



**HAL**  
open science

# Spherical Region-Based Matching of Vanishing Points in Catadioptric Images

Jean-Charles Bazin, Inso Kweon, Cedric Demonceaux, Pascal Vasseur

► **To cite this version:**

Jean-Charles Bazin, Inso Kweon, Cedric Demonceaux, Pascal Vasseur. Spherical Region-Based Matching of Vanishing Points in Catadioptric Images. The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS, Rahul Swaminathan and Vincenzo Caglioti and Antonis Argyros, Oct 2008, Marseille, France. inria-00325320

**HAL Id: inria-00325320**

**<https://inria.hal.science/inria-00325320>**

Submitted on 28 Sep 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Spherical Region-Based Matching of Vanishing Points in Catadioptric Images

Jean-Charles Bazin<sup>1</sup>, Inso Kweon<sup>1</sup>, Cedric Demonceaux<sup>2</sup>, and Pascal Vasseur<sup>2</sup>

<sup>1</sup> RCV Lab, KAIST, Daejeon, Korea

jcbazin@rcv.kaist.ac.kr, iskweon@ee.kaist.ac.kr

<sup>2</sup> MIS, UPJV, Amiens, France

{pascal.vasseur, cedric.demonceaux}@u-picardie.fr

**Abstract.** A large literature exists for rotation estimation using vanishing points (VP). All these VP-based methods require to match the vanishing points in consecutive frames. It is usually considered that this matching step is trivial. However whereas some techniques exist, they suffer from some important limitations and might not work correctly in real robotic applications. For example, the continuity constraint (which aims to match the pair of VPs having the lowest angular distance) cannot handle large rotations and the line matching technique is very slow to execute and assumes accurate/stable line detection. In this paper, we present a fast and robust method to build the correspondences of VPs in catadioptric images. This work is strongly motivated by our research on real-time rotation estimation for dynamic vehicles using catadioptric vision. The underlying idea of the proposed method consists in matching, by histogram comparison, the spherical regions defined by the VPs in the equivalent sphere. Experiments have demonstrated the efficiency of this approach in terms of speed and robustness to translation, rotation, dynamic environment and image blurring.

## 1 Introduction

Vanishing Points (VP) are invariant to translation and thus provide key information to estimate the rotation of a camera. That is why there exists a large literature for rotation estimation using vanishing points. This paper particularly focuses on catadioptric images. These images are obtained from a vision system composed of a mirror with a specific shape and a conventional camera. Independently of the vision system that is used (e.g. traditional or catadioptric camera), most of the existing methods for rotation estimation proceeds in two steps: first, VP extraction and second, VP matching. The VPs are usually extracted by detecting lines and their intersections [1][2][3][4]. To match the VPs, two methods are widely used. The first one refers to the continuity constraint which matches the pairs of VPs having the lowest angular distance [5]. This approach is extremely fast but assumes that each rotation angle is less than  $45^\circ$ . This assumption might not be verified for highly dynamic robotic systems or low frame-rate videos for example. The second method consists in matching the

lines associated to the VPs. Several types of constraints have been used to match the lines like neighborhood correlation, epipolar geometry, orientation, length, overlapping, average contrast or mid-point (due to space limitation, cf [6] and [7] for a review). Whereas these constraints may provide interesting results, they suffer from at least one of the following limitations. First, many of them assume the camera geometry is known to impose epipolar constraint. Second, it is hard to extend some of these methods to catadioptric images for taking distortions into account. Third, they assume a reliable line detection but due to camera calibration error, a line might be split into two parts during the line extraction [2][8][9]. Fourth, after matching the lines, these methods usually assume a rotation lower than  $90^\circ$  to match the VPs. Finally, they are computationally expensive, especially when many lines are extracted, and thus cannot be used for real-time applications.

Regarding the existing works for VP matching, we aim to develop a new method that can overcome the listed limitations. After recalling the sphere equivalence, we will introduce our proposed method for VP matching. Finally, we will test its performance on several catadioptric videos.

## 2 Proposed Method

The problem of VP matching is motivated by our application of real-time rotation estimation for dynamic vehicles using catadioptric vision. Due to the specificities of this application, our goal is to develop a method for VP matching that must verify the following constraints:

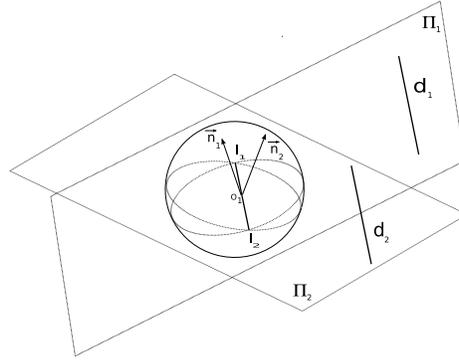
- automatic: no manual interaction during the execution
- robust to motion
- fixed complexity: to guarantee execution time
- fast: for real-time application

Our solution is based on the fact that VPs provide strong photometric information. More precisely, our proposed method consists in matching, by histogram comparison, the spherical regions defined by the VPs in the equivalent sphere. In the following, we recall the sphere equivalence and present the details of our proposed algorithm.

### 2.1 Sphere equivalence

Geyer and Daniilidis have demonstrated the equivalence for the central catadioptric sensors (single viewpoint constraint) with a two-step projection via a unitary sphere centered on the focus of the mirror [10]. A key advantage of the sphere equivalence is that it allows working in a general framework, i.e. the sphere, independently on the fact that a hyperbolic or a parabolic mirror is used. A second important advantage is that the sphere space provides some very interesting projection properties for lines. It has been proven that a line is projected as a great circle in the sphere and the great circles associated to a pencil of

3D parallel lines intersect in two antipodal points in the sphere [10]. These two important properties are depicted on Figure 1 and will be used for our proposed method of VP matching.

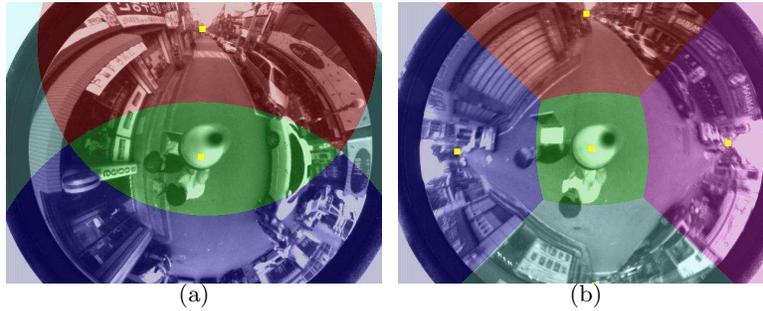


**Fig. 1.** A line in the world is projected as a great circle in the sphere and the projections of parallel lines intersect in two antipodal points in the sphere. These two antipodal points correspond to the vanishing points of the parallel lines.

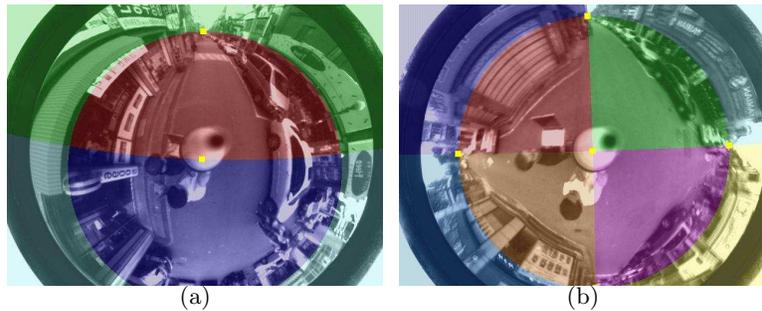
## 2.2 Region definition

In this section, we investigate some region definitions for efficient VP matching. Our initial proposition was based on the remark that the neighborhoods (in sphere or image space) of the VPs are different enough to distinguish, match and track the VPs. However early experiments have shown an important difficulty of this approach: the VPs might appear/disappear in the image so that it is not possible to build and/or match their neighborhoods. That is why we preferred developing another region definition. An important comment is that, even if the VPs do not lie in the image, they lie in the sphere space and can provide strong photometric information. An intuitive region definition for VP matching consists in clustering each pixel with respect to its nearest VP. Mathematically, each  $i^{th}$  vanishing point  $V_i$  defines a region  $R_i$  and a pixel projected on the sphere at  $P_s$  belongs to the region  $R_k$  where  $k$  is the solution of  $\arg_i \max V_i \cdot P_s$ . Since the vanishing points of parallel lines correspond to 2 antipodal points in the sphere, we also take into account the list of opposite VPs  $-V_i$ . Therefore, given  $N$  VPs, the sphere is split into  $2N$  regions. An example of regions obtained by this definition (referred as definition1 in the following) is shown in Fig 2. One may notice that, in Fig 2(b), whereas the sphere is split into  $2 \times 3 = 6$  regions, only 5 regions are contained in the image. This issue will be analyzed in section 2.4.

We also investigate a second region definition. Based on the properties of parallel lines projection in the sphere, each VP associated to a group of parallel



**Fig. 2.** Example of splitting the image into regions according to definition1 with 2 (left) and 3 (right) dominant vanishing points.



**Fig. 3.** Same than figure 2 but the image splitting is done according to definition2.

lines can be represented by a unit vector in sphere space or a plane normal to this vector. Therefore, given  $N$  vanishing points, the sphere is split by  $N$  planes. The spherical points are clustered into regions depending on their position with respect to the planes defined by each VP. The algorithm for  $N = 2$  is given in Algorithm1. The splitting of the sphere depends on the order of the vanishing points, i.e. switching  $VP_1$  and  $VP_2$  in Algorithm1 leads to a different splitting. This region definition is referred as definition2 in the following and some examples of split images are shown in Figure 3. In these examples, the projections of all the spherical regions lie in the image and experiments have shown that it is generally the case. Exemptions will be analyzed in section 2.4.

Both region definitions provide some important photometric cues and can lead to interesting results. One may notice that definition1 and 2 are equivalent for  $N = 1$ . They will be experimentally studied in section 3.2. It is worthwhile to note that the sphere provides a simplified framework for splitting the image into regions. First, the VPs are usually obtained in the sphere since the algorithms of line detection for general catadioptric images [2][8][9] perform in this space.

And second, it is needed to compute only  $N$  dot products to classify a pixel in its corresponding region, where  $N$  is the number of vanishing points ( $N$  usually equals to 2 or 3).

---

**Algorithm 1** FnSphereSplitting ( $VP_1, VP_2$ )
 

---

```

// (VP1,VP2) : the 1st and 2nd vanishing points
// FnProjectionSphere(i,j) : projects the point (i,j) on the sphere
// FnAddInRegion(i,j,k) : add the point (i,j) to the region k
// s : sampling step for speed purpose
for i = 1 : s + 1 : height do
  for j = 1 : s + 1 : width do
    Ps = FnProjectionSphere(i, j)
    pos1 = VP1 · Ps
    pos2 = VP2 · Ps
    if pos1 > 0 and pos2 > 0 then FnAddInRegion(i, j, 1)
    if pos1 > 0 and pos2 ≤ 0 then FnAddInRegion(i, j, 2)
    if pos1 ≤ 0 and pos2 > 0 then FnAddInRegion(i, j, 3)
    if pos1 ≤ 0 and pos2 ≤ 0 then FnAddInRegion(i, j, 4)
  end for
end for

```

---

### 2.3 Region Comparison

In order to match the regions of 2 split spheres, we need a representation of the regions and a method to compare them. We assume that the number of spherical regions is the same in the 2 images, i.e. same number of VPs. One approach would be to extract feature points (e.g. Harris corners [11] or SIFT [12]) in each region and match them. However it is computationally expensive and thus cannot verify our real-time constraint. Our approach is to represent each region by a histogram using the intensity of every pixel inside this region and then match the histograms. This technique of region-based matching has been previously used for stereo vision [13], optical flow [14] or egomotion estimation [15]. It provides the key advantage to be very fast and further experiments will also demonstrate its robustness. Several distances have been defined to compute the similarity between two histograms (cf [16] for a review). For experiments, we have implemented and evaluated two popular methods: Minkowski-form distance of order 1 ( $L_1$ ) and histogram intersection [17]:

$$d(H, K) = \sum_{i=1}^{Nb} |H(i) - K(i)| \quad \text{and} \quad d(H, K) = 1 - \frac{\sum_{i=1}^{Nb} \min(H(i), K(i))}{\sum_{i=1}^{Nb} K(i)} \quad (1)$$

where  $H$  and  $K$  are two histograms composed of  $Nb$  bins.

As the sizes of the regions can be different, our algorithm must be able to compare histograms containing a different number of observations. This problem can be solved by histogram normalization, which can be done in two ways:

$$\overline{H}(n) = \frac{H(n)}{\sum_{i=1}^{Nb} H(i)} \quad \text{and} \quad \overline{H}(n) = \frac{H(n)w(n)}{\sum_{i=1}^{Nb} H(i)} \quad (2)$$

where  $w(n)$  is the width of the  $n^{th}$  bin. The basic and intuitive solution is to divide the number of observation of each bin by the total number of observations (cf eq (2)-left). Thus each bin corresponds to the proportion of data and their sum is 1. Another solution consists in dividing each bin by the total number of observations and multiplying by the bin width  $w$  (cf eq (2)-right). Whereas this normalization is less intuitive, it permits to use the histogram to model a probability density function (the integral of the histogram is 1).

## 2.4 Matching of regions and VPs

To build the correspondences of regions between the previous and current images, we had initially performed the following procedure: for each region in the current image, test all the regions in the previous image and select the one maximizing the histogram similarity. However this basic method has several problems. First of all, two regions in the current image can be matched to the same region in the previous image. One solution would be to perform a disambiguation process by double-cross checking (i.e. matching from current to previous images, then from previous to current images and finally keep the matches that have the same single correspondence). However it might lead to the fact that a region has no correspondence. Second, contrary to what could be thought, once the regions are matched, it is not trivial to get the combination of VPs that leads to this matching. Third, we could obtain a solution that is not physically possible. Indeed, the list of matched VPs in the previous and current images ( $v_{i-1}^i$  and  $v_i^i$ ) must verify the relation  $v_i^i = Rv_{i-1}^i$ , where  $R$  is the rotation matrix between the two images. If this relation is verified, the matching is said *physically possible* otherwise *physically impossible* because a rotation that maps the VPs does not exist. Regarding all these difficulties, we preferred the following approach: we test all the physically possible matching solutions and pick up the one that maximizes the sum of the histogram comparisons for all the regions. Given a VP, it exists only 4 combinations of the two other VPs thanks to the right-hand constraint. So for 3 VPs and their opposite direction, it exists  $2 \times 3 \times 4 = 24$  physically possible hypothesis. However some hypothesis are repeated. There exists a total number of 16 non-repeated combinations of VPs:  $(x, y, z)$ ,  $(x, z, -y)$ ,  $(x, -z, y)$ ,  $(x, -y, -z)$ ,  $(y, x, -z)$ ,  $(y, -x, z)$ ,  $(z, y, -x)$ ,  $(z, -y, x)$ ,  $(-z, y, x)$ ,  $(-z, -y, -x)$ ,  $(-y, x, z)$ ,  $(-y, -x, -z)$ ,  $(-x, y, -z)$ ,  $(-x, z, y)$ ,  $(-x, -z, -y)$ ,  $(-x, -y, z)$ . The number of non-repeated combinations can be reduced to only 4 if we assume that the vertical vanishing point remains stable (planar motion).

In some cases, the number of regions between two images might be different because some spherical regions might not be projected inside the image. To handle this situation, we used the following simple technique: if the projection of a spherical region in the image contains less than a few number of pixels (less than 20 in our code), its histogram distance is set to 0 for all its physically possible matching regions in the other image.

In a theoretical point of view, the method proposed in this paper assumes the regions are invariant to camera motion. In practice, the histograms representing the regions could change a lot when the camera suddenly approaches or leaves an object that represents a large part of the field of view (e.g. wall, near car, etc...). In our application, the vehicle (ground or aerial) is expected to approach no objects very suddenly. Therefore the histograms should not change dramatically. However in some real cases, this constraint might not be verified and the camera may approach some large objects, such as a car, but experimental results will show that our algorithm can successfully handle such situations.

### 3 Experimental Results

#### 3.1 Methods used for comparison

To compare our method, we have implemented two existing techniques: the continuity constrain and one method of line matching. The continuity constraint matches the pair of VPs having the lowest angular distance. Formally, the  $i^{th}$  vanishing point  $v_i^t$  of frame  $t$  is matched to the vanishing point  $v_j^{t-1}$  of frame  $t - 1$  which is solution of:

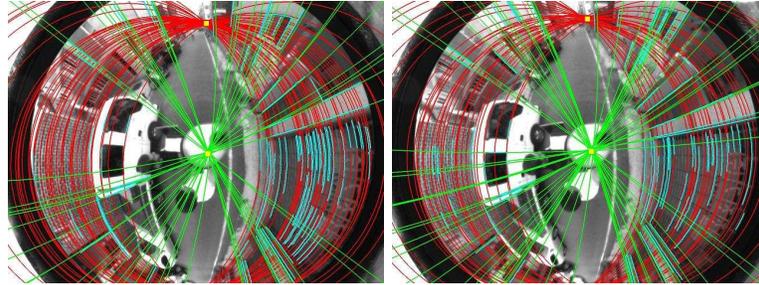
$$\arg \max_j |v_i^t \cdot v_j^{t-1}| \quad (3)$$

The choice of a method for line matching was more difficult. Indeed the epipolar geometry is unknown and basic experiments have shown that simple constraints based on orientation, length, overlapping, average contrast or mid-point did not work in our catadioptric videos. Therefore we have selected the method of [6] for short range motion. The idea is to treat each line as a list of points to which neighborhood correlation is applied as a measure of similarity (cf Fig 4). The point-to-point correspondence with an other line is obtained by searching along the line with a winner-takes-all matching strategy. In practice, the matching cost of two lines  $l_i$  and  $l_j$  is defined as:

$$cost(l_i, l_j) = \sum_{P_i \in l_i} \max_{P_j \in l_j} ZNCC(P_i, P_j) \quad (4)$$

where  $ZNCC(P_i, P_j)$  computes the zero mean normalized cross-correlation between the 2 square neighborhoods centered at  $P_i$  and  $P_j$ . Two lines  $l_i$  and  $l_j$  are matched if  $cost(l_i, l_j) > thresh$  and  $l_j$  is the solution of  $\arg \min_{l_k} cost(l_i, l_k)$ . Since some lines in the left image may be matched to a same line in the right image, a global disambiguation process for selecting the most relevant correspondences of lines is applied by double-cross checking. The final step matches

each VP of the current frame with the VP of the previous frame having the highest number of corresponding lines. An important remark is that the line matching permits to build the correspondences between the vanishing points (and between the groups of parallel lines) but only up to a sign. For example this approach cannot differentiate the matchings  $v^{t-1} \leftrightarrow v^t$  and  $v^{t-1} \leftrightarrow -v^t$ . This is very important because an error on the sign of the VP leads to a totally wrong estimation of the rotation. Generally, the choice between  $v^t$  and  $-v^t$  is made on the maximum of  $v^{t-1} \cdot v^t$  and  $v^{t-1} \cdot -v^t$ . This technique can deal with rotation lower than  $90^\circ$ . Finally, since the line matching procedure is very slow, our implementation aims to match only the 10 longest (i.e. the most stable) lines of each group of parallel lines.



**Fig. 4.** Extraction of VPs (displayed in yellow squares), their associated lines (red and green conics) and associated points (cyan) by [2] in two consecutive images of the sequence *Gungdong1*. The line matching technique of [6] for short range motions aims to build the line correspondences by matching the points associated to the lines.

### 3.2 Results

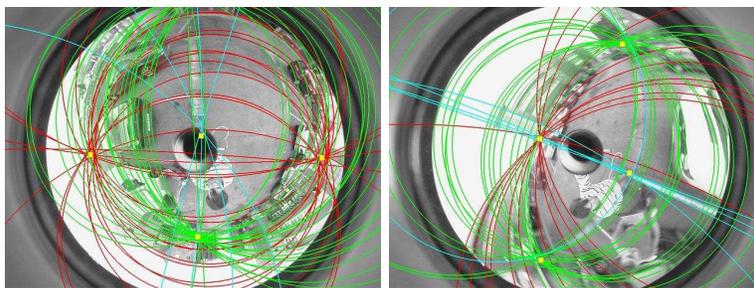
For clarity to general readers, we explicitly write the procedure of the proposed method. It is composed of the following main steps:

1. given a list of VPs for the previous and current images, build the spherical regions of both images (cf section 2.2) and their histograms.
2. for each physically possible solution
  - (a) compute the histogram similarity between each region in the current image and their corresponding region in the previous image (cf section 2.3)
  - (b) compute the total score as the sum of the similarities
3. the VP matching corresponds to the solution having the highest score

For the experiments, the camera is calibrated by [18] and we have used the framework of [2] to detect the lines and the vanishing points. To validate our

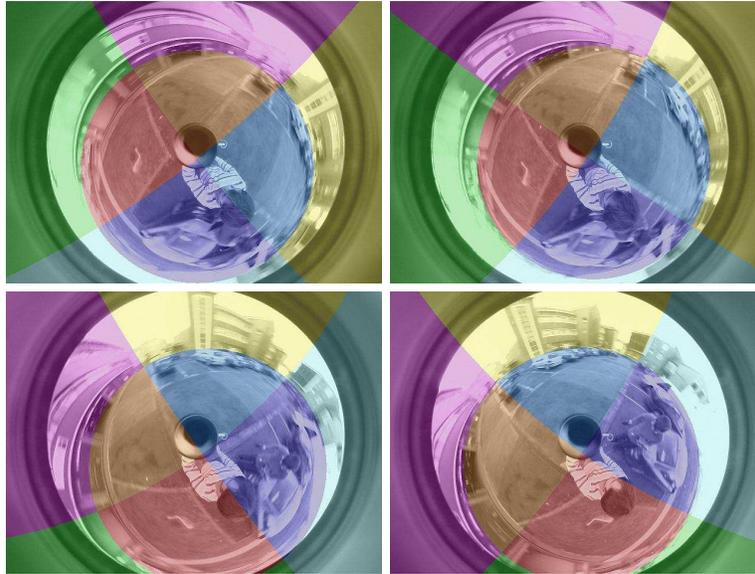
proposed method, we have performed several series of experiments and for each experiment, we have tested the two region definitions and the two histogram distances. Due to space limitation, we only include the results obtained by the region definition2. For more results, the readers are invited to refer to our website <http://rcv.kaist.ac.kr/~jcbazin/>.

In a first experiment, we have tested two important aspects on the sequence *ParkingStraight*. This sequence contains a large rotation around the z-axis and in some frames, the VPs lie outside the image (Fig 5). As depicted in Fig 6, our method is able to match the regions and thus can correctly track the associated VPs. In Fig 7, we compare the rotation angles estimated by the continuity constraint and our proposed method with a calibrated Inertial Measurement Unit (IMU). It is important to note that the continuity constraint failed for this sequence due to the rotation higher than  $45^\circ$  around the z-axis (yaw angle) at the 69<sup>th</sup> frame. The line matching based approach managed to track the VPs for small intra-frame motions. However as soon as rotation occurs, correlation fails because the ZNCC is not invariant to rotation.

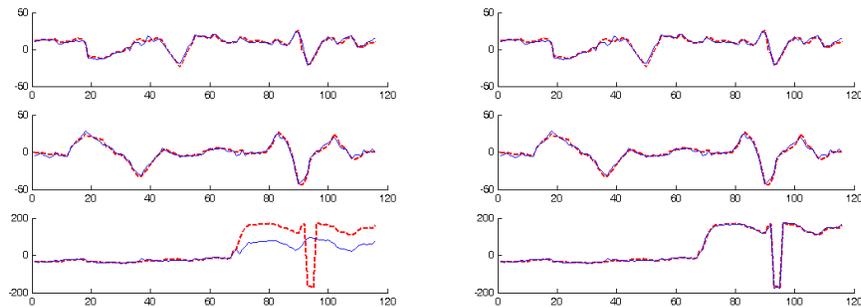


**Fig. 5.** Extraction of VPs (displayed in yellow squares) and their associated lines (one color per group of parallel lines) in some frames of the sequence *ParkingStraight* where some VPs lie outside the image.

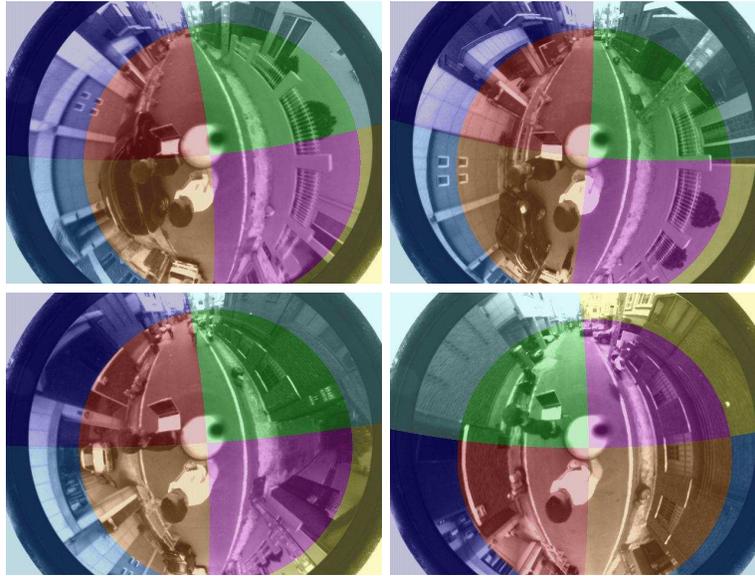
In a second test, we have analyzed the dependency of the histogram matching to the translation on the sequence *Gungdong1*. Experiments showed that translation has no effect at the motion speed of the original sequence (about 30cm between each image at 3fps). Thus the regions were correctly matched for the whole sequence. Obviously, if the translation speed is very large, then it becomes very hard to match the regions. To study faster motion (i.e. larger translation), we have skipped some images of the sequence and applied the same method. Results have shown that we can handle up to about 4meters between each image (40 images were skipped) and still correctly track the VPs. Figure 8 shows two pairs of images with 40 (success) and 50 (failure) skipped frames. The continuity constraint and the line matching based approach worked properly in the original sequence because there are only very small rotations and translations.



**Fig. 6.** Correct matching of the spherical regions in 4 consecutive images of the sequence *ParkingStraight* with a large rotation motion using definition2. The matched regions are drawn in the same color. Note that due to a rotation higher than  $45^\circ$  around z-axis, the continuity constraint fails for this sequence.



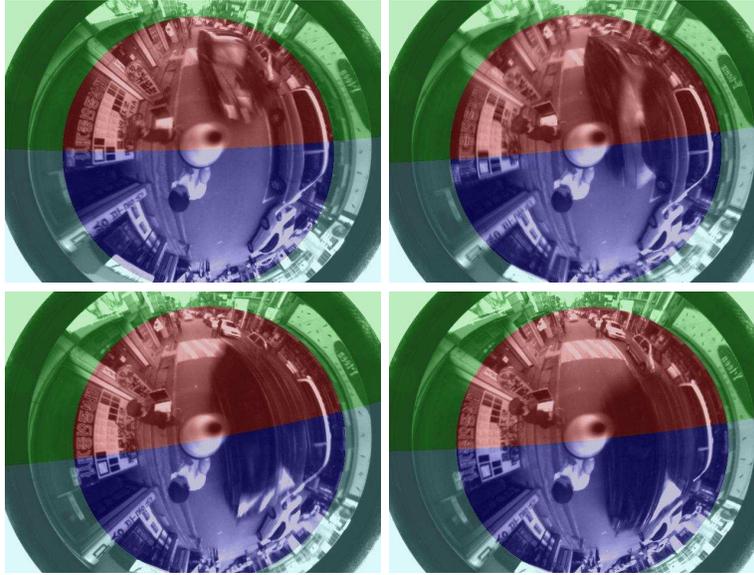
**Fig. 7.** Comparison of the three rotation angles roll (top), pitch (middle) and yaw (bottom) in degrees obtained by the continuity constraint (left) and the proposed method (right) along the 120 frames of the sequence *ParkingStraight*. These angles are drawn in blue and the ground truth data obtained by a calibrated IMU in red. The error of the continuity constraint occurs in yaw angle because the vertical VP is correctly tracked and the 2 horizontal VPs have been mismatched.



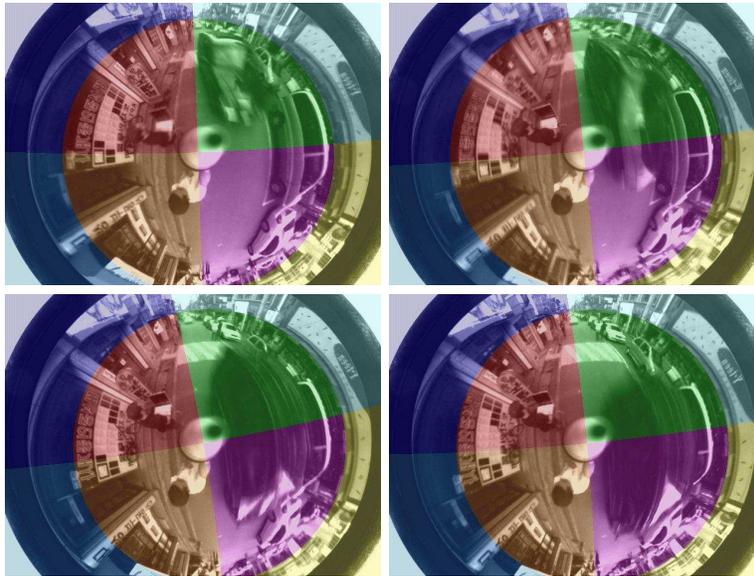
**Fig. 8.** Matching of the spherical regions in 2 pairs of images of the sequence *Gungdong1* with large translation using definition2. Top-row: success with around 4 meters displacement (40 skipped frames). Bottom-row: failure with around 5 meters displacement (50 skipped frames).

Finally, we have applied our algorithm in a difficult experiment. The sequence *Gungdong2* contains two important difficulties. First, it is a dynamic environment composed of moving cars and pedestrians. And second, large objects (cars) approach the camera. This sequence contains also an additional specificity: only 2 VPs are detected. As depicted in Fig 9, our approach works successfully and can correctly handle the two difficult aspects of dynamic environment and large objects approaching the camera. To test our algorithm with 3 VPs, we have synthesized a virtual third VP which is orthogonal to the 2 other VPs. Fig 10 shows that our proposed method works also with 3 vanishing points. The continuity constraint worked correctly for this sequence because the intra-frame rotation is small. The line matching based approach failed for some frames especially because of the image blur.

For all the studied sequences, both region definitions and both histogram distances have lead to the same results of region matching. We have tested several different values for the number  $Nb$  of histogram bins and experiments showed that this parameter does not play a key role as long as  $Nb > 20$ bins. The continuity constraint runs almost instantaneously. On the contrary, matching lines is extremely time consuming, especially when the scene contains a large number of lines or long lines. Thus it is totally impossible to use this approach for real-time systems. In terms of complexity, to split the sphere, the proposed



**Fig. 9.** Matching of the spherical regions in 4 consecutive images of the sequence *Gungdong2* with a dynamic environment (car and pedestrians) and large object approaching the camera with 2 vanishing points using region definition2.



**Fig. 10.** Same than Figure 9 but using a virtual third vanishing point.

algorithm performs of the order of  $N * height * width / (s+1)^2$  dot products where  $s$  is the sampling rate (in pixels) and  $N$  the number of vanishing points. We have tested the importance of  $s$  and experiments have shown that  $s = 0, 5, 10$  pixels reach similar results for region matching so we finally set  $s = 10$ . The execution time using non-optimized C++ code was about 6ms only per frame for sphere splitting and histogram comparison with  $s = 10$  pixels and  $N = 3$  vanishing points. Therefore the proposed method can be easily used for real-time applications.

## 4 Conclusion

Estimating the rotation by vanishing points is very popular. This approach usually consists of two steps: first, extract the VPs and second, match them in a sequence. It is widely considered that the second step is so simple that there is no reason to work on this aspect. However we have shown that existing methods (essentially continuity constraint and line matching) suffer from many limitations: they cannot handle large rotations, are very slow to execute and/or assume stable line detection. As a consequence, these methods cannot be applied for our work on real-time rotation estimation of dynamic vehicles using catadioptric vision. That is why in this paper, we have presented an original method to match the VPs in catadioptric videos. Our approach offers three important advantages. First its complexity depends only on the image size. Therefore it is fixed for any images, which permits to guarantee the execution time, and it is independent of the number/length of lines which is an important aspect for catadioptric images because a high number of long lines could be detected. Second it is very fast and can be applied for real-time applications since it runs in 6ms only. Finally, experiments on real data have demonstrated that our algorithm is robust to large rotation, translation, dynamic environment and image blurring.

## 5 Acknowledgement

This work has been performed within the STAR project of the Hubert Curien (Egide) partnership between RCV Lab at KAIST-Korea and MIS at UPJV-France. Authors also thank the anonymous reviewers for their useful comments and suggestions.

## References

1. Antone, M.E., Teller, S.J.: Automatic recovery of relative camera rotations for urban scenes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00). 2282–2289
2. Bazin, J.C., Kweon, I.S., Démonceaux, C., Vasseur, P.: Rectangle extraction in catadioptric images. In: ICCV Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS'07)

3. Magee, M.J., Aggarwal, J.K.: Determining vanishing points from perspective images. *Computer Vision, Graphics and Image Processing* (1984) 256–267
4. Quan, L., Mohr, R.: Determining perspective structures using hierarchical hough transform. *Pattern Recognition Letters* **9**(4) (1989) 279–286
5. Bazin, J.C., Kweon, I.S., Démonceaux, C., Vasseur, P.: UAV attitude estimation by vanishing points in catadioptric image. In: *IEEE International Conference on Robotics and Automation (ICRA'08)*. (2008)
6. Schmid, C., Zisserman, A.: Automatic line matching across views. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*. (1997) 666–671
7. Karimian, G., Raie, A.A., Faez, K.: A new efficient stereo line segment matching algorithm based on more effective usage of the photometric, geometric and structural information. *IEICE - Transactions on Information and Systems* **E89-D**(7) (2006) 2012–2020
8. Ying, X., Hu, Z.: Catadioptric line features detection using hough transform. In: *International Conference on Pattern Recognition (ICPR'04)*. IV: 839–842
9. Vasseur, P., Mouaddib, E.M.: Central catadioptric line detection. In: *Proceedings of the British conference on Machine vision (BMVC'04)*
10. Geyer, C., Daniilidis, K.: Catadioptric projective geometry. *International Journal of Computer Vision (IJCV'01)* **45**(3) (2001) 223–243
11. Harris, C., Stephens, M.: A combined corner and edge detection. In: *Proceedings of The Fourth Alvey Vision Conference*. (1988) 147–151
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision (IJCV'03)*. Volume 20. (2003) 91–110
13. Marapane, S.B., Trivedi, M.M.: Region-based stereo analysis for robotic applications. *IEEE Transactions on Systems, Man and Cybernetics* **19**(6) (1989) 1447–1464
14. Fuh, C.S., Maragos, P.: Region-based optical flow estimation. *Computer Vision and Pattern Recognition (CVPR'89)* (1989) 130–135
15. Irani, M., Rousso, B., Peleg, S.: Recovery of ego-motion using region alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'97)* **19**(3) (1997) 268–272
16. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision (IJCV'00)* **40**(2) (2000) 99–121
17. Swain, M., Ballard, D.: Color indexing. *International Journal of Computer Vision (IJCV'91)* **7**(1) (1991) 11–32
18. Barreto, J.P., Araujo, H.: Geometric properties of central catadioptric line images and their application in calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'05)* **27**(8) 1327–1333