBMA, for all contents apart from content 3. Also, the gain obtained by using the proposed method is limited and metric-dependent, for content 4. This can be explained by the fact that these two videos are those where the proposed motion model is less accurate, due to the presence of strong non-translational motion. Quality improvements are higher for the $TATA$ configuration, as expected, due to the fact that we considered a value of displacement Δx and Δy on the motion plane that is small, thus more accurate when the anchor and target frames are closer in time.

Fig. 8 shows one visual example of a portion of anchor panoramic frame and its prediction obtained by using EBMA-360 and EBMA. Fig. 9 shows where the blocks for which a better matching is achieved when using 360-EBMA are localised, for each content. The quality of the matching is quantified as the Mean Squared Error (MSE) between the matching candidate selected by each algorithm and the block in the anchor frame. The improvement obtained by using our method is linked to the object motion model, i.e., a better matching block is selected by exploiting the assumptions on the block warping due to the projections, as well as to the azimuthal continuity, i.e., matching across the equirectangular frame left and right borders. To quantify how much each factor is contributing to the performance improvement, Table III reports the total percentage of blocks on which the proposed algorithm performs a better matching than classical EBMA with one pixel accuracy, and vice versa, as average computed across all frames and the two configurations, for each content. It can be noticed that the improvement achieved by using our method is mainly due to the accurate object motion model.



(a) Content 1 (b) Content 2

(c) Content 3 (d) Content 4

(e) Content 5

Fig. 7. First frame of each video sequence considered as test material, in equirectangular format.

Content ID	Avg PSNR gain		Avg SSIM gain		Avg S-PSNR gain	
	$TAAA$	$TATA$	$TAAA$	$TATA$	$TAAA$	$TATA$
1	4.71	5.05	0.016	0.01	1.141	1.413
2	1.63	1.97	0.023	0.012	0.861	1.017
3	-0.48	0.05	-0.016	-0.003	-0.045	-0.029
4	0.18	1.24	0.017	0.013	-0.142	0.755
5	6.14	6.1	0.043	0.022	2.421	2.967

TABLE I
AVERAGE QUALITY IMPROVEMENT ON PREDICTED ANCHOR FRAMES WHEN USING 360-EBMA VERSUS EBMA WITH ONE-PIXEL ACCURACY.

VI. CONCLUSION

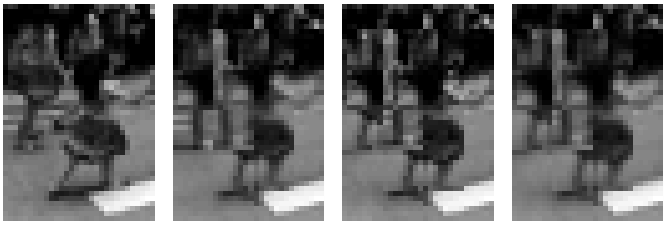
In this paper we have presented an extension of block-based motion estimation for omnidirectional video sequences, based on a translational object motion model that accounts for the spherical geometry of the imaging system. We have used this model to design a new algorithm (360-EBMA) to perform block matching in sequences of panoramic frames that

Content ID	Avg PSNR gain		Avg SSIM gain		Avg S-PSNR gain	
	$TAAA$	$TATA$	$TAAA$	$TATA$	$TAAA$	$TATA$
1	3.29	4.06	0.008	0.007	0.18	0.64
2	0.54	1.14	0.013	0.009	-0.12	0.31
3	-1.34	-0.63	-0.035	-0.015	-0.92	-0.69
4	-0.15	0.46	0.004	0.006	-0.4	0.06
5	4.97	5.24	0.032	0.015	1.35	1.87

TABLE II
AVERAGE QUALITY IMPROVEMENT ON PREDICTED ANCHOR FRAMES WHEN USING 360-EBMA VERSUS EBMA WITH HALF-PIXEL ACCURACY.

Content ID	360-EBMA > EBMA		360-EBMA < EBMA
	total	az. continuity	
1	7.39 %	0.15 %	2.19 %
2	8.54 %	0.17 %	2.7 %
3	23.15 %	0.55 %	30.01 %
4	19.74 %	0.74 %	6.92 %
5	22.88 %	1.35 %	4.44 %

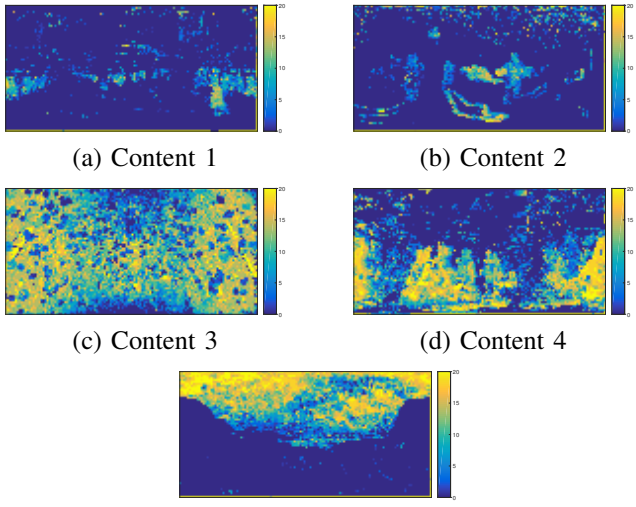
TABLE III
AVERAGE PERCENTAGE OF BETTER MATCHING BLOCKS OBTAINED BY USING 360-EBMA VERSUS CLASSICAL EBMA AT THE SAME ACCURACY (360-EBMA > EBMA - THE PORTION OF BETTER MATCHING BLOCKS DUE TO THE AZIMUTHAL CONTINUITY IS INDICATED) AND VICE VERSA (360-EBMA < EBMA).



(a) Anchor (b) 360-EBMA (c) EBMA1pix (d) EBMA0.5pix

Fig. 8. Example of visual improvements in predicted anchor frame: portion of anchor frame 248 of Content 1 (a) and its predicted version from frame 240 using 360-EBMA (b), EBMA with one pixel accuracy (c) and half pixel accuracy (d).

are the result of the equirectangular projection. Experimental results demonstrate that significant gains can be achieved with respect to the classical EBMA in terms of accuracy of motion prediction, even when the accuracy of the search is set to a higher value for the EBMA. As future work, the model will be extended to other map projections and enriched to take into account motions other than translational ones as well as camera ego-motion. Implementation in a complete video encoder and analysis of complexity will also be considered.



(a) Content 1 (b) Content 2

(c) Content 3 (d) Content 4

(e) Content 5

Fig. 9. Heat maps of 8×8 blocks for which EBMA-360 selects a better matching block than classical EBMA with one or half pixel accuracy, for configuration $TAAA$. The range is from 0 (EBMA-360 never outperforms EBMA on a specific block) to 20 (EBMA-360 always outperforms EBMA on a specific block).

REFERENCES

[1] F. Pearson, *Map Projections: Theory and Applications*, 1990.

[2] K.-T. Ng, S.-C. Chan, and H.-Y. Shum, "Data compression and transmission aspects of panoramic videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, 2005.

[3] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video processing and communications*, 2002.

[4] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012.

[5] C. Heithausen and J. H. Vorwerk, "Motion compensation with higher order motion models for HEVC," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[6] L. Li, H. Li, D. Liu, Z. Li, H. Yang, L. Sixin, H. Chen, and F. Wu, "An efficient four-parameter affine motion model for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, 2017.

[7] J. Zheng, Y. Shen, Y. Zhang, and G. Ni, "Adaptive selection of motion models for panoramic video coding," in *Proc. of the IEEE International Conference on Multimedia and Expo*, 2007.

[8] I. Tosic, I. Bogdanova, P. Frossard, and P. Vandergheynst, "Multiresolution motion estimation for omnidirectional images," in *Proc. of the European Signal Processing Conference*, 2005.

[9] A. Eichenseer, M. Batz, J. Seller, and A. Kaup, "A hybrid motion estimation technique for fisheye video sequences based on equisolid re-projection," in *Proc. of the IEEE International Conference on Image Processing*, 2015.

[10] A. Eichenseer, M. Batz, and A. Kaup, "Motion estimation for fisheye video sequences combining perspective projection with camera calibration information," in *Proc. of the IEEE International Conference on Image Processing*, 2016.

[11] G. Jin, A. Saxena, and M. Budagavi, "Motion estimation and compensation for fisheye warped video," in *Proc. of the IEEE International Conference on Image Processing*, 2015.

[12] D. Alouache, Z. Ameur, and D. Kachi, "An adapted block-matching method for optical flow estimation in catadioptric images," in *Proc. of the International Conference on Multimedia Computing and Systems*, 2014.

[13] B. Micusik, "Two view geometry of omnidirectional cameras," Ph.D. dissertation, Czech Technical University in Prague, 2004.

[14] C.-W. Fu, L. Wan, T.-T. Wong, and C.-S. Leung, "The rhombic dodecahedron map: An efficient scheme for encoding panoramic video," *IEEE Transactions on Multimedia*, vol. 11, no. 4, 2009.

[15] F. D. Simone, P. Frossard, P. Wilkins, N. Birkbeck, and A. C. Kokaram, "Geometry-driven quantization for omnidirectional image coding," in *Proc. of the Picture Coding Symposium*, 2016.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, 2004.

[17] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *Proc. of the IEEE International Symposium on Mixed and Augmented Reality*, 2015.