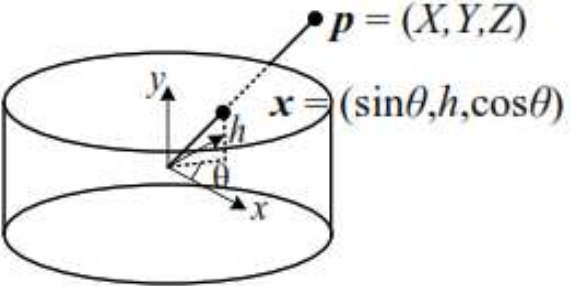


ANSWER KEY SUBMISSION

Date of Exam & Session	17/11/2022 & FN	Category of Exam	CLA3
Course Name	COMPUTER VISION	Course Code	18CSE390T
Name of the Faculty submitting	Dr.C.G.Balaji	Date of submission of Answer Key	19/11/2022
Set	ODD	Total Marks	50

	Part-A (1 x 10 = 10)	Marks
1.	B	1
2.	C	1
3.	A	1
4.	B	1
5.	A	1
6.	B	1
7.	C	1
8.	A	1
9.	D	1
10.	A	1
	Part-B (4 x 4 = 16)	
11.	<p>What is the need for Projective Reconstruction?</p> <p>In many cases, such as when trying to build a 3D model from internet or legacy photos taken by unknown cameras without any EXIF tags, we do not know ahead of time the intrinsic calibration parameters associated with the input images. In such situations, we can still estimate a two-frame reconstruction, although the true metric structure may not be available, e.g., orthogonal lines or planes in the world may not end up being reconstructed as orthogonal.</p> <p>Consider the derivations we used to estimate the essential matrix \mathbf{E} (11.30–11.32). In the uncalibrated case, we do not know the calibration matrices \mathbf{K}_j, so we cannot use the normalized ray directions $\hat{\mathbf{x}}_j = \mathbf{K}_j^{-1} \mathbf{x}_j$. Instead, we have access only to the image coordinates \mathbf{x}_j, and so the essential matrix equation (11.30) becomes</p> $\hat{\mathbf{x}}_1^T \mathbf{E} \hat{\mathbf{x}}_1 = \mathbf{x}_1^T \mathbf{K}_1^{-T} \mathbf{E} \mathbf{K}_0^{-1} \mathbf{x}_0 = \mathbf{x}_1^T \mathbf{F} \mathbf{x}_0 = 0, \quad (11.48)$ <p>where</p> $\mathbf{F} = \mathbf{K}_1^{-T} \mathbf{E} \mathbf{K}_0^{-1} \quad (11.49)$	2

	<p>is called the <i>fundamental matrix</i> (Faugeras 1992; Hartley, Gupta, and Chang 1992; Hartley and Zisserman 2004).</p> <p>Like the essential matrix, the fundamental matrix is (in principle) rank two,</p> $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{e}_1 \end{bmatrix} \begin{bmatrix} \sigma_0 & & \\ & \sigma_1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_0^T \\ \mathbf{v}_1^T \\ \mathbf{e}_0^T \end{bmatrix}. \quad (11.50)$ <p>Its smallest left singular vector indicates the epipole \mathbf{e}_1 in the image 1 and its smallest right singular vector is \mathbf{e}_0 (Figure 11.11). The fundamental matrix can be factored into a skew-symmetric cross product matrix $[\mathbf{e}]_{\times}$ and a homography $\tilde{\mathbf{H}}$,</p> $\mathbf{F} = [\mathbf{e}]_{\times} \tilde{\mathbf{H}}. \quad (11.51)$ <p>The homography $\tilde{\mathbf{H}}$, which in principle from (11.49) should equal</p> $\tilde{\mathbf{H}} = \mathbf{K}_1^{-T} \mathbf{R} \mathbf{K}_0^{-1}, \quad (11.52)$	2
12.	<p>Bundle Adjustment techniques involved in accurately recovering structure and motion.</p> <p>¹⁸The term “bundle” refers to the bundles of rays connecting camera centers to 3D points and the term “adjustment” refers to the iterative minimization of re-projection error. Alternative terms for this in the vision community include <i>optimal motion estimation</i> (Weng, Ahuja, and Huang 1993) and <i>non-linear least squares</i> (Appendix A.3) (Taylor, Kriegman, and Anandan 1991; Szeliski and Kang 1994).</p> <p>As we have mentioned several times before, the most accurate way to recover structure and motion is to perform robust non-linear minimization of the measurement (re-projection) errors, which is commonly known in the photogrammetry (and now computer vision) communities as <i>bundle adjustment</i>.¹⁸ Triggs, McLauchlan <i>et al.</i> (1999) provide an excellent overview of this topic, including its historical development, pointers to the photogrammetry literature (Slama 1980; Atkinson 1996; Kraus 1997), and subtle issues with gauge ambiguities. The topic is also treated in depth in textbooks and surveys on multi-view geometry (Faugeras and Luong 2001; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010).</p> <p>We have already introduced the elements of bundle adjustment in our discussion on iterative pose estimation (Section 11.2.2), i.e., Equations (11.14–11.20) and Figure 11.7. The biggest difference between these formulas and full bundle adjustment is that our feature location measurements \mathbf{x}_{ij} now depend not only on the point (track) index i but also on the camera pose index j,</p> $\mathbf{x}_{ij} = \mathbf{f}(\mathbf{p}_i, \mathbf{R}_j, \mathbf{c}_j, \mathbf{K}_j), \quad (11.61)$ <p>and that the 3D point positions \mathbf{p}_i are also being simultaneously updated. In addition, it is common to add a stage for radial distortion parameter estimation (2.78),</p> $\mathbf{f}_{\text{RD}}(\mathbf{x}) = (1 + \kappa_1 r^2 + \kappa_2 r^4) \mathbf{x}, \quad (11.62)$ <p>if the cameras being used have not been pre-calibrated, as shown in Figure 11.14.</p>	2

13.	<p>Brightness Constancy Constraint equation.</p> <p>The linearized form of the incremental update to the SSD error (9.28) is often called the <i>optical flow constraint</i> or <i>brightness constancy constraint</i> equation (Horn and Schunck 1981)</p> $I_x u + I_y v + I_t = 0, \quad (9.31)$ <p>where the subscripts in I_x and I_y denote spatial derivatives, and I_t is called the <i>temporal derivative</i>, which makes sense if we are computing instantaneous velocity in a video sequence. When squared and summed or integrated over a region, it can be used to compute optical flow (Horn and Schunck 1981).</p>	2
14.	<p>Projection from 3D to Cylindrical coordinate</p> <p>Assume for now that the camera is in its canonical position, i.e., its rotation matrix is the identity, $\mathbf{R} = \mathbf{I}$, so that the optical axis is aligned with the z-axis and the y-axis is aligned vertically. The 3D ray corresponding to an (x, y) pixel is therefore (x, y, f).</p>  <p>We wish to project this image onto a <i>cylindrical surface</i> of unit radius (Szeliski 1996). Points on this surface are parameterized by an angle θ and a height h, with the 3D cylindrical coordinates corresponding to (θ, h) given by</p> $(\sin \theta, h, \cos \theta) \propto (x, y, f), \quad (8.45)$ <p>as shown in Figure 8.8a. From this correspondence, we can compute the formula for the <i>warped</i> or <i>mapped</i> coordinates (Szeliski and Shum 1997),</p> $x' = s\theta = s \tan^{-1} \frac{x}{f}, \quad (8.46)$ $y' = sh = s \frac{y}{\sqrt{x^2 + f^2}}, \quad (8.47)$ <p>where s is an arbitrary scaling factor (sometimes called the <i>radius</i> of the cylinder) that can be set to $s = f$ to minimize the distortion (scaling) near the center of the image.¹⁷ The inverse of this mapping equation is given by</p> $x = f \tan \theta = f \tan \frac{x'}{s}, \quad (8.48)$ $y = h \sqrt{x^2 + f^2} = \frac{y'}{s} f \sqrt{1 + \tan^2 x'/s} = f \frac{y'}{s} \sec \frac{x'}{s}. \quad (8.49)$	2

15	<p>The process of Recognizing Panoramas.</p> <p>To recognize panoramas, Brown and Lowe (2007) first find all pairwise image overlaps using a feature-based method and then find connected components in the overlap graph to “recognize” individual panoramas (Figure 8.11). The feature-based matching stage first extracts scale invariant feature transform (SIFT) feature locations and feature descriptors (Lowe 2004) from all the input images and places them in an indexing structure, as described in Section 7.1.3. For each image pair under consideration, the nearest matching neighbor is found for each feature in the first image, using the indexing structure to rapidly find candidates and then comparing feature descriptors to find the best match. RANSAC is used to find a set of <i>inlier</i> matches; pairs of matches are used to hypothesize similarity motion models that are then used to count the number of inliers. A RANSAC algorithm tailored specifically for rotational panoramas is described by Brown, Hartley, and Nistér (2007).</p>	2
16	<p>Parallax</p> <p>Once we have optimized the global orientations and focal lengths of our cameras, we may find that the images are still not perfectly aligned, i.e., the resulting stitched image looks blurry or ghosted in some places. This can be caused by a variety of factors, including unmodeled radial distortion, 3D parallax (failure to rotate the camera around its front nodal point), small scene motions such as waving tree branches, and large-scale scene motions such as people moving in and out of pictures.</p> <p>3D parallax can be handled by doing a full 3D bundle adjustment, i.e., by replacing the projection Equation (8.59) used in Equation (8.62) with Equation (2.68), which models camera translations. The 3D positions of the matched feature points and cameras can then be simultaneously recovered, although this can be significantly more expensive than parallax-free image registration. Once the 3D structure has been recovered, the scene could (in theory) be</p>	2
	PART C (2 × 12 = 24 marks)	
17.a.	<p>FOURIER BASED ALIGNMENT:</p> <p>When the search range corresponds to a significant fraction of the larger image (as is the case in image stitching, see Section 8.2), the hierarchical approach may not work that well, as it is often not possible to coarsen the representation too much before significant features are blurred away. In this case, a Fourier-based approach may be preferable.</p> <p>Fourier-based alignment relies on the fact that the Fourier transform of a shifted signal has the same magnitude as the original signal, but a linearly varying phase (Section 3.4), i.e.,</p> $\mathcal{F}\{I_1(\mathbf{x} + \mathbf{u})\} = \mathcal{F}\{I_1(\mathbf{x})\} e^{-j\mathbf{u} \cdot \boldsymbol{\omega}} = \mathcal{I}_1(\boldsymbol{\omega}) e^{-j\mathbf{u} \cdot \boldsymbol{\omega}}, \quad (9.17)$ <p>where $\boldsymbol{\omega}$ is the vector-valued angular frequency of the Fourier transform and we use calligraphic notation $\mathcal{I}_1(\boldsymbol{\omega}) = \mathcal{F}\{I_1(\mathbf{x})\}$ to denote the Fourier transform of a signal (Section 3.4).</p> <p>Another useful property of Fourier transforms is that convolution in the spatial domain corresponds to multiplication in the Fourier domain (Section 3.4).⁶ The Fourier transform of the cross-correlation function E_{CC} can thus be written as</p>	2

	$\mathcal{F}\{E_{CC}(\mathbf{u})\} = \mathcal{F}\left\{\sum_i I_0(\mathbf{x}_i)I_1(\mathbf{x}_i + \mathbf{u})\right\} = \mathcal{F}\{I_0(\mathbf{u})\bar{*}I_1(\mathbf{u})\} = \mathcal{I}_0(\boldsymbol{\omega})\mathcal{I}_1^*(\boldsymbol{\omega}), \quad (9.18)$ <p>where</p> $f(\mathbf{u})\bar{*}g(\mathbf{u}) = \sum_i f(\mathbf{x}_i)g(\mathbf{x}_i + \mathbf{u}) \quad (9.19)$ <p>is the <i>correlation</i> function, i.e., the convolution of one signal with the reverse of the other, and $\mathcal{I}_1^*(\boldsymbol{\omega})$ is the <i>complex conjugate</i> of $\mathcal{I}_1(\boldsymbol{\omega})$. This is because convolution is defined as the summation of one signal with the reverse of the other (Section 3.4).</p> <p>To efficiently evaluate E_{CC} over the range of all possible values of \mathbf{u}, we take the Fourier transforms of both images $I_0(\mathbf{x})$ and $I_1(\mathbf{x})$, multiply both transforms together (after conjugating the second one), and take the inverse transform of the result. The Fast Fourier Transform algorithm can compute the transform of an $N \times M$ image in $O(NM \log NM)$ operations (Bracewell 1986). This can be significantly faster than the $O(N^2M^2)$ operations required to do a full search when the full range of image overlaps is considered.</p> <p>While Fourier-based convolution is often used to accelerate the computation of image correlations, it can also be used to accelerate the sum of squared differences function (and its variants). Consider the SSD formula given in (9.1). Its Fourier transform can be written as</p> $\begin{aligned} \mathcal{F}\{E_{SSD}(\mathbf{u})\} &= \mathcal{F}\left\{\sum_i [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2\right\} \\ &= \delta(\boldsymbol{\omega}) \sum_i [I_0^2(\mathbf{x}_i) + I_1^2(\mathbf{x}_i)] - 2\mathcal{I}_0(\boldsymbol{\omega})\mathcal{I}_1^*(\boldsymbol{\omega}). \end{aligned} \quad (9.20)$ <p>Thus, the SSD function can be computed by taking twice the correlation function and subtracting it from the sum of the energies in the two images (or patches).</p>	2
	(OR)	
17.b.	<p>Hierarchical Motion Estimation:</p> <p>Now that we have a well-defined alignment cost function to optimize, how can we find its minimum? The simplest solution is to do a <i>full search</i> over some range of shifts, using either integer or sub-pixel steps. This is often the approach used for <i>block matching</i> in <i>motion compensated video compression</i>, where a range of possible motions (say, ± 16 pixels) is explored.⁴</p> <p>To accelerate this search process, <i>hierarchical motion estimation</i> is often used: an image pyramid (Section 3.5) is constructed and a search over a smaller number of discrete pixels (corresponding to the same range of motion) is first performed at coarser levels (Quam 1984; Anandan 1989; Bergen, Anandan <i>et al.</i> 1992). The motion estimate from one level of the pyramid is then used to initialize a smaller <i>local</i> search at the next finer level. Alternatively, several seeds (good solutions) from the coarse level can be used to initialize the fine-level search. While this is not guaranteed to produce the same result as a full search, it usually works almost as well and is much faster.</p>	4

	<p>More formally, let</p> $I_k^{(l)}(\mathbf{x}_j) \leftarrow \tilde{I}_k^{(l-1)}(2\mathbf{x}_j) \quad (9.15)$ <p>be the <i>decimated</i> image at level l obtained by subsampling (<i>downsampling</i>) a smoothed version of the image at level $l-1$. See Section 3.5 for how to perform the required downsampling (pyramid construction) without introducing too much aliasing.</p> <p>At the coarsest level, we search for the best displacement $\mathbf{u}^{(l)}$ that minimizes the difference between images $I_0^{(l)}$ and $I_1^{(l)}$. This is usually done using a full search over some range of displacements $\mathbf{u}^{(l)} \in 2^{-l}[-S, S]^2$, where S is the desired <i>search range</i> at the finest (original) resolution level, optionally followed by the incremental refinement step described in Section 9.1.3.</p> <p>Once a suitable motion vector has been estimated, it is used to <i>predict</i> a likely displacement</p> $\hat{\mathbf{u}}^{(l-1)} \leftarrow 2\mathbf{u}^{(l)} \quad (9.16)$ <p>for the next finer level.⁵ The search over displacements is then repeated at the finer level over a much narrower range of displacements, say $\hat{\mathbf{u}}^{(l-1)} \pm 1$, again optionally combined with an incremental refinement step (Anandan 1989). Alternatively, one of the images can be <i>warped</i> (resampled) by the current motion estimate, in which case only small incremental motions need to be computed at the finer level. A nice description of the whole process, extended to parametric motion estimation (Section 9.2), is provided by Bergen, Anandan <i>et al.</i> (1992).</p>	<p>4</p> <p>4</p>
<p>18.a.</p>	<p>Parametric Motion Models:</p> <p>Before we can register and align images, we need to establish the mathematical relationships that map pixel coordinates from one image to another. A variety of such <i>parametric motion models</i> are possible, from simple 2D transforms, to planar perspective models, 3D camera rotations, lens distortions, and mapping to non-planar (e.g., cylindrical) surfaces.</p> <p>We already covered several of these models in Sections 2.1 and 8.1. In particular, we saw in Section 2.1.4 how the parametric motion describing the deformation of a planar surface as viewed from different positions can be described with an eight-parameter homography (2.71) (Mann and Picard 1994; Szeliski 1996). We also saw how a camera undergoing a pure rotation induces a different kind of homography (2.72).</p> <p>Planar perspective motion</p> <p>The simplest possible motion model to use when aligning images is to simply translate and rotate them in 2D (Figure 8.4a). This is exactly the same kind of motion that you would use if you had overlapping photographic prints. It is also the kind of technique favored by David Hockney to create the collages that he calls <i>joiners</i> (Zelnik-Manor and Perona 2007; Nomura, Zhang, and Nayar 2007). Creating such collages, which show visible seams and inconsistencies that add to the artistic effect, is popular on websites such as Flickr, where they more commonly go under the name <i>panography</i> (Section 8.1.2). Translation and rotation are also usually adequate motion models to compensate for small camera motions in applications such as photo and video stabilization and merging (Exercise 8.1 and Section 9.2.1).</p>	<p>4</p> <p>2</p>

	<p>In Section 2.1.4, we saw how the mapping between two cameras viewing a common plane can be described using a 3×3 homography (2.71). Consider the matrix \mathbf{M}_{10} that arises when mapping a pixel in one image to a 3D point and then back onto a second image,</p> $\tilde{\mathbf{x}}_1 \sim \tilde{\mathbf{P}}_1 \tilde{\mathbf{P}}_0^{-1} \tilde{\mathbf{x}}_0 = \mathbf{M}_{10} \tilde{\mathbf{x}}_0. \quad (8.33)$ <p>When the last row of the \mathbf{P}_0 matrix is replaced with a plane equation $\hat{\mathbf{n}}_0 \cdot \mathbf{p} + c_0$ and points are assumed to lie on this plane, i.e., their disparity is $d_0 = 0$, we can ignore the last column of \mathbf{M}_{10} and also its last row, since we do not care about the final z-buffer depth. The resulting homography matrix $\tilde{\mathbf{H}}_{10}$ (the upper left 3×3 sub-matrix of \mathbf{M}_{10}) describes the mapping between pixels in the two images,</p> $\tilde{\mathbf{x}}_1 \sim \tilde{\mathbf{H}}_{10} \tilde{\mathbf{x}}_0. \quad (8.34)$ <p>This observation formed the basis of some of the earliest automated image stitching algorithms (Mann and Picard 1994; Szeliski 1994, 1996). Because reliable feature matching techniques had not yet been developed, these algorithms used direct pixel value matching, i.e., direct parametric motion estimation, as described in Section 9.2 and Equations (8.19–8.20).</p> <p>More recent stitching algorithms first extract features and then match them up, often using robust techniques such as RANSAC (Section 8.1.4) to compute a good set of inliers. The final computation of the homography (8.34), i.e., the solution of the least squares fitting problem given pairs of corresponding features,</p> $x_1 = \frac{(1 + h_{00})x_0 + h_{01}y_0 + h_{02}}{h_{20}x_0 + h_{21}y_0 + 1} \quad \text{and} \quad (8.35)$ $y_1 = \frac{h_{10}x_0 + (1 + h_{11})y_0 + h_{12}}{h_{20}x_0 + h_{21}y_0 + 1}, \quad (8.36)$ <p>uses iterative least squares, as described in Section 8.1.3 and Equations (8.21–8.23).</p>	2
	OR	
18.b.	<p>Bundle Adjustment:</p> <p>One way to register a large number of images is to add new images to the panorama one at a time, aligning the most recent image with the previous ones already in the collection (Szeliski and Shum 1997) and discovering, if necessary, which images it overlaps (Sawhney and Kumar 1999). In the case of 360° panoramas, accumulated error may lead to the presence of a gap (or excessive overlap) between the two ends of the panorama, which can be fixed by stretching the alignment of all the images using a process called <i>gap closing</i> (Section 8.2.4). However, a better alternative is to simultaneously align all the images using a least-squares framework to correctly distribute any misregistration errors.</p>	2

The process of simultaneously adjusting pose parameters and 3D point locations for a large collection of overlapping images is called *bundle adjustment* in the photogrammetry community (Triggs, McLauchlan *et al.* 1999). In computer vision, it was first applied to the general structure from motion problem (Szeliski and Kang 1994) and then later specialized for panoramic image stitching (Shum and Szeliski 2000; Sawhney and Kumar 1999; Coorg and Teller 2000).

2

In this section, we formulate the problem of global alignment using a feature-based approach, since this results in a simpler system. An equivalent direct approach can be obtained either by dividing images into patches and creating a virtual feature correspondence for each one (Shum and Szeliski 2000) or by replacing the per-feature error metrics with per-pixel metrics (Irani and Anandan 1999).

Before we describe this in more details, we should mention that a simpler, although less accurate, approach is to compute pairwise rotation estimates between overlapping images, and to then use a *rotation averaging* approach to estimate a global rotation for each camera (Hartley, Trumpf *et al.* 2013). However, since the measurement errors in each feature point location are not being counted correctly, as is the case in bundle adjustment, the solution will not have the same theoretical optimality.

2

Consider the feature-based alignment problem given in Equation (8.2), i.e.,

$$E_{\text{pairwise-LS}} = \sum_i \|\mathbf{r}_i\|^2 = \|\tilde{\mathbf{x}}'_i(\mathbf{x}_i; \mathbf{p}) - \hat{\mathbf{x}}'_i\|^2. \quad (8.58)$$

For multi-image alignment, instead of having a single collection of pairwise feature correspondences, $\{(\mathbf{x}_i, \hat{\mathbf{x}}'_i)\}$, we have a collection of n features, with the location of the i th feature point in the j th image denoted by \mathbf{x}_{ij} and its scalar confidence (i.e., inverse variance) denoted by c_{ij} .²¹ Each image also has some associated pose parameters.

In this section, we assume that this pose consists of a rotation matrix \mathbf{R}_j and a focal length f_j , although formulations in terms of homographies are also possible (Szeliski and Shum 1997; Sawhney and Kumar 1999). The equation mapping a 3D point \mathbf{x}_i into a point \mathbf{x}_{ij} in frame j can be re-written from Equations (2.68) and (8.38) as

2

$$\tilde{\mathbf{x}}_{ij} \sim \mathbf{K}_j \mathbf{R}_j \mathbf{x}_i \quad \text{and} \quad \mathbf{x}_i \sim \mathbf{R}_j^{-1} \mathbf{K}_j^{-1} \tilde{\mathbf{x}}_{ij}, \quad (8.59)$$

where $\mathbf{K}_j = \text{diag}(f_j, f_j, 1)$ is the simplified form of the calibration matrix. The motion mapping a point \mathbf{x}_{ij} from frame j into a point \mathbf{x}_{ik} in frame k is similarly given by

	$\tilde{\mathbf{x}}_{ik} \sim \tilde{\mathbf{H}}_{kj} \tilde{\mathbf{x}}_{ij} = \mathbf{K}_k \mathbf{R}_k \mathbf{R}_j^{-1} \mathbf{K}_j^{-1} \tilde{\mathbf{x}}_{ij}. \quad (8.60)$ <p>Given an initial set of $\{(\mathbf{R}_j, f_j)\}$ estimates obtained from chaining pairwise alignments, how do we refine these estimates?</p> <p>One approach is to directly extend the pairwise energy $E_{\text{pairwise-LS}}$ (8.58) to a multiview formulation,</p> $E_{\text{all-pairs-2D}} = \sum_i \sum_{jk} c_{ij} c_{ik} \ \tilde{\mathbf{x}}_{ik}(\hat{\mathbf{x}}_{ij}; \mathbf{R}_j, f_j, \mathbf{R}_k, f_k) - \hat{\mathbf{x}}_{ik}\ ^2, \quad (8.61)$ <p>where the $\tilde{\mathbf{x}}_{ik}$ function is the <i>predicted</i> location of feature i in frame k given by (8.60), $\hat{\mathbf{x}}_{ij}$ is the <i>observed</i> location, and the “2D” in the subscript indicates that an image-plane error is being minimized (Shum and Szeliski 2000). Note that since $\tilde{\mathbf{x}}_{ik}$ depends on the $\hat{\mathbf{x}}_{ij}$ observed value, we actually have an <i>errors-in-variable</i> problem, which in principle requires more sophisticated techniques than least squares to solve (Van Huffel and Lemmerling 2002; Matei and Meer 2006). However, in practice, if we have enough features, we can directly minimize the above quantity using regular non-linear least squares and obtain an accurate multi-frame alignment.</p> <p>An alternative way to formulate the optimization is to use true bundle adjustment, i.e., to solve not only for the pose parameters $\{(\mathbf{R}_j, f_j)\}$ but also for the 3D point positions $\{\mathbf{x}_i\}$,</p> $E_{\text{BA-2D}} = \sum_i \sum_j c_{ij} \ \tilde{\mathbf{x}}_{ij}(\mathbf{x}_i; \mathbf{R}_j, f_j) - \hat{\mathbf{x}}_{ij}\ ^2, \quad (8.62)$ <p>where $\tilde{\mathbf{x}}_{ij}(\mathbf{x}_i; \mathbf{R}_j, f_j)$ is given by (8.59). The disadvantage of full bundle adjustment is that there are more variables to solve for, so each iteration and also the overall convergence may be slower. (Imagine how the 3D points need to “shift” each time some rotation matrices are updated.) However, the computational complexity of each linearized Gauss–Newton step can be reduced using sparse matrix techniques (Section 11.4.3) (Szeliski and Kang 1994; Triggs, McLauchlan <i>et al.</i> 1999; Hartley and Zisserman 2004).</p>	<p>2</p> <p>2</p>
--	--	-------------------

Signature of the HoD/CSE