Information Fusion for Multi-camera and Multi-body Structure and Motion

Alexander Andreopoulos and John K. Tsotsos

York University,
Dept. of Computer Science & Engineering,
Toronto, Ontario, M3J 1P3, Canada
{alekos,tsotsos}@cse.yorku.ca

Abstract. Information fusion algorithms have been successful in many vision tasks such as stereo, motion estimation, registration and robot localization. Stereo and motion image analysis are intimately connected and can provide complementary information to obtain robust estimates of scene structure and motion. We present an information fusion based approach for multi-camera and multi-body structure and motion that combines bottom-up and top-down knowledge on scene structure and motion. The only assumption we make is that all scene motion consists of rigid motion. We present experimental results on synthetic and non-synthetic data sets, demonstrating excellent performance compared to binocular based state-of-the-art approaches for structure and motion.

1 Introduction

Multi-body and multi-camera structure and motion establishes the structure and motion of a scene that consists of multiple moving rigid objects that are observed from multiple views [1], [2]. Stereo vision analysis and image motion analysis provide information with complementary uncertainties which can depend on the motion of the camera platform, the scene structure and the spatio-temporal baselines. There are four fundamental problems with the extractable information from motion data or from stereo data [3]: (i) Image motion and disparity, with an unknown camera translation, allow us to infer object range only up to a scale ambiguity, since image motion and disparity depend on the ratio of camera translation to object range. (ii) Image motion and disparity tend towards zero near the focus of expansion (FOE). Since object range is inversely proportional to image motion and disparity, scene structure estimation is ill-conditioned near the FOE. (iii) The more closely aligned the local image structure is with the epipolar directions – *i.e.*, directions pointing towards the FOE – the more ill-conditioned scene structure estimation becomes in those regions. (iv) Whereas large spatio-temporal baselines give better depth estimates for distant objects, the greater disparity and occlusion makes such cameras unsuitable for nearby objects. The severity of these problems is reversed when dealing with small baselines. The spatio-temporal baselines might be defined with respect to a monocular camera in motion –structure from motion–, a static stereo camera, or some other combination of static and non-static cameras.

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 385–396, 2007.

A method for fusing the structure and motion estimates of different cameras by preserving the accurate estimates and diminishing the effect of inaccurate estimates is highly desirable. For example, whereas in one camera the optical flow near the FOE might be poorly estimated, from another camera's viewpoint the optical flow of the same scene region might not be as ill-conditioned, since the FOE will likely have changed. We present an information fusion based approach for dealing with all these problems in a unified framework. We model the above mentioned errors as originating from ambiguities in the estimation of stereo image correspondences and in the optical flow across all cameras. The only assumption we make as to the scene motion is that we are dealing with rigidly moving objects.

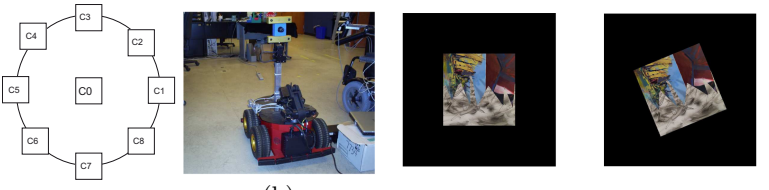
The rest of the paper is organized as follows. Section 2 presents some related work. Section 3 introduces an approach for representing the motion and stereo data from a network of cameras. Section 4 describes how to combine this data in a single reference frame. Section 5 outlines a simple extension of the approach to camera rigs with arbitrary intrinsic and extrinsic parameters. Section 6 presents experimental results demonstrating the robustness of the approach. Section 7 concludes the paper.

2 Related Work

Richards [4] shows how the integration of changing disparity and object velocity can solve many of the ambiguitites inherent in stereopsis and motion under orthographic projection. Waxman [5] demonstrates the importance of the ratio of the rate of change of disparity over disparity, by using this quantity to unify stereo and motion analysis. As it is elaborated in [6], the importance of this ratio has been demonstrated numerous other times. Hanna and Okamoto [3] demonstrate how motion and stereo could be combined in a multi-camera system for egomotion and scene structure estimation. Their work is further expanded upon by Mandelbaum *et al.* [7]. Zhang and Kambhamettu [8] present a system which integrates 3D scene flow and structure recovery in order to complement the performance of each other, using a number of calibrated cameras. Singh and Allen [9] employ the *Best Linear Unbiased Estimator* (BLUE) to fuse local motion. Comaniciu [10], [11] developed a method for motion estimation under multiple source models. Neumann *et al.* [12] present a method for establishing a hierarchy of cameras based upon the stability and complexity of structure and motion estimation. To the best of our knowledge, the work we present is the first approach using information fusion for multi-camera and multi-body structure and motion.

3 Fusing Multiple Cameras

Assume we have a multi-camera rig composed of N monocular cameras. A maximum of $\binom{N}{2}$ camera pairs exist. The coordinate system of camera C_0 is referred



(a) (b) (c) (d)

Fig. 1. (a) Diagram of a hypothetical nine camera rig. (b) A five camera rig mounted on a mobile robotic platform. (c)A planar textured region we used in some of the experiments for structure and motion estimation at a depth of 300cm. (d)The region after a 20 degree rotation around the camera's optical axis.

to as the *basis coordinate system*. By convention a vector's superscript will denote the coordinate system with respect to which we are expressing the vector. The camera rig is calibrated and therefore, for each pair of cameras C_i, C_j, we know a rotation matrix $\mathbf{R}_{ij}$ and translation vector $\mathbf{T}_{ij} = (T_{ij}^x, T_{ij}^y, T_{ij}^z)^T$ that describes the rotation and translation that aligns camera C_i's coordinate axes with camera C_j's coordinate axes. See Fig. 1(a),(b) for examples of camera rigs where $\mathbf{R}_{ij} = \mathbf{I}$ (the identity matrix) $\forall i, j$. For each pixel $\mathbf{p}_0$ in camera C_0, and for each camera pair (C_j, C_i) such that $i \neq 0$, $j \neq i$, we can use a stereo correspondence algorithm, such as [13], to obtain estimates of the pixels $\mathbf{p}_j$, $\mathbf{p}_i$ in cameras C_j, C_i respectively, corresponding to pixel $\mathbf{p}_0$ in basis camera C_0. Similarly, we can obtain motion flow estimates for each pixel $\mathbf{p}_j$, $\mathbf{p}_i$ in C_j, C_i.

With each such pair of image pixels $\mathbf{p}_j$, $\mathbf{p}_i$, we can associate a $6D$ vector $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$, containing the $3D$ coordinates $\mathbf{X}_1^{C_0} = (X_1^{C_0}, Y_1^{C_0}, Z_1^{C_0})$ of a point $\mathbf{P}$ that is imaged by $\mathbf{p}_j$, $\mathbf{p}_i$ in camera pair (C_j, C_i). We can also associate with $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$ a $3D$ vector $\mathbf{u}^{C_0}$ corresponding to the $3D$ displacement vector of $\mathbf{P}$ that was extracted using the camera pair (C_j, C_i). The displacement vector might be due to camera movement, an independent motion of scene point $\mathbf{P}$ or a combination of both. As we have indicated above, the superscript C_0 in $\mathbf{X}_1^{C_0}$, $\mathbf{u}^{C_0}$ indicates that the vectors are expressed with respect to the coordinate system of C_0. Let $\mathbf{X}_2^{C_0} = (X_2^{C_0}, Y_2^{C_0}, Z_2^{C_0})$ denote the coordinate of $\mathbf{P}$ with respect to camera's C_0 coordinate system, obtained after an arbitrary camera rig or scene motion. The context will always make it clear with respect to which camera pair (C_j, C_i) we estimated $\mathbf{X}_1^{C_0}$, $\mathbf{X}_2^{C_0}$. We can then obtain the 3D motion estimate $\mathbf{u}^{C_0}$ for point $\mathbf{P}$ by $\mathbf{u}^{C_0} = \mathbf{X}_2^{C_0} - \mathbf{X}_1^{C_0}$. Then $\mathbf{V}(\mathbf{p}_0, C_j, C_i) \triangleq (\mathbf{X}_1^{C_0}, \mathbf{u}^{C_0})^T$.

Given a small neighborhood $\Delta_{\mathbf{p}_0}$ of pixels around a pixel $\mathbf{p}_0$ in C_0 – we use 3×3 pixel neighborhoods in this paper –, the set $\bigcup_{\mathbf{p}\in\Delta_{\mathbf{p}_0}} \bigcup_{j=0}^{N} \bigcup_{i=1,i>j}^{N} \mathbf{V}(\mathbf{p}, C_j, C_i)$ contains estimates of scene structure and motion over all camera pairs. If we need to enforce a hard real-time constraint, we can select to process a subset of the camera pairs. For each camera pair C_j, C_i and each pixel $\mathbf{p}_0$ in C_0 that we process, we assign the covariance matrix $Cov(\mathbf{V}(\mathbf{p}_0, C_j, C_i))$. In the next section we will show how to estimate this covariance matrix and how to use it to assign a weight of importance to each one of those vectors. We will also show how to use

information fusion techniques to get a robust estimate of the true scene structure and motion. Notice that in the above mentioned set, mainly due to occlusions, $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$ will not always contribute a vector for all $\mathbf{p}_0$, C_j, C_i.

4 Fusing the Camera Data

We need to model the uncertainty in each of the 6D vectors $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$ in order to obtain each vector's 6×6 covariance matrix. These covariance matrices are used by the BLUE estimator to obtain a reliable estimate of the scene structure and motion. For example, an image pixel that is near the focus of expansion in one monocular camera needs to assign a high uncertainty to its motion elements and assign a 3D structure uncertainty that depends on the scene depth relative to the camera pair used. From a different stereo camera's point of view, these uncertainties will differ. By combining bottom-up and top-down information related to the scene uncertainty we obtain the noise model used by our BLUE estimator. For notational simplicity, we initially assume a perspective projection camera model where all cameras have the same focal length f, the aspect ratio is 1, the skew is 0, and the principal point is set to (0,0). The camera set up similar to Fig. 1(a),(b) where $\mathbf{R}_{ij} = \mathbf{I}$ and $T^z_{ij} = 0 \ \forall i, j$. The extension to arbitrary camera setups is presented in Section 5. Every pair of cameras (C_j, C_i) can be viewed as a stereo camera with a focal length f, such that the projection of a point $\mathbf{X}_1^{C_0} = (X_1^{C_0}, Y_1^{C_0}, Z_1^{C_0})$ in camera C_i is given by:

$$x_r = \frac{(X_1^{C_0} - T^x_{0i}) * f}{Z_1^{C_0}}, \quad y_r = \frac{(Y_1^{C_0} - T^y_{0i}) * f}{Z_1^{C_0}} \tag{1}$$

and the projection of the same point in camera C_j is:

$$x_l = \frac{(X_1^{C_0} - T^x_{0j}) * f}{Z_1^{C_0}}, \quad y_l = \frac{(Y_1^{C_0} - T^y_{0j}) * f}{Z_1^{C_0}}. \tag{2}$$

If $|-T^x_{0j} + T^x_{0i}| \geq |-T^y_{0j} + T^y_{0i}|$, we have:

$$X_1^{C_0} = \frac{(-T^x_{0j} + T^x_{0i})}{2} \frac{(x_r + x_l)}{x_l - x_r} + \frac{T^x_{0j} + T^x_{0i}}{2} \tag{3}$$

$$Y_1^{C_0} = (-T^x_{0j} + T^x_{0i}) \frac{y_r}{x_l - x_r} + T^y_{0i} \tag{4}$$

$$Z_1^{C_0} = \frac{(-T^x_{0j} + T^x_{0i}) f}{x_l - x_r}. \tag{5}$$

Conversely, if $|-T^x_{0j} + T^x_{0i}| < |-T^y_{0j} + T^y_{0i}|$, we have:

$$X_1^{C_0} = (-T^y_{0j} + T^y_{0i}) \frac{x_r}{y_l - y_r} + T^x_{0i} \tag{6}$$

$$Y_1^{C_0} = \frac{(-T^y_{0j} + T^y_{0i})}{2} \frac{(y_r + y_l)}{y_l - y_r} + \frac{T^y_{0j} + T^y_{0i}}{2} \tag{7}$$

$$Z_1^{C_0} = \frac{(-T^y_{0j} + T^y_{0i}) f}{y_l - y_r}. \tag{8}$$

Notice that in Eqs.(3)-(5) and Eqs.(6)-(8), y_l and x_l respectively, are not used. This provides a simple approximation for $\mathbf{X}_1^{C_0}$ when due to small errors (x_l, y_l), (x_r, y_r) are not corresponding pixels. The corresponding image coordinates in the next frame are given by $(x_l', y_l') = (x_l, y_l) + (v_x^{C_j}, v_y^{C_j})$, $(x_r', y_r') = (x_r, y_r) + (v_x^{C_i}, v_y^{C_i})$ where $(v_x^{C_j}, v_y^{C_j})$, $(v_x^{C_i}, v_y^{C_i})$ denote the motion flow vectors in cameras C_j, C_i respectively. We can use (x_l', y_l'),(x_r', y_r'), in conjunction with Eqs.(3)-(8), to estimate $\mathbf{X}_2^{C_0} = (X_2^{C_0}, Y_2^{C_0}, Z_2^{C_0})$ and calculate $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$.

We now show how Eqs. (3)-(8) can be used to define a covariance matrix for $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$. We only describe the covariance matrix derivation for $|-T_{0j}^x + T_{0i}^x| \geq |-T_{0j}^y + T_{0i}^y|$, since the case $|-T_{0j}^x + T_{0i}^x| < |-T_{0j}^y + T_{0i}^y|$ is similar. We model the error in the correspondences of the image points as $(x_r + n_{x_r}, y_r + n_{y_r})$, $(x_l + n_{x_l}, y_l + n_{y_l})$ where n_{x_r},n_{y_r},n_{x_l},n_{y_l} are zero mean Gaussian random variables. Their standard deviation can depend on how noisy the images are and on prior knowledge regarding the accuracy of the correspondences –*e.g.,* the sample variance of the correspondences within $\Delta_{\mathbf{p}_0}$. In this paper we assume a variance of $\frac{1}{2}$ pixel for each of the four random variables. We also assume that the random variables are independent. Furthermore, we notice that in Eqs. (3)-(8) we can view $X_1^{C_0}$, $Y_1^{C_0}$, $Z_1^{C_0}$ as functions in terms of n_{x_r},n_{y_r},n_{x_l},n_{y_l}. We obtain first order Taylor expansions of $X_1^{C_0}$, $Y_1^{C_0}$, $Z_1^{C_0}$ and we use these Taylor expansions to obtain variance/covariance measures for vector $\mathbf{X}_1^{C_0}$. It can be shown that within first order:

$$Var(X_1^{C_0}) \approx \frac{((-T_{0j}^x + T_{0i}^x)\widehat{x}_l)^2}{(\widehat{x}_l - \widehat{x}_r)^4} Var(x_r) + \frac{((T_{0j}^x - T_{0i}^x)\widehat{x}_r)^2}{(\widehat{x}_l - \widehat{x}_r)^4} Var(x_l) \tag{9}$$

$$Var(Y_1^{C_0}) \approx \frac{(-T_{0j}^x + T_{0i}^x)^2}{(\widehat{x}_l - \widehat{x}_r)^2} Var(y_r) + \frac{((-T_{0j}^x + T_{0i}^x)\widehat{y}_r)^2}{(\widehat{x}_l - \widehat{x}_r)^4} Var(x_r) + \frac{((T_{0j}^x - T_{0i}^x)\widehat{y}_r)^2}{(\widehat{x}_l - \widehat{x}_r)^4} Var(x_l) \tag{10}$$

$$Var(Z_1^{C_0}) \approx \frac{((-T_{0j}^x + T_{0i}^x)f)^2}{(\widehat{x}_l - \widehat{x}_r)^4} Var(x_r) + \frac{((T_{0j}^x - T_{0i}^x)f)^2}{(\widehat{x}_l - \widehat{x}_r)^4} Var(x_l) \tag{11}$$

where $\widehat{x}_l$, $\widehat{x}_r$, $\widehat{y}_l$ and $\widehat{y}_r$ are estimated using a trimmed mean estimator, with the top and bottom, 25% of the samples being rejected before calculating the mean. The samples used to estimate $\widehat{x}_l$, $\widehat{x}_r$, $\widehat{y}_l$ and $\widehat{y}_r$ are the pixels in C_i, C_j corresponding to the neighborhood $\Delta_{\mathbf{p}_0}$ in C_0. For example, to estimate $\widehat{x}_r$ we use the stereo matching algorithm to find the pixels in C_i that correspond to the pixels $\Delta_{\mathbf{p}_0}$ in C_0, and then apply the trimmed mean estimator to get $\widehat{x}_r$.

$$Cov(\mathbf{V}(\mathbf{p}_0, C_j, C_i) = (1-a)\begin{pmatrix} Var(X_1^{C_0}) & 0 & 0 & -Var(X_1^{C_0}) & 0 & 0 \\ 0 & Var(Y_1^{C_0}) & 0 & 0 & -Var(Y_1^{C_0}) & 0 \\ 0 & 0 & Var(Z_1^{C_0}) & 0 & 0 & -Var(Z_1^{C_0}) \\ -Var(X_1^{C_0}) & 0 & 0 & Var(X_2^{C_0} - X_1^{C_0}) & 0 & 0 \\ 0 & -Var(Y_1^{C_0}) & 0 & 0 & Var(Y_2^{C_0} - Y_1^{C_0}) & 0 \\ 0 & 0 & -Var(Z_1^{C_0}) & 0 & 0 & Var(Z_2^{C_0} - Z_1^{C_0}) \end{pmatrix} + a\begin{pmatrix} \overline{Var}(X_1^{C_0}) & 0 & 0 & 0 & 0 & 0 \\ 0 & \overline{Var}(Y_1^{C_0}) & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{Var}(Z_1^{C_0}) & 0 & 0 & 0 \\ 0 & 0 & 0 & \overline{Var}(X_2^{C_0} - X_1^{C_0}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \overline{Var}(Y_2^{C_0} - Y_1^{C_0}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \overline{Var}(Z_2^{C_0} - Z_1^{C_0}) \end{pmatrix}$$

Fig. 2. The covariance matrix encoding the uncertainties

In order to obtain a covariance matrix for $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$, we also need to obtain an estimate of the variance of the elements of $\mathbf{u}^{C_0}$. We know that for a physical point $\mathbf{P}^{C_j}$ that is moving with velocity $\mathbf{S}^{C_j}$ with respect to camera C_j and its coordinate frame, we can decompose the velocity as $\mathbf{S}^{C_j} = -\mathbf{T}^{C_j} - \mathbf{\Omega}^{C_j} \times \mathbf{P}^{C_j}$ where $\mathbf{T}^{C_j} = (T_x^{C_j}, T_y^{C_j}, T_z^{C_j})^T$ and $\mathbf{\Omega}^{C_j} = (\Omega_x^{C_j}, \Omega_y^{C_j}, \Omega_z^{C_j})^T$ denote the translational and angular velocity vectors of camera C_j that would cause the same apparent motion of the particle $\mathbf{P}^{C_j}$ with respect to camera C_j's coordinate frame. Then, the image velocity of the projection (x_l, y_l) of $\mathbf{P}^{C_j}$ in camera C_j is given by

$$\begin{pmatrix} v_x^{C_j} \\ v_y^{C_j} \end{pmatrix} = B^{C_j} \mathbf{\Omega}^{C_j} + d^{C_j} A^{C_j} \mathbf{T}^{C_j} \tag{12}$$

where d^{C_j} is the inverse of the scene depth with respect to camera C_j's coordinate system – it is estimated using Eqs.(3)-(8) and the current camera pair – and

$$B^{C_j} = \begin{pmatrix} \frac{x_l y_l}{f} & -(f + \frac{x_l^2}{f}) & y_l \\ (f + \frac{y_l^2}{f}) & -\frac{x_l y_l}{f} & -x_l \end{pmatrix} \quad A^{C_j} = \begin{pmatrix} -f & 0 & x_l \\ 0 & -f & y_l \end{pmatrix}. \tag{13}$$

Similar conditions hold for camera C_i. We use Eq.(12) to model the noise sensitivity of $\mathbf{X}_2^{C_0}$, as we did for $\mathbf{X}_1^{C_0}$ in Eqs.(9)-(11). This allows us to weigh the suitability of each camera for tracking a particular object. In the case of multi-body structure and motion and due to the reasons mentioned in the introduction, it is quite feasible to end up with degenerate situations of objects whose motion estimation is ill-conditioned from a particular viewpoint. If we model $\mathbf{T}^{C_j}$, $\mathbf{\Omega}^{C_j}$ as being corrupted by Gaussian noise, we can view $\mathbf{X}_2^{C_0}$ as a function of n_{x_r}, n_{y_r}, n_{x_l}, n_{y_l}, $n_{\mathbf{T}^{C_j}}$, $n_{\mathbf{\Omega}^{C_j}}$, $n_{\mathbf{T}^{C_i}}$, $n_{\mathbf{\Omega}^{C_i}}$, where $n_{\mathbf{T}^{C_j}}$, $n_{\mathbf{\Omega}^{C_j}}$, $n_{\mathbf{T}^{C_i}}$, $n_{\mathbf{\Omega}^{C_i}}$ denote zero mean Gaussian noise vectors.

For each camera pair (C_j, C_i) and for each corresponding pixel pair $\mathbf{p}_j$, $\mathbf{p}_i$ in the two cameras, we obtain approximations for $\mathbf{T}^{C_j}$, $\mathbf{\Omega}^{C_j}$, $\mathbf{T}^{C_i}$, $\mathbf{\Omega}^{C_i}$ – denoted $\widehat{\mathbf{T}}^{C_j}$, $\widehat{\mathbf{\Omega}}^{C_j}$, $\widehat{\mathbf{T}}^{C_i}$, $\widehat{\mathbf{\Omega}}^{C_i}$ – and the variances of $n_{\mathbf{T}^{C_j}}$, $n_{\mathbf{\Omega}^{C_j}}$,$n_{\mathbf{T}^{C_i}}$, $n_{\mathbf{\Omega}^{C_i}}$ as follows: For each local image region centered at $\mathbf{p}_\alpha$, $\alpha \in \{i, j\}$, or for each image region containing $\mathbf{p}_\alpha$ and undergoing independent rigid motion – estimated using any popular motion segmentation algorithm – we estimate $\widehat{\mathbf{T}}^{C_\alpha}$, $\widehat{\mathbf{\Omega}}^{C_\alpha}$, the approximation of the camera's translational and rotational velocity that would lead to the motion flow observed in that particular image region using camera pair (C_j, C_i). For each such image region, we use a least squares pseudo-inverse based approach on a random subset of the estimated displacement vectors to approximate the translational and rotational velocity. We repeat this approach a number of times and the mean of the results is used as $\widehat{\mathbf{T}}^{C_\alpha}$, $\widehat{\mathbf{\Omega}}^{C_\alpha}$ and their variance provides an estimate for the variance used in the noise model described above. If we take the partial derivatives of $X_2^{C_0}$, $Y_2^{C_0}$, $Z_2^{C_0}$ with respect to the above mentioned random variables and expand around $\widehat{x}_l$, $\widehat{x}_r$, $\widehat{y}_l$, $\widehat{y}_r$, $\widehat{\mathbf{T}}^{C_j}$, $\widehat{\mathbf{\Omega}}^{C_j}$, $\widehat{\mathbf{T}}^{C_i}$, $\widehat{\mathbf{\Omega}}^{C_i}$ we obtain the desired expressions for $Var(X_2^{C_0})$, $Var(Y_2^{C_0})$ and $Var(Z_2^{C_0})$. In the appendix we list the derived expressions for these variances. The above mentioned variances are referred to

as the "top-down" information. Note that in our experiments, when modeling $Var(W_2^{C_0} - W_1^{C_0})$, we make the assumption of independence between $W_1^{C_0}$ and $W_2^{C_0}$ for all $W \in \{X, Y, Z\}$. Notice that $Var(X_2^{C_0})$, $Var(Y_2^{C_0})$ and $Var(Z_2^{C_0})$ are calculated using the derivatives of velocities $v_x^{C_j}$, $v_y^{C_j}$ and might have very different magnitudes from $Var(X_1^{C_0})$, $Var(Y_1^{C_0})$, $Var(Z_1^{C_0})$. To guarantee the numerical stability of the covariance matrices, we perform two simple modifications to the top-down variances. We first set an upper bound $maxvar$ to each of the variances by setting $Var(W_1^{C_0}) \leftarrow \min(Var(W_1^{C_0}), maxvar)$, $Var(W_2^{C_0} - W_1^{C_0}) \leftarrow \min(Var(W_2^{C_0} - W_1^{C_0}), maxvar)$. Secondly, for each $W \in \{X, Y, Z\}$ and each pixel in C_0, we scale the variances $Var(W_2^{C_0})$ acquired across all camera pairs by $a_W \triangleq c \cdot \min(\bigcup_{allpairs} Var(W_1^{C_0}))/\min(\bigcup_{allpairs} Var(W_2^{C_0}))$ for some constant c (we set $c = 2$ in our experiments).

For each pair $(C_j,\ C_i)$, we also estimate the sample variances $\overline{Var}(X_1^{C_0})$, $\overline{Var}(Y_1^{C_0})$, $\overline{Var}(Z_1^{C_0})$, $\overline{Var}(X_2^{C_0} - X_1^{C_0})$, $\overline{Var}(Y_2^{C_0} - Y_1^{C_0})$ and $\overline{Var}(Z_2^{C_0} - Z_1^{C_0})$ by using the samples in $\bigcup_{\mathbf{p} \in \Delta_{\mathbf{p}_0}} \mathbf{V}(\mathbf{p}, C_j, C_i)$ and using the mean of the vectors in $\bigcup_{\mathbf{p} \in \Delta_{\mathbf{p}_0}} \bigcup_{j=0}^{N} \bigcup_{i=1, i>j}^{N} \mathbf{V}(\mathbf{p}, C_j, C_i)$ as the sample mean. We refer to these sample variances as the "bottom-up" information. We define the final covariance matrix corresponding to each vector $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$ as a linear combination of their corresponding top-down and bottom-up variances. For each point $\mathbf{p}_0$ in camera C_0 and by using the two cameras C_j,C_i for depth estimation, we use the set $\bigcup_{\mathbf{p} \in \Delta_{\mathbf{p}_0}} \mathbf{V}(\mathbf{p}, C_j, C_i)$ in conjunction with the variances defined above, to model the covariance matrix of $\mathbf{V}(\mathbf{p}_0, C_j, C_i)$ as given by Fig.2, where $0 \leq a \leq 1$.

Assume we have n vectors $\mathbf{V}_{i(1),j(1)}$,...,$\mathbf{V}_{i(n),j(n)}$, where for each $k \in \{1, ..., n\}$, $\mathbf{V}_{i(k),j(k)}$ is the average of all the vectors in $\bigcup_{\mathbf{p} \in \Delta_{\mathbf{p}_0}} \mathbf{V}(\mathbf{p}, C_{j(k)}, C_{i(k)})$. Also with each of the vectors $\mathbf{V}_{i(k),j(k)}$ we associate a covariance matrix $\mathbf{N}_k$ indicating our confidence in this measure, as described in this section. If we ignore any potential cross-correlation between the n vectors, the *Best Linear Unbiased Estimator* (BLUE) [9] is the vector $\mathbf{X}$ that minimizes the sum of the Mahalanobis distances $\sum_{k=1}^{n} D(\mathbf{X}, \mathbf{V}_{i(k),j(k)}, \mathbf{N}_k)$. It can be shown that $\mathbf{X}^T = (\mathbf{V}_{i(1),j(1)}^T \mathbf{N}_1^{-1} + ... + \mathbf{V}_{i(n),j(n)}^T \mathbf{N}_n^{-1})(\mathbf{N}_1^{-1} + ... + \mathbf{N}_n^{-1})^{-1}$. In the next section we extend our approach to camera rigs with arbitrary intrinsic and extrinsic parameters.

5 Arbitrary Camera Rig Setup

Let us suppose that for a camera pair (C_j, C_i) with intrinsic camera parameters $(\mathbf{K}_j, \mathbf{K}_i)$ and for pixels $\mathbf{p}_j = (x_l, y_l)^T$, $\mathbf{p}_i = (x_r, y_r)^T$ imaging a common scene point $\mathbf{P} = (X_1^{C_0}, Y_1^{C_0}, Z_1^{C_0})^T$, the following equations hold:

$$\begin{pmatrix} x_l \\ y_l \end{pmatrix} = \begin{pmatrix} \frac{K_j^{1,1}(R_{j,0}^{1,1}(X_1^{C_0} - T_{0,j}^x)) + K_j^{1,2}(R_{j,0}^{1,2}(Y_1^{C_0} - T_{0,j}^y)) + K_j^{1,3}(R_{j,0}^{1,3}(Z_1^{C_0} - T_{0,j}^z))}{K_j^{3,3}(R_{j,0}^{3,3}(Z_1^{C_0} - T_{0,j}^z))} \\ \frac{K_j^{2,1}(R_{j,0}^{2,1}(X_1^{C_0} - T_{0,j}^x)) + K_j^{2,2}(R_{j,0}^{2,2}(Y_1^{C_0} - T_{0,j}^y)) + K_j^{2,3}(R_{j,0}^{2,3}(Z_1^{C_0} - T_{0,j}^z))}{K_j^{3,3}(R_{j,0}^{3,3}(Z_1^{C_0} - T_{0,j}^z))} \end{pmatrix} \quad (14)$$

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} \frac{K_i^{1,1}(R_{i,0}^{1,1}(X_1^{C_0}-T_{0,i}^x))+K_i^{1,2}(R_{i,0}^{1,2}(Y_1^{C_0}-T_{0,i}^y))+K_i^{1,3}(R_{i,0}^{1,3}(Z_1^{C_0}-T_{0,i}^z))}{K_i^{3,3}(R_{i,0}^{3,3}(Z_1^{C_0}-T_{0,i}^z))} \\ \frac{K_i^{2,1}(R_{i,0}^{2,1}(X_1^{C_0}-T_{0,i}^x))+K_i^{2,2}(R_{i,0}^{2,2}(Y_1^{C_0}-T_{0,i}^y))+K_i^{2,3}(R_{i,0}^{2,3}(Z_1^{C_0}-T_{0,i}^z))}{K_i^{3,3}(R_{i,0}^{3,3}(Z_1^{C_0}-T_{0,i}^z))} \end{pmatrix} \quad (15)$$

where $K_j^{m,n}/R_{j,0}^{m,n}$ denote the m,n^{th} entry of $\mathbf{K}_j/\mathbf{R}_{j0}$. As we did in Eqs.(3)-(8), if $|-T_{0j}^x+T_{0i}^x| \geq |-T_{0j}^y+T_{0i}^y|$, we can express $\mathbf{X}_1^{C_0}$ in terms of x_l, x_r, y_r. Conversely, if $|-T_{0j}^x+T_{0i}^x| < |-T_{0j}^y+T_{0i}^y|$ we can express $\mathbf{X}_1^{C_0}$ in terms of y_l, y_r, x_r Thus, we can define a function $\mathbf{g}(\mathbf{p}_j,\mathbf{p}_i) \triangleq \mathbf{X}_1^{C_0}$ with respect to camera C_0's coordinate system. By using $\mathbf{g}(\cdot)$ and the approach described in Section 4, we can obtain the desired variance approximations. We also need to redefine Eqs.(12)-(13) in order to obtain variance estimates for the motion error. We will only deal with the case of camera C_j, as the case of camera C_i is similar. As indicated in Section 4, $\mathbf{S}^{C_j} = -\mathbf{T}^{C_j} - \mathbf{\Omega}^{C_j} \times \mathbf{P}^{C_j}$. Then:

$$\begin{pmatrix} v_x^{C_j} \\ v_y^{C_j} \end{pmatrix} = \frac{d}{dt}\begin{pmatrix} K_j^{1,1}\frac{\mathbf{X}^{C_j}}{\mathbf{Z}^{C_j}} + K_j^{1,2}\frac{\mathbf{Y}^{C_j}}{\mathbf{Z}^{C_j}} + K_j^{1,3} \\ K_j^{2,2}\frac{\mathbf{Y}^{C_j}}{\mathbf{Z}^{C_j}} + K_j^{2,3} \end{pmatrix} = \begin{pmatrix} K_j^{1,1}\frac{d}{dt}\frac{\mathbf{X}^{C_j}}{\mathbf{Z}^{C_j}} + K_j^{1,2}\frac{d}{dt}\frac{\mathbf{Y}^{C_j}}{\mathbf{Z}^{C_j}} \\ K_j^{2,2}\frac{d}{dt}\frac{\mathbf{Y}^{C_j}}{\mathbf{Z}^{C_j}} \end{pmatrix} \quad (16)$$

assuming $K_j^{2,1}=0$, $K_j^{3,3}=1$. The derivatives are taken with respect to time t, and by using the expression for $\mathbf{S}^{C_j}$ we can express Eq.(16) in terms of $\mathbf{T}^{C_j}$ and $\mathbf{\Omega}^{C_j}$. Then the variance derivation proceeds as described in Section 4. The derivatives can be determined analytically, or via common numerical methods such as finite differences.

6 Experiments

We present our camera setup and results in Figs. 1, 3, 4. We test our approach on a number of synthetic and non-synthetic datasets. Synthetic dataset (i) consists of a $30cm \times 30cm$ planar surface on a black background (Fig. 1(c),(d)) centered at camera C_0, moving in depth, along the optical axis, by $15cm$ per frame. Synthetic dataset (ii) consists of the planar surface, rotated by 4 degrees around the optical axis between each frame. Syntethic dataset (iii) consists of a ($2cm$, $2cm$) translation of the planar surface, parallel to the image plane, between each frame. The camera setup is similar to that of Fig. 1(a). All cameras C_i, $i>0$ are radially distributed around camera C_0 at a radius of $12cm$, have a focal length of $4mm$, and have corrupting Gaussian noise added to their images. We fuse all (C_0,C_i) camera pairs and demonstrate the performance of the algorithm with an increasing object range from $300cm$ to $800cm$ by setting $a=0.5$ in Fig.2. The stereo correspondence and optical flow algorithm used is described in [13] and is available by the authors online[1]. Our results are illustrated in Fig. 3(a)-(f). We also test our algorithm using a five camera rig, as shown in Fig. 1(b). The corresponding results are presented in Fig. 4(a)-(h). In the synthetic data set we used the entire planar surface to estimate each $\mathbf{T}^{C_\alpha}$, $\mathbf{\Omega}^{C_\alpha}$

[1] http://www.cs.umd.edu/users/ogale/download/code.html

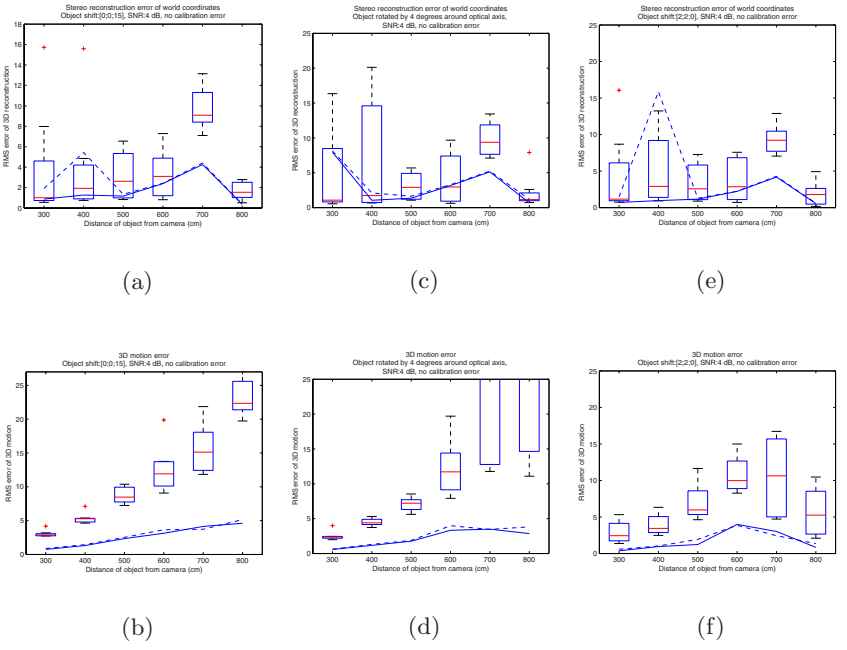


Fig. 3. (a)-(f):The results of our tests on the synthetic dataset. The x-axes represent the depth of the object in cm, and the y-axes represent the RMS error of the stereo reconstructed coordinates and the 3D motion vector (in cm). The RMS error for a particular camera pair is calculated by estimating the error across all pixels in the base camera C_0 that fall within the textured region. The solid/dashed lines correspond to the errors of our information fusion based approach using the BLUE/mean estimator, and the boxplots represent the distribution of the errors across each of the camera pairs used. The red crosses represent outliers. Note that in some figures the outliers are not displayed as they fall outside the vertical range of our error axes. (a),(b) correspond to the stereo reconstruction and 3D motion error respectively, when the planar object was translated by $15cm$ in depth along the optical axis. (c),(d) correspond to the stereo reconstruction and 3D motion error respectively, when the planar object was rotated by 4 degrees around the optical axis between frames. (e),(f) correspond to the stereo reconstruction and 3D motion error respectively when the translation occured parallel to the image plane. The object was translated by $2cm$ along the x and y axes of the world coordinate system. Notice how, even though gross outliers exist in most of the figures, the effect of those outliers on the estimated scene structure and motion is minimal in general. We also performed a number of experiments with modest errors in the external parameters' calibration and similar observations were made. The mean RMS error of the stereo reconstruction using the information fusion/mean approach for all instances of the reconstructed planar surface is $2.05 \pm 1.71/3.71 \pm 3.87$ cm respectively. The respective values for the motion data are $2.35 \pm 1.43/2.56 \pm 1.41$ cm. In both cases the improvement compared to the mean approach is statistically significant using a paired-samples t-test ($p \approx 0.01$).

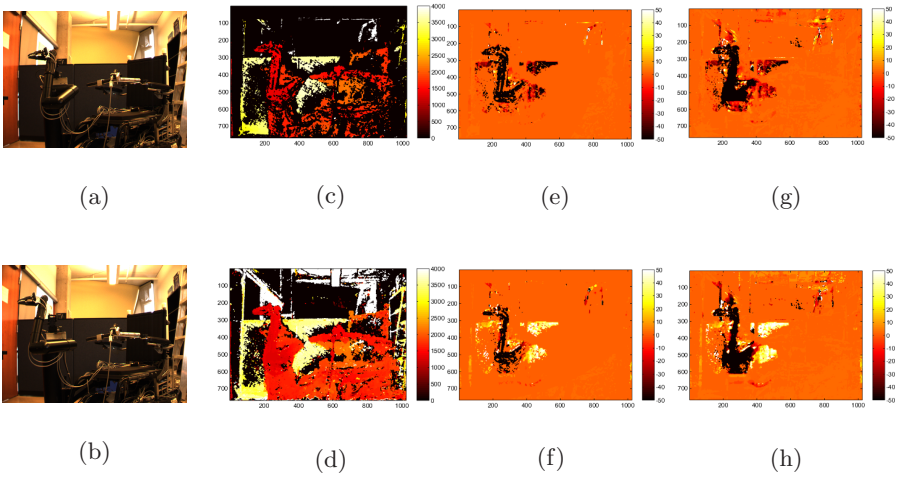


Fig. 4. (a)-(h):Experimental results from an image sequence showing a robotic wheelchair that is equipped with a 6-d.o.f. robotic arm. The robotic arm is moving diagonally towards the top left image corner. (a)-(b): Adjacent frames from the respective sequence (before correcting for radial/tangential distortions). (c): The reconstructed scene depth using a single pair of cameras to reconstruct each scene. Image regions in black denote pixels where the left-right consistency constraint could not be enforced. (d): The reconstructed scene depth of frames (a),(b) using the five camera rig setup shown in Fig. 1 in conjunction with our information fusion based algorithm. The colorbar depths of (c),(d) represent *mm*. Notice the significant decrease in occlusions. (e)-(f): Image motion of the sequence after projecting the estimated 3D motion on the image plane using a single camera pair in conjunction with our information fusion based algorithm. Image motion is represented in pixel units. (g)-(h): Image motion of the respective image sequences after using the five camera rig setup shown in Fig. 1 in conjunction with our information fusion based algorithm. (e),(g): The image motion component parallel to the horizontal axis and (f),(h): The image motion component parallel to the vertical axis.

(simulating perfect motion segmentation) and in the non-synthetic data set we used 21×21 pixel regions centered at the current pixel of interest.

From Fig. 3, we observe that the multi-camera approach provides a significant decrease of the RMS error in both structure and motion estimation compared to the errors achieved using the stereo camera pairs. In almost all cases the quality of the results surpasses that obtained by any one of the camera pairs. As indicated in the caption of Fig. 3 the BLUE estimator provides better results than the results obtained by the mean vector across all cameras and their neighborhoods. For both the structure and motion data the improvement is judged statistically significant. In Fig. 3(a),(b) where the plane is moving along the z-axis and we are dealing with ill-conditioned motion estimation near the focus of expansion, we observe significant improvements. In Fig. 3(c)-(d) we present results after a pure rotation of the plane around the optical axis. It is interesting

to notice however, that for depths $700cm$, $800cm$ the optical flow estimation algorithm we used performs poorly on about half of our camera pairs, thus, resulting in a big RMS error, as the boxplots show. Our algorithm is capable of ignoring the erroneous data and gives us a relatively robust estimate of the 3D motion. This indicates that if we are using a multi-camera rig with cameras that break down quite often and provide gross outliers, our algorithm remains reliable. We observe that the mean estimator is severely affected by outliers at various depths, while the information fusion based algorithm is more robust in the presence of outliers.

In Fig. 4(a)-(h) we compare the performance of our algorithm using a two camera rig versus a five camera rig (Fig. 1(b)). The five camera rig consists of two Point Grey Research Bumblebee stereo cameras and a Point Grey Research Flea camera. The coordinate system of the Flea camera is used as our basis coordinate system and represents camera C_0. The two camera rig is represented using the Flea camera and one of the four Bumblebee monocular cameras. The robotic wheelchair presented in Fig. 3 is equipped with a 6-d.o.f. robotic arm providing a number of independent rigid motions to test our algorithm. We used the algorithm described in [13] to determine the correspondences. We notice a dramatic increase in the number of pixels satisfying the left-right consistency constraint as the number of cameras in our rig increases.

7 Conclusion

We presented an algorithm for multi-camera and multi-body structure and motion. The algorithm combines top-down and bottom-up knowledge on scene structure and motion to model the respective uncertainties. An information fusion based algorithm uses these uncertainties to obtain competitive results demonstrating that our algorithm performs robustly in situations where a number of camera pairs provide severely degraded results. Such situations arise in practice due to hardware failures and poor environmental conditions. We are currently investigating the use of other information fusion algorithms for solving this problem [10]. Some potential application areas in future research are dynamic scene interpretation, vision based simultaneous localization and mapping (SLAM) and dynamic rendering.

Acknowledgments. JKT holds the Canada Research Chair in Computational Vision and gratefully acknowledges its financial support. AA would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for its financial support through the PGS-D scholarship program.

References

1. Schindler, K., Suter, D.: Two-view multibody structure and motion. In: Proc. Conf. Computer Vision and Pattern Recognition (2005)
2. Zhang, W., Kosecka, J.: Nonparametric estimation of multiple structures with outliers. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, Springer, Heidelberg (2006)

3. Hanna, K.J., Okamoto, N.E.: Combining stereo and motion analysis for direct estimation of scene structure. In: Proc. Int. Conf. on Computer Vision (1993)
4. Richards, W.: Structure from stereo and motion. Journal of the Optical Society of America A. 2(2), 343–349 (1985)
5. Waxman, A., Duncan, J.: Binocular image flows. IEEE Trans. Patt. Anal. Mach. Intell. 8(6), 715–729 (1986)
6. Grosso, E., Tistarelli, M.: Active dynamic stereo vision. IEEE Trans. Patt. Anal. Mach. Intell. 17(11), 1117–1128 (1995)
7. Mandelbaum, R., Salgian, G., Sawhney, H.: Correlation-based estimation of ego-motion and structure from motion and stereo. In: Proc. Int. Conf. on Computer Vision (1999)
8. Zhang, Y., Kambhamettu, C.: Integrated 3D scene flow and structure recovery from multiview image sequences. In: IEEE Conf. Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2000)
9. Singh, A., Allen, P.: Image-flow computation: An estimation-theoretic framework and a unified perspective. CVGIP: Image Understanding 56(2), 152–177 (1992)
10. Comaniciu, D.: Nonparametric information fusion for motion estimation. In: IEEE Conf. Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2003)
11. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Patt. Anal. Mach. Intell. 24, 603–619 (2002)
12. Neumann, J., Fermuller, C., Aloimonos, Y.: A hierarchy of cameras for 3D photography. Computer Vision and Image Understanding 96, 274–293 (2004)
13. Ogale, A.S., Aloimonos, Y.: A roadmap to the integration of early visual modules. International Journal of Computer Vision: Special Issue on Early Cognitive Vision 72(1), 9–25 (2007)

Appendix

In this section we derive expressions for $Var(X_2^{C_0})$, $Var(Y_2^{C_0})$, $Var(Z_2^{C_0})$ for the case $|-T_{0j}^x + T_{0i}^x| \geq |-T_{0j}^y + T_{0i}^y|$. From Eqs.(9)-(11) we can derive the corresponding expressions for $Var(X_2^{C_0})$, $Var(Y_2^{C_0})$, $Var(Z_2^{C_0})$: $Var(X_2^{C_0}) \approx \frac{((-T_{0j}^x+T_{0i}^x)\widehat{x}_l')^2}{(\widehat{x}_l'-\widehat{x}_r')^4}Var(x_r') + \frac{((T_{0j}^x-T_{0i}^x)\widehat{x}_r')^2}{(\widehat{x}_l'-\widehat{x}_r')^4}Var(x_l')$, $Var(Y_2^{C_0}) \approx \frac{(-T_{0j}^x+T_{0i}^x)^2}{(\widehat{x}_l'-\widehat{x}_r')^2}Var(y_r') + \frac{((-T_{0j}^x+T_{0i}^x)\widehat{y}_r')^2}{(\widehat{x}_l'-\widehat{x}_r')^4}Var(x_r') + \frac{((T_{0j}^x-T_{0i}^x)\widehat{y}_r')^2}{(\widehat{x}_l'-\widehat{x}_r')^4}Var(x_l')$, $Var(Z_2^{C_0}) \approx \frac{((-T_{0j}^x+T_{0i}^x)f)^2}{(\widehat{x}_l'-\widehat{x}_r')^4}Var(x_r') + \frac{((T_{0j}^x-T_{0i}^x)f)^2}{(\widehat{x}_l'-\widehat{x}_r')^4}Var(x_l')$. We have previously noted that $(x_r', y_r') = (x_r, y_r) + (v_x^{C_i}, v_y^{C_i})$, $(x_l', y_l') = (x_l, y_l) + (v_x^{C_j}, v_y^{C_j})$. If we let $a \in \{x, y\}$, $b \in \{r, l\}$ and $k = i/j$ if $b = r/l$ we obtain the following approximations for $Var(x_r')$, $Var(x_l')$, $Var(y_r')$: $Var(a_b') \approx (\frac{\partial a_b'}{\partial x_r})^2 Var(x_r) + (\frac{\partial a_b'}{\partial x_l})^2 Var(x_l) + (\frac{\partial a_b'}{\partial y_r})^2 Var(y_r) + (\frac{\partial a_b'}{\partial \Omega_x^{C_k}})^2 Var(\Omega_x^{C_k}) + (\frac{\partial a_b'}{\partial \Omega_y^{C_k}})^2 Var(\Omega_y^{C_k}) + (\frac{\partial a_b'}{\partial \Omega_z^{C_k}})^2 Var(\Omega_z^{C_k}) + (\frac{\partial a_b'}{\partial T_a^{C_k}})^2 Var(T_a^{C_k}) + (\frac{\partial a_b'}{\partial T_z^{C_k}})^2 Var(T_z^{C_k})$. By using Eqs. (12)-(13) we can derive the expressions for the partial derivatives. By expanding these expressions for the partial derivatives around $\widehat{x}_l$, $\widehat{x}_r$, $\widehat{y}_l$, $\widehat{y}_r$, $\widehat{\mathbf{T}}^{C_i}$, $\widehat{\mathbf{\Omega}}^{C_i}$, $\widehat{\mathbf{T}}^{C_j}$, $\widehat{\mathbf{\Omega}}^{C_j}$, as appropriate, we can obtain the desired expressions.