Human Tracking in Top-view Fisheye Images with Color Histograms via Deep Learning Detection

Olfa Haggui, Marina Vert, Bastien Brieussel, Kieran Mcnamara, Baptiste Magnier

Human Tracking in Top-view Fisheye Images with Color Histograms via Deep Learning Detection

Olfa Haggui, Marina Vert, Kieran McNamara, Bastien Brieussel and Baptiste Magnier
EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, 30100, Ales, France
olfa.haggui@mines-ales.fr, {marina.vert,kieran.mcnamara,bastien.brieussel}@mines-ales.org, baptiste.magnier@mines-ales.fr

***Abstract*—Fisheye cameras produce panoramic images. For a while, classical people detection algorithms were not optimal in fisheye images because detection bounding boxes were non-oriented. People detection algorithms for topview fisheye images have been developed recently. However, these algorithms only detect the people present in the different frames but do not follow them through a video sequence. First, we based our work on the RAPiD (Rotation-Aware People Detection in Over-head Fisheye Images) method to detect people in video frames. Then, in order to track the target throughout the video, we use a comparison method for color histograms based on Bhattacharyya distance. This distance is computed with several histograms with different properties relating to the number of bins or the colorspace to compare the efficiency. Finally, their position is assessed by computing an angle and a distance to the camera. As a result, in a video where several people are detected, we are able to follow the path of one single person throughout the video.**

***Index Terms*—People detection/tracking, deep learning, fisheye.**

I. Introduction

Detecting and tracking persons represents an important area of research in the computer vision domain [1]. Indeed, endowing machines with the ability to interact with people is considered as one of the most useful challenges for modern engineering. In recent years, the number of approaches to detect pedestrians in monocular images has grown substantially [2]. In this context, numerous papers have recently been published on people or pedestrian detection using deep learning techniques. Indeed, in recent research investigations, deep neural networks have frequently improved the detection performance even though they require a large amount of annotated data. In this paper, our aim is to detect and track persons in top-view fisheye images. As the camera axis is directed vertically, people standing straight may appear oriented in the image, pointing towards the center due to the distortion of the camera lens. Therefore, a conventional perspective human detection techniques cannot be directly used, such as Histogram of Oriented Gradients (HOG) [3]. Various methods have been implemented for distorted perspectives [4]–[8], but they do not enable to track the movement of a single person

In this article, we present a simple method to track the moving position of a single person. It combines a deep learning algorithm [4] for the detection part, on the one hand, with distances of color histograms for the tracking part, on the other hand. Eventually, polar coordinates of the person within the camera framework can be returned, in order to track their position. Both this angle and this distance are quantitatively evaluated in the experimental part.

II. People Detection in Fisheye Images

A. Fisheye model

Usually, omnidirectional and fisheye cameras offer panoramic views of 2π or π radian angles [9]. They represent a major asset for several applications. In this way, these wide angle cameras are popular in many fields of computer vision, robotics and photogrammetric tasks such as navigation, localization, tracking, mapping and so on, not only because of their panoramic views but also because the vanishing points indicate very useful information [10], [11].

A fisheye camera is a camera fixed to a front lens group which appears as a single "big" lens, as shown in Fig. 1 on the left. This device enables a far greater negative refraction power than usual lenses, allowing to greatly increase the back focal distance and embrace wider fields of view [12]. The wide field of vision provided by these cameras makes people look inclined and distorted. Consequently, standard detection and tracking techniques are not reliable to these types of images and specific detectors are hard to design because they need a calibration stage which could be difficult. Even though many algorithms already exist for standard images, people detection and tracking regarding top-view images acquired by fisheye cameras are not a very documented topic and demand very specific involvement for it to work consistently.

B. People detection by RAPiD method with fisheye camera

The Rotation-Aware People Detection in Overhead Fisheye Images (RAPiD) method [4], developed in 2019, provides much faster and more accurate results than previous algorithms aiming at tracking people in fisheye images. Its goal is to predict bounding boxes of people, with certain center and size, but also the angle of the bounding box. The RAPiD method

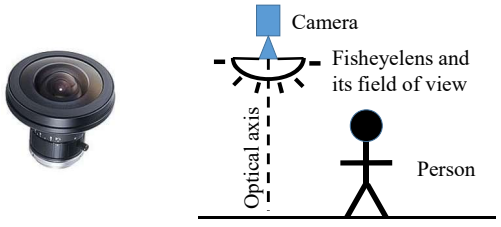
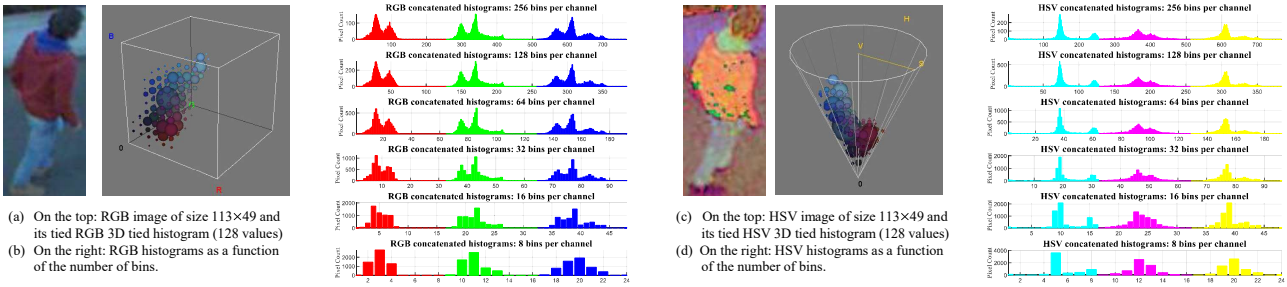


Fig. 1. Fisheye camera. On the left: fisheye lens used in our experiments: 2/3" Format C-Mount fisheye lens 1.8mm FL, with a horizontal field of view, 1/2" sensor for 185°. On the right: diagram of the experimental protocol.



(a) On the top: RGB image of size 113×49 and its tied RGB 3D tied histogram (128 values)
(b) On the right: RGB histograms as a function of the number of bins.
(c) On the top: HSV image of size 113×49 and its tied HSV 3D tied histogram (128 values)
(d) On the right: HSV histograms as a function of the number of bins.

Fig. 2. Visualization of RGB and HSV concatenated histograms as a function of the number of bins per channel (256, 128, 64, 32,16 or 8 bins).

is based on a neural network structured in three parts. The first one is the backbones network, trained on the ImageNet database [13]. Its main goal is to extract features at different spatial resolutions. The second part is the features pyramid network (FPN) [14], which contains information about small and large objects. Finally, the third part is the detection head, which allows to build a tensor containing information on the bounding box position, including its angle of rotation. RAPiD implements a loss function which combines binary cross entropy (BCE), described in YOLOv3 [15], and a periodic loss function that regresses the angle of each bounding box, accounting for angle periodicities. Having the possibility to predict angled bounding boxes is essential for people detection in fisheye images since most targets have an orientation neither vertical nor horizontal. When given a single frame of a video, or just a simple image, the algorithm returns the position, size and angle of all individuals it has detected as such, each one being associated with a confidence score that quantifies how confident the algorithm is about the target actually being a human. There is a confidence threshold, determined by the user but usually set to 0.3, and the algorithm only returns the bounding boxes whose confidence score is higher than this threshold. Examples of images analyzed by the RAPiD algorithm are shown in Figs. 3 and 4.

The RAPiD algorithm was widely evaluated by its authors. They showed that for various datasets and various evaluation methods, RAPiD always seemed to score much better than other algorithms, and found that it was optimal when the threshold was defined at 0.3. We have taken numerous videos in a lot of different situations, conditions and environments and applied the RAPiD algorithm. For all the frames where at least one person's body is almost entirely in the image, the algorithm was able to detect it with a confidence score much higher than the 0.3 threshold, most of the time between 0.7 and 1. The algorithm sometimes also detected the person holding the camera. In some rare cases, the algorithm detected non-human objects, with a confidence score up to 0.6. In that case, it can be corrected by setting the confidence threshold to approximately 0.6 or higher, so that it only returns the accurate bounding boxes.

III. Tracking people with color histograms

Once the persons have been detected in the fisheye image by the technique above, a tracking strategy begins between the different objects (i.e., the number of objects corresponds to the number of detected persons). Indeed, during the video, the goal is to follow the detected person in order to detect its trajectory. Here, the simplest idea for the tracking is the use of the color histogram for each detected object.

A. Color histograms

A color histogram of a digital image represents the distribution of existing colors in this matrix. It is determined by counting the number of pixels corresponding to each pixel values for each color channel. There are different types of color spaces and color histograms in these different color spaces can he easily computed. In this preliminary study, we focused on the RGB (Red-Green-Blue) and the HSV (Hue-Saturation-Value) color spaces.

The RGB color space is an additive color space. Colors are made by adding red, green and blue lights, with different intensities, represented by a pixel value. The Fig. 2(a) represents a 3D histogram of an RGB image. We computed the histograms for each channel (Red, Green and Blue) and then concatenated these three histograms, as illustrated in Fig. 2(b).

An image can also be represented through other color spaces, such as HSV, as illustrated in Fig. 2(b). This color transformation allows to represent how colors appear under light in the image and is a close representation of the human visual perception of colors. Fig. 2(c) represents a 3D histogram of a HSV image. It is an hexacone where the central axis represents the Intensity. Hue represents an angle in the range [0,2π] relative to the Red axis. Red is at angle 0, Green is at angle $2\pi/3$, Blue is at angle $4\pi/3$ and we have Red again at angle 2π. Saturation measures the depth of the color (or the purity) is a radial distance between 0 at the center and 1 at the outer surface. [16] Once again, we computed the histograms for each channel (Hue, Saturation and Value) and concatenated these three histograms, as illustrated in Fig. 2(d).

B. Tracking people with color histograms

The goal here is to compare the similarity between two images. Usually, in reality, two pictures of two following frames contain almost the same objects, these objects are similar and should have close color histograms. Given this hypothesis, the detected bounding boxes in the frames are simply compared using their color histograms. Color-based tracking methods have already been studied, in the probabilistic field for instance with the Sequential Monte Carlo Tracking [17]. However, we decided to use histogram comparison to track people who are detected by a reliable technique regarding fisheye cameras [4]. Several similarity functions exist to compare color

histograms regardless of the colorspace (RGB, HSV or other) such as histogram intersection, the chi-square measure (χ^2) or the Pearson Product-Moment Correlation Coefficient [18]. We chose to compute the Bhattacharyya Distance to compare the histograms [19]. Let H_1 and H_2 be two histograms to be compared. The smaller the distance, the more similar the images are. The Bhattacharyya distance between them is computed by:

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \cdot \bar{H}_2 N^2}} \sum_{k=1}^{N} H_1(k) \cdot H_2(k)}, \quad (1)$$

where $\bar{H}_i = \frac{1}{N} \sum_{k=1}^{N} H_i(k)$ represents a normalized histogram and N the number of bins in the histogram $H_{i,i\in\{1,2\}}$.

In our experiments, we applied the tracking technique to the detected objects in multiple videos in order to evaluate the efficiency of our method. For each video, the goal is to follow the detected people throughout the frames. To do so, the bounding boxes of the detected objects in each frame are easily isolated and their color histograms are computed. The histograms of the detected objects in the previous frame is also computed in order to compare these histograms together. This step is crucial because the persons are detected with reliability but they are not associated along the video because the RAPiD algorithm relates only to people detection. As a matter of fact, the smallest distance is tied to the closest bounding box for the person we are trying to follow, thus the target. Let P_1^t be the person P_1 we are following in the frame at time t, P_2^t the second detection on the same frame, P_1^{t+1} the first detection and P_2^{t+1} the second detection in the frame at time $t+1$, i.e., the person P_2. If the distance between the histograms of P_1^t and P_1^{t+1} is the smallest, then this is the person P_1 we are following and P_2^{t+1} is the person P_2. On the contrary, P_2^{t+1} is the person P_1 we are following if the distance between the histograms of P_1^t and P_2^{t+1} is smallest.

IV. Experimental Results

A. Experimental protocol

In order to test the efficiency of the proposed technique, we tried the comparison with several histograms, by changing the number of bins. We tried to compute three different RGB histograms and three different HSV histograms before applying the Bhattacharyya distance. For the RGB histograms, we tried:

- 256 bins (one bin per pixel value),
- 85 bins (one bin for 3 pixel values),
- 8 bins (on bin for 32 pixel values)

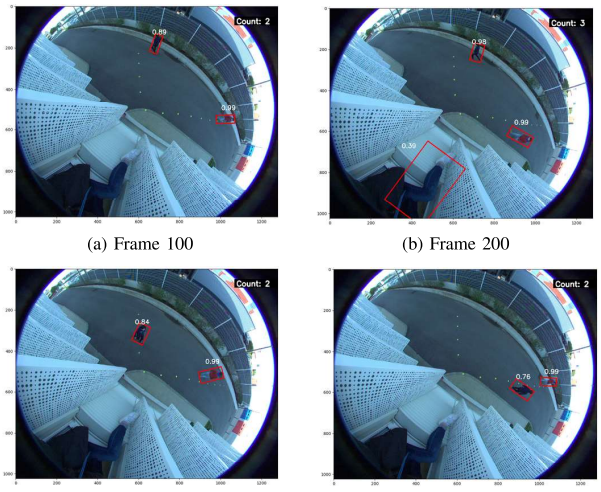
For the HSV histograms, we tried

- 180 bins for H and 256 bins for S and V,
- 50 bins for H and 60 bins for S and V,
- 9 bins for H and 15 bins for S and V.

Two videos are presented here. The first one, video 1 in Fig. 3, was acquired outside with natural lightening, while the second one, video 2, corresponds to inside building conditions, see Fig. 4. The videos are taken from above with a fisheye camera, as diagrammed in Fig. 1.

B. Evaluation of the tracking technique

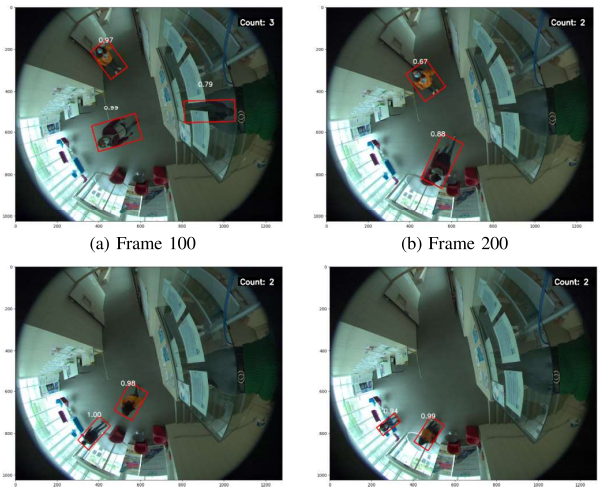
The first video was acquired outside from a window on the second floor. Two people are walking under the camera, they are quite visible and easily-detected as seen in Fig. 3. Figs. 5(a)-(c) illustrate the Bhattacharyya distance computed with the different types of histograms and Tab. I sums up its characteristic values. Here, the RGB histogram with 8 bins per channel and the HSV histogram with 9 bins the Hue and 15 bins for Saturation and Value appear to be the most reliable histograms [17], regarding the Bhattacharyya distance and its characteristics of minimization. Let's consider P_1 as the person we want to follow and P_2 as a second person detected in



(a) Frame 100 (b) Frame 200

(c) Frame 300 (d) Frame 400

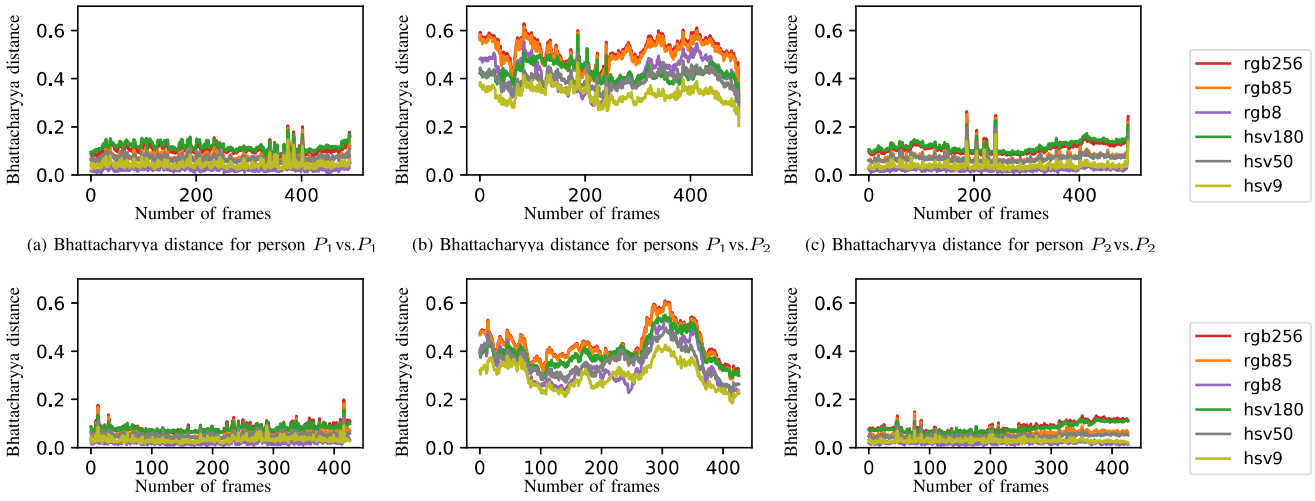
Fig. 3. Detected people in the video 1 acquired by a fisheye camera using the RAPiD method. Originally, images are of size 1024×1280 and downsampled here for displaying.



(a) Frame 100 (b) Frame 200

(c) Frame 300 (d) Frame 400

Fig. 4. Detected persons in the video 2 acquired by a fisheye camera using the RAPiD method. Originally, images are of size 1024×1280 and downsampled here for displaying.



(a) Bhattacharyya distance for person P_1 vs.P_1 (b) Bhattacharyya distance for persons P_1 vs.P_2 (c) Bhattacharyya distance for person P_2 vs.P_2

(d) Bhattacharyya distance for person P_1 vs.P_1 (e) Bhattacharyya distance for persons P_1 vs.P_2 (f) Bhattacharyya distance for person P_2 vs.P_2

Fig. 5. Bhattacharyya distance P_1P_2 comparison with different histograms properties along video 1 in (a)-(c) and video 2 in (d)-(f).

TABLE I
CHARACTERISTIC VALUES OF BHATTACHARYYA DISTANCES IN VIDEO 1

	Minimum	Maximum	Average	Median
Distance P_1P_1				
RGB256	0.067	0.205	0.105	0.103
RGB85	0.046	0.195	0.073	0.069
RGB8	**0.008**	**0.117**	**0.029**	**0.024**
HSV180	0.080	0.186	0.117	0.115
HSV50	0.045	0.146	0.069	0.067
HSV9	0.023	0.146	0.042	0.040
Distance P_1P_2				
RGB256	**0.330**	**0.629**	**0.516**	**0.521**
RGB85	0.308	0.615	0.504	0.510
RGB8	0.238	0.553	0.420	0.422
HSV180	0.302	0.581	0.428	0.422
HSV50	0.289	0.508	0.395	0.397
HSV9	0.204	0.466	0.336	0.339
Distance P_2P_2				
RGB256	0.080	0.264	0.111	0.107
RGB85	0.051	0.253	0.073	0.069
RGB8	**0.007**	0.179	**0.026**	**0.022**
HSV180	0.086	0.227	0.119	0.116
HSV50	0.045	0.195	0.069	0.068
HSV9	0.017	**0.170**	0.038	0.035

TABLE II
CHARACTERISTIC VALUES OF BHATTACHARYYA DISTANCES IN VIDEO 2

	Minimum	Maximum	Average	Median
Distance P_1P_1				
RGB256	0.054	0.199	0.078	0.074
RGB85	0.032	0.183	0.058	0.053
RGB8	**0.008**	0.137	**0.025**	**0.020**
HSV180	0.058	0.156	0.078	0.076
HSV50	0.032	0.128	0.051	0.049
HSV9	0.015	**0.091**	0.033	0.030
Distance P_1P_2				
RGB256	**0.297**	**0.609**	**0.429**	**0.410**
RGB85	0.284	0.603	0.422	0.404
RGB8	0.21	0.543	0.346	0.326
HSV180	0.281	0.551	0.397	0.384
HSV50	0.230	0.497	0.356	0.358
HSV9	0.184	0.426	0.300	0.297
Distance P_2P_2				
RGB256	0.043	0.148	0.083	0.079
RGB85	0.0283	0.142	0.056	0.055
RGB8	0.007	0.104	**0.022**	0.019
HSV180	0.050	0.120	0.079	0.074
HSV50	0.031	0.091	0.047	0.046
HSV9	**0.014**	**0.071**	0.029	**0.027**

the video. Overall, the Bhattacharyya distance P_1P_2 is much higher than the distance P_1P_1 or P_2P_2. This emphasizes the fact the algorithm tracks correctly the desired person.

The second video was taken in a hall, with two people walking around as seen in Fig. 4. Once again, Figs. 5(c)-(e) and Tab. II show that histograms that give minimize the Bhattacharyya distances between the bounding boxes of a same person are the RGB histogram with 8 bins for each channel of the HSV histogram with 9 bins for the Hue channel and 15 bins for Saturation and Value channels. Consequently, short histograms allow an efficient matching of the desired targets between two consecutive frames.

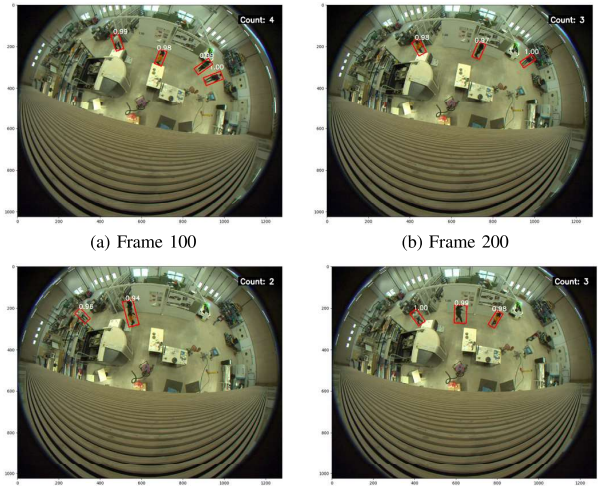
Yet, the most efficient histogram is the one that minimizes the distance P_1P_1 between two frames and also maximizes the distance P_1P_2, computed here on Tab. I and Tab. II. In order to determine the most efficient histogram, we computed a quantitative score [18] for each video defined as follows:

$$S = \frac{\textit{inter-distance}}{\textit{intra-distance}} = \frac{2 \cdot \hat{d}(H_{P_1}, H_{P_2})}{\hat{d}(H_{P_1}, H_{P_1}) + \hat{d}(H_{P_2}, H_{P_2})}, \quad (2)$$

where H_{P_1} and H_{P_2} represent the color histograms of P_1 and P_2 respectively and $\hat{d}$ the average of the distance d (see Eq. (1)) along the video. As shown in Tab. III, the RGB histogram with 8 bins per channel is indeed the most efficient, followed

TABLE III
QUANTITATIVE SCORES S OF THE TWO VIDEOS 1 AND 2, EQ. (2)

	S scores in Video 1	S scores in Video 2
RGB256	4.778	5.329
RGB85	6.904	7.404
RGB8	**15.273**	**14.723**
HSV180	3.627	5.237
HSV50	5.725	7.265
HSV9	8.400	9.667



(a) Frame 100 (b) Frame 200

(c) Frame 300 (d) Frame 400

Fig. 6. Detected persons in the video 3 acquired by a fisheye camera using the RAPiD method. Originally, images are of size 1024×1280 and downsampled here for displaying.

by the HSV histogram with 9 bins for the Hue channel and 15 bins for the Saturation and the Value channels.

The next video, video 3 in Fig. 6, was taken inside a hall. We tried to follow one person when four people are walking simultaneously. The main target to track corresponds to the second person from the right in Fig. 6(a). To do so, for two following frames, we computed the four Bhattacharyya distances between the first detection of the first frame and the four detections of the following frames. Then the model is updated by keeping the histogram of the minimum distance as reference for the next frame. Tab. IV show Bhattacharyya distances with the different histograms. With values under 0.3 and medians values between 0.028 and 0.147, we can assume that the person has been correctly followed throughout the video. Moreover, no undesirable target was ever tracked. However, Figs. 6(c) or 6(d) illustrates that sometimes, people are not detected when the view is busy with many objects and quite far away from the camera. Indeed, the detection reliability depends on object sampling and cluttered background.

C. Angle and distance computation

In this part, we define a method to calculate the position of the person within the camera framework. The position of the person shall refer to the center of the bounding box and is defined with two parameters: an angle α and a distance r:

- α, the angle defined by the vertical axis crossing the image center and the bounding box center,
- r a pixel distance calculated between the bounding box center and the image center.

Figs. 7(a) and 8(a) illustrate some examples in images of our experiments. A distance d is computed, representing the physical distance in meters between the optical axis projection on the ground and the border of the bounding box (usually the feet of the person if this person stands vertically). In short, we

TABLE IV
CHARACTERISTIC VALUES OF BHATTACHARYYA DISTANCE BETWEEN P_1 AND P_1 IN VIDEO 3 WITH FOUR DETECTIONS

	Minimum	Maximum	Average	Median
Distance P_1P_1				
RGB256	0	0.272	0.135	0.135
RGB85	0	0.249	0.087	0.084
RGB8	0	**0.207**	**0.034**	**0.028**
HSV180	0	0.251	0.145	0.147
HSV50	0	0.449	0.081	0.076
HSV9	0	0.496	0.045	0.039

are looking for a model that evaluates d in meters as a function of r in pixels. Let v_1 and v_2 be two vectors in the image, both starting at the image center. v_1 is fixed in the image framework, pointing towards the image top point (vertically), while v_2 is pointed at the person. Examples are given in Fig. 7(a). Let x_B be the x coordinates (column) of the bounding box. The angle α can be calculated using the following equation which uses the dot product of v_1 and v_2:

$$\alpha = \begin{cases} \arccos(v_1.v_2) & \text{if} \quad x_B > \frac{1024}{2} \\ 2\pi - \arccos(v1.v2) & \text{elsewhere.} \end{cases} \tag{3}$$

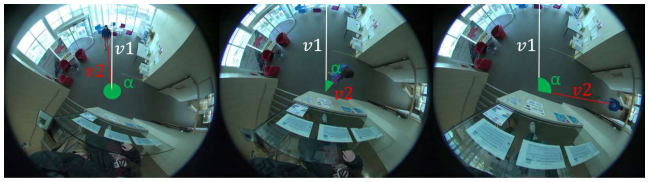
This equation is separated in two cases: the first one when the person is on the right side of the image ($x_B > \frac{1024}{2}$ here) - because the resolution is of 1024×1024 pixels - and the other when the person is on the left ($x_B < \frac{1024}{2}$). An example of the evolution of α during a video is shown in Fig. 7.

To calculate d, we have to calculate the norm r in pixels of the vector v_2. When converting r into d, we must take into account the geometrical distortion of the camera lens. Because our camera uses an equidistant projection, the distance r is proportional to the angle θ between the optical axis of the camera and the person [20]. Besides, the Thales theorem gives the following relation : $\theta = \arctan(\frac{d}{h})$, with h being the height of the camera. This means that the evolution of d according to r, can be modeled by a tangent function f:

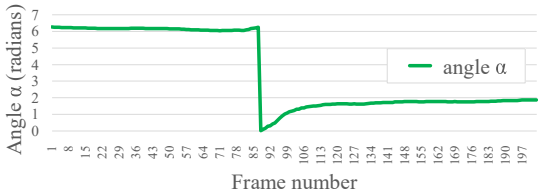
$$f(x) = a \cdot \tan(b \cdot x), \tag{4}$$

where a and b are two parameters to be determined, a being proportional the the camera focal length and b being inversely proportional to the camera resolution. In order to evaluate the right values of a and b in our model, yellow markers were placed on the ground every meter for 8 meters as specified, starting from the center of the frame. A person then walked along this ground marking with a constant speed, as shown in Fig. 8(a). The component r is computed for each frame and d is calculated using the function f with different values for a and b. At end, the most accurate values for those parameters were $a = 3.6$ and $b = \frac{1}{350}$, it complies with the ground truth, as illustrated in Fig. 8(b).

Finally, in our evaluation model of the person's position, we determined an angle and a distance. We can thus make an analogy with polar coordinates, also based on an angle and a distance. In addition, we have taken into account the camera lens geometrical distortion in our distance calculation, which allows us to know precisely the person's position in relation to the camera, which can serve many use cases.



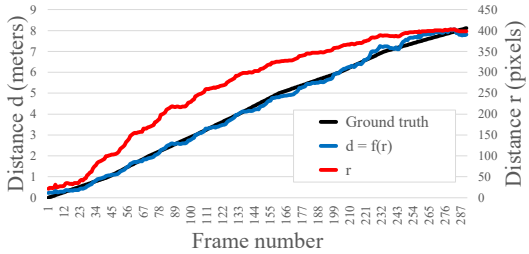
(a) Visualization of α over time



(b) Measures of the angle α (in radians)

Fig. 7. Evolution of the angle in radians where 0 is tied to the vertical axis.

(a) Visualization of r and of the ground truth over time



(b) Measures of the distances d (in meters) and r (in pixels) compared to the ground truth.

Fig. 8. Experimental results showing distances comparison between our model calculations and ground truth.

V. Conclusion

In this communication, we present a model which enables us to locate and track a person in a scene filmed by a top-view fisheye camera. First, this model detects the people in each image using the RAPiD deep learning method [4], returning bounding boxes coordinates (horizontal and vertical coordinates, width, heigh and angle of the box). It then isolates each bounding box, applies color histograms on it, calculates the distance between each histogram using the Bhattacharyya method. We compared the Bhattacharyya distances of several histograms to see which one would give better scores. Overall, regarding our experiments, the most efficient comparison is obtained with the RGB histogram with 8 bins for each channel. The target to track is tied to the shortest histogram distance between each frame, ensuring the tracking of one and only person throughout the video, without taking into account the movement of the other people present in the video. Finally, it provides two pieces of information on the person's position in relation to the camera, an angle in radians and a distance in meters without the need of any calibration toolbox. These information can be used to track the movement of the person over time as they pass through the field of view of the camera.

The tracking process using the Bhattacharyya distance is not expensive in computation. However, some investigations must be led in the future in order for the detection algorithm [4] to be able to perform in real time.

References

[1] O. Haggui, M. Agninoube Tchalim, and B. Magnier, "A comparison of opencv algorithms for human tracking with a moving perspective camera," *IEEE EUVIP -to appear-*, 2021.

[2] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE TPAMI*, vol. 34, no. 4, pp. 743–761, 2011.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE CVPR*, vol. 1, 2005, pp. 886–893.

[4] Z. Duan, O. Tezcan, H. Nakamura, P. Ishwar, and J. Konrad, "Rapid: rotation-aware people detection in overhead fisheye images," in *IEEE/CVF CVPR Workshops*, 2020, pp. 636–637.

[5] S.-H. Chiang, T. Wang, and Y.-F. Chen, "Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches," *Image and Vision Computing*, vol. 105, p. 104069, 2021.

[6] A.-T. Chiang and Y. Wang, "Human detection in fish-eye images using hog-based detectors over rotated windows," in *IEEE ICME Workshop*, 2014, pp. 1–6.

[7] V. Srisamosorn, N. Kuwahara, A. Yamashita, T. Ogata, S. Shirafuji, and J. Ota, "Human position and head direction tracking in fisheye camera using randomized ferns and fisheye histograms of oriented gradients," *The Visual Computer*, pp. 1–14, 2019.

[8] O. Krams and N. Kiryati, "People detection in top-view fisheye imaging," in *IEEE AVSS*, 2017, pp. 1–6.

[9] D. Scaramuzza and K. Ikeuchi, "Omnidirectional camera," 2014.

[10] B. Magnier, F. Comby, O. Strauss, J. Triboulet, and C. Demonceaux, "Highly specific pose estimation with a catadioptric omnidirectional camera," in *IEEE IST*, 2010, pp. 229–233.

[11] P. Hansen, P. Corke, and W. Boles, "Wide-angle visual feature matching for outdoor localization," *IJRR*, vol. 29, no. 2-3, pp. 267–297, 2010.

[12] J. J. Kumler and M. L. Bauer, "Fish-eye lens designs and their relative performance," in *Current developments in lens design and optical systems engineering*, vol. 4093. International Society for Optics and Photonics, 2000, pp. 360–369.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009, pp. 248–255.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE CVPR*, 2017, pp. 2117–2125.

[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[16] G. Stockman and L. Shapiro, "Computer vision," 2001, pp. 209–233.

[17] J. V. P Pérez, C. Hue and M. Gangnet, "Color-based probabilistic tracking," in *ECCV*. Springer Berlin Heidelberg, 2002, pp. 661–675.

[18] T. R. A. Zweng and M. Kampel, "Evaluation of histogram-based similarity functions for different color spaces," in *CAIP*, 2011, pp. 455–462.

[19] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE TPAMI*, vol. 25, no. 5, pp. 564–577, 2003.

[20] W. Hou, M. Ding, N. Qin, and X. Lai, "Digital deformation model for fisheye image rectification," *Optics express*, vol. 20, no. 20, pp. 22 252–22 261, 2012.