Omni-Probe: Mosaicing the Interior of Tubular Shapes

Luís Ruivo, José Gaspar, José Santos Victor
Instituto Superior Técnico, Instituto de Sistemas e Robótica, Lisboa, Portugal

September 29, 2007

Abstract

In this work we approach the problem of imaging the complete interior surface of a tubular shape with one single image. This involves dewarping and combining (mosaicing) images acquired by a moving camera.

Our framework for mosaicing the interior of a tubular shape involves acquiring images, finding and reconstructing corresponding feature points, fitting a simple 3D model to the reconstructed points, estimating the camera path and dewarping to one single mosaic image. This work focus on how to reconstruct the scene, fit the scene into the simple 3D model and build the mosaic image.

We use a 3D model that is based on cylindrical sections, and is useful both for generating simulated data and implementing the fitting procedure. This 3D model was also constructed to make easier the dewarping to the mosaic image.

keywords: *Mosaicing tubular shapes, eight-point Algorithm, simple 3D model fitting.*

1 Introduction

Mosaicing the interior of tubular shapes consists in combining multiple images into a single one (mosaic) representing all the interior. Mosaicing finds applications in e.g. simplifying the inspection of pipelines - one single mosaic image completely covering the interior of a pipeline is definitely much faster to read than one (eventually long) video.

In the case of tubular shapes perfectly cylindrical (no curves), watched by cameras perfectly positioned and aligned with the cylinder axis, mosaicing the interior would be just a polar to cartesian dewarping followed by the registration of corresponding image features. Registration would than be just computing an homography [2] between each pair of images.

When the cameras are not positioned or aligned with the cylinder axis homography and corresponded image features do not do the work for them selves. This is the case of this work. The camera travels freely inside the tubular shape and therefore the images have very different perspectives.

Correct image dewarping implies knowing the camera motion between consecutive images, or in other words the path of the camera inside the tubular shape. Using only corresponded image features this motion can be found by estimating and factorizing the essential matrix [1]. Given the estimated camera path we can reconstruct the 3D tubular shape. Than, the dewarping procedure consists just in opening the 3D shape to a mosaic image.

The dewarping procedure has some complications because the loss of one coordinate will impose some decisions on how to represent on a single image all the interior surface. The idea is to open the tubular shape by the top and rollover the surfaces to a 2D plane that will be the 2D image mosaic. But if the tubular shape can change his radius, go up or down, left or right, how we can show that information on the mosaic image? The purpose of this work is to make fast reports of tubular shapes. So what we do is to represent all the surface in a image that can be placed on a document.

Opening the tubular shape to the mosaic image is not simple if the 3D points don't have a pre-adjustment. What we do is fitting the 3D points to the various sections of the tubular shape, constructing

a simple 3D model that will make the dewarping much easier. This 3D model organizes all the tubular shape structure as a list of circles representing tubular sections. This list has 3 parameters: 3D center point of the circle, the scalar radius and the 3D direction vector.

The mosaic image shows all the internal surface of the tubular shape and is based on simple perspective images taken along the structure by a standard camera.

This work is organized as follows: after a small review of related work we refer in section 3 how to reconstruct a scenario; in section 4 we propose a geometric model for a tubular shape, detail how to simulate a sequence of pictures taken inside that tube and how to fit the tubular model to real 3D points data; finally on section 5 the tube is opened and then conclusions are made.

2 Related work

Scene reconstruction is a well known area in computer vision. Reconstructing a scene traditionally starts by selecting in a video-sequence some features that can be corresponded in a robust manner, e.g. the SIFT features [5]. The motion of the camera between consecutive images is than determined for instance by estimating and factorizing the essential matrix [2], assuming that the camera is calibrated.

Currently, scene reconstruction is further improved with SLAM *(Simultaneous localization and Mapping)* or vSLAM *(Visual SLAM)* processes by assuming a smooth camera motion model. In essence, the SLAM and vSLAM algorithms use an autonomous vehicle that starts at an unknown location and then incrementally build a map of the environment while simultaneously uses this map to compute absolute vehicle location [3].

Our work involves reconstructing the scene by one of the previous methods. In particular we present reconstruction results based on the estimation and the factorization of the essential matrix. Given the reconstructed scene, i.e. a cloud of 3D points, we want to fit a simple 3D model to it. The 3D model has to be established in a manner that simplifies the dewarping of the images into a mosaic.

3 Camera Motion and Scene Reconstruction

With corresponding points in stereo matching and the intrinsic parameters of the camera it is possible to reconstruct the scene. With epipolar geometry and the eight-point algorithm from Longuet-Higgins [4], the extrinsic parameters R and t can be determinated.

The fundamental matrix is a basic tool in the analysis of scenes taken with two uncalibrated cameras, and the eight-point algorithm is a frequently cited method for computing the fundamental matrix from a set of eight or more point matches. It has the advantage of simplicity of implementation and with a very simple normalization (translation and scaling) of the coordinates of the matched points gives good estimations of the fundamental matrix.

3.1 Essential Matrix Estimation

The epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose [2]. The fundamental matrix F encapsulates this intrinsic geometry. It is a (3×3) matrix of rank 2. If a point in 3D space p is imaged as x_1 in the first view, and x_2 in the second, then the image points satisfy the relation:

$$x_2^T F x_1 = 0 \tag{1}$$

The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with a point having the baseline as axis (the baseline is the line joining the camera centers), Fig.1. This geometry is usually motivated by considering the search for corresponding points in stereo matching.

To reconstruct the scene it is used the essential matrix which is the specialization of the fundamental matrix to the case of normalized image coordinates. Historically, the essential matrix was introduced

(by Longuet-Higgins) before the fundamental matrix, and the fundamental matrix may be thought as the generalization of the essential matrix in which the (inessential) assumption of calibrated cameras is removed.

So to have the scene and the cameras path we need the camera extrinsic paramaters rotation and translation present on the essential matrix, $E = [R, t]$.

The essential matrix is related to fundamental matrix by $E = K_2^T F K_1$ where K is the intrinsic parameters. So E is equally defined with

$$\hat{x}_2^T E \hat{x}_1 = 0 \tag{2}$$

where the points x are normalized to remove the intrinsic parameters on the coordinates $\hat{x}_i = K_i x_i$.

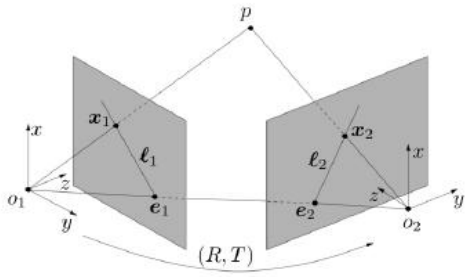


Figure 1: Two views x_1,x_2 of 3D-point p. Euclidean transformation between two images given by (R, T). Epipoles e_1,e_2 are the points intersection of the line (o_1, o_2) with each image. The interception of the plane (o_1, o_2, p) with the two images planes are the epipolar lines l_1, l_2.

Given a pair of images, to each point x_1 in one image, there exists a corresponding epipolar line l_2 in the other image. Any point x_2 in the second image matching the point x_1 must lie on the epipolar line l_1.

The great advantage of the eight-point algorithm is that it is linear, hence fast and easily implemented. If eight point matches are known, then the solution of a set of linear equations is involved. With more than eight points, a linear least squares minimization problem must be solved. The essential matrix is defined by the eq.2 and using at least eight matched points $(x_1, y_1, 1) \rightarrow (x_2, y_2, 1)$ it is possible to compute the matrix E. The coefficients of eq.4 are easily written in terms of the known coordinates x_1,x_2:

$$A_{n\times 9} = \begin{bmatrix} x_1x_2 & x_1y_2 & x_1 & y_1x_2 & y_1y_2 & y_1 & x_2 & y_2 & 1 \end{bmatrix}_n \tag{3}$$

From all the point matches, we obtain a set of linear equations of the form $Ae = 0$ where e is a nine-vector containing the entries of the matrix E, and A is the equation matrix. The essential matrix E is a reshape of the solution vector e and e is defined only up to an unknown scale. For this reason, and to avoid the trivial solution e, we make the additional constraint $||e|| = 1$, resulting in the following system:

$$\begin{cases} min_E \sum_{j=1}^{n} (x_{2_j}^T E x_{1_j})^2 \\ \|E\|_E = 1 \end{cases} \tag{4}$$

The essential matrix $E = [R, t]$ has only five degrees of freedom: both the rotation matrix R and the translation t have three degrees of freedom, but there is an overall scale ambiguity – like the fundamental matrix, the essential matrix is a homogeneous quantity.

The essential matrix is also a (3×3) matrix *but* with two of its singular values has to be equal and the third zero. This is deduced from the decomposition of $E = RS$, where S is skew-symmetric.

3.2 Factorization

To reconstruct the path of the camera is necessary to factorize E to find the rotation R and the translations t. Using

$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

(notice that W is orthogonal and Z is skew-symmetric) and a SVD decomposition of E, the paramaters R and t can be computed as follows:

$$E = UDV^T \quad where \quad D = diag(1, 1, 0) \tag{6}$$

$$R = UWV^T \qquad or \qquad R = UW^TV^T \tag{7}$$

$$and \qquad S = VZV^T \tag{8}$$

$$S = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{9}$$

$$t = (t_x, t_y, t_z) \qquad or \qquad t = U\,(:, 3) \tag{10}$$

A last check is necessary to have the essential matrix estimated: the R and t have norm=1, so a correspondent point is reconstructed in both cameras ($P1 = [I, 0], P2 = E$) and the both cameras have to see it.

After the essential matrix factorization the 3D-scene can be reconstructed applying the back-projection to the image points.

At start it was mention a "very simple normalization (translation and scaling) of the matched points gives good estimation of the fundamental matrix". It is a normalization of the absolute values of the matrix A eq 3. Instead of the image coordinates $(1, 1)$ start at top left they start at the center point, and the average norm of all points is rescaled to be $\sqrt{2}$. This makes matrix A more balanced and noise robust and therefor making a better estimation of matrix E

3.3 Results

A simulated camera was placed traveling inside the tube shown in figure 4-bottom. It travels on the axis of the tube and is always facing the positive Oz direction. Therefore the extrinsic parameters only have a translation component. For this reason, in some images there is no complete view of the tubular section (see figure 2-right). The camera has 44 degrees field of view and a fixed 640x480 pixel resolution.

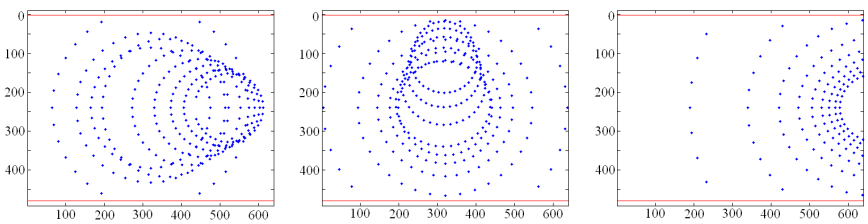


Figure 2: Three images of the simulated camera inside the tube

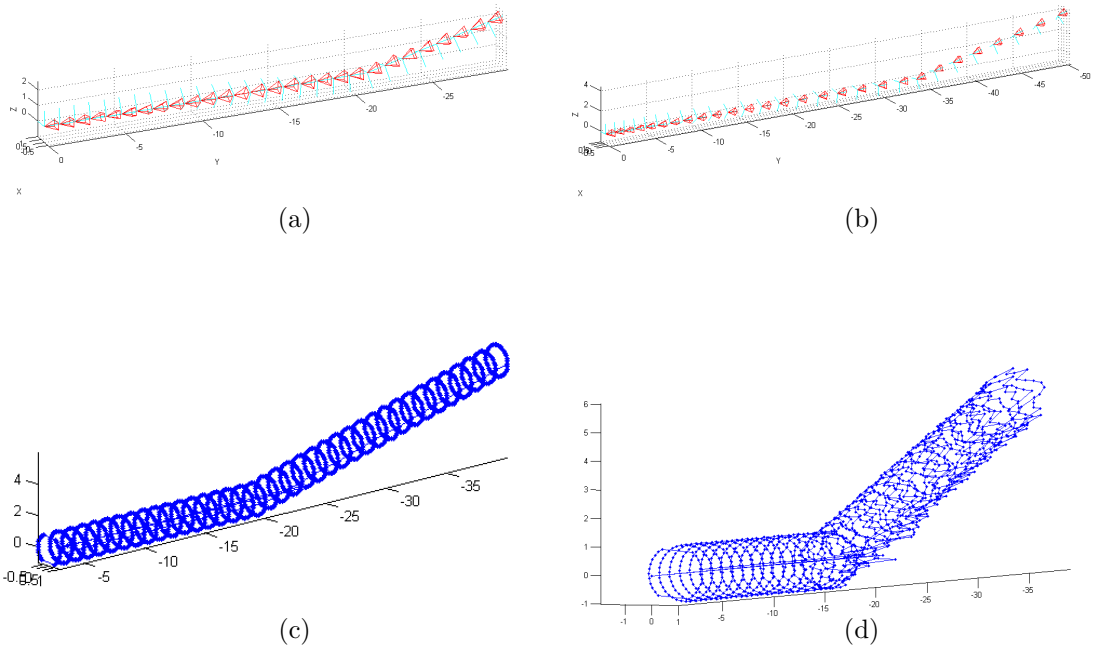


Figure 3: Estimated camera motion without (a) and with noise (b), reconstructed scene points without (c) and with noise (d).

The next images show the scenario reconstruction of a tubular shape with and without noise. Is also estimated the camera motion.

The figure 3(a) shows the camera motion. In figure 3(b) is visible the same camera motion but with noisy correspondent points. Along the path the accumulation of the noise is making a worst estimation of the camera location. The input noise is ten percent of pixel dimension. The figure 3(c) shows the reconstruction of the scenario using only the two first cameras. Without noise the reconstruction is perfect but with noise 3(d) there is a growing bad reconstruction along the tube. This is espected because the relevance of the noise is much bigger at the end of the tube than to the starting of the tube.

4 Tubular Shape Modeling

With the reconstruction of the scenario is important to organize all the 3D data in order to do an efficient dewarping. This organization is made by fitting a simple 3D model that reduces the 3D points into a list of circles representing the tubular shape in conical sections.

4.1 Tubular Shape Model

We define the tubular shape model as a set of circular sections. Each circular section is defined by its center point, m_i, its radius, r_i, and the normal vector to the circle plane, v_i (see figure 4-top). This model allows representing tubular shapes with straight and curved segments, and variable section-diameters.

The tube model can be used both for creating a virtual world to obtain simulated navigation images and for fitting (representing) real data. Simulating navigation images involves specifying the camera intrinsic parameters and motion. Fitting real data is described in the next section.

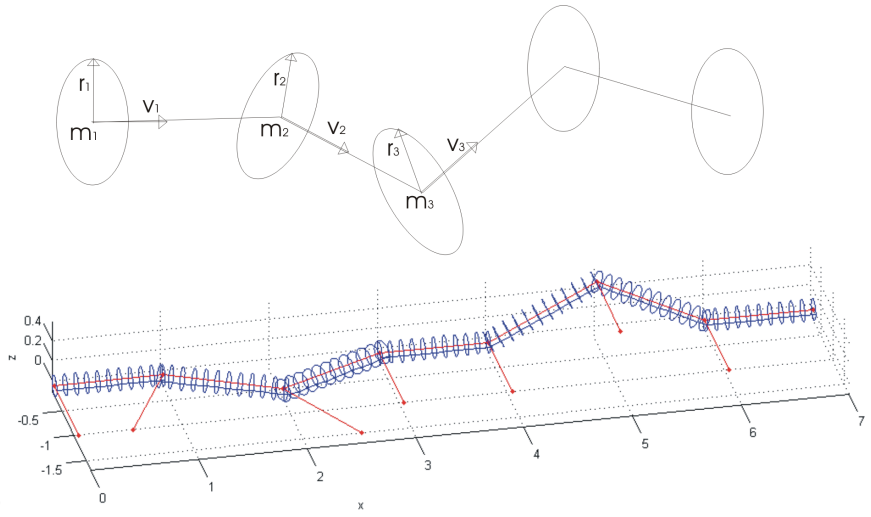


Figure 4: *(Top)* Tubular shape model. *(Bottom)* Example of a tubular shape.

4.2 3D model fitting

Given 3D points representing a tubular shape, we fit to that data the simple model defined in the previous section. This fitting process comprises three steps: (i) searching the best fitting cylinders to each of the 3D clouds-of-points (reconstructed between each pair of pair of consecutive images) (ii) removing the overlapping between consecutive cylinders (iii) defining a continuous path along the cylinder sections, i.e. matching the ending-face of each cylinder section with the starting-face of the next cylinder section. See figure 5.

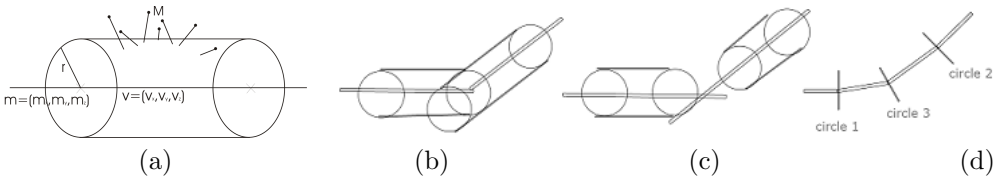


Figure 5: Fitting cylinders: (a) fit one cylinder to a cloud of points (b) remove the overlapping between cylinders (c) refit cylinders to make the central path continuous (d) circles separating the cylinder sections.

Figure 5a represents how a cylindrical shape is fitted to a cloud of points. This fitting corresponds to finding the cylinder yielding the smallest average distance to the observed 3D points. Letting $\{M_i\}$ be the 3D reconstructed points, (m, v, r) the center point of the basis, the axis direction and the radius of the cylinder, then the optimization problem is:

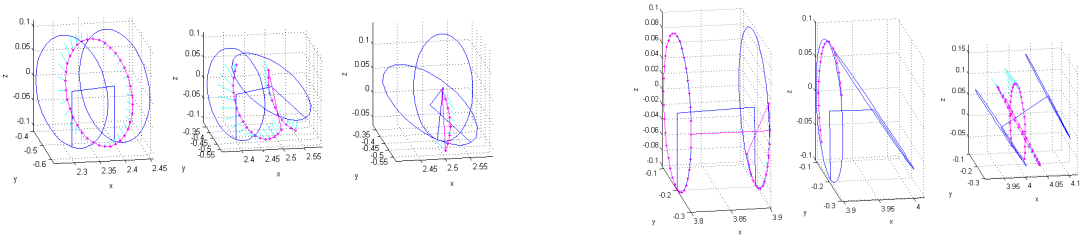
$$\hat{\theta} = \arg_\theta \min \sum_i \left(\|(M_i - m) - proj_v\,(M_i - m)\| - r \right)^2 \tag{11}$$

where θ contains just only a minimal set of degrees of freedom of (m, v, r), namely $\theta = [m_x\ m_y\ v_x\ v_y\ r]$.

Removing the overlapping between consecutive cylinders is just a truncation of the length of each cylinder such that the truncation-plane is *before* the next cylinder (see figure 5c). The set of non-overlapping cylinders usually does not have a continuous path linking their axis. We enforce this by defining circles separating the cylinder surfaces (see figure 5d). Note that as we allow these circles to

have free orientations and radius, we allow for more general shapes (as e.g. sections of cones) instead of just cylinders.

4.3 Results



(a) (b)

Figure 6: Fitting cylinders: (a) (b) Optimizing the connections between sections. *Blue* is the reconstructed tube/circles, *Magenta* is original 3D image points, *Light Blue* is the distance between the original 3D points and the tubular surface.

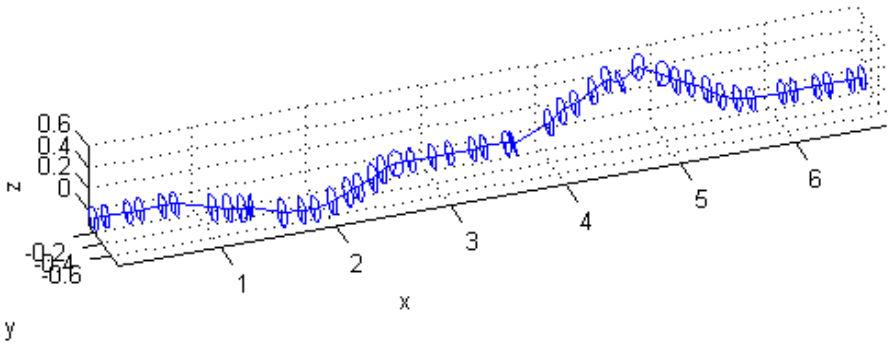


Figure 7: Reconstructed tubular shape.

The figure 6 represents refitting the tube sections after truncating the cylinders to be non-overlapping. This optimization process adjusts the circles defining the tube sections. By maintaining fixed the first and the fourth circle, the two in the middle are readjusted to keep a continuous/smooth tube. The next iteration maintains the second and the fifth, and the third circle is again readjusted.

The figure 7 represents the reconstructed tubular shape. The structure is close to the original despite the local removal of sections overlapping and refitting procedures.

5 Tubular Shape Opening

The tube opening is a dewarp procedure that transforms the 3D minimal tube fitting to a mosaic image. It receives the list of circles and builts the interior shape image.

5.1 General Dewarping

This dewarp is quite simple because is a transformation of polar coordinates to cartesian coordinates, or in other words, the circles of the tube fitting are opened into lines that will fill the columns of the rectangular mosaic image.

Linking each circle is defined the ground line. The ground line is the representation of the tube floor and it is the minimal 3D point on the vertical axis. This ground line is the middle line of the mosaic and the perpendicular lines (columns) are the dewarped circles.

With the reduction of one dimension some information is lost. This image gives a fast idea of the tubular interior surface but changes is proportions. Curves and radius are hard to notice on this image.

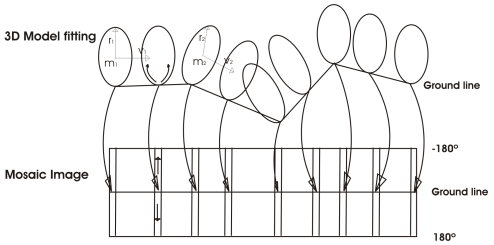


Figure 8: Dewarping 3D tube fitting to mosaic

5.2 Results

A simple case of this opening is when the tube is perfectly cylindrical and the camera inside is perfectly positioned and aligned with the cylinder axis. This case was simulated and the results can be seen on the next subsection. The perspective images are placed on the 3D model but some overlapping of the images occurs. To remove the overlapping we use homography.

The estimation of the homography matrix needs at least four correspondent not collinear points. Each par of points constructs two equations eq.12 and eq.13 of the equations matrix $Ah = b$:

$$x_1 h_{11} + y_1 h_{12} + h_{13} - x_2 x_1 h_{31} - x_2 y_1 h_{32} = x_2 \tag{12}$$

$$x_1 h_{21} + y_1 h_{22} + h_{23} - y_2 x_1 h_{31} - y_2 y_1 h_{32} = y_2 \tag{13}$$

Notice that the homography matrix have the entry h_{33} fixed at 1 because the number of equations are eight and not nine.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{14}$$

This perfect simulated tube was built with a picture of a calendary on the surface.

In Fig. 9-left is shown the perspective images taken inside the tube. In Fig. 9-right is shown the panoramic overlapping images which will be included properly on the final image with the help of homography. The Fig. 10 is the final image were the interior of the tube can be seen in a 2D mode, in this case a calendary.

The perspective images have 640x480 pixels resolution.

6 Conclusion

The Mosaicing is a way to enlarge the information in one single image. The perspective images are very easy and cheap to get comparing to the panoramic images taken with catadrioptic or angular lens, so this work lies only in computer image processing. Simple Mosaicing is very restricting to the type of

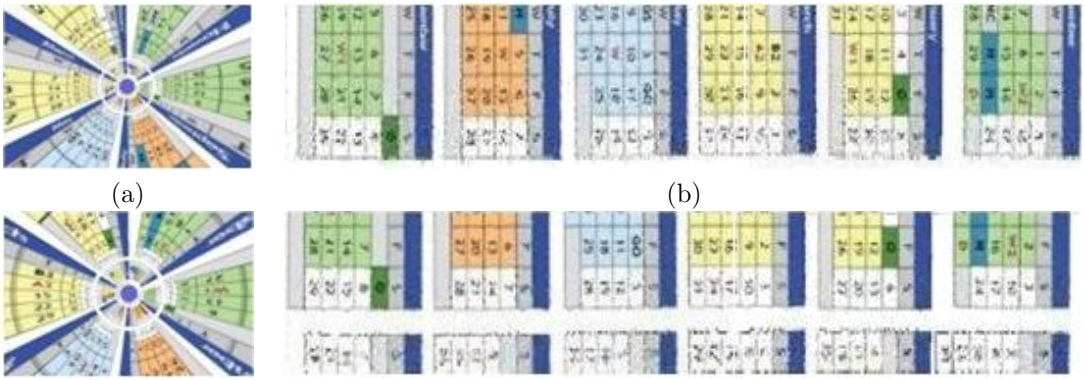


Figure 9: Images captured by the camera (a,c). Dewarped images, (b,d). Notice the overlapping in the dewarped images.

images that ca be processed. So the idea as to reconstruct the scenario and instruct the dewarping to build panoramic images that are sensitive to the localization of the camera inside the tube. But that requires a well thought algorithm to construct panoramic images that will respect the geometry of the tube based on the pre disposal of camera.

Instead of using homography the mosaic image is built based on the reconstructed scenario and a dewarping of a simplified 3D model of the tubular shape to 2D coordinates.

The proposed 3D model based on circles defining tubular sections, is a convenient representation as it allows optimizing locally the fitting process. It is also a useful representation for dewarping as it allows defining a *ground line* as the intersection of the circles with a vertical plane. The line will than be the center line of the mosaic image and the circles are broken at the top and rollover to a 2D plane that will be the 2D image mosaic.

References

[1] R. Hartley. In defense of the eight-point algorithm. *IEEE-T. PAMI*, 1997.

[2] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2000.

[3] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. Munich. The vslam algorithm for robust localization and mapping. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 24 – 29, 2005.

[4] Longuet-Higgins. A computer algorithm for reconstructing from two projections. *Nature*.

[5] David Lowe. Distinctive images features from scale-invariant keypoints. *International Journal of computer Vision*, 2004.

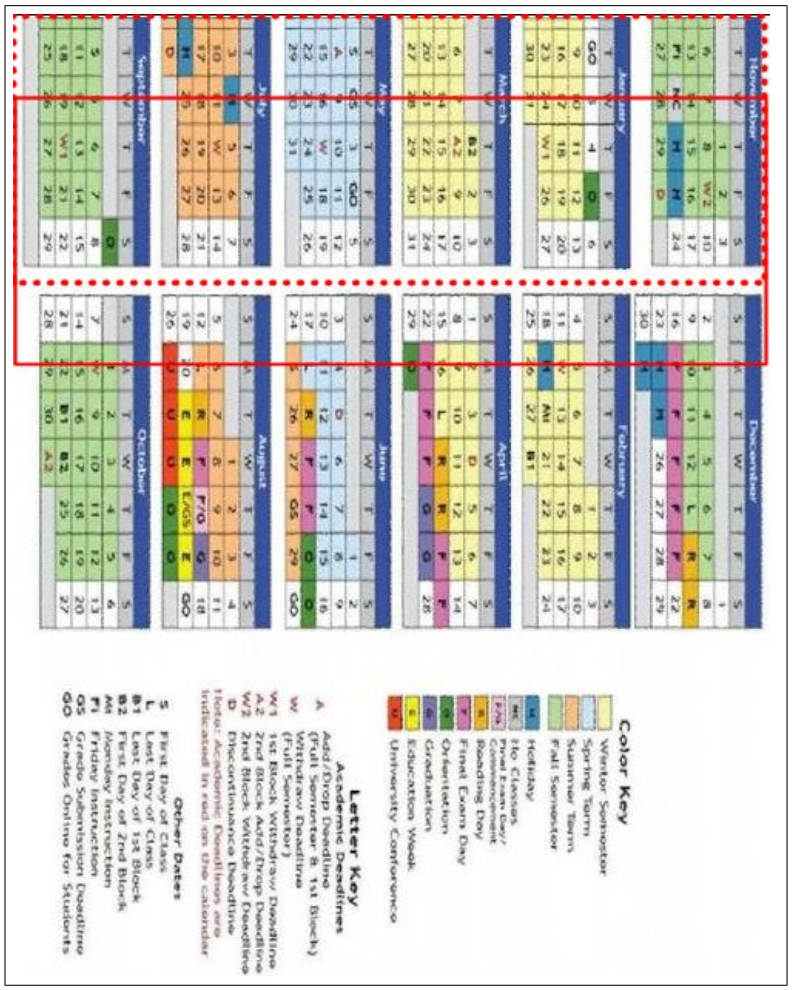


Figure 10: Final Mosaic Image. The red dotted- and solid-rectangular frames show the locations of the dewarped images shown in Fig.9(b,d), respectively.