ORIENT-CAM, A Camera That Knows Its Orientation and Some Applications

Bertrand Vandeportaele[1,2], Christophe Dehais[1], Michel Cattoen[2], and Philippe Marthon[1]

[1] Institut de Recherche en Informatique de Toulouse, France
bvdp@enseeiht.fr
http://www.enseeiht.fr/~bvdp/

[2] Laboratoire d'Electronique de l'ENSEEIHT, Toulouse, France

Abstract. We introduce a new type of smart cameras. These cameras have an embedded orientation sensor which provides an estimate of the orientation of the camera. In this paper, we describe our prototype orientation sensor and propose some methods for the calibration of the whole camera. We then show two applications. First, the camera is used to create oriented spherical panoramas. Second, it is used for image based localization, in which only the position of the camera has to be retrieved.

1 Introduction

Many researchers are working on methods providing the camera pose (orientation and translation) from the image. These methods use either some knowledge about the observed scene (for example, a 3D model) or more than one image in order to determine the camera pose and a partial model of the scene at the same time. In the first case, the recovered pose is relative to the 3D model. In the second one, the different poses are generally expressed as a function of the first pose and are defined up to a unknown scale.

In this paper, we propose to use an electronic device fastened to a camera in order to determine its orientation. This absolute orientation is defined in a fixed orthonormal world coordinates system in which two axes correspond to the vertical and magnetic north directions.

Some commercial digital cameras contain an embedded orientation sensor. This sensor is not of the same type that the one described here, as it only allows to determine if the captured image is in portrait or landscape orientation. Our sensor, on the other hand, allows to know in which direction each pixel of the camera is pointing.

In [7], a system combining a camera and inertial sensors is proposed to detect the vertical reference. The authors of [8] proposed a more advanced solution, using both inertial sensors an gyroscopes. The magnetic sensors we use do not deviate in time as gyroscopes do, and we can so expect to obtain the correct absolute orientation whereas they can only expect a relative one.

The paper is organized as follows. First, we describe the camera and the orientation sensor. Second, we propose some methods for the calibration of the

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 267–276, 2006.

complete sensor. Third, we use the camera to create automatically omnidirectional oriented panoramas. Finally, we show how this camera can be used for image based localization and conclude with some perspectives.

2 Description of the Camera

Our aim was to create an handheld and low cost smart camera. The figure 1 shows the resulting prototype whose dimensions are 100*60*22mm and cost is about 150 USD. Both can be greatly reduced if the sensor is mass produced.

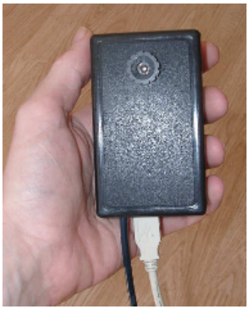


Fig. 1. The complete camera whose thickness is about 2cm

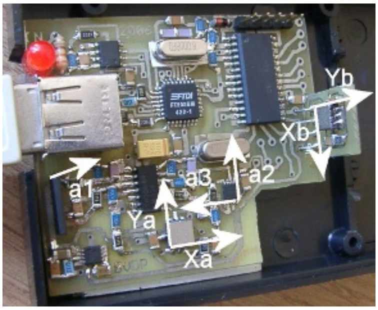


Fig. 2. The orientation sensor and its different axes of sensor. Both a1 and Yb are orthogonal to the main board.

2.1 The Orientation Sensor

The figure 2 shows the orientation sensor prototype we have built and integrated inside the camera. Its dimensions are 45*60*10mm. It is mainly composed of low cost integrated Micro Electro Mechanical Systems (iMEMS), namely accelerometers and magnetometers.

We use the magnetic sensors HONNEYWELL HMC1051Z and HMC1052 in order to have a 3D measurement of the magnetic field (on the axes a1,a2 and a3 shown on the figure 2). We also use two double axis accelerometers MEMSIC MXD3334UL mounted orthogonally to sense the gravity. Each MXD3334UL includes 2 orthogonal accelerometers. A single chip would lead to inaccuracies when one of the two sensor's axis is near the gravity, because each measurement corresponds to the sinus of the angle between the sensor axis and the horizontal plane. When this angle is near 90 degrees, a change of orientation of a few degrees cannot be detected as its sinus changes very few. Using at least 3 sensors allows to always have some sensor axes far away from the gravity axis and hence ensure more accuracy. Finally, the acceleration is sensed on the axes Xa, Ya, Xb and Yb as shown on the figure 2.

A Microchip PIC 18F252 microcontroller gathers the different sensors' measures and send them to the computer using a USB connexion thanks to the

FTDI FT232BM circuit. The complete set of measures is sent at 100Hz allowing a realtime orientation estimation in our computer vision task.

It is noticeable that the sensor provides redundant information as only 3 parameters (the rotations) have to be retrieved from 7 measurements. This allows to compute a more accurate orientation and to detect some incoherences such as parasite accelerations due to movements or magnetic field perturbations. Hence a confidence measure in the retrieved orientation is provided.

2.2 The Camera

The imaging sensor can be either a digital camera or a Webcam. In our prototype, we use the electronic parts of a Logitech QuickCam Pro 4000 at a resolution of 352*288 pixels and at 15 fps. This camera provides the ability to control the shutter speed and gain. Moreover, our model has very low radial distortions, and so can be well approximated by a simple pinhole model.

3 Calibration

3.1 The Orientation Sensor

The data provided by the different sensors have to be corrected in order to determine the orientation. The adequate correction is computed thanks to a calibration procedure that we describe hereafter.

Each magnetometer provide an analogous output. Offsets and gains have to be applied to each channel to ensure that the norm of the magnetic field is approximately constant when the sensor is rotated in any direction. They are estimated by acquiring a sequence of measurements while the sensor is rotating. A cost function whose value is minimal when the variance of the corrected norm is the lowest is then minimized to find the best set of parameters.

Oppositely, the accelerometers needs to be calibrated in static positions to avoid parasite accelerations. The orientation sensor is held in different known orientations and many measurements are done on each channel to compute mean values corresponding to extremal orientations (vertical and horizontal on each axis). The outputs of these circuits are Pulse Width Modulated with a duty cycle between 20 and 70%. The calibration consists in the estimation of a rectification function that provides a duty cycle of 20 or 70% when the sensor is held vertically and of 50% when the sensor is held horizontally.

The next calibration step is optional and only provides better accuracy. It consists in the estimation of the 3D direction of the earth magnetic field at the current location. The field has a different vertical component depending on the latitude on earth. The angle between the magnetic field and the horizontal is estimated by holding the sensor horizontally and deducing the angle directly from the measures given by the sensor.

Once the calibration has been achieved, a first estimate of the orientation is computed using only two accelerometers and a 2D projection of the magnetic field to the estimated horizontal plane. The orientation is then refined by taking into account the data from the 7 sensors. The orientation is parameterized with a quaternion to avoid discontinuity problems and gimbal lock caused by Euler angles parametrization.

Because of the noise in the measurements, we apply a temporal filter to increase the stability and accuracy. Our first idea was to implement a Kalman filter as proposed in [2], but as we have no idea of the movement of the sensor at a given time, we would have modeled it as static. Under this condition, the Kalman filter is equivalent to a standard least squares estimator that we have preferred to implement. Let w defines the size of a temporal window in which the measures of the sensors previously acquired between times $t - w$ and t are used to compute the orientation at instant t. The samples in this windows are weighted differently in order to make the samples which are close in time to the last acquired sample more influent. Increasing (resp. decreasing) w makes the detected orientation more stable (resp. reactive). Values of w between 10 and 20 have provided good results in our applications.

3.2 The Camera and Its Relative Orientation

Once the orientation sensor has been calibrated, it is necessary to calibrate the whole camera, *ie.* estimate its internal parameters and orientation relatively to the orientation sensor.

This is achieved by acquiring pictures I_j of a checkerboard pattern whose one axis is oriented to the magnetic north. We then use the Jean-Yves Bouguet's Complete Camera Calibration Toolbox for Matlab [5] to estimate the pose of the camera for each image (Ri_j, Ti_j) and its intrinsic parameters. Using the orientation given by the orientation sensor for each image Rs_j, we then compute a rotation $R_\triangle$ which minimizes $\sum_{j=1}^{n} (R_\triangle Ri_j - Rs_j)^2$ over the n images. This provides the relative orientation of the camera much more accurately than the measure from only one view. Moreover, in order to avoid bias in the estimation of $R_\triangle$, it is better to sample the complete space of possible orientations.

The final detected standard deviation between $R_\triangle Ri_j$ and Rs_j is about 1.5 degree and the max deviation is about 4 degrees.

4 Oriented Panoramas

4.1 Their Inherent Advantages

The most obvious applications for our camera is the automatic stitching of images to create panoramas. It consists in generating a panoramic image from the different images acquired by the camera without needing to detect some common points in the different images, as Brown and Lowe do in [9]. Our approach has mainly four main advantages, thanks to the orientation sensor.

First, the reconstructed panoramas are oriented, so the horizon of the scene is projected on a straight horizontal line and the first column of the panoramic image can point to the magnetic north for example.

Second, as the orientation of the camera is known a priori, homologous points detection is not needed to stitch the images. This allows to deal with panoramas which contains big portions of non textured areas whose images cannot be automatically stitched by standard methods.

Third, the knowledge of the orientation simplifies greatly the creation of complete spherical panoramas. The problem of closings usually appears when one try to stitch images under a full 360 degrees rotation of the camera. Because of the accumulated errors, it is quite hard to stitch the last image and the first one. This problem can be avoided by using bundle adjustment to estimate the whole solution for the stitching instead of stitching iteratively new images to the panorama as proposed in [1] but this method requires a good initial guess for the whole correspondances.

Finally, our camera allows to deal very efficiently with scenes containing moving object which are outliers in the panorama reconstruction. Methods generally used to compute panoramas of such scenes involve a background/foreground segmentation in order to detect the camera orientation relatively to the background. With our sensor, this is done very simply by detecting zones in the images whose movements are incoherent with the orientation provided by the orientation sensor. The pixels of these zones are labeled as foreground and are discarded in the panorama reconstruction.

4.2 How to Create the Panoramas

We recall that a panorama can be created from images if the viewpoint of the camera does not move (or at least, its displacement should be very small compared with the distance to the observed scene), id. that the images are only related by a rotation of the camera. In this paper, we consider that the intrinsic parameters of the camera do not change between the images.

Using the orientation provided by the orientation sensor, it is possible to project each image of the camera to the right location on the panorama. However, due to imprecisions in the orientations of the different images, the different images do not exactly connect together. We first show how to obtain an approximate panorama in real-time and then propose to use the bundle adjustment method to improve its quality.

The resulting panorama we create is spherical, id. each pixel of the resulting image corresponds to a point on the sphere whose spherical coordinates are related to the pixel position. Any surface could be used to sample the image but the sphere allows an omnidirectional field of view.

Let the point P_i be one 3D position on the sphere corresponding to a pixel i. Let the quaternion corresponding to the orientation of the image j be defined by $Q_j = [abcd]^T$. The j indices are omitted in the quaternion parameters for readability purpose. The corresponding rotation matrix R_j is defined in the equation (1).

Let the camera matrix K be defined in the equation (2), α_u (resp. α_v) being the horizontal (resp. vertical) focal length and $[p_u p_v]^T$ being the position of the principal point. The points $P_i = [x_i\ y_i\ z_i]^T$ of the sphere is projected in the image j to the homogeneous point $p_{Hij} = [u_{ij}\ v_{ij}\ w_{ij}]^T$ whose inhomogeneous coordinates are $p_{ij} = [u_{ij}/w_{ij}\ v_{ij}/wij]^T$ obtained by the function $p_{ij} = D(p_{Hij})$. The image p_{ij} of the point P_i in the image j is obtained as shown in the equation (2).

$$R_j = \begin{bmatrix} 1-2\,b^2-2\,c^2 & 2\,ab-2\,cd & 2\,ac+2\,bd \\ 2\,ab+2\,cd & 1-2\,a^2-2\,c^2 & 2\,bc-2\,ad \\ 2\,ac-2\,bd & 2\,bc+2\,ad & 1-2\,a^2-2\,b^2 \end{bmatrix} \tag{1}$$

$$K = \begin{pmatrix} \alpha_u & 0 & p_u \\ 0 & \alpha_v & p_v \\ 0 & 0 & 1 \end{pmatrix} ; p_{ij} = D(KR_jP_i) \tag{2}$$

If p_{ij} lies inside the image boundaries and if the z component of R_jP_i is positive (id. the camera is facing the point), then the point P_i is viewable on the image j and the RGB values of p_{ij} are used in the panorama for the pixel i.

This computation is achieved for the whole sphere at each frame in order to update the panorama. The figure 3 shows an image from the camera and the reconstructed low resolution (1024*512 pixels) oriented panorama obtained in real time from a sequence of approximately 10 seconds. No post processing is applied; the panorama only results from the projections of the different images to the sphere. The left column of the reconstructed panorama points to the north and the horizon of the scene corresponds to an horizontal line in the panorama. Note the narrow field of view of the camera. The scene contains large un-textured areas on the roof, walls and a over exposed area at the window. The images of these parts would be impossible to stitch automatically as they do not contain any discriminant points.

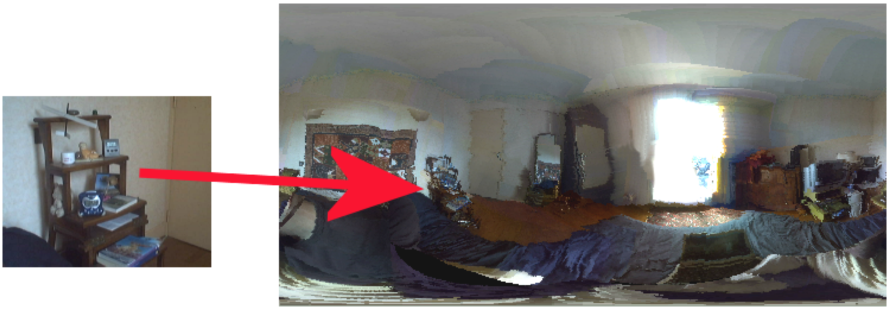


Fig. 3. Left: An image from the camera. Right: The reconstructed oriented panorama.

4.3 Refinement of the Panoramas

The bundle adjustment is used to improve the estimation of the different images orientations. This has already been proposed in [1] but they were not using the a priori knowledge provided by the orientation sensor. The reader can find more information about the bundle adjustment techniques in [6]. This problem can be stated as an iterative minimization of a cost function f defined in equation (3). A detected projection of the point i in the image j is denoted $\widetilde{p_{ij}} = [\widetilde{u_{ij}}\ \widetilde{v_{ij}}]^T$ and is called an observation. Each observation of a point projection provides 2 residuals (res_u and res_v).

$$f = \sum_{j=1}^{n}(\sum_{i=1}^{m}(||D(KR_jP_i) - [\widetilde{u_{ij}}\ \widetilde{v_{ij}}]^T||^2)) = \sum_{j=1}^{n}(\sum_{i=1}^{m}(res_u^2 + res_v^2)) \qquad (3)$$

Not all the images (the camera is acquiring images at 15fps) are needed to refine the panorama, so the set of images have to be decimated in order to avoid useless computations. The decimation is very simple and consists in selecting images such that they have at least half their surface in common with the other images. The approximate orientation from the sensor is accurate enough to compute the set of images to keep.

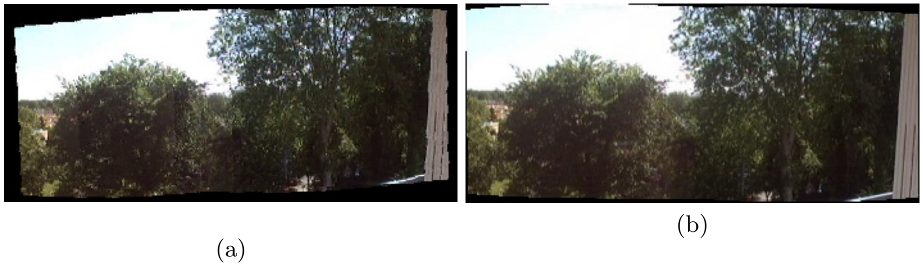
The computations cannot be achieved on every pixels of the retained images and so some of them have to be selected. We use the famous Harris corner detector [3] to detect the projections of corners. Thanks to the knowledge of the approximate orientation of each image, there is no need to use descriptors to detect correspondences between the points in the different images. Instead of this, we simply project the detected Harris points of the different images to the sphere in order to detect the correspondences. A correspondence is established if some projected points from different images are closer than a threshold and if there is no other Harris points in the same images too close in their neighborhood.

Once the correspondences have been detected, the cost function is iteratively minimized. This way, we refine the intrinsic parameters, the quaternions Q_j and the P_i. The convergence is very fast in general and the computation can be achieved very efficiently thanks to the sparse structure of the jacobian. As the accuracy of the orientation sensor is known, we define some limits on the possible values for the quaternion parameters during the iterations. If a quaternion exceed these limits, we take it back to the limit. This ensure that the orientations of the different images cannot deviate too much during the bundle adjustment.

Some images contain no Harris point at all. These images will not have any observation and will not be related in the jacobian matrix. So their orientation will not change. Hopefully, the images which do not contains Harris point are likely to be untextured, so small misalignments due to the orientation sensor inaccuracies are not usually noticeable.

Finally, a post processing is applied in order to smooth the discontinuities at the image boundaries.

The figure (4) shows the results of the bundle adjustment and post processing on a part of the panorama. Only five iterations where necessary to adjust the



(a)

(b)

Fig. 4. (a) A part of the panorama from 4 images before the bundle adjustment and post processing. (b) The same part after refinement and post processing.

whole panorama . Only four images are rendered here in order to see the details. Note that the discontinuities are attenuated in the refined image.

We have not deal with radial distortions because our camera does not require it. However it can be included very easily in the model, both in the calibration and bundle adjustment steps.

5 Image Based Localization

Our camera can also be used as an image based substitute for the GPS inside buildings, where the satellite signals cannot be received. In our previous works, we were using catadioptric camera in order to achieve this task. However, these cameras are quite expensive and fragile. Their size do not really allows to use the term "handheld" and special care must be taken in order to avoid scratches and dust that reduce the image quality. Finally, their resolution is lower than the one available from stitched images and the exposure has to be set for the whole scene, resulting in over or under exposed areas. Their main advantage is their ability to acquire more than an hemisphere of the scene in only one shot. Another advantage is that they allow to retrieve the orientation relatively to the horizon using the projection of vertical lines which are numerous inside buildings. However, the orientation to the magnetic north obviously cannot be determined automatically from the image.

In this section, we propose a very simple system to illustrate the ability of our camera. A database B of images I_P of the building in which we want to localize our camera (from its image I_C) is acquired using a catadioptric camera with a parabolic mirror. The acquired area is sampled at a given altitude and every 30cm in an horizontal plane. The paracatadioptric camera is held horizontally and with the columns of the image sensor pointing to the north (using a compass) so the database is at least approximately oriented.

The localization of I_C is achieved after computing a rectified image I_R (which is of the same type than I_P) and finding the closest image I_P in B. Instead of the sphere (as in the section 4), the surface of the paracatadioptric image is sampled. This way I_C is projected to a portion of a paracatadioptric image I_R and can be

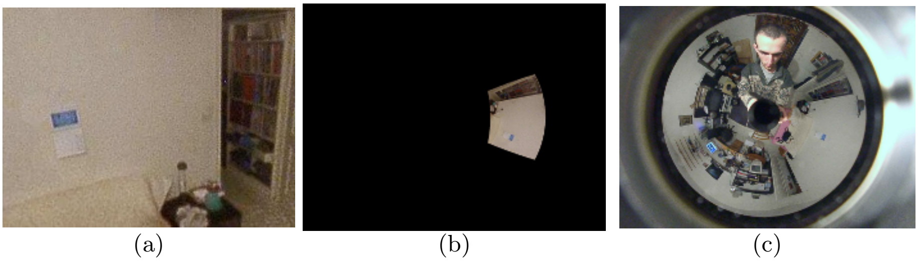

(a) (b) (c)

Fig. 5. (a) The image I_C from the camera to be localized. (b) The projection I_R of the image to the surface of the paracatadioptric mirror. (c) The closest panoramic image I_P found in the database.

matched with images from B. Thanks to the orientation knowledge, I_R should superpose to a I_P acquired from the same location.A color rectification is also applied in order to make the images matchable.

In order to avoid costly comparisons of I_R with all the I_P, a first detection is achieved using corners detected by the Harris corner detector. The database is processed offline. Local descriptors are computed to describe every corner points. A famous descriptor widely used in the computer vision community is the Scale Invariant Feature Transform (SIFT) from David Lowe [4]. It provides very good results for matching image points related by affine transformations. However, thanks to the orientation sensor, such a complex descriptor is useless, as the points are observed in similar orientations and from very close positions (less than 20cm). In this experiment, we only use square windows of pixels to describe the points. These descriptors are simpler to compute and are also more discriminant, as they do not give good matching scores to couples which are related by affine transformation.

When an image I_C has to be localized, the Harris points H_C are detected in I_C using a variable size for the gaussian filter in order to deal with the varying resolution of the paracatadioptric camera. I_C and H_C are then projected to the paraboloid and the descriptors corresponding to the projected H_C are computed in I_R.

In order to match the images efficiently, we use a decomposition of the images in annular portions of sectors. The potential Harris points correspondences between I_R and the whole set of I_P are only compared if the points lie in the same or one of the 8 neighbor portions. The computed score is a ZNCC measure between the descriptors, which allows to deal with light variations. Harris points which do not have any correspondence get the worst score. The partial correlation score computed for an image I_P is the sum of the scores of the different points.

A final verification step is applied in order to check the correspondences which have the highest scores, and to decide between different solutions if needed (when many correspondences have nearly the same scores). This step consists in trying to match every pixels of I_R with one pixel in I_P. The ZNCC score is computed for each pixel of the I_R with pixels from I_P in a centered window neighborhood. The lowest score is kept for each pixel, allowing small displacement in the images. The resulting score for each image correspondence is the sum of the pixels' scores.

During our preliminary experiments, we have observed that this simple localization is robust to occlusions up to 30% of the surface of the image. When we integrate many images I_R to create a paracatadioptric panorama from the current location, the robustness of the localization is greatly increased. Obviously, the localization fails if the camera is pointed only to a region which is not discriminant, such as an uniform wall for example.

The figure 5 shows an example of the method. Less than 5 seconds are necessary to retrieve the correct I_P from a database of more than 200 images using a pentium 4 computer. The panoramic images were divided in 10 angular sectors, each one divided in 4 portions.

6 Conclusions and Perspectives

As far as we know, this paper presents the first handheld camera containing a complete absolute orientation sensor. We have shown that this camera is compact, low cost, practical and quite easy to calibrate. This sensor could be improved to obtain more accurate measures but it is sufficient for the proposed applications. We believe that this kind of camera have many other applications and should interest many researchers. Future work will focus on improving the image based localization system in order to avoid the dense sampling of panoramic images while acquiring the database. We are trying to use either a reconstructed 3D model of the scene or a sparse set of panoramic images.

References

1. Philip F.Mc Lauchlanand, Allan Jaenicke: Image Mosaicing using Sequential Bundle Adjustment, BMVC 2000.
2. E.H. Shin: A Quaternion-based unscented Kalman filter for the integration of GPS and MEMS, ION GNSS 2004.
3. C. Harris and M.J. Stephens: A combined corner and edge detector, In Alvey Vision Conference, pages 147-152, 1988.
4. David G. Lowe: Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.
5. Jean-Yves Bouguet, Complete Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/
6. C. Slama: Manual of Photogrammetry, American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.
7. Jorge Lobo, Jorge Dias:Vision and Inertial Sensor Cooperation Using Gravity as a Vertical Reference, PAMI, December 2003 (Vol. 25, No. 12), pp. 1597-1608.
8. Thomas Schon and Fredrik Gustafsson: Integrated Navigation of Cameras for Augmented Reality, IFAC 2005.
9. M. Brown, D. G. Lowe: Recognising Panoramas, In Proceedings of the 9th International Conference on Computer Vision (ICCV2003).