

# 360° Tracking Using a Virtual PTZ Camera

Luca Greco and Marco La Cascia<sup>(✉)</sup>

DIID, Università degli Studi di Palermo, Viale delle Scienze, Palermo, Italy  
{luca.greco,marco.lacascia}@unipa.it

**Abstract.** Object tracking using still or PTZ cameras is a hard task for large spaces and needs several devices to completely cover the area or to track multiple subjects. The introduction of 360° camera technology offers a complete view of the scene in a single image and can be useful to reduce the number of devices needed in the tracking problem. In this paper we present a framework using 360° cameras to simulate an unlimited number of PTZ cameras and to be used for tracking. The proposed method to track a single target process an equirectangular view of the scene and obtains a model of the moving object in the image plane. The target is tracked analyzing the next frame of the video sequence and estimating the P,T and Z shifts needed to keep the target in the center of the virtual camera view. The framework allows to use a single 360° device to obtain an equirectangular video sequence and to apply the proposed tracking strategy on each target simulating several virtual PTZ cameras.

**Keywords:** 360° cameras · Equirectangular projection · PTZ cameras · Object tracking

## 1 Introduction

The recent availability of 360° camera has provided a new type of images and videos and several visualization models. For still images, it is common to use the expanded version of the image (i.e. cylindrical or equirectangular projection) with the whole content directly visible to the user. Navigable players are mainly used for videos, allowing to change the direction of the view, different users can focus on different parts of the 360° video and have a totally different information and experience from the media.

In this paper we use 360° videos for tracking. Following the navigable players paradigm, we don't use the complete 360° information, but we simulate a traditional PTZ camera that can freely move in a 360° world. Using this approach it is possible to reduce the computational complexity of tracking analyzing only a small region of the video frames.

In real-time systems, a PTZ camera can miss the target during tracking. The re-acquisition of the target can be hard and time consuming due to mechanical limitations of PT motors. In 360° video, on the other hand, the target is still present (in a unknown position) and the re-acquisition of the target might be as simple as object detection in images.

Finally, the use of this type of videos can provide practical advantages: reducing the number of cameras needed to cover a space (just one in a small place) and allowing multiple subjects tracking using only one video.

## 2 Related Works

Object tracking has a long history in computer vision. There are a lot of papers and surveys, [12, 18] for example, explaining which are the main aspects to evaluate developing a new approach. Different methods use different information of the observed video, for example the appearance of the target, a model of the target, color information, keypoints detection and so on. 360° tracking is nowadays an open problem and the related works in this field is limited and in fast evolution. In this paper we use 360° video to simulate PTZ virtual cameras. In next paragraphs we report an overview on 360° video capturing and handling, and on classical PTZ tracking techniques.

### 2.1 360° Image Representation

Panoramic images are used to give a single representation of a wide scene, changing the representation coordinate system from the classic planar projection to a different one. Most commonly used coordinate systems are the cylindrical and the spherical ones. They were mainly used in 3D Graphics to reproduce a real environment or in panoramic stitching to create a synthetic wide-horizontal image from a collection of pictures, augmenting the real pan of the camera.

An overview on panoramic imaging, including both hardware and software specific tools, is presented in [8]. In particular, this work describes the capturing, processing, stitching [17] and reconstruction steps for single cameras or pairs of stereo cameras. Nowadays this function is often available on smartphones. 360° cameras can be considered off-the-shelf products. Image stitching [4] is now a less important problem, because these devices can directly provide a 360° image from a small number (two, in the principal commercial devices) of sensors. In our work we give only a brief description of the geometry of a specific panoramic image projection and we assume that the output of the 360° camera is calibrated and with a negligible distortion.

### 2.2 PTZ Tracking

The introduction of PTZ cameras offered the capability of focusing on the interesting part of the scene moving the camera and zooming in, having a higher resolution with respect to static cameras. The main drawback is the control of the PTZ parameters and the camera calibration. The typical surveillance system or simple tracking scenario usually uses a master-slave (static-PTZ) couple of camera [7, 16], or a network of devices [14].

In this paper we want to focus only on tracking approaches that use PTZ or 360° cameras and, therefore, can observe a wide part of the scene. In [6] tracking

is performed using a histogram based on the HSV values of the frame as target feature and the mean shift method to search the target in the next frame. The result of tracking is then used, separately, to estimate when the target is going out of the center of the field of view and which are the correct camera PTZ shifts to keep it inside.

In [3] authors work on the estimation of the PTZ parameters tracking a moving planar target. The dimensions of the object are known and the focus of the paper is in recovering from errors in moving the camera due to incorrect projection and to camera motors. This is done using an extended Kalman filter and estimating the camera state and the object position in a recursive Bayesian filter network.

An alternative method, proposed in [10], is based on adaptive background modelling and the detection of moving objects. Once moving blobs are detected, tracking is performed using color distribution by histogram intersection.

Authors of [13] proposed a virtual framework for PTZ tracking algorithm testing. They collect a large number of views of a specific background and generate a panoramic image. The moving target (foreground) is then inserted in the image and the PTZ camera can virtually move in it. In this case, possible camera motions is limited (left, right, up, down, zoom in and out) and with fixed steps.

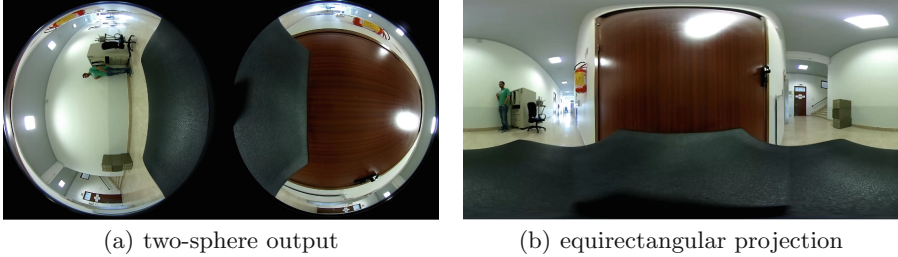
An experimental framework to evaluate different tracking algorithms is presented in [15], consisting mainly in a specific hardware setup and a known ground-truth. Authors uses the following configuration: a PTZ camera is placed in front of a screen where a projector displays the ground-truth video, so the whole system is calibrated to allow the comparison of the tracking result (i.e. the velocities of the motors and the part of the screen seen by the camera) and the real position of the target on the screen. The paper provides the setting and calibration parameters estimated using a Cam-Shift based tracking approach and a Particle Filter one.

A system that combine the use of a face detector and a tracker, based on a single PTZ camera, is shown in [5]. Authors avoided the use of a static camera dividing the work of the PTZ one in two modes: zoom-out mode, that shows the whole scene, and the zoom-in mode, that points on a single face detection using the known faces in the scene and the calculated trajectories. So the camera passes from the wide angle mode to the zoomed mode on a single person using a scheduling algorithm. The authors focused mainly on the face to face and face to person association, on the camera mode handling and scheduling and on the real-time implementation of the system.

### 3 360° Camera Model

There are two principal ways to show the output of a 360° camera in a single video: the two-sphere version and the equirectangular projection. Examples of the former and the latter are shown in Fig. 1. In this case we are considering a 2-sensor camera (a common configuration), but it is possible to use cameras with

a higher number of sensors obtaining several overlapping spheres. The equirectangular version, on the other hand, is always a single image so we decided to use it for the tracking problem.



**Fig. 1.** An example of output of a 360° device: (a) the two-sphere view, (b) the equirectangular projection.

### 3.1 Equirectangular Projection

Given a point  $P = [x, y, z]$  in world coordinates, the equirectangular projection is defined by two angles:  $\theta$  (polar angle), that is the angle defined by the projection of the P vector on the x-z plane and x axis;  $\phi$  (azimuth angle) that is the one formed by the P vector and the x-z plane. Given P, the equirectangular coordinates  $\theta$  and  $\phi$  can be obtained using the following formulas:

$$\theta = \text{acos}(z/d)$$

$$\phi = \text{atan}(y/d)$$

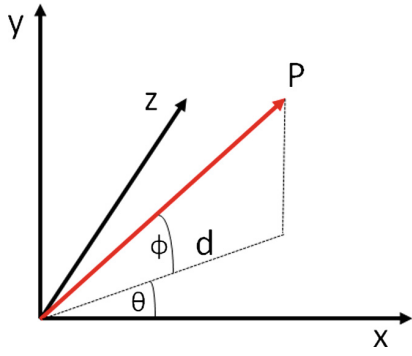
where  $d = \text{sqrt}(x^2 + z^2)$ .

The equirectangular projection basically maps the  $\theta$  and  $\phi$  coordinates obtained by a spherical projection, as shown in Fig. 2(a), in a coordinate plane (the equirectangular projection) where  $\theta$  varies from  $-\pi$  to  $\pi$  and  $\phi$  from  $-\pi/2$  to  $\pi/2$ .

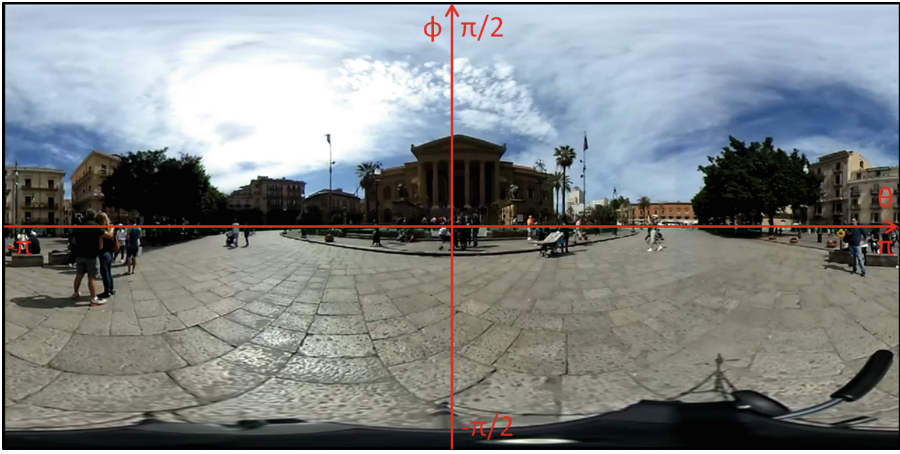
### 3.2 Equirectangular to Virtual Camera Plane Projection

The equirectangular projection is used to have in a single image a complete information of the scene. To simulate the PTZ camera it is necessary to retrieve the virtual output of the camera given the PTZ values. The part of the real world seen in the virtual camera plane according to the PTZ values, can be computed from a rectangular grid of points. The grid is projected in the equirectangular representation and the final camera plane image is computed interpolating the color value of the projected points.

One example of this transformation is shown in Fig. 3, where the virtual camera setting is:  $P = -120^\circ$ ,  $T = 10^\circ$  and  $Z = 90^\circ$ . The projection of the virtual camera plane is the blue grid of Fig. 3(a), these values are then interpolated to obtain the planar surface in Fig. 3(b).



(a) World to spherical coordinates



(b) equirectangular projection

**Fig. 2.** The world coordinates are transformed to spherical ones and then projected in the equirectangular version



(a)



(b)

**Fig. 3.** The blue grid in (a) is the projection of the part of the scene seen by the virtual camera, (b) is the planar reconstruction. (Color figure online)

## 4 Tracking Model

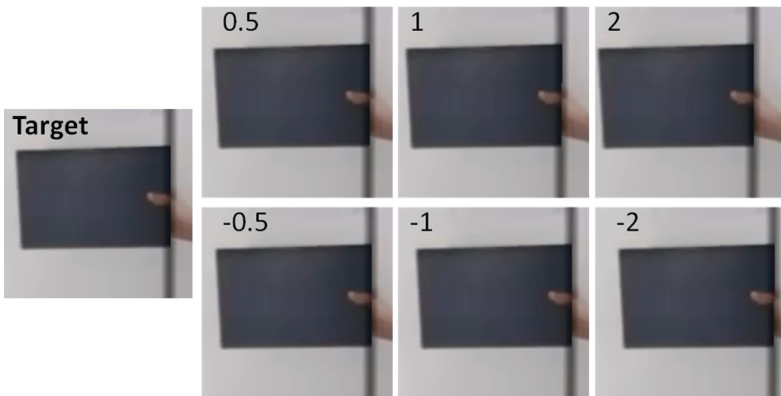
The tracking model takes the equirectangular projection as input, but the actual tracking is performed in the virtual camera plane. The idea is to use the equirectangular image as a world representation and the virtual camera plane as if it was the output of a traditional PTZ camera.

Supposing that the virtual camera parameters Pan and Tilt coincide with the  $\theta$  and  $\phi$  of the equirectangular image and that Zoom coincides with the field of view of the projection, our tracking method estimates, for each frame, how to update the parameters to keep the target in the center of the virtual camera plane.

Our method initially computes the differences between the virtual camera plane with the initial values of PTZ set by hand or by any object detector and the views obtained shifting the virtual camera parameters, one at once, by a fixed value. These images, shown in Fig. 5 are assumed as representative of the moving object relatively to the virtual camera. In this case we move the camera to simulate the object motion, during tracking the object moves and the tracker has to adjust PTZ parameters to keep the object in the center of camera plane.

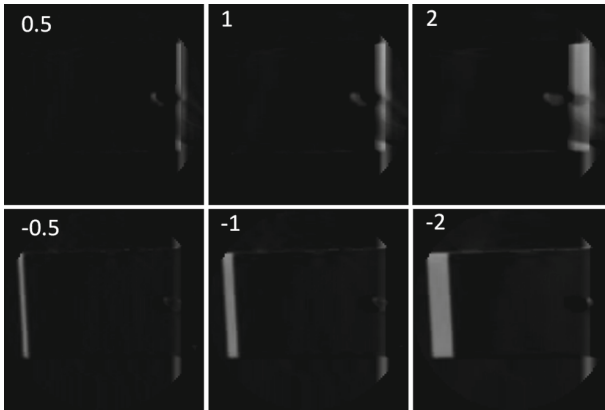
Our method is based on the assumption that the variation in virtual camera plane due to small object motion is approximately equivalent to the variation due to small virtual camera parameters shifts.

Figure 4 contains the original target image and the images obtained with virtual camera P parameter shifted by  $0.5^\circ$ ,  $1^\circ$  and  $2^\circ$  in both directions. Figure 5 contains the difference images. It is possible to notice that small shifts in the virtual camera parameters leads to small values in the difference images (the brighter the value in the figure, the higher is the value of the difference) and vice versa. For this reason the difference associated to small values of parameter shift (0.5) are useful to describe slower movements and the ones associated to higher



**Fig. 4.** The target and the virtual camera planes changing P value by 0.5, 1, 2, -0.5, -1 and -2.

values (2) are representative of faster movements. These difference images can then be considered as a sort of motion templates.



**Fig. 5.** Difference images changing P value by 0.5, 1, 2, -0.5, -1 and -2.

Our method tries to estimate the motion of the object from a weighted sum of basic motion templates. For each frame the motion is estimated selecting the most similar difference image calculated in the off-line step of the algorithm (Fig. 5), or selecting several difference images creating a mixture. If the target is not moving, no difference image is selected.

An example of this is present in Fig. 6: the left image is the target object, the central one is the new frame seen by the camera using the parameters at previous time instant and the right image is the difference. Looking at the latter, the position of the brighter parts (the horizontal and vertical segments in the figure) indicates that the object is moving upwards and slightly to the left.

It is easy to notice that a similar image can be obtained as the mixture of different difference images: a  $0.5^\circ$  in P and a  $1^\circ$ T, representing the motion in horizontal and vertical direction. The actual oblique motion is detected as the weighted mixture of the horizontal and vertical motion. A similar idea has been exploited in [9, 11].



**Fig. 6.** Residual image obtained from the target and the current output of the camera

## 4.1 Tracking Algorithm

The proposed method implements this motion decomposition stating the problem as a least-square minimization one. We define:

- $[P_i, T_i, Z_i]$  as the camera parameters at time  $t_i$ ;
- $I(P, T, Z, t)$  as the virtual camera plane with P,T,Z parameters and at time  $t$ ;
- $s = [\Delta P, \Delta T, \Delta Z]$  as the parameter shifts needed to keep the target in the center of the virtual camera plane;
- $w = [w_P, w_T, w_Z]$  as the vector containing the difference steps (both positive and negative) used for P, T and Z parameter in computing the difference images;
- $A$  as the matrix containing the difference images, represented as column vectors, repeated for different steps in the positive and negative direction.

The tracking algorithm can then be stated as follows:

1. Set  $t_i = t_0$  (the first frame). Set the initial Pan, Tilt, Zoom parameters  $(P_0, T_0, Z_0)$  on the target obtaining  $I_0 = I(P_0, T_0, Z_0, t_0)$  for the first frame;
2. Calculate  $A$  using the differences in  $w$ ;
3. Calculate  $I = I(P_i, T_i, Z_i, t_{i+1})$ ;
4. calculate  $R = I - I_0$ ;
5. solve  $x = (A^T A)^{-1} A^T R$ ;
6.  $s = -xw^T$ ;
7. Set  $[P_{i+1}, T_{i+1}, Z_{i+1}] = [P_i, T_i, Z_i] + s$ ,  $t_i = t_{i+1}$  and go to step 3.

## 5 Results

As discussed in Sect. 2, it is difficult to test the performances of 360° and PTZ tracking for different reasons: the lack of a common accepted benchmark for the former, the use of “real” scenes and actual camera movement for the latter. In particular for PTZ methods, the only way to compare different algorithms is testing them in real scenes and visually looking at the result, having a not repeatable experiment.

To test our method, we initially created a dataset of 59 360° videos using two cameras: a Ricoh Theta S (2 sensors with effective 12 MPixels [1], 26 videos) and a Samsung Gear 360° (2 sensors with effective 15 MPixels [2], 33 videos). The content of video is composed by outdoor (10) and indoor (49) scenes and the targets are in uniform or cluttered background.

The method shows good performance for well-shaped targets (i.e. object with corners) also in cluttered scenes, but the performances are worse when the target speed is too high (i.e. sudden changes of position).

We tested the method with our dataset using six difference images per parameter (three steps in positive and negative direction), so matrix  $A$  contains  $6 \times 3$  difference images. A higher number of steps could lead to a more precise tracking, but with a higher computational complexity and the risk of numerical instability in matrix inversion.





Fig. 7. An example of two simulated PTZ cameras using a single equirectangular video

## 6 Conclusions

The large diffusion of the  $360^\circ$  technology will probably lead to a significant use of  $360^\circ$  videos in next years. This type of cameras could reduce the number of devices needed to cover a large area.

In this paper we showed how to simulate one or several PTZ cameras using a single  $360^\circ$  video and a tracking method based on this idea. The first task is important because  $360^\circ$  videos are very useful to compress in a single (equirectangular) video the informations of the whole  $360^\circ$  scene without significant loss of information. In fact the strong distortion, and consequent loss of information deriving from projection, is mainly in those part (top, bottom) that contains less important information (roof, sky, ground).

Moreover, simulating a PTZ or still cameras from a  $360^\circ$  video allows the use of well-known methods for tracking and video analytics for conventional video.

The contribution of the paper is two-fold: the realization of a simulated PTZ camera from a  $360^\circ$  video whit the explication of the geometric transformations needed to implement it and a simple object tracking method for  $360^\circ$  videos. The first one can be used as a starting point to apply known methods of the state of the art of PTZ or still cameras on  $360^\circ$  videos, the second one can be used to implement a low-complexity tracking method that can run in real time and on low power devices.

The entire proposed framework, in addition, uses a single equirectangular video as input and allows to have a dedicated virtual PTZ camera for each target to track. An example is shown in Fig. 7: on the left there is the equirectangular video, on the right the two simulated PTZ cameras tracking the book and the face simultaneously.

As future work, it is possible to fuse sophisticated tracking methods (i.e. using more informations of motion, keypoint features, recognition and so on) on the virtual camera plane.

This paper is (partially) funded on a research grant by the Italian Ministry of University and Research, namely project NEPTIS (Grant no. PON03PE 002143).

## References

1. Ricoh theta s. <https://theta360.com/en/about/theta/s.html>
2. Samsung gear 360 specifications. <http://www.samsung.com/global/galaxy/gear-360/#!/spec>
3. Al Haj, M., Bagdanov, A.D., Gonzalez, J., Roca, F.X.: Reactive object tracking with a single PTZ camera. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 1690–1693. IEEE (2010)
4. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **74**(1), 59–73 (2007)
5. Cai, Y., Medioni, G.: Demo: persistent people tracking and face capture using a PTZ camera. In: 2013 7th International Conference on Distributed Smart Cameras (ICDSC), pp. 1–3. IEEE (2013)
6. Chang, F., Zhang, G., Wang, X., Chen, Z.: PTZ camera target tracking in large complex scenes. In: 2010 8th World Congress on Intelligent Control and Automation (WCICA), pp. 2914–2918. IEEE (2010)
7. Funahasahi, T., Tominaga, M., Fujiwara, T., Koshimizu, H.: Hierarchical face tracking by using PTZ camera. In: 2004 Proceedings of 6th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 427–432. IEEE (2004)
8. Gledhill, D., Tian, G.Y., Taylor, D., Clarke, D.: Panoramic imaging—a review. *Comput. Graph.* **27**(3), 435–445 (2003)
9. Gleicher, M.: Projective registration with difference decomposition. In: 1997 Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 331–337. IEEE (1997)
10. Kang, S., Paik, J.K., Koschan, A., Abidi, B.R., Abidi, M.A.: Real-time video tracking using PTZ cameras. In: Quality Control by Artificial Vision, pp. 103–111. International Society for Optics and Photonics (2003)
11. La Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(4), 322–336 (2000)
12. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol. (TIST)* **4**(4), 58 (2013)
13. Liu, C., Cao, R., Jia, S., Zhang, Y., Wang, B., Zhao, Q.: The PTZ tracking algorithms evaluation virtual platform system. In: 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), pp. 1–6. IEEE (2014)
14. Micheloni, C., Rinner, B., Foresti, G.L.: Video analysis in pan-tilt-zoom camera networks. *IEEE Sig. Process. Mag.* **27**(5), 78–90 (2010)
15. Salvagnini, P., Cristani, M., Del Bue, A., Murino, V.: An experimental framework for evaluating PTZ tracking algorithms. In: Crowley, J.L., Draper, B.A., Thonnat, M. (eds.) ICVS 2011. LNCS, vol. 6962, pp. 81–90. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-23968-7\\_9](https://doi.org/10.1007/978-3-642-23968-7_9)
16. Stillman, S.T., Tanawongsuwan, R., Essa, I.A.: A system for tracking and recognizing multiple people with multiple camera. Technical report, Georgia Institute of Technology (1998)

17. Szeliski, R., Shum, H.Y.: Creating full view panoramic image mosaics and environment maps. In: Proceedings of 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 251–258. ACM Press/Addison-Wesley Publishing Co. (1997)
18. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. *ACM Comput. Surv. (CSUR)* **38**(4), 13 (2006)