Robust Motion Estimation and Structure Recovery from Endoscopic Image Sequences With an Adaptive Scale Kernel Consensus Estimator

Hanzi Wang[1], Daniel Mirota[1], Masaru Ishii[2], Gregory D. Hager[1]
[1]Computer Science Department, Johns Hopkins University, Baltimore, MD, 21218
[2]Department of Otolaryngology-Head and Neck Surgery, Johns Hopkins Bayview Medical Center, Baltimore, MD, 21224

Abstract

To correctly estimate the camera motion parameters and reconstruct the structure of the surrounding tissues from endoscopic image sequences, we need not only to deal with outliers (e.g., mismatches), which may involve more than 50% of the data, but also to accurately distinguish inliers (correct matches) from outliers. In this paper, we propose a new robust estimator, Adaptive Scale Kernel Consensus (ASKC), which can tolerate more than 50 percent outliers while automatically estimating the scale of inliers. With ASKC, we develop a reliable feature tracking algorithm. This, in turn, allows us to develop a complete system for estimating endoscopic camera motion and reconstructing anatomical structures from endoscopic image sequences. Preliminary experiments on endoscopic sinus imagery have achieved promising results.

1. Introduction

Endoscopic anterior skull based surgery has the potential to significantly reduce patient morbidities associated with operating on the undersurface of the front third of the brain. Of the anterior skull based approaches, the endoscopic transnasal approach to the sphenoid sinus, which is a small structure and is surrounded by major blood vessels, is most mature and utilized. Surgery in this area is technically challenging and requires an accurate appreciation of the patient's anatomy. Failure to correctly interpret a patient's anatomy can result in catastrophic outcomes.

Traditional navigation systems [1, 2] rely on an external tracking system and fiducial or anatomical landmarks for registration. These systems have many fundamental limitations [3] in terms of accuracy and flexibility with the workflow in the operating room. Another approach to surgical navigation systems is to directly register endoscopic images to the patient anatomy [3, 4, 5]. However, this is nontrivial because endoscopic images involve a number of challenges such as low texture, abundant specularities and extreme illumination changes from the light source attached to the endoscope, and blurring from the movement of the endoscope. These difficulties may result in a number of outliers (including both feature localization errors and mismatches) which can not be easily handled by traditional robust statistical methods such as LMedS [6] and RANSAC [7].

To recover the surface structure of surrounding tissues and further to register this information against a preoperative volumetric image (such as CT or MRI), we need not only to accurately estimate the motion of an endoscopic camera from endoscopic image sequences but also to correctly distinguish inliers from outliers. This can be realized by employing advanced techniques from robust statistics.

1.1. Background on Robust Statistics

Various robust estimation techniques have appeared in the literature during the last decades. Maximum-likelihood estimators (M-estimators) [8] minimize the sum of symmetric, positive-definite functions of residuals with a unique minimum at zero. The Least Median of Squares (LMedS) estimator [6] minimizes the median of squared residuals. However, it has been shown that the breakdown points of M-estimators and LMedS are no more than 50%,. Chen and Meer [9] modified the cost function of the M-estimators to create a projection based M-estimator (pbM-estimator). The authors of [10] and [11] further improved the performance of the pbM-estimator by modifying its objective function. All of these modifications are concentrated on the projection pursuit paradigm [9]. RANSAC [7] and its variant MSAC [12] can resist the influence of more than 50% outliers. However, the performance of RANSAC and MSAC depends on a user-specified *error tolerance* (or the scale of inliers), which is not known *a priori* in many practical environments. MUSE [13], MINPRAN [14], ALKS [15], RESC [16] and ASSC [17] can deal with more than 50% outliers. However, MUSE needs a lookup table for the scale estimator correction. MINPRAN and ALKS are computationally expensive and cannot effectively deal with multiple structures with extreme outliers. RESC needs the user to tune many parameters. ASSC weights all inliers equally, thus it is less efficient.

The main contributions in this paper are: (1) we employ kernel density estimation techniques to create a new robust estimator, Adaptive Scale Kernel Consensus (ASKC)

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

which can simultaneously estimate both the model parameters and the scale of inliers. ASKC can be treated as a generalized form of RANSAC [7] and ASSC [17] (see Section 2 for details); (2) we propose an effective feature tracking approach; and, (3) we integrate the robust ASKC estimator and the feature tracking approach into a complete system for estimating endoscopic camera motion and performing surface reconstruction of sinus anatomy from endoscopic image sequences. Experiments show our system has achieved promising results.

2. The Adaptive Scale Kernel Consensus (ASKC) estimator

2.1. The kernel density estimation

Given a model parameter estimate $\hat{\boldsymbol{\theta}}$, the fixed bandwidth kernel density estimate with the kernel $K(.)$ and a bandwidth h can be written as [18]:

$$\hat{f}_{\hat{\boldsymbol{\theta}}}(r) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{r - r_{i,\hat{\boldsymbol{\theta}}}}{h}\right) \tag{1}$$

where $\{r_{i,\hat{\boldsymbol{\theta}}}\}_{i=1,\dots,n}$ is the residuals and n is the number of data points.

An alternative is to select a different bandwidth $h = h(\hat{\boldsymbol{\theta}}) \equiv h_{\hat{\boldsymbol{\theta}}}$ for each value of $\hat{\boldsymbol{\theta}}$. The variable bandwidth kernel density estimate can be written as:

$$\hat{f}_{\hat{\boldsymbol{\theta}}}(r) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_{\hat{\boldsymbol{\theta}}}} K\left(\frac{r - r_{i,\hat{\boldsymbol{\theta}}}}{h_{\hat{\boldsymbol{\theta}}}}\right) \tag{2}$$

In this paper, we consider two popular kernels, the Epanechnikov kernel $K_E(r)$ and the normal kernel $K_N(r)$:

$$K_E(r) = \begin{cases} \frac{3}{4}(1 - \|r\|^2) & \|r\| \le 1 \\ 0 & \|r\| > 1 \end{cases}, \text{ with } k_E(r) = \begin{cases} 1 - r & 0 \le r \le 1 \\ 0 & r > 1 \end{cases} \tag{3}$$

$$K_N(r) = \frac{1}{(2\pi)^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\|r\|^2\right), \text{ with } k_N(r) = \exp\left(-\frac{r}{2}\right) \; r \ge 0 \tag{4}$$

The Epanechnikov kernel yields the minimum asymptotic mean integrated square error (AMISE) measure. However, the Epanechnikov profile is not differentiable at the boundary. As pointed out by the authors of [19], the path of the mean shift procedure employing a normal kernel follows a smooth trajectory.

Although we are interested in investigating the properties of ASKC with the Epanechnikov kernel (termed as ASKC1) and the normal kernel (termed as ASKC2) in this paper, our method can employ arbitrary kernels.

2.2. Estimating the bandwidth/the scale of inliers

As noted above, the bandwidth h is a crucial parameter in kernel density estimation. An over-smoothed bandwidth selector with the scale estimate $\hat{\sigma}_{\boldsymbol{\theta}}$ is suggested in [20].

$$\hat{h}_{\boldsymbol{\theta}} = \left[\frac{243\int_{-1}^{1} K(r)^2 dr}{35n\int_{-1}^{1} r^2 K(r) dr}\right]^{1/5} \hat{\sigma}_{\boldsymbol{\theta}} \tag{5}$$

It is recommended that the bandwidth is set as $c_h \hat{h}_{\boldsymbol{\theta}}$ ($0 < c_h < 1$) to avoid over-smoothing ([20], p.62).

Robust scale estimators (such as the median [6], the MAD [9], or the robust k scale estimator [15]) can be employed to yield a scale estimate. The authors of [17] have shown that TSSE, which employs the mean shift and the mean shift valley procedure, can effectively estimate the scale under multiple modes. The valley closest to zero detected by the mean shift valley procedure on the ordered absolute residuals can be a sensitive point to determine the inliers/outliers dichotomy.

In our method, we use a procedure similar to TSSE. We use a robust k scale estimator (the k value is set to 0.1 so that at least 10 percent of the data points are included in the shortest window) to yield an initial scale estimate. In [17], the authors use the Epanechnikov kernel for both the mean shift and the mean shift valley approaches. This can be different in our case when we use different kernels.

Figure 1 shows the procedure of the TSSE-like scale estimator. When the model parameter estimate $\hat{\boldsymbol{\theta}}$ is incorrect, the detected valley is far away from the origin and the kernel density estimate at the origin is lower. In contrast, when the $\hat{\boldsymbol{\theta}}$ estimate is correct, the residual value corresponding to the detected valley is closer to the origin and the kernel density at the origin is higher.

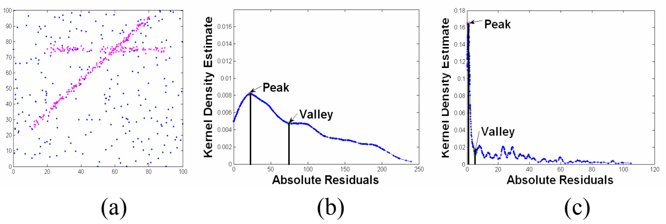

Figure 1: Simultaneous scale estimate of inliers and outlier detection. (a). The detected peaks and valleys with incorrect model parameters (b) and correct model parameters (c).

2.3. The ASKC estimator

We assume that inliers involve a relative majority of the data, i.e., inliers may involve less than 50% of the data but they involve more data points than structured pseudo-outliers. Our method considers the kernel density at the origin point as its objective function. Given a set of residuals $\{r_{i,\hat{\boldsymbol{\theta}}}\}_{i=1,\dots,n}$ subject to $\hat{\boldsymbol{\theta}}$, the objective function of ASKC is:

$$\hat{f}_{\hat{\theta}}(r_0^*) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_{\hat{\theta}}}K\left(\frac{r_{i,\hat{\theta}}}{h_{\hat{\theta}}}\right) \tag{6}$$

The ASKC estimator can be written as:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\hat{\theta}} \hat{f}_{\hat{\theta}}(r_0^*) = \arg\max_{\hat{\theta}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_{\hat{\theta}}}K\left(\frac{r_{i,\hat{\theta}}}{h_{\hat{\theta}}}\right) \tag{7}$$

If we consider the RANSAC estimator [7]:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\hat{\theta}} \hat{\mathsf{n}}_{\hat{\theta}} \tag{8}$$

and the ASSC estimator [17]:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\hat{\theta}} \left(\hat{\mathsf{n}}_{\hat{\theta}} / \hat{S}_{\hat{\theta}}\right) \tag{9}$$

where $\hat{\mathsf{n}}_{\hat{\theta}}$ is the number of inliers within an error tolerance (for RANSAC) or the scale of inliers (for ASSC) and $\hat{S}_{\hat{\theta}}$ is the estimated scale of inliers given a set of residuals relative to $\hat{\boldsymbol{\theta}}$, we can see that RANSAC and ASSC are actually special cases of ASKC with the uniform kernel:

$$K_U(r) = \begin{cases} C & \|r\| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where C is a normalization constant.

More specifically, RANSAC is one case of ASKC with the uniform kernel and a fixed bandwidth, and ASSC is another case of ASKC with the uniform kernel and a variable bandwidth. However, the efficiency of the uniform kernel is low as it weights all inliers equally.

To get the solution of equation (7), we need to sample a set of candidates. We can employ a random sampling scheme [6, 7], or a guided sampling technique [21].

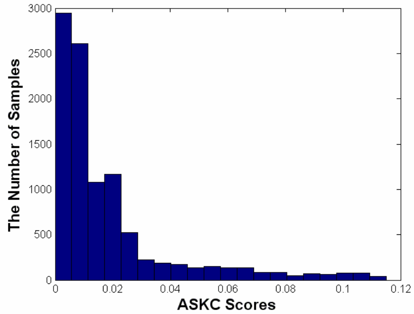

Figure 2: The histogram of ASKC scores of 10000 random samples.

By way of illustration,

Figure 2 shows a histogram of ASKC scores (equation 6) computed from 10000 random samples from the data in Figure 1 (a). It shows that most of the samples have small score values which means that the samples are most likely contaminated with outliers. To improve the computational efficiency, it is not necessary to run the TSSE-like procedure for all samples. We only run the TSSE-like procedure for the samples with high ASKC scores. With this strategy, only about 7% of the 10000 samples are further processed with the TSSE-like procedure.

2.4. The ASKC procedure

The procedure of the ASKC estimator is shown in Figure 3. In step 3, the purpose using data other than the sample candidate is to avoid extreme low scale estimates. In step 5, an additional TSSE-like procedure may refine the scale estimate for heavily contaminated data.

Step 1: Select a sample candidate M_I

Step 2: Estimate the model parameters $\hat{\boldsymbol{\theta}}_I$ from M_I.

Step 3: Derive the residuals of the data points other than M_I.

Step 4: Get an initial scale by the robust k scale estimator. Calculate a coarse estimate of the ASKC score by (6). If it is larger than a certain value (say, half of the largest score so far), go to the next step. Otherwise, go to step 1.

Step 5: Run the TSSE-like procedure to estimate the inliers' scale and the bandwidth. If the valley is valid (i.e., the ratio of the kernel density at the peak and valley is large enough), go to the next step. Otherwise, go to step 1.

Step 6: Compute the ASKC score by (6).

Step 7: If the computed score is larger the current largest score, update the largest score and save the estimated parameters and the inliers' scale. Otherwise, go to step 1.

Step 8: Run step 1 to 7 many times, output the parameters and the inliers' scale corresponding to the largest score.

Figure 3. The procedure of the ASKC estimator

2.5. The performance of ASKC

In this subsection, we test the performance of the ASKC estimator employing the Epanechnikov kernel (ASKC1) and the normal kernel (ASKC2) and we compare the performance of ASKC1/ASKC2 with those of several other robust estimators (ASSC, RESC, and LMedS).

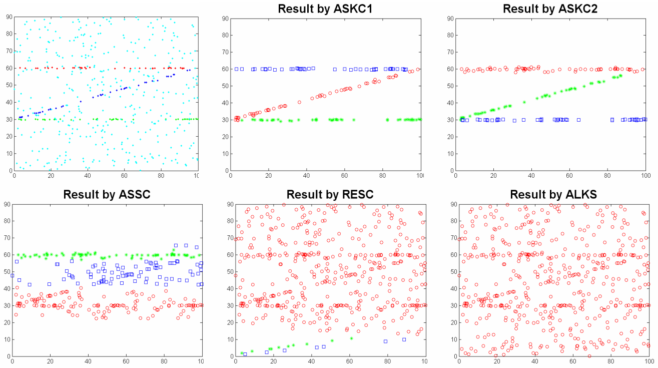

Figure 4: Lines extracted by the robust estimators.

In the first example, we generate three lines (each line contains 40 data points) and 380 random outliers. We apply the robust estimators to sequentially extract all three lines. As shown in Figure 4, both RESC and LMedS fail to extract any line. ASSC extracts one line but fails in two. ASKC1/ASKC2 successfully extract all three lines.

In the second example, we use 3D data. There are 500 data points including 4 planes (each contains 50 data points) and 300 randomly distributed outliers. Likewise, we sequentially extract all planes with the robust estimators.

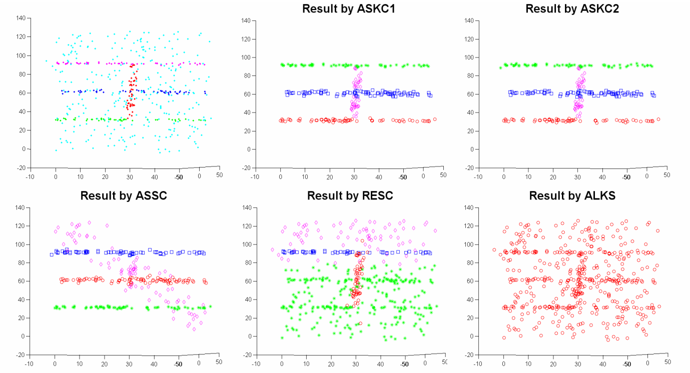

Figure 5: Planes extracted by the robust estimators.

Figure 5 shows that ASKC1/ASKC2 correctly extract all planes. In contrast, RESC succeeds on 2 planes while ASSC succeeds on 3. LMedS fails to extract any plane.

3. Structure and motion recovery with ASKC

We now consider a camera observing a 3D point $\mathbf{X}$ on a surface from two camera positions, the point $\mathbf{X}$ will project to two image locations $\mathbf{x}_1 = (u_1, v_1, 1)^{\mathrm{T}}$ and $\mathbf{x}_2 = (u_2, v_2, 1)^{\mathrm{T}}$. The following condition holds [22]:

$$\mathbf{x}_2^{\mathrm{T}}\mathcal{K}_2^{-\mathrm{T}}[\boldsymbol{\Gamma}]_{\mathrm{x}}\mathbf{R}\mathcal{K}_1^{-1}\mathbf{x}_1 = 0 \qquad (11)$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are respectively the intrinsic camera matrices corresponding to the two images. $[\boldsymbol{\Gamma}]_{\mathrm{x}}$ is the skew matrix of the translation vector $\boldsymbol{\Gamma}$ and $\mathbf{R}$ is the rotation matrix.

The essential matrix $\mathbf{E}=[\boldsymbol{\Gamma}]_{\mathrm{x}}\mathbf{R}$ encodes the motion information of the camera. Given the camera matrices, the essential matrix $\mathbf{E}$ can be estimated using the nonlinear five-point algorithm [23]. The camera motion ($\mathbf{R}$ and $\boldsymbol{\Gamma}$) can be recovered from $\mathbf{E}$ by the Singular Value Decomposition (SVD) approach [22] and the translation can only be estimated up to a scale factor (we use $\tilde{\boldsymbol{\Gamma}}$ to represent the estimated scaled translation vector and $\boldsymbol{\Gamma} = \lambda\tilde{\boldsymbol{\Gamma}}$). The scale λ can be recovered by registering the reconstructed 3D model to a pre-operative CT scan [3].

3.1. Feature detection and matching

To estimate the motion parameters of a camera between a pair of images, we need to robustly detect features in the images and then match these features. We employ the SIFT feature detector [24] in our method. To find the matches between feature points, we use the SVD matching algorithm [25]. The reason that we employ the SVD matching algorithm rather than the SIFT matching function [24] is that we have found that the SVD matching approach can return more correct matches.

Figure 6 shows one example where ASKC can correctly estimate the epipolar geometry and the scale of inliers, and select most correct matches even when outlier percentage is larger than 70%.

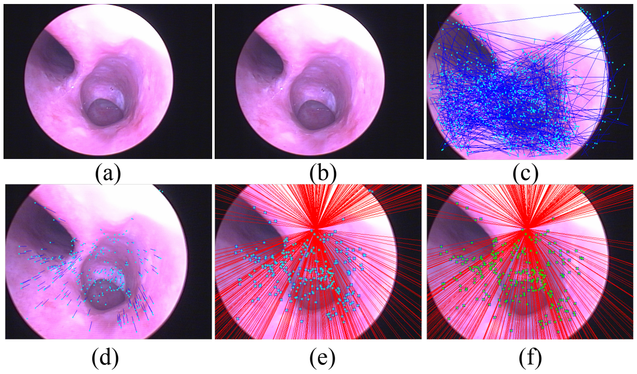

Figure 6: (a) and (b) a pair of original sinus endoscopic images; (c) the matches obtained by the SVD-matching algorithm; (d) the matches selected by the ASKC estimator on the left undistorted image; (e) and (f) the recovered epipolar geometry.

3.2. SIFT feature tracking

```
Step 0: Initialize the SIFT feature list ( $\mathcal{L}^{F=1}$ ) and the
        chain matrix ( $\mathcal{T}^{F=1}$ ).
For $F = 2,\ldots N$
    Step 1: Compute a set of potential matches between
            frame $F$ and frame $F$-1
    Step 2: Select the matches which are consistent to the
            relative majority of data by the robust ASKC
            estimator
    Step 3: Maintain the SIFT feature list $\mathcal{L}^{F}$
    Step 4: Maintain the chain matrix $\mathcal{T}^{F}$
End
    Step 5: Output the trajectories of the tracked SIFT
features in $\mathcal{T}^{F}$
```

Figure 7. Overview of the SIFT feature tracking algorithm.

We need to track SIFT features through a video sequence to derive the projection matrix at each frame and further recover the structure. To track a set of SIFT features $\{S_i\}_{i=1,\ldots,m'}$, we maintain a feature list $\mathcal{L} = \{l_i\}_{i=1,\ldots,m'} = \{u_i, v_i, w_i, s_i\}_{i=1,\ldots,m'}$ which records, for each frame, the feature locations (u_i, v_i), the number of the frames that a feature is continuously tracked (w_i), and the status (s_i) of each feature. For the F^{th} frame, we also maintain a chain matrix $\mathcal{T}^F = \{\mathcal{T}_i^F\}_{i=1,\ldots m'}$, where $\mathcal{T}_i^F = \{(u_i^f, v_i^f)\}_{f=F-w_i,\ldots,F}$, to record all past locations (trajectories) of each tracked feature. The SIFT features at the frames F and F-1 are robustly matched and we can obtain newly selected matches $\{(\mathcal{S}_j^{F\text{-}1}, \mathcal{S}_j^F)\}_{j=1,\ldots,n}$. The status of each SIFT feature $\mathcal{S}_j^F$ may have three possibilities: (1) "*active*", (2) "*inactive*", and (3) "*new*".

(1) If a feature $\mathcal{S}_j^{F\text{-}1}$ of the match $(\mathcal{S}_j^{F\text{-}1}, \mathcal{S}_j^F)$ has a correspondence with $\ell_i^{F\text{-}1} = (u_i^{F\text{-}1}, v_i^{F\text{-}1}, w_i^{F\text{-}1}, s_i^{F\text{-}1})$ in the feature list $\mathrm{L}^{F\text{-}1}$, the status of $\mathcal{S}_j^F$ in the list L^F is labeled as "*active*" and s_i^F =1. In this case, the location (u_i^F, v_i^F) of ℓ_i^F is updated by the image coordinates $(u_{\mathcal{S}_j}^F, v_{\mathcal{S}_j}^F)$ of $\mathcal{S}_j^F$, and $w_i^F = w_i^{F\text{-}1} + 1$ .C^F is updated with $\mathcal{C}_i^F = \mathcal{C}_i^{F\text{-}1} \bigcup (u_{\mathcal{S}_j}^F, v_{\mathcal{S}_j}^F)$.

(2) If there is no correspondence between $\ell_i^{F\text{-}1}$ and $\{\mathcal{S}_j^{F\text{-}1}\}_{j=1,\ldots,n}$, s_i^F is labeled as "*inactive*" and $s_i^F = -1$. We set w_i^F =0. When the number of times that the value of w_i^F continuously remains zero is larger than a threshold, we assume the feature is out of view and it is removed from the list $\mathcal{L}^F$ and the chain matrix $\mathcal{C}^F$.

(3) If there is no correspondence between $\mathcal{S}_j^{F\text{-}1}$ and $\{\ell_i^{F\text{-}1}\}_{i=1,\ldots,m'}$, we add the new feature $\mathcal{S}_j^F$ to L^F and s_i^F is labeled as "*new*". We set $(u_{m'+1}^F, v_{m'+1}^F) = (u_{S_j}^F, v_{S_j}^F)$, $w_{m'+1}^F$=1 and $s_{m'+1}^F = 0$. C^F is initialized with $\mathcal{C}_{m'+1}^F = (u_{S_j}^F, v_{S_j}^F)$ and the value of *m'* is update (m'=m'+1).

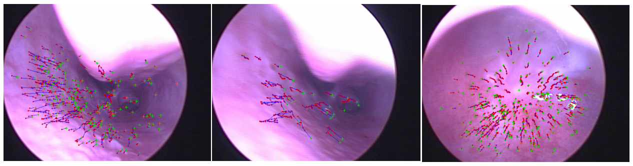

Figure 8. The trajectories of the tracked SIFT features.

Figure 7 summarizes the procedures of the SIFT feature tracking algorithm. The trajectories of the tracked SIFT features on an endoscopic sinus image sequence are shown in Figure 8. We can see that most significant SIFT features are tracked. Even when the image is seriously blurred, there are still sufficient SIFT features tracked.

3.3. Structure recovery from endoscopic images

We assume a calibrated camera is used and the optical distortion is removed by undistortion [27].

Let $\mathbf{X}_i = (X_i, Y_i, Z_i, 1)^{\mathrm{T}}$ be a 3D point in the world system. The 3D point $\mathbf{X}_i$ is projected to an image point $\mathbf{x}_i^F$ at the frame F by a 3x4 projection matrix $\mathbf{P}_F$. We have:

$$\mathbf{x}_i^F = \mathbf{P}_F \mathbf{X}_i \tag{12}$$

Let the first camera be at the center of the world coordinate, we have:

$$\mathbf{P}_1 = \mathcal{K}[\mathbf{I}|0] \text{ and } \mathbf{P}_F = \mathcal{K}[{}^1\mathbf{R}_F|\,{}^1\mathbf{\Gamma}_F] \tag{13}$$

where ${}^1\mathbf{R}_F$ and ${}^1\mathbf{\Gamma}_F$ are respectively the rotation and the translation of the camera at the Fth frame relative to those of the camera at the first frame. Note: The camera matrix $\mathcal{K}$ of the endoscope remains fixed throughout the sequence.

Step 1: Extract the SIFT features by the SIFT detector [24].
Step 2: Initialize the structure
step 2.1: Choose two initial frames and detect potential matches by the SVD-matching algorithm [25].
step 2.2: Select the correct matches by ASKC and calculate the motion parameters of the endoscopic camera.
step 2.3: Initialize the structure $\{\mathbf{X}_i\}$ by triangulation [22].
Step 3: Maintain the structure.
step 3.1: Obtain matches between the frames F and F-1.
step 3.2: Track the SIFT features using the feature tracking algorithm proposed in subsection 3.2.
step 3.3: Compute the projection matrix $\mathbf{P}_F$ by the method proposed in subsection 3.3.
step 3.4: Compute the 3D points corresponding to the new SIFT features and add them to the 3D structure.
step 3.5: Refine the existing 3D points that correspond to the tracked SIFT features
step 3.6: Repeat step 3.1 to 3.5 until the last frame.
Step 4: Output the reconstructed 3D structure $\{\mathbf{X}_i\}_{i=1,\ldots,\mathrm{M}}$ and the projection matrices $\{\mathbf{P}_i\}_{i=1,\ldots,N}$.

Figure 9. Overview of the reconstruction algorithm.

At the beginning, the structure is initialized using two selected frames through triangulation [22].

For a new frame F, we relate it to its previous frame $F-1$. Assuming we have known $\mathbf{P}_{F-1} = \mathcal{K}[{}^1\mathbf{R}_{F-1}|\,{}^1\mathbf{\Gamma}_{F-1}]$ at the frame F-1, $\mathbf{P}_F$ can be written as:

$$\mathbf{P}_F = \mathcal{K}[{}^{F-1}\mathbf{R}_F\,{}^1\mathbf{R}_{F-1}|{}^{F-1}\mathbf{R}_F\,{}^1\mathbf{\Gamma}_{F-1} + \lambda_F\,{}^{F-1}\tilde{\mathbf{\Gamma}}_F] \tag{14}$$

Let:

$$\mathcal{C}_F = {}^{F-1}\mathbf{R}_F\,{}^1\mathbf{\Gamma}_{F-1}, \mathcal{K} \equiv [k_1\ k_2\ k_3]^{\mathrm{T}} \text{and } \mathbf{P}_F = \begin{bmatrix} p_{11}^F & p_{12}^F & p_{13}^F \\ p_{21}^F & p_{22}^F & p_{23}^F \\ p_{31}^F & p_{32}^F & p_{33}^F \end{bmatrix} \tag{15}$$

From equations (12), (14) and (15), we can derive:

$$\begin{aligned} u_i^F &= \frac{p_{11}^F X_i + p_{12}^F Y_i + p_{13}^F Z_i + k_1 \mathcal{C}_F + \lambda_F(i) k_1\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F}{p_{31}^F X_i + p_{32}^F Y_i + p_{33}^F Z_i + k_3 \mathcal{C}_F + \lambda_F(i) k_3\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F} \\ v_i^F &= \frac{p_{21}^F X_i + p_{22}^F Y_i + p_{23}^F Z_i + k_2 \mathcal{C}_F + \lambda_F(i) k_2\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F}{p_{31}^F X_i + p_{32}^F Y_i + p_{33}^F Z_i + k_3 \mathcal{C}_F + \lambda_F(i) k_3\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F} \end{aligned} \tag{16}$$

If we define the following:

$$\begin{aligned} \mathrm{A}_i^F &= \begin{pmatrix} u_i^F k_3\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F - k_1\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F \\ v_i^F k_3\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F - k_2\ {}^{F-1}\tilde{\mathbf{\Gamma}}_F \end{pmatrix} \\ \mathrm{B}_i^F &= \begin{pmatrix} p_{11}^F X_i + p_{12}^F Y_i + p_{13}^F Z_i + k_1 \mathcal{C}_F - u_i^F p_{31}^F X_i - u_i^F p_{32}^F Y_i - u_i^F p_{33}^F Z_i - u_i^F k_3 \mathcal{C}_F \\ p_{21}^F X_i + p_{22}^F Y_i + p_{23}^F Z_i + k_2 \mathcal{C}_F - v_i^F p_{31}^F X_i - v_i^F p_{32}^F Y_i - v_i^F p_{33}^F Z_i - v_i^F k_3 \mathcal{C}_F \end{pmatrix} \end{aligned} \tag{17}$$

We can calculate the scale value λ_F by:

$$\lambda_F(i) = (A_{F,i}^{T} A_{F,i})^{-1} A_{F,i}^{T} B_{F,i} \tag{18}$$

However, as both the feature's location $\{\mathbf{x}_i\}$ and the 3D points may be in error, we estimate λ_F in a robust way:

$$\hat{\lambda}_F = \underset{\lambda_F(i)}{\operatorname{argmax}} \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h_j} K\left(\frac{r_j}{h_j}\right) \tag{19}$$

where $r_j = \sum \left| \mathbf{x}_j^F - \mathbf{P}_F(\lambda_F(i))\hat{\mathbf{X}}_i \right|$ and h_j is estimated from equation (5) with the robust *k* scale estimator.

After $\hat{\mathbf{P}}_F$ is estimated, the 3D points $\{\mathbf{X}_i\}$ having correspondences to the tracked SIFT features are refined:

$$\hat{\mathbf{X}}_i = Minimize \sum_{j=0}^{w_i-1} \sum \left| \mathbf{x}_i^{F-j} - \hat{P}_{F-j} \hat{\mathbf{X}}_i \right| \tag{20}$$

Newly appearing 3D points are initialized and added to the structure. Figure 9 gives an outline of the reconstruction algorithm.

4. Experiments

4.1. Data Collection

We collected endoscopic sinus image data on a cadaverous porcine specimen. Images were captured using a Storz Telecam, 202212113U NTSC with a zero degree rigid rod monocular endoscope, 7210AA. An external tracking system (Optotrack, Northern Digital Corp. Waterloo) was used to measure and record the motion of the endoscope during the procedure of image acquisition and we use the Optotrack motion data as the ground truth to which the estimated endoscopic motion was compared. Images from a standard optical calibration target were also recorded using the endoscope before the data collection was performed. We perform an offline calibration [26] of the endoscope using a Matlab Camera Calibration Toolkit [27].

4.2. Motion estimation

To evaluate the performance of our system, first, we compare our proposed robust estimator ASKC (ASKC1/ASKC2) with five other robust estimators (LMedS, MSAC, RANSAC, RESC and ASSC) in motion estimation. Following [12], we used a median scale estimator for MSAC. For RANSAC, we specify the error tolerance value with which optimal results are achieved.

To get quantitative results, we apply the methods to one hundred pairs of endoscopic sinus images. The distance between the positions of the endoscopic camera in each pair of images is larger than 1mm. To measure the accuracy of the motion estimation, both translation error and rotation error are tested. We use a formula similar to that of [28].

Each of the methods is run for the 100 pairs of images. The median error values, the mean error values and the standard variances of the estimate errors in translation and rotation are used to evaluate the performance of the methods.

From Table 1, our methods (ASKC1/ASKC2) achieve the most accurate results among the comparative methods. LMedS and MSAC achieve the worst results as the median scale estimator is not robust to more than 50% outliers. RANSAC with a user-specified error tolerance achieves better results than LMedS and MSAC, but worse than the rest. This is because RANSAC requires different error tolerance values for different image pairs and it is hard to find a global optimal value. The results of ASSC are better than those of RESC but less accurate than those of ASKC1/ASKC2. Between ASKC1 and ASKC2, ASKC2 outperforms ASKC1 in the translation estimation while ASKC1 is slightly better in rotation estimation.

	Translation Error			Rotation Error		
	Median	Mean	Std.Var.	Median	Mean	Std.Var.
LMedS	27.904	28.895	13.971	2.722	3.238	2.269
MSAC	26.554	27.182	12.877	2.660	3.038	2.016
RANSAC	7.520	7.828	4.837	0.636	0.676	0.375
RESC	6.009	10.833	15.068	0.262	0.952	1.955
ASSC	5.125	5.652	3.855	0.303	0.360	0.247
ASKC1	4.401	5.215	3.643	0.231	0.273	0.173
ASKC2	4.196	4.997	3.615	0.255	0.299	0.200

Table 1. Quantitative evaluation of the different methods on 100 pairs of sinus images. Both the translation error and the rotation error are in degrees.

4.3. Structure reconstruction

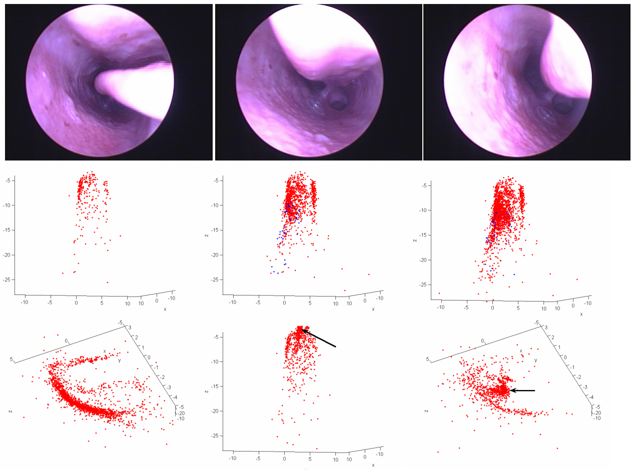

Figure 10. Top row: the first, middle and last frame of the image sequence. Middle row: the recovered structure corresponding to the top row. Blue points are the newly recovered 3D points. Bottom row: (left) another view of the final recovered structure; (middle and right) two views of the final recovered structure by M1.

We test our reconstruction algorithm with a sinus image sequence including 130 frames with the frame size of

640x480. The endoscope performed several movements (sideways, forward, and backward) during the acquisition of the image sequence. The image sequence was digitally captured at a rate of roughly 30 frames per second. As a result, the baselines between the consecutive frames are too close together which results in ill-conditioned epipolar geometry estimation. To avoid this problem, we only consider a set of key frames that are far enough apart for the motion and structure recovery.

We use the reconstruction algorithm proposed in subsection (3.3) to recover the structure of the sinus. We choose to use ASKC2 in the system but ASKC1 can also be employed in the system.

Figure 10 shows the reconstruction results on the sinus image sequence. As we can see the main structure of the surrounding tissues of the sinus is recovered by our system. In comparison, when we use the LMedS estimator and estimate the projection matrices $\{\mathbf{P}_i\}_{i=1,2,\ldots,N}$ by the approach in [29] (we call it as M1), it fails to recover the structure and most recovered 3D points are clustered in a small area pointed out by the arrow (see the middle and right column of the bottom row in Figure 10).

5. Conclusions

In this paper, we present a new robust estimator (ASKC) that can tolerate more than 50% (or even 80%) outliers. We also propose a reliable feature tracking algorithm that can track features even when images involve significant blurring, illumination changes and geometry distortion. We integrate ASKC and the feature tracking approach to a complete system for motion and structure recovery from sinus endoscopic image sequences. The primarily experiments show that ASKC outperforms several other robust estimators (including LMedS, MSAC RANSAC, RESC, and ASSC) and our system has achieved promising results.

Acknowledgement

This work has been supported by the National Institutes of Health under grant number 1R21EB005201 - 01A1.

References

[1] Y. Kosugi, et al. An Articulated Neurosurgical Navigation System Using MRI and CT Images. T-BME.147-152, 1988.

[2] M. Scholtz, et al. Development of an Endoscopic Navigating System Based on Digital Image Processing. Journal of Computer Aided Surgery. 3(3):134-143, 1998.

[3] D. Burschka, et al. Scale-Invariant Registration of Monocular Endoscopic Images to CT-Scans for Sinus Surgery Medical Image Analysis. 9(5):413-439, 2005.

[4] J.P. Helferty and W.E. Higgins. Technique for Registering 3D Virtual CT Images to Endoscopic Video. ICIP, 893-896, 2001.

[5] K. Mori, et al. A Method for Tracking the Camera Motion of Real Endoscope by Epipolar Geometry Analysis and Virtual Endoscopy System. MICCAI, 1-8, 2001.

[6] P.J. Rousseeuw and A. Leroy. Robust Regression and outlier detection. John Wiley & Sons, New York. 1987.

[7] M.A. Fischler and R.C. Rolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. ACM. 24(6):381-395, 1981.

[8] P.J. Huber. Robust Statistics. New York, Wiley. 1981.

[9] H. Chen and P. Meer. Robust Regression with Projection Based M-estimators. ICCV, 878-885, 2003.

[10] R. Subbarao and P. Meer. Heteroscedastic projection based M-estimators. Workshop on EEMCV, 2005.

[11] S. Rozenfeld and I. Shimshoni. The Modified pbM-estimator Method and a Runtime Analysis Technique for the RANSAC Family. CVPR, 1113-1120, 2005.

[12] P. Torr and D. Murray. The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. IJCV. 24(3):271-300, 1997.

[13] J.V. Miller and C.V. Stewart. MUSE: Robust Surface Fitting Using Unbiased Scale Estimates. CVPR, 300-306, 1996.

[14] C.V. Stewart. MINPRAN: A New Robust Estimator for Computer Vision. PAMI. 17(10):925-938, 1995.

[15] K.-M. Lee, P. Meer and R.-H. Park. Robust Adaptive Segmentation of Range Images. PAMI. 20(2):200-205, 1998.

[16] X. Yu, T.D. Bui and A. Krzyzak. Robust Estimation for Range Image Segmentation and Reconstruction. PAMI. 16(5):530-538, 1994.

[17] H. Wang and D. Suter. Robust Adaptive-Scale Parametric Model Estimation for Computer Vision. PAMI. 26(11):1459-1474, 2004.

[18] B.W. Silverman. Density Estimation for Statistics and Data Analysis. 1986. Chapman and Hall. London.

[19] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach towards Feature Space A Analysis. PAMI. 24(5):603-619, 2002.

[20] M.P. Wand and M. Jones. Kernel Smoothing. Chapman & Hall. 1995.

[21] B. Tordoff and D.W. Murray. Guided Sampling and Consensus for Motion Estimation. ECCV. 82-96, 2002.

[22] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. 2004. Cambridge University Press.

[23] D. Nistér. An Efficient Solution to the Five-point Relative Pose Problem. PAMI. 26(6):756-770, 2004.

[24] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. IJCV. 60(2):91-110, 2004.

[25] E. Delponte, et al. SVD-matching using SIFT Features. Graphical Models. 68(5-6):415-431, 2006.

[26] Z. Zhang. A Flexible New Technique for Camera Calibration. PAMI. 22(11):1330-1334, 2000.

[27] J.-Y. Bouget. The matlab camera calibration toolkit. http://www.vision.caltech.edu/bouguetj/calib_doc/.

[28] T. Tian, C. Tomasi and D. Heeger. Comparison of Approaches to Egomotion Computation. CVPR, 315-320, 1996.

[29] M. Pollefeys, et al. Visual Modeling with a Hand-held Camera. IJCV. 59(3):207-232, 2004.