Quantitative Endoscopy: Uncertainty Analysis of Motion Estimation with Robust Feature Matching

Group 15: Xiang Xiang
Mentors: Prof. Gregory Hager, Dr. Daniel Mirota and Prof. Russell Taylor

May 10, 2013

Abstract

Feature matching based 3D reconstruction is a standard technique in 3D Computer Vision. An natural extension is to reconstruct dynamic surfaces from videos, such as reconstructing sinus surfaces from endoscopic videos. However, since the camera is moving and the sinus surfaces are normally deformable and non-planar, the feature matching is usually unsatisfactory. We will employ a state-of-the-art feature matching strategy in the domain of minimally invasive image analysis. Instead of restricting inliers using a global affine transformation, multiple affine components are hierarchically clustered. Qualitative results verify that this Hierachical Multi-Affine (HMA) strategy works well for non-planar and deformable surfaces. Also conducted are the empirical uncertainty analysis of the estimated motion in a leaving-one-out cross validation setting. A series of comparison between HMA matching and SIFT matching are presented as well.

1 Introduction

From [1], it is estimated that there are more than 200,000 functional endoscopic sinus surgeries (FESS) procedures performed annually in the United States at a cost of several billion dollars annually. As the name implies, all of these procedures are performed under endoscopic guidance, and a large fraction employ surgical navigation systems to visualize critical structures that must not be disturbed during the surgery. Although navigation is widely employed for FESS, its capabilities are far from optimal. In particular, the sinuses contain structures that are smaller than a millimeter in size, and yet delineate critical anatomy such as the optic nerve or the carotid artery. However, the accuracy of navigation is 2 mm under near ideal conditions [1]. As a result, navigation can provide a qualitative sense of location, but final confirmation of anatomic structures ultimately relies on the surgeon's ability to interpret and relate the CT image to the endoscopic view. This process, which is further complicated when the anatomy is distorted or otherwise altered by surgery, requires time, skill and experience and can lead to errors in judgement that adversely affect outcome [1].

According to [1], the significance of the endoscopic visualization and navigation is the introduction of a paradigm shift in surgical navigation by using a device present in every endoscopic surgery, namely the endoscope, to improve registration and visualization of anatomy. This will have numerous positive impacts. Most importantly, it will provide an inexpensive, non-invasive, radiation-free method to enhance registration accuracy at any point of the procedure. Enhancements in registration will reduce ambiguity for the surgeon during surgery, enhancing confidence, and improve workflow by reducing the need to re-register or re-image the patient. The endoscope will also be used as a measurement device to update anatomic models during a procedure. This not only will improve the ability of the surgeon to visualize the progress of the surgery, but it will accrue additional benefits to the patient and hospital, as it may reduce the level of radiation exposure and cost by eliminating the need for intraoperative CT imaging. Figure 1 presents our proposed pipeline to achieve endoscopic 3D visualization.

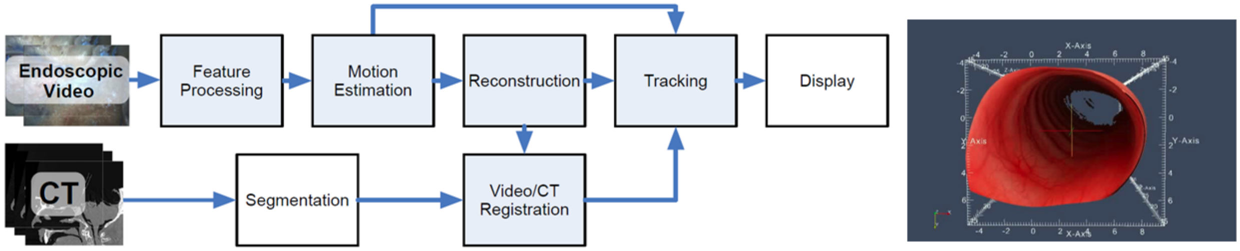


Figure 1: The pipeline for 3D visualization from endoscopic videos and pre-operative CT scan images [7].

3D reconstruction has been deeply explored in the computer vision community [2, 5] and surface rendering has been widely examined in the computer graphics community. Most results are shown qualitatively. However, results on endoscopic reconstruction are seldom reported, while the demand for clinical use is actually large as introduced above. Data collection is not the only difficulty. Simply applying multi-view geometry techniques may not work well. It is an interdisciplinary field between computer vision and microsurgery, so domain problems such as precision are crucial. To satisfy the precision requirement, We hope to build a sense of quantitative endoscopy and make algorithmic improvement such as more robust feature matching and more accurate motion estimation.

2 Approach

In the course project, we focus on feature matching and motion estimation, which are elaborated in Sec. 2.1 and Sec. 2.2, respectively. Sec. 2.3 briefly introduces some standard techniques employed in Sec. 2.1 and Sec. 2.2.

2.1 Hierarchical Multi-Affine Feature Matching

Image matching are key to 3D reconstruction, stereo, tracking and recognition. Generally speaking, it include feature detection, description and matching. Feature description is generally expected to be invariant to image scaling and rotation, at least partially invariant to changes in illumination and 3D camera viewpoint and highly discriminative. Scale Invariant Feature Transform (SIFT) [4] just provides such a keypoint detector and a feature descriptor, which normally satisfies those requirements.

Matching is expected to be robust to outliers and deformation (e.g., non-planar surface). In the original SIFT framework [4], Lowe estimates a global affine transformation between two images and defines outliers as those point pair which do not subject to the affine transformation. A recent Hierarchical Multi-Affine (HMA) matching algorithm's basic idea is to represent a plane or surface using multiple affine-transformation components. Multiple affine estimation will slow down the processing, so hierarchical K-means clustering is incorporated. Binary search in a tree structure outperforms exhaustive search in efficiency. We need to highlight our own contributions on uncertainty analysis in this report and thus refer interested reader to [9] for details of HMA.

2.2 Empirical Uncertainty Analysis of Motion Estimation

In this section, we will quantitatively analyze the feature matching algorithm. and use the matched features in camera motion estimation . The basic idea of empirical covariance analysis is to compute statistics from results in a number of experiments, either by cross validation or Monte Carlo simulation [2]. The **uncertainty** of the computed rotation is captured in the covariance matrix of the rotation [2]. Here, we perform the leave-one-out cross validation (LOOCV), partially due to lack of groud truth matches. The exact algorithm is elaborated as followed.

Algorithm 1. Covariance analysis of motion estimation by LOOCV.

```
for $k = 1 ... FrmNum$
    Compute SIFT feature keypoints to form a candidate feature pool.
    Perform HMA matching to select keypoints, which are grouped into affine components.
    Perform image rectification considering radial distortion.
    Convert image coordinates to World's coordinates using camera's intrinsic parameters.
    if $MatchFeaNum > 4$
        for $trial = 1 ... MatchFeaNum$
            Leave the $trial$-th keypoint out as a query point.
            Perform RANSAC on the left keypoints to generate an inlier set.
            Perform 5-point algorithm to estimate the essential matrix $E$ using the inlier set.
            Decompose $E$ into a rotation matrix $R$ and a translation vector $t$.
            Convert $R$ to a quaternion.
            Compute square of projection error for the held-out query point:
                $residual = (R * X_{left}^{query} + t) - X_{right}^{query}$
            $sqErr = L2norm(residual)$
        **end for**
        Compute the mean and standard deviation of a sequence of $sqErr$.
        Compose a $R_{mean}$ from $mean(quaternion)$.
        **for** $trial = 1...MatchFeaNum$
            $R_{mean} * R^{-1}$ is approximately a skew-symmetric matrix $skew(\alpha, \beta, \gamma)$,
                where $\alpha, \beta, \gamma$ are the rotation angle (scalar) in X, Y, Z axis, respectively.
        **end for**
        Compute the standard deviations of $\alpha, \beta, \gamma$, respectively.
        Compute the covariance matrix of a sequence of vector $< \alpha, \beta, \gamma >$.
    **end if**
**end for**
```

2.3 Standard Techniques Employed

This section briefly introduce some terminologies appeared in Algorithm 1. Most of them are standard techniques in 3D computer vision [5]. Thus, Wikipedia is a good resource for detailed explanations. For readers who are farmiliar with them, please skip this section.

Coordinate transformation from image coordinates' to World's coordinates follows the standard geometric model of image formation [5] as shown below.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ I \end{bmatrix}$$

In the right-hand side of the above equation, the first matrix encodes the scaling information and the second encodes the focus information. The product of these two matrices is termed the intrinsic parameter matrix. Here the projection matrix is not included since the original coordinates are uncalibrated image coordinates, instead of not World's 3D coordinates. However, only linear distortion is considered in this equation.

Radial distortion [5] is also compensated in our program using calibration outputs, such as the focus length and image optical center.

RANSAC. Since not all matched feature keypoints are inliers, simply using all the points will not induce a good estimation. We need a robust estimator such as RANSAC, which is short for RANdom SAmple Consensus. Its basic idea is to use a minimal number of data points needed to estimate the model [5].

Essential matrix comes from the epipolar constraint equation and encodes the relative pose/motion $[R, t]$ between two cameras [5]. By using eight-point algorithm [5] or the improved five-point algorithm, the camera motion can be re recovered from an essential matrix.

Rotation quarternion. Unit quarternion is a four-element vector. A rotation matrix can be represented by a quarternion as:

$$\boldsymbol{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$

Euler's rotation theorem implies that the composition of two rotations is also a rotation. According to [12], suppose we specify an axis of rotation by a unit vector $[x, y, z]$ and we have an infinitely small rotation of angle $\Delta\theta$ about the vector. Expanding the rotation matrix as an infinite addition, and taking the first order approach, the rotation matrix ΔR is represented as:

$$\Delta R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \Delta\theta = \mathbf{I} + \mathbf{A}\,\Delta\theta.$$

Notice that ΔR is a skew symmetric matrix, where the element x, y, z denotes the rotation angle in X, Y, Z axis, respectively. In our case, $\Delta R = R_{mean} * R^{-1}$ and x, y, z correspond to α, β, γ, respectively. An empirical analysis of ΔR's uncertainty is a core task in this course project.

3 Experiments

A fair comparison between HMA and SIFT macthing algorithms will be evaluating each algorithm's matching accuracy with respect to the ground truth. However, it is much too time-consuming to manually pick up pairs of feature keypoints. Thus, we do not collect the ground truth data for feature matching evaluation.

3.1 External Libraries

Camera calibration is performed by using Caltech Matlab calibration toolkit [3].
SIFT features are extracted using VLfeat Matlab library [11].
HMA matching are performed using HMA Matlab toolbox [8].
RANSAC based E matrix estimation is performed using OpenCV's findFundamentalMat.
Camera motion recovery from E is done using Structure and Motion Matlab toolkit [10].

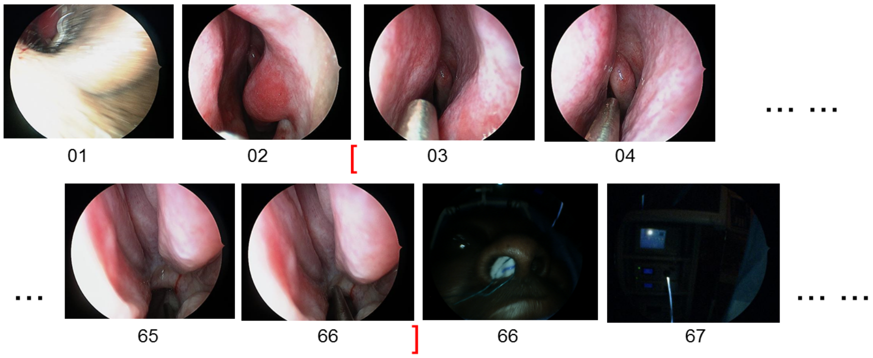


Figure 2: A sample of patient endoscopic video data. In this sample, we make use of frame 3 to frame 66 for testing. All frames in this continuous part represent endoscopic scenarios.

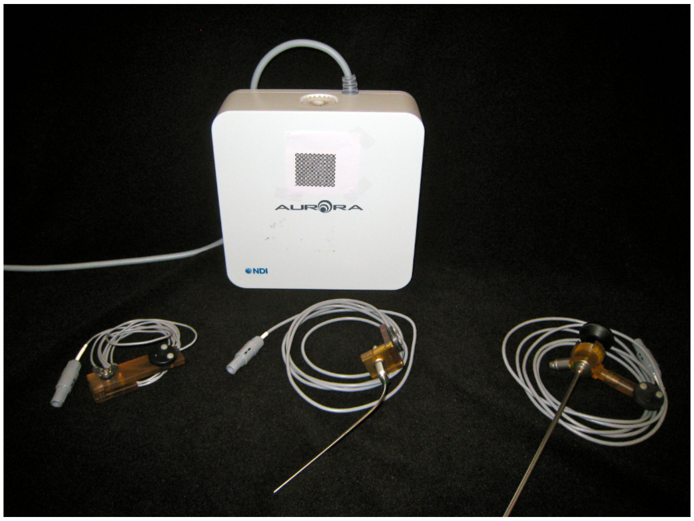


Figure 3: Endoscopic sensor and data collection devices. The top box is the processor produced by NDI. The bottom left is a high-precision optically tracked endoscope, The bottom middle and right form a EM tracked scope for use in airway data collection [1]. Picture courtesy of Dr. Daniel Mirota.

3.2 Patient Data

Patient data are collected at Johns Hopkins Hospital on December 19, 2012. The endoscopic video is hours long. As shown in Figure 2, we pick up a sample sequence consisting of 64 frames for testing. The design of the data collection device (see Figure 3) is another task in the grant project. A data collection system has been developed to simultaneously capture both the endoscopic video and external motion tracking data. However, data collection is out of the scope of this course project, which focuses on the algorithm design and testing. Images are processed in the original size (1280 × 1024 pixels, 3.8MB per image).

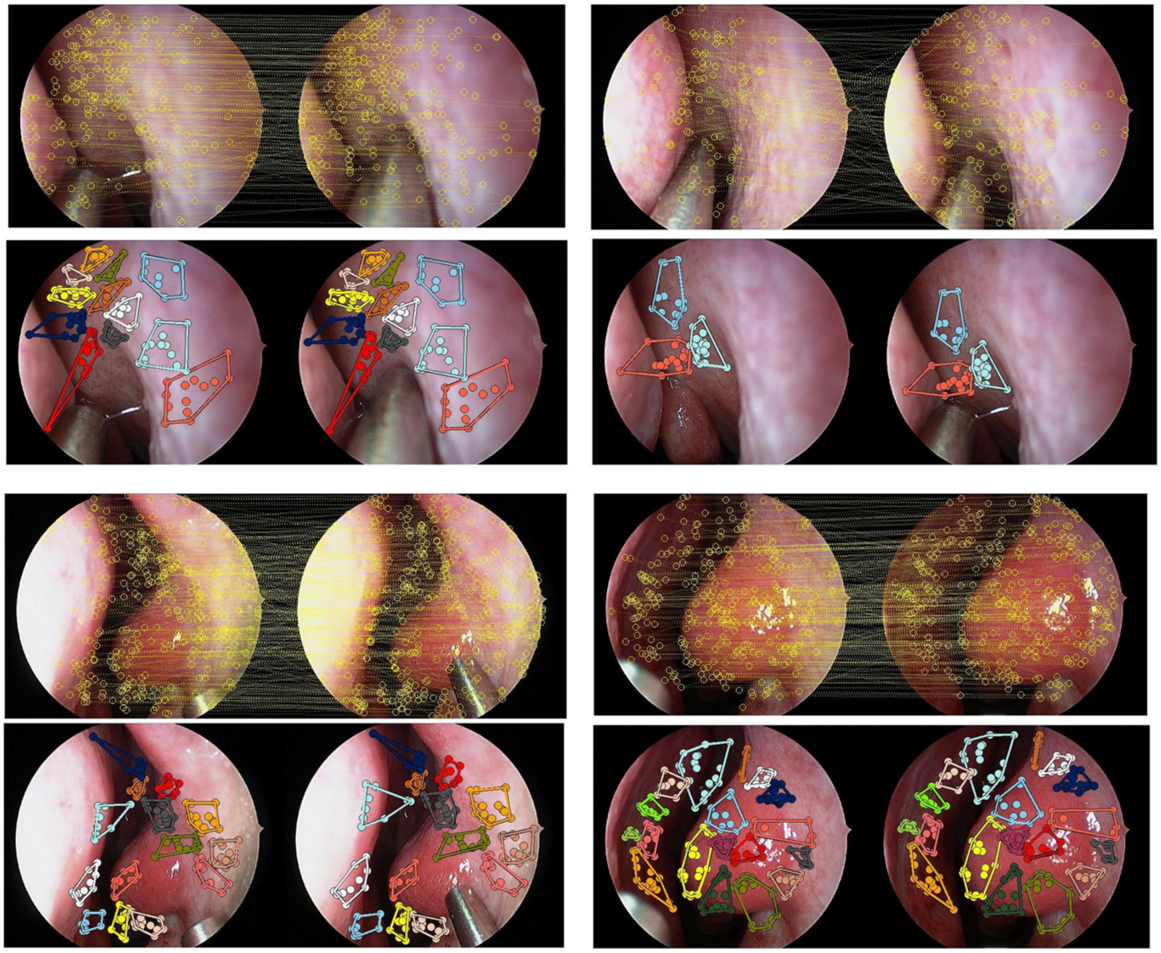


Figure 4: Example results of HMA matching vs. SIFT matching. In each group, the top row shows SIFT's result, in which line crossings normally imply mismatches. The bottom row shows HMA's result, in which different affine components are displayed in different color.

3.3 Feature Matching Results and Comparison

Figure 4 presents a qualitative comparison between HMA and SIFT feature matching. HMA's results are with high confidences. Namely, the matches found are likely to be correct matches. and there are no obvious mismatches. For example, there is no point match which corss regions in HMA's results. Moreover, it is clear to see that HMA finds enough correct matches. These results verify that local deformation constraints are useful in restricting point matches.

Although there are more matches in SIFT's results (see Figure 5), the confidences of matches are relatively low. Namely, there are more outliers among the matched pairs. There can be obvious mismatches. For instance, a point in a left region can be matched to another in a right region. This pair is still possible to satisfy the inaccurate global deformation.

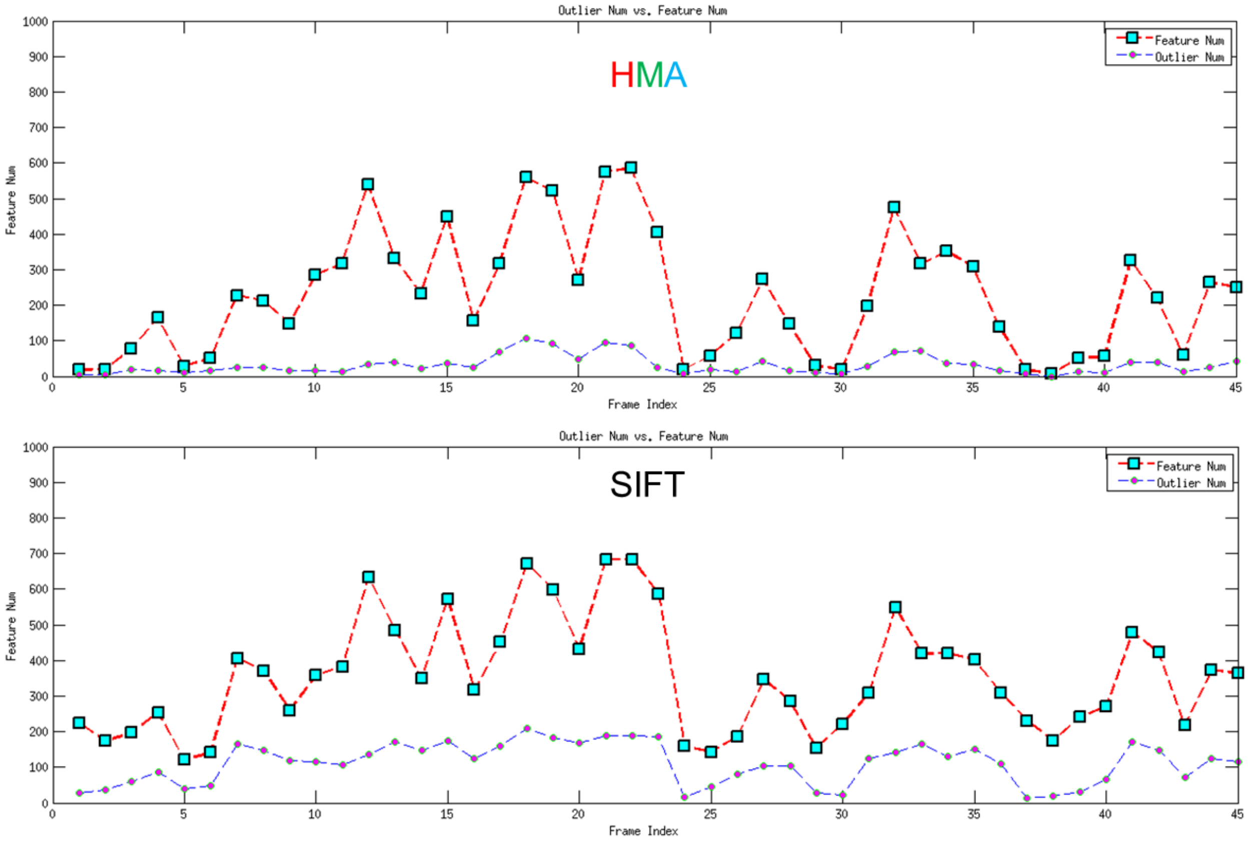


Figure 5: Examples results of HMA matching vs. SIFT matching. In each group, the top row shows SIFT's result, in which line crossings normally imply mismatches. The bottom row shows HMA's result, in which different affine components are displayed in different color.

Figure 5 presents the detected outlier number given by RANSAC vs. the total matched feature number given by the matching algorithm. Although RANSAC's estimation is not guaranteed to be right, Figure 5 more or less implies: while there are more matches found by SIFT, there are actually fewer inliers.

3.4 Variance and Covariance Analysis for Estimated Motion

In this section, we will examine estimated motion - primarily the rotation matrix ΔR, which encodes the rotation angles. Detailed algorithm has been shown in Algorithm 1. Here, we opt the closely related part out and give Algorithm 3.

Algorithm 2. Estimating covariance matrix by quarternions.

```
for $k = 1...FrmNum$
    for $trial = 1...MatchFeaNum$
        Decompose $E$ to get $R$ and $t$
        Convert $R$ to quaternion
    end for
    Compose a $R_{mean}$ from $mean(quaternion)$
    for $trial = 1...MatchFeaNum$
        $Delta_R = R_{mean} * R^{-1}$, approximately a skew-symmetric matrix $skew(\alpha, \beta, \gamma)$.
    end for
    Compute the standard deviations of $\alpha, \beta, \gamma$, respectively.
    Compute the covariance matrix of a sequence of $\alpha, \beta, \gamma$.
end for
```

Actually, another way is to estimate the covariance matrix by Euler angles. Theoretically, quarternions induces higher precision than Euler angles. However, in cases where rotation angles are small, these two ways are normally equivalent.

Algorithm 3. Estimating covariance matrix by Euler angles.

```
for $k = 1...FrmNum$
    for $trial = 1...MatchFeaNum$
        Decompose $E$ to get $R$ and $t$
        Decompose $R$ to get yaw, pitch and roll angles: $rx$, $ry$, $rz$
    end for
    Compose a $R_a$ from $mean(rx), mean(ry), mean(rz)$ and set it as $R_{mean}$
    for $trial = 1...MatchFeaNum$
        $Delta_R = R_{mean} * R^{-1}$, approximately a skew-symmetric matrix $skew(\alpha, \beta, \gamma)$.
    end for
    Compute the standard deviations of $\alpha, \beta, \gamma$, respectively.
    Compute the covariance matrix of a sequence of $\alpha, \beta, \gamma$.
end for
```

When we refer to variance, we actually compute the its square root - the standard deviation. Both the variance and covariance encode the uncertainty in the estimation of the rotation. Figure 6 jointly displays all three rotation angles together with the number of features and presents a comparison between HMA and SIFT. We can see that for both HMA and SIFT, the standard deviation of α, β, γ almost have the the same trends. The trend of feature number generally satisfy their pattern as well.

Discussion on Uncertainty. Comparing HMA with SIFT in Figure 6, we cannot safely say that the angular standard deviation in HMA's results are generally smaller than that in SIFT's. Given that the variance of estimated motion is affected by the variance of the locations of feature keypoints, the uncertainty should still be somewhat essential for the

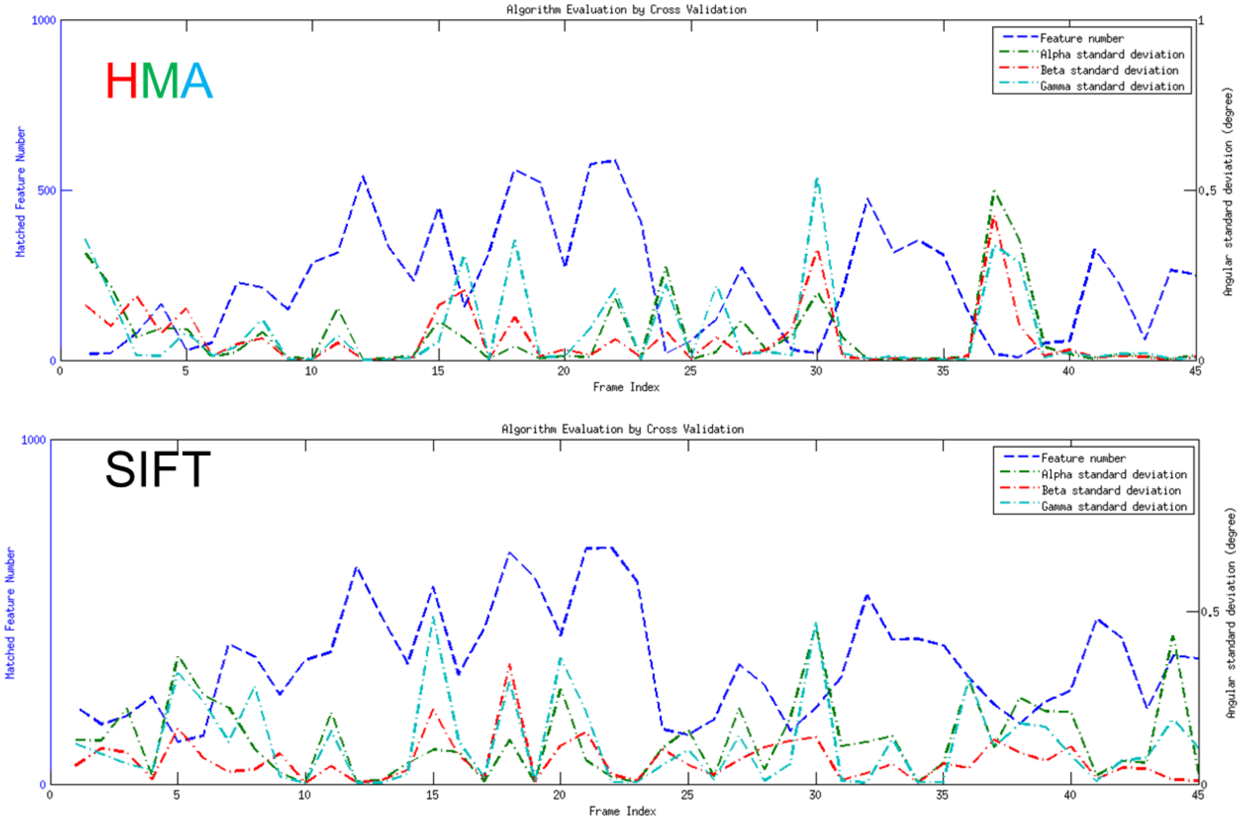
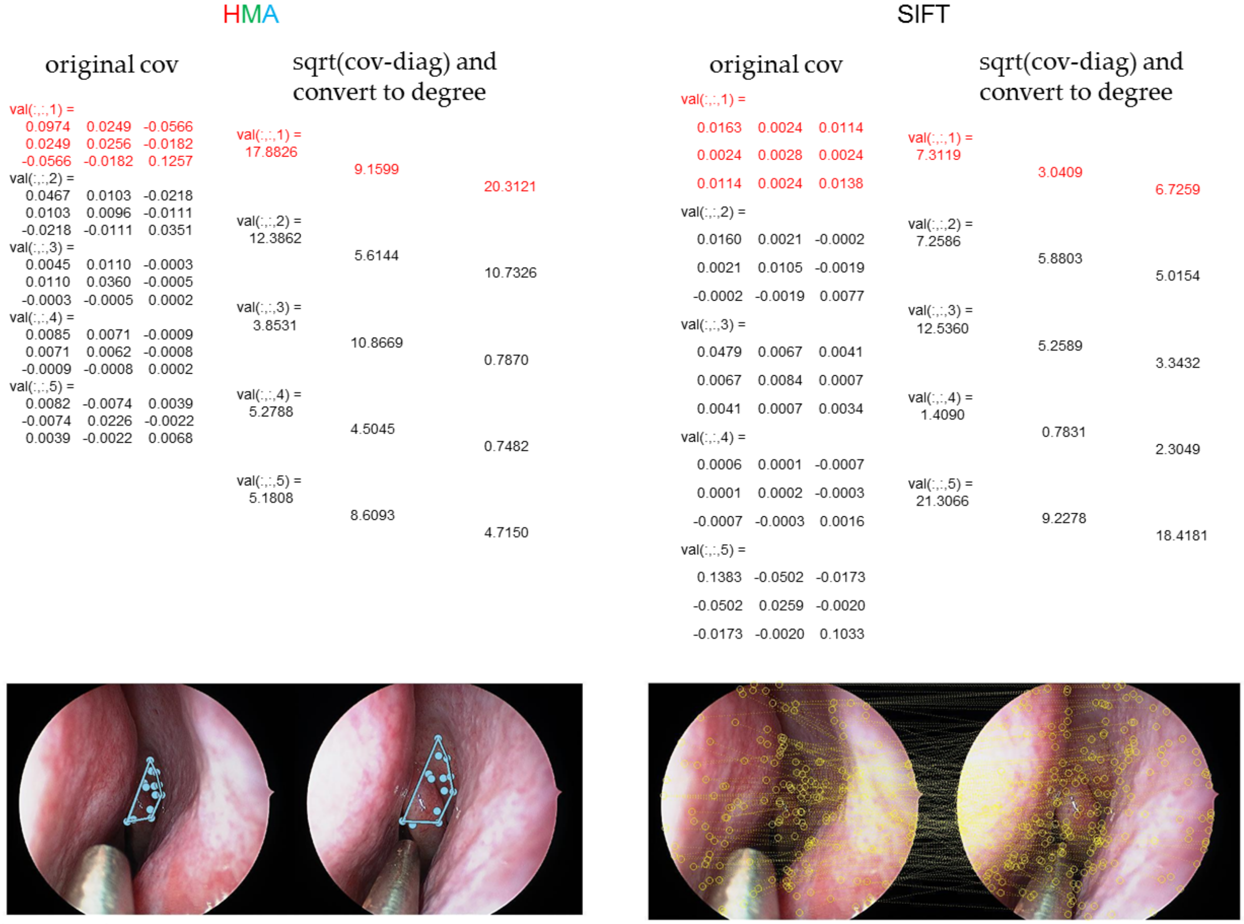


Figure 6: Examples results of HMA matching vs. SIFT matching. In each group, the top row shows SIFT's result, in which line crossings normally imply mismatches. The bottom row shows HMA's result, in which different affine components are displayed in different color.

motion estimation algorithm such as five-point algorithm. From [2], the uncertainty of the estimated transformation depends on many factors, including the number of points used to compute it, the accuracy of the given point matches, as well as the configuration of the points in question. Analytically, we can apply direct differentiation of the epipolar constraint and finally estimate the covariance of the motion given keypoint correspondences [6]. For this issue, much more in-depth analysis will be performed as future work. We hope to give theoretic explanation as well.

Now, let us examine the covariance matrix for two example cases. It requires experiences about rotations and deep understanding about uncertainty to analyze the estimated covariance matrices. It is unclear for me to compare the covariance matrices given by HMA and those given by SIFT matching.



HMA

original cov

```
val(:,:,1) =
   0.0974   0.0249  -0.0566
   0.0249   0.0256  -0.0182
  -0.0566  -0.0182   0.1257
val(:,:,2) =
   0.0467   0.0103  -0.0218
   0.0103   0.0096  -0.0111
  -0.0218  -0.0111   0.0351
val(:,:,3) =
   0.0045   0.0110  -0.0003
   0.0110   0.0360  -0.0005
  -0.0003  -0.0005   0.0002
val(:,:,4) =
   0.0085   0.0071  -0.0009
   0.0071   0.0062  -0.0008
  -0.0009  -0.0008   0.0002
val(:,:,5) =
   0.0082  -0.0074   0.0039
  -0.0074   0.0226  -0.0022
   0.0039  -0.0022   0.0068
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
 17.8826
              9.1599
                         20.3121
val(:,:,2) =
 12.3862
              5.6144
                         10.7326
val(:,:,3) =
  3.8531
             10.8669
                          0.7870
val(:,:,4) =
  5.2788
              4.5045
                          0.7482
val(:,:,5) =
  5.1808
              8.6093
                          4.7150
```

SIFT

original cov

```
val(:,:,1) =
   0.0163   0.0024   0.0114
   0.0024   0.0028   0.0024
   0.0114   0.0024   0.0138
val(:,:,2) =
   0.0160   0.0021  -0.0002
   0.0021   0.0105  -0.0019
  -0.0002  -0.0019   0.0077
val(:,:,3) =
   0.0479   0.0067   0.0041
   0.0067   0.0084   0.0007
   0.0041   0.0007   0.0034
val(:,:,4) =
   0.0006   0.0001  -0.0007
   0.0001   0.0002  -0.0003
  -0.0007  -0.0003   0.0016
val(:,:,5) =
   0.1383  -0.0502  -0.0173
  -0.0502   0.0259  -0.0020
  -0.0173  -0.0020   0.1033
```

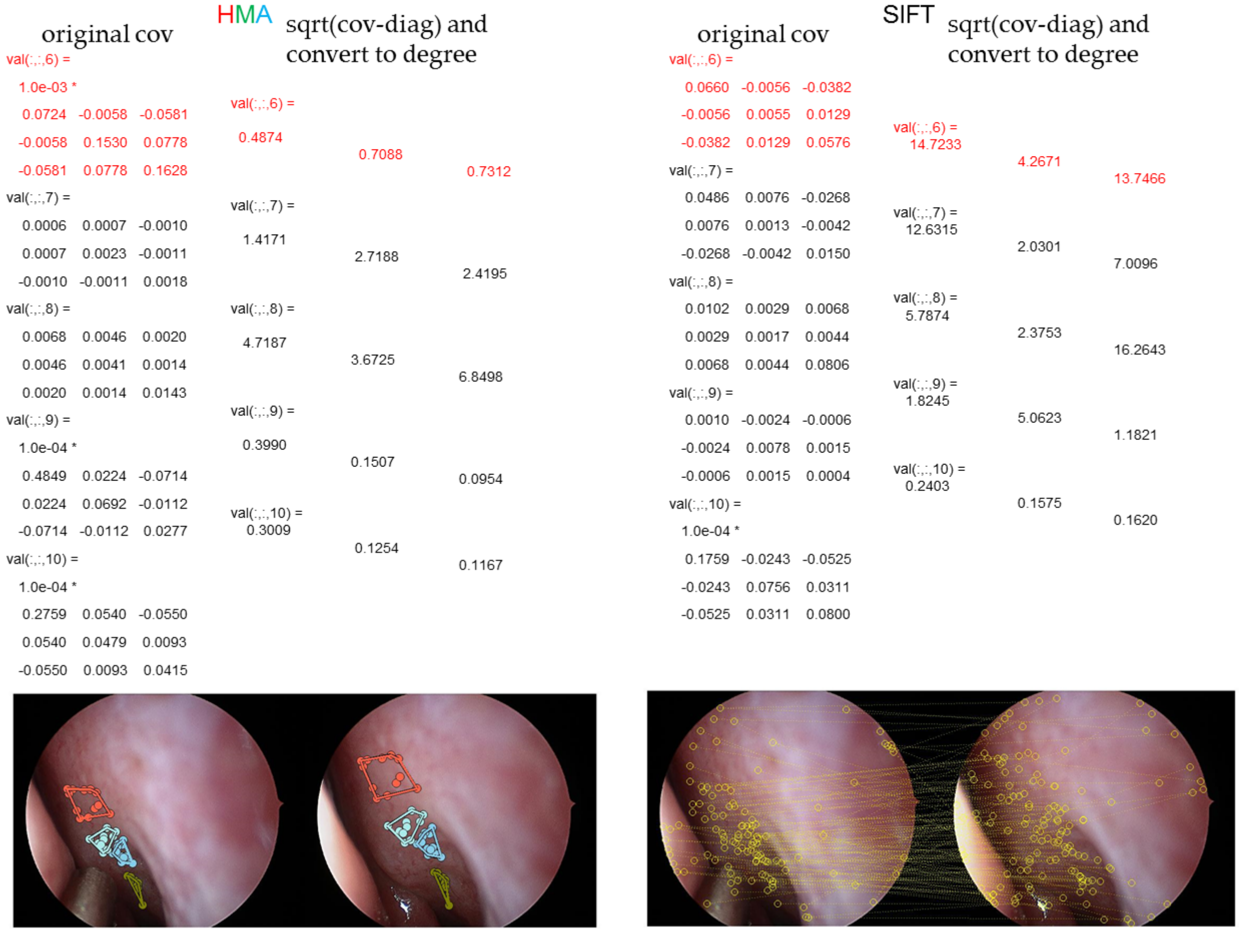
sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
  7.3119
              3.0409
                          6.7259
val(:,:,2) =
  7.2586
              5.8803
                          5.0154
val(:,:,3) =
 12.5360
              5.2589
                          3.3432
val(:,:,4) =
  1.4090
              0.7831
                          2.3049
val(:,:,5) =
 21.3066
              9.2278
                         18.4181
```

Figure 7: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 1, frame 2). These two frames are mainly different in scale. The rotation angles should be relatively large. The SIFT matching algorithm works rather poorly, as we see a number of line crossing. The way SIFT matching judge outlier is to globally estimate an affine transformation and then see if points fit it or not. A single affine transformation is difficult to describe the deformation between these two frames. HMA matching apply group SIFT keypoints into several clusters and estimate an affine transformation for each cluster. Although there is only one local affine transformation fit well the points, this cluster provides enough point matches to estimate the motion. Normally speaking, the more matches, the better motion estimation.

3.5 Tentative Accuracy Analysis for Estimated Motion

In this section, we will project the held-out query keypoint using the estimated R and t. For each pair of adjacent frames with over 4 matched features, the input of the this testing are coordinates of feature keypoints in the frame t (on the left) and those in the frame $t + 1$ (on the right), together with the essential matrix E estimated previously.



HMA

original cov

```
val(:,:,6) =

   1.0e-03 *

    0.0724   -0.0058   -0.0581
   -0.0058    0.1530    0.0778
   -0.0581    0.0778    0.1628

val(:,:,7) =

    0.0006    0.0007   -0.0010
    0.0007    0.0023   -0.0011
   -0.0010   -0.0011    0.0018

val(:,:,8) =

    0.0068    0.0046    0.0020
    0.0046    0.0041    0.0014
    0.0020    0.0014    0.0143

val(:,:,9) =

   1.0e-04 *

    0.4849    0.0224   -0.0714
    0.0224    0.0692   -0.0112
   -0.0714   -0.0112    0.0277

val(:,:,10) =

   1.0e-04 *

    0.2759    0.0540   -0.0550
    0.0540    0.0479    0.0093
   -0.0550    0.0093    0.0415
```

sqrt(cov-diag) and convert to degree

```
val(:,:,6) =
    0.4874    0.7088    0.7312

val(:,:,7) =
    1.4171    2.7188    2.4195

val(:,:,8) =
    4.7187    3.6725    6.8498

val(:,:,9) =
    0.3990    0.1507    0.0954

val(:,:,10) =
    0.3009    0.1254    0.1167
```

SIFT

original cov

```
val(:,:,6) =

    0.0660   -0.0056   -0.0382
   -0.0056    0.0055    0.0129
   -0.0382    0.0129    0.0576

val(:,:,7) =

    0.0486    0.0076   -0.0268
    0.0076    0.0013   -0.0042
   -0.0268   -0.0042    0.0150

val(:,:,8) =

    0.0102    0.0029    0.0068
    0.0029    0.0017    0.0044
    0.0068    0.0044    0.0806

val(:,:,9) =

    0.0010   -0.0024   -0.0006
   -0.0024    0.0078    0.0015
   -0.0006    0.0015    0.0004

val(:,:,10) =

   1.0e-04 *

    0.1759   -0.0243   -0.0525
   -0.0243    0.0756    0.0311
   -0.0525    0.0311    0.0800
```

sqrt(cov-diag) and convert to degree

```
val(:,:,6) =
   14.7233    4.2671   13.7466

val(:,:,7) =
   12.6315    2.0301    7.0096

val(:,:,8) =
    5.7874    2.3753   16.2643

val(:,:,9) =
    1.8245    5.0623    1.1821

val(:,:,10) =
    0.2403    0.1575    0.1620
```

Figure 8: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 6, frame 7). There is no much change between the two frames. Correspondingly, the diagonal elements of the estimated covariance matrix are small.

Algorithm 4. Computing projection error for the held-out query point.

```
for $k = 1 \ldots FrmNum$
    for $trial = 1 \ldots MatchFeaNum$
        Decompose $E$ to get $R$ and $t$
        Compute projection error for the held-out testing point:
            $residual = (R * X_{left}^{query} + t) - X_{right}^{query}$
            $err = L2 - norm(residual)$
    end for
    Compute the mean and standard deviation of a sequence of $err$.
end for
```

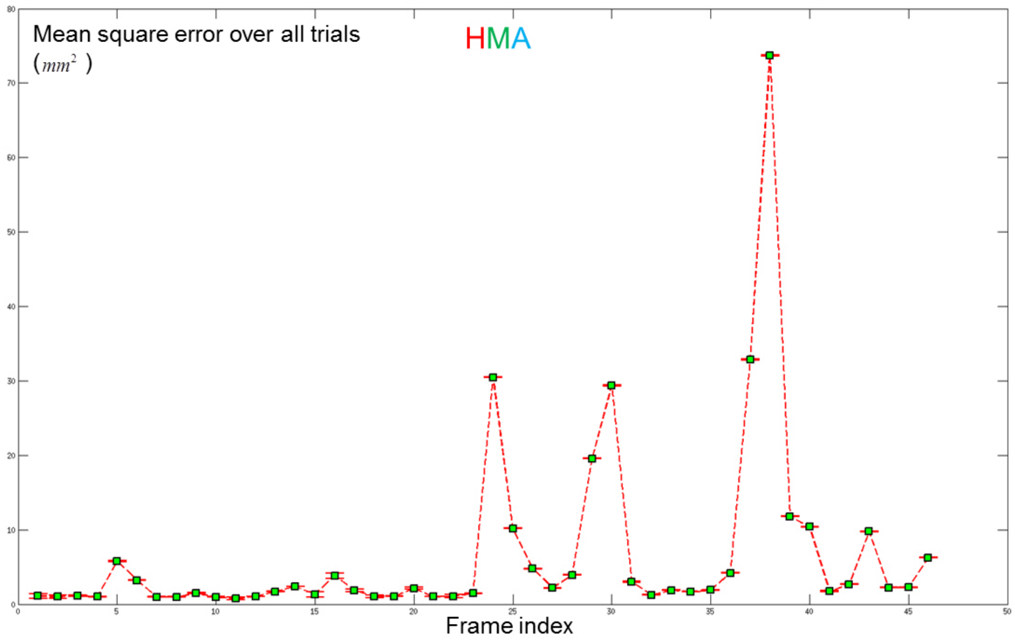


Figure 9: Projection error of the held-out query keypoint with HMA matching.

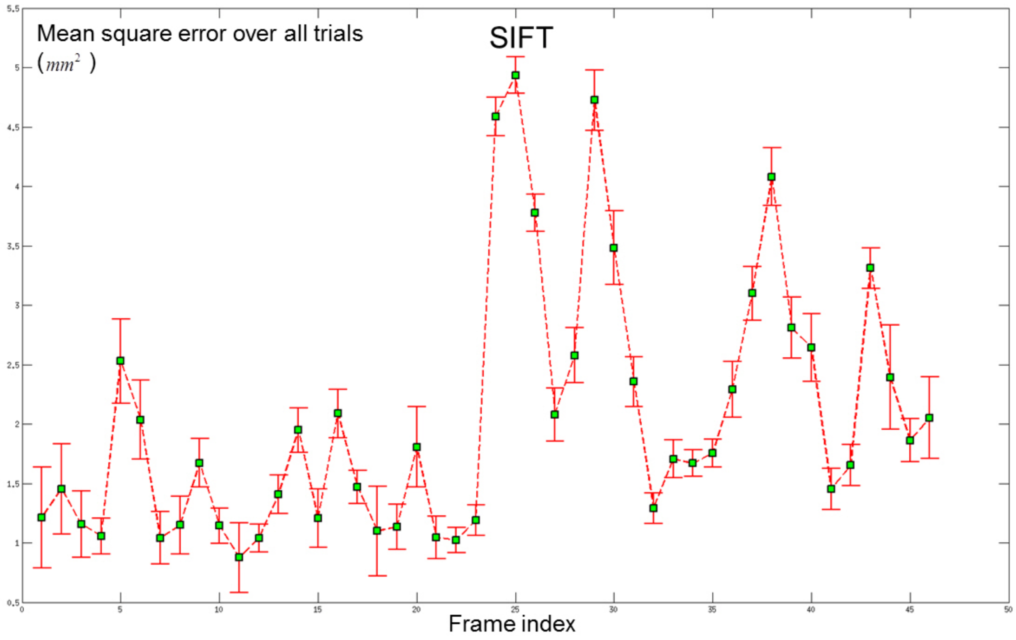


Figure 10: Projection error of the held-out query keypoint with SIFT matching.

First of all, let us see Figure 9 to examine the results for HMA matching. Except 9 frames with much larger mean of square error (MSE), all other frames' MSE are within 10 mm^2, and the majority's MSE are within 5 pixel square. Then, let us check Figure 10 to examine the results for SIFT matching. Nearly all frames' MSE are within 5 mm^2.

It seems that SIFT performs better than HMA, which contradicts our motivation to improve SIFT matching by HMA matching. However, since in both cases the majority frames' MSE are both within 5 pixel square, the performances seem to be bascially similar. Actually, since the query point varies in each trial, the MSE does not strictly represent the matching performance. It is also affected by the number of trials. Nonetheless, we analyze this to have an impression of the performance quantitatively.

4 Discussion

Till now, I feel that I cannot safely draw a conclusion about uncertainty in motion estimation. However, it verifies that HMA generally provides more accurate and more robust feature matching than SIFT does. About the further step - motion estimation, our expectation is that HMA induces better motion estimation than SIFT does, which is not verified by the comparison of the mean square projection error in the LOOCV experiment. Nonetheless, we have an impression of both's performance qualitatively and quantitatively.

Following up in the future is much more in-depth uncertainty analysis. We hope to give theoretic explanation as well. In the same time, we will test how HMA feature matching affect the reconstruction pipeline's perfomance and how it complete against the SIFT matching.

References

[1] G. D. Hager and etal. Enhanced navigation for endoscopic sinus surgery through video analysis. *National Institutes of Health grant project proposal*, Grant No. R01 EB015530, 2012.

[2] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[3] C. V. Lab. Camera Calibration Toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`, July 2010.

[4] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2005.

[5] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision.* Springer, 2004.

[6] D. Mirota. *Video-Based Navigation with Application to Endoscopic Skull Base Surgery.* Johns Hopkins University Computer Science Department Ph.D. Dissertation, 2012.

[7] D. Mirota, H. Wang, R. H. Taylor, M. Ishii, G. L. Gallia, and G. D. Hager. A system for video-based navigation for endoscopic endonasal skull base surgery. *IEEE Trans. Med. Imaging*, 31.

[8] G. A. Puerto and G. L. Mariottini. Hma feature-matching toolbox. `http://ranger.uta.edu/~gianluca/feature_matching/`, 2012.

[9] G. A. Puerto and G. L. Mariottini. A fast and accurate feature-matching algorithm for minimally invasive endoscopic images. *IEEE Transactions on Medical Imaging*, 2013.

[10] P. Torr. Hma feature-matching toolbox. `http://www.mathworks.com/matlabcentral/fileexchange/4576-structure-and-motion-toolkit-in-matlab`, 2004.

[11] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[12] Wikipedia. Euler's Rotation Theorem. `http://en.wikipedia.org/wiki/Euler's_rotation_theorem`.

5 Appendix

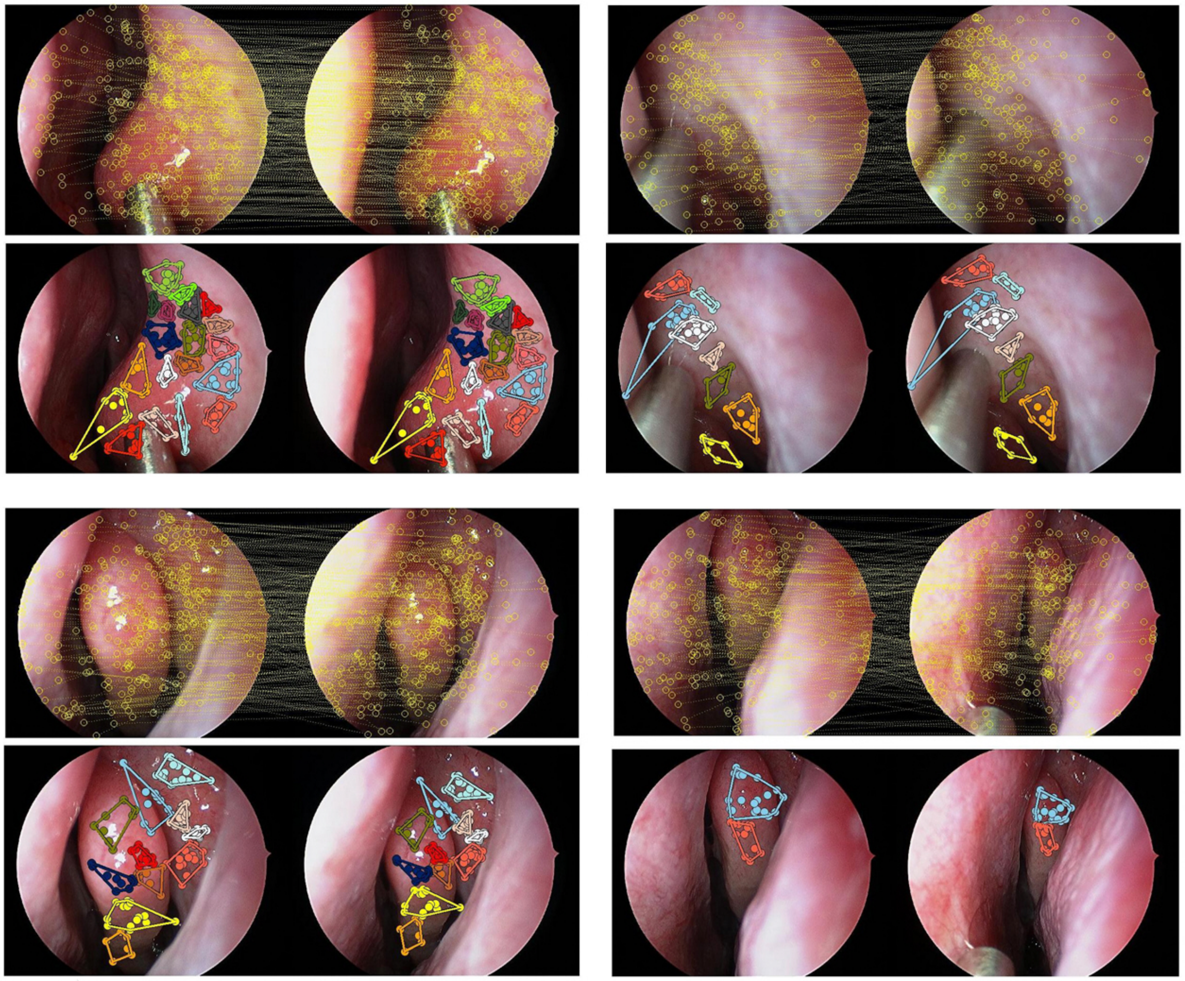
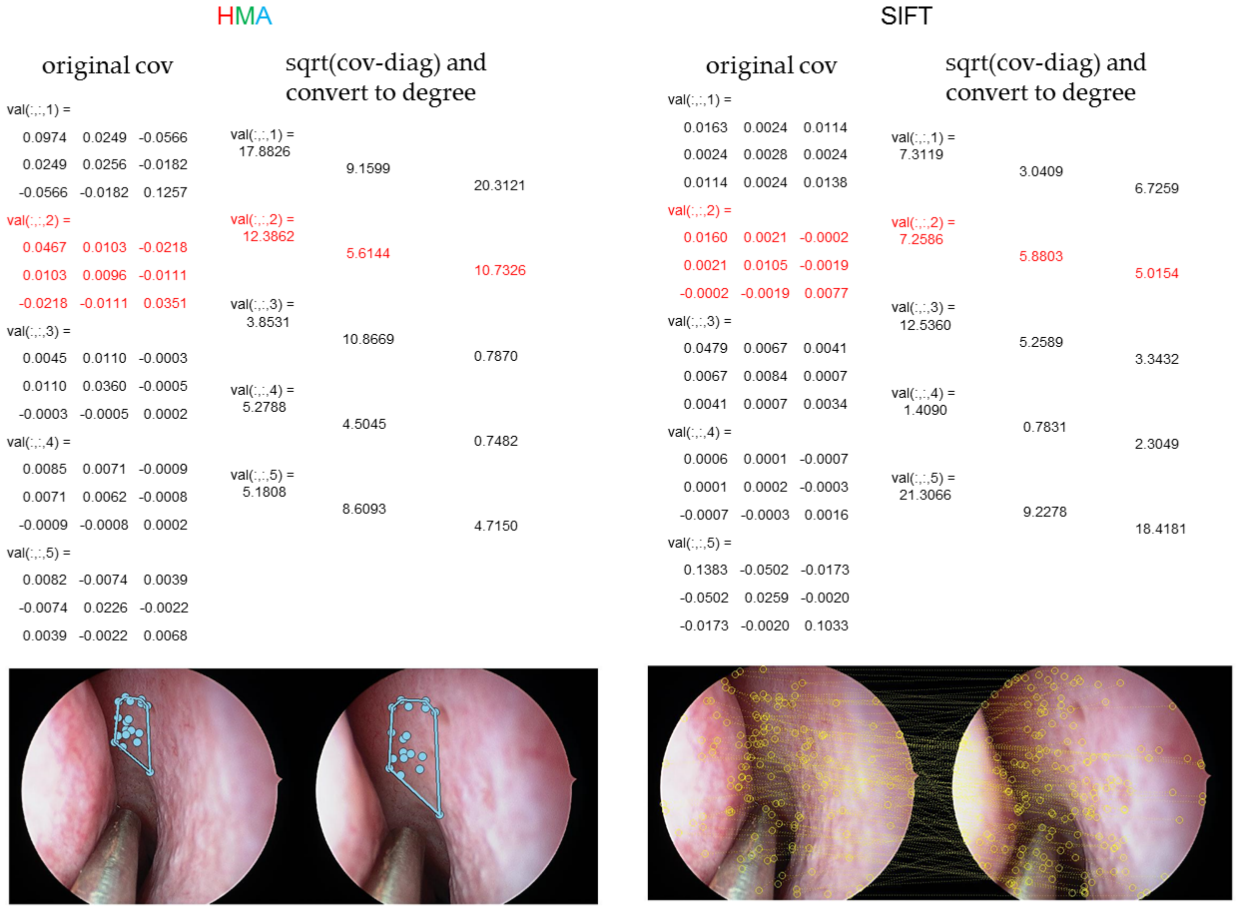


Figure 11: More example results of HMA matching vs. SIFT matching.



HMA

original cov

```
val(:,:,1) =

    0.0974    0.0249   -0.0566
    0.0249    0.0256   -0.0182
   -0.0566   -0.0182    0.1257

val(:,:,2) =

    0.0467    0.0103   -0.0218
    0.0103    0.0096   -0.0111
   -0.0218   -0.0111    0.0351

val(:,:,3) =

    0.0045    0.0110   -0.0003
    0.0110    0.0360   -0.0005
   -0.0003   -0.0005    0.0002

val(:,:,4) =

    0.0085    0.0071   -0.0009
    0.0071    0.0062   -0.0008
   -0.0009   -0.0008    0.0002

val(:,:,5) =

    0.0082   -0.0074    0.0039
   -0.0074    0.0226   -0.0022
    0.0039   -0.0022    0.0068
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
   17.8826

             9.1599

                        20.3121

val(:,:,2) =
   12.3862

             5.6144

                        10.7326

val(:,:,3) =
    3.8531

            10.8669

                         0.7870

val(:,:,4) =
    5.2788

             4.5045

                         0.7482

val(:,:,5) =
    5.1808

             8.6093

                         4.7150
```

SIFT

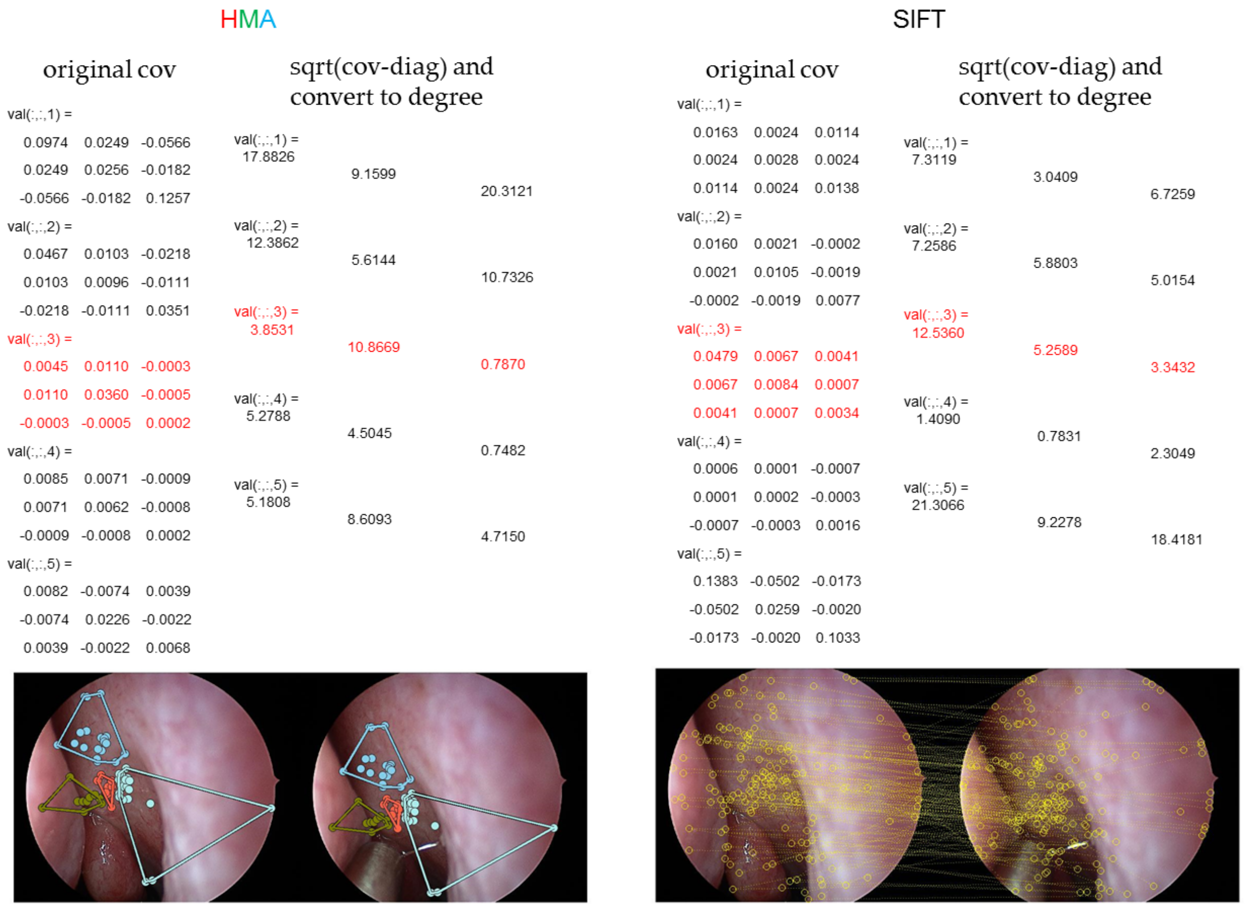
original cov

```
val(:,:,1) =

    0.0163    0.0024    0.0114
    0.0024    0.0028    0.0024
    0.0114    0.0024    0.0138

val(:,:,2) =

    0.0160    0.0021   -0.0002
    0.0021    0.0105   -0.0019
   -0.0002   -0.0019    0.0077

val(:,:,3) =

    0.0479    0.0067    0.0041
    0.0067    0.0084    0.0007
    0.0041    0.0007    0.0034

val(:,:,4) =

    0.0006    0.0001   -0.0007
    0.0001    0.0002   -0.0003
   -0.0007   -0.0003    0.0016

val(:,:,5) =

    0.1383   -0.0502   -0.0173
   -0.0502    0.0259   -0.0020
   -0.0173   -0.0020    0.1033
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
    7.3119

             3.0409

                         6.7259

val(:,:,2) =
    7.2586

             5.8803

                         5.0154

val(:,:,3) =
   12.5360

             5.2589

                         3.3432

val(:,:,4) =
    1.4090

             0.7831

                         2.3049

val(:,:,5) =
   21.3066

             9.2278

                        18.4181
```

Figure 12: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 2, frame 3). The different between the frames lies more in scale. This pair are challenging. For HMA matching, there is one group subjecting to an affine transformation. For SIFT matching, most are mismatches.



HMA

original cov

```
val(:,:,1) =

   0.0974   0.0249  -0.0566
   0.0249   0.0256  -0.0182
  -0.0566  -0.0182   0.1257

val(:,:,2) =

   0.0467   0.0103  -0.0218
   0.0103   0.0096  -0.0111
  -0.0218  -0.0111   0.0351

val(:,:,3) =

   0.0045   0.0110  -0.0003
   0.0110   0.0360  -0.0005
  -0.0003  -0.0005   0.0002

val(:,:,4) =

   0.0085   0.0071  -0.0009
   0.0071   0.0062  -0.0008
  -0.0009  -0.0008   0.0002

val(:,:,5) =

   0.0082  -0.0074   0.0039
  -0.0074   0.0226  -0.0022
   0.0039  -0.0022   0.0068
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
  17.8826
                 9.1599
                              20.3121

val(:,:,2) =
  12.3862
                 5.6144
                              10.7326

val(:,:,3) =
   3.8531
                10.8669
                               0.7870

val(:,:,4) =
   5.2788
                 4.5045
                               0.7482

val(:,:,5) =
   5.1808
                 8.6093
                               4.7150
```

SIFT

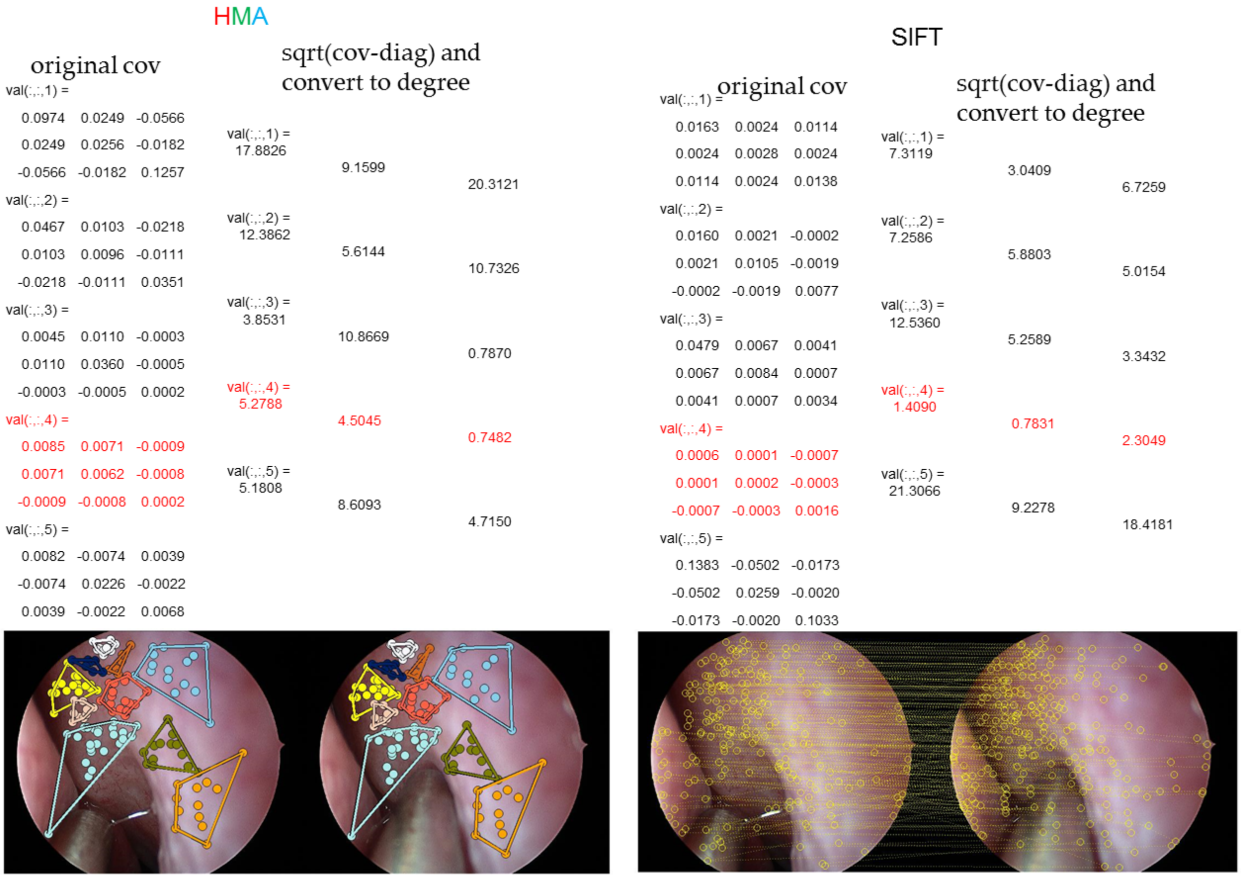
original cov

```
val(:,:,1) =

   0.0163   0.0024   0.0114
   0.0024   0.0028   0.0024
   0.0114   0.0024   0.0138

val(:,:,2) =

   0.0160   0.0021  -0.0002
   0.0021   0.0105  -0.0019
  -0.0002  -0.0019   0.0077

val(:,:,3) =

   0.0479   0.0067   0.0041
   0.0067   0.0084   0.0007
   0.0041   0.0007   0.0034

val(:,:,4) =

   0.0006   0.0001  -0.0007
   0.0001   0.0002  -0.0003
  -0.0007  -0.0003   0.0016

val(:,:,5) =

   0.1383  -0.0502  -0.0173
  -0.0502   0.0259  -0.0020
  -0.0173  -0.0020   0.1033
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
   7.3119
                 3.0409
                               6.7259

val(:,:,2) =
   7.2586
                 5.8803
                               5.0154

val(:,:,3) =
  12.5360
                 5.2589
                               3.3432

val(:,:,4) =
   1.4090
                 0.7831
                               2.3049

val(:,:,5) =
  21.3066
                 9.2278
                              18.4181
```

Figure 13: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 3, frame 4).



HMA

original cov

```
val(:,:,1) =

    0.0974    0.0249   -0.0566
    0.0249    0.0256   -0.0182
   -0.0566   -0.0182    0.1257

val(:,:,2) =

    0.0467    0.0103   -0.0218
    0.0103    0.0096   -0.0111
   -0.0218   -0.0111    0.0351

val(:,:,3) =

    0.0045    0.0110   -0.0003
    0.0110    0.0360   -0.0005
   -0.0003   -0.0005    0.0002

val(:,:,4) =

    0.0085    0.0071   -0.0009
    0.0071    0.0062   -0.0008
   -0.0009   -0.0008    0.0002

val(:,:,5) =

    0.0082   -0.0074    0.0039
   -0.0074    0.0226   -0.0022
    0.0039   -0.0022    0.0068
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
   17.8826
              9.1599
                         20.3121

val(:,:,2) =
   12.3862
              5.6144
                         10.7326

val(:,:,3) =
    3.8531
             10.8669
                          0.7870

val(:,:,4) =
    5.2788
              4.5045
                          0.7482

val(:,:,5) =
    5.1808
              8.6093
                          4.7150
```

SIFT

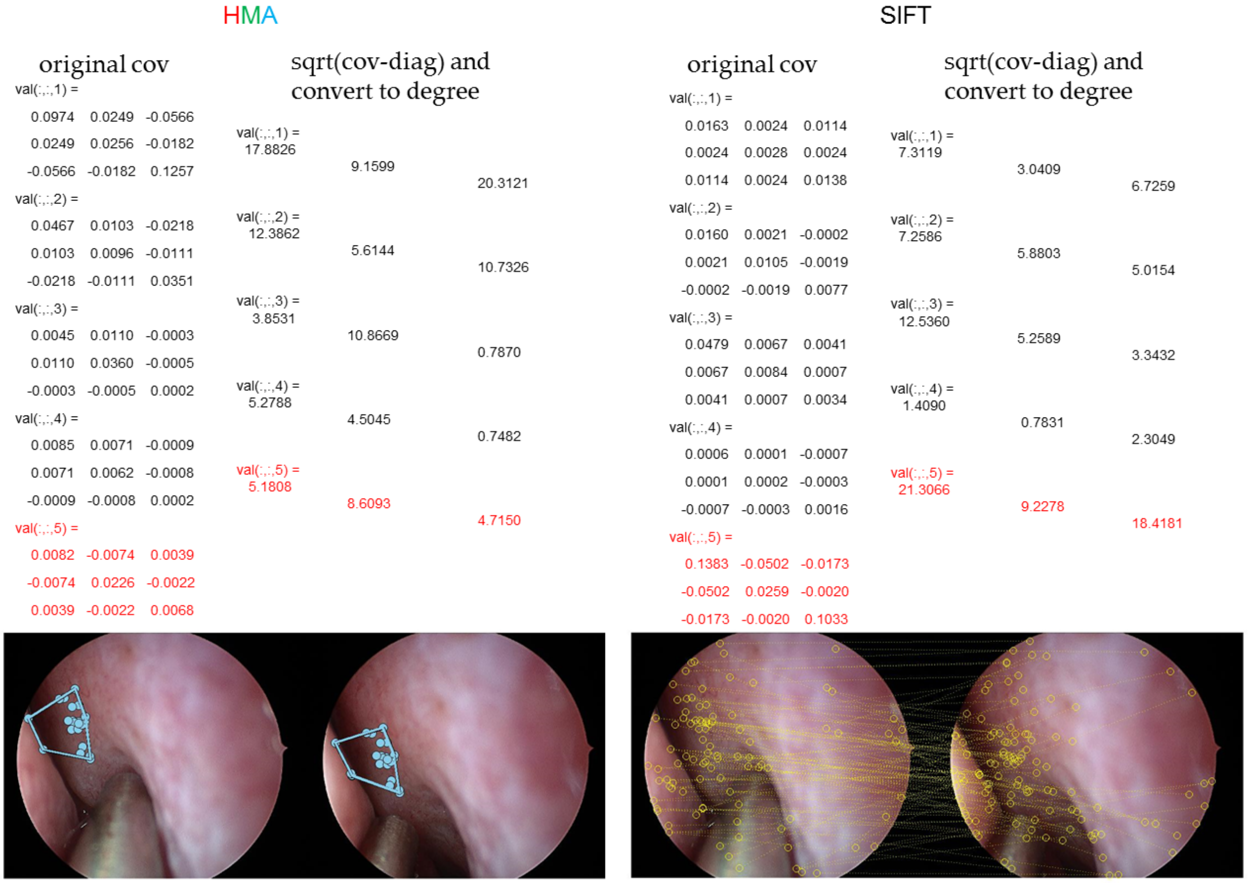
original cov

```
val(:,:,1) =

    0.0163    0.0024    0.0114
    0.0024    0.0028    0.0024
    0.0114    0.0024    0.0138

val(:,:,2) =

    0.0160    0.0021   -0.0002
    0.0021    0.0105   -0.0019
   -0.0002   -0.0019    0.0077

val(:,:,3) =

    0.0479    0.0067    0.0041
    0.0067    0.0084    0.0007
    0.0041    0.0007    0.0034

val(:,:,4) =

    0.0006    0.0001   -0.0007
    0.0001    0.0002   -0.0003
   -0.0007   -0.0003    0.0016

val(:,:,5) =

    0.1383   -0.0502   -0.0173
   -0.0502    0.0259   -0.0020
   -0.0173   -0.0020    0.1033
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
    7.3119
              3.0409
                          6.7259

val(:,:,2) =
    7.2586
              5.8803
                          5.0154

val(:,:,3) =
   12.5360
              5.2589
                          3.3432

val(:,:,4) =
    1.4090
              0.7831
                          2.3049

val(:,:,5) =
   21.3066
              9.2278
                         18.4181
```

Figure 14: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 4, frame 5).



HMA

original cov

```
val(:,:,1) =

   0.0974   0.0249  -0.0566
   0.0249   0.0256  -0.0182
  -0.0566  -0.0182   0.1257

val(:,:,2) =

   0.0467   0.0103  -0.0218
   0.0103   0.0096  -0.0111
  -0.0218  -0.0111   0.0351

val(:,:,3) =

   0.0045   0.0110  -0.0003
   0.0110   0.0360  -0.0005
  -0.0003  -0.0005   0.0002

val(:,:,4) =

   0.0085   0.0071  -0.0009
   0.0071   0.0062  -0.0008
  -0.0009  -0.0008   0.0002

val(:,:,5) =

   0.0082  -0.0074   0.0039
  -0.0074   0.0226  -0.0022
   0.0039  -0.0022   0.0068
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
   17.8826
                    9.1599
                                    20.3121

val(:,:,2) =
   12.3862
                    5.6144
                                    10.7326

val(:,:,3) =
    3.8531
                   10.8669
                                     0.7870

val(:,:,4) =
    5.2788
                    4.5045
                                     0.7482

val(:,:,5) =
    5.1808
                    8.6093
                                     4.7150
```

SIFT

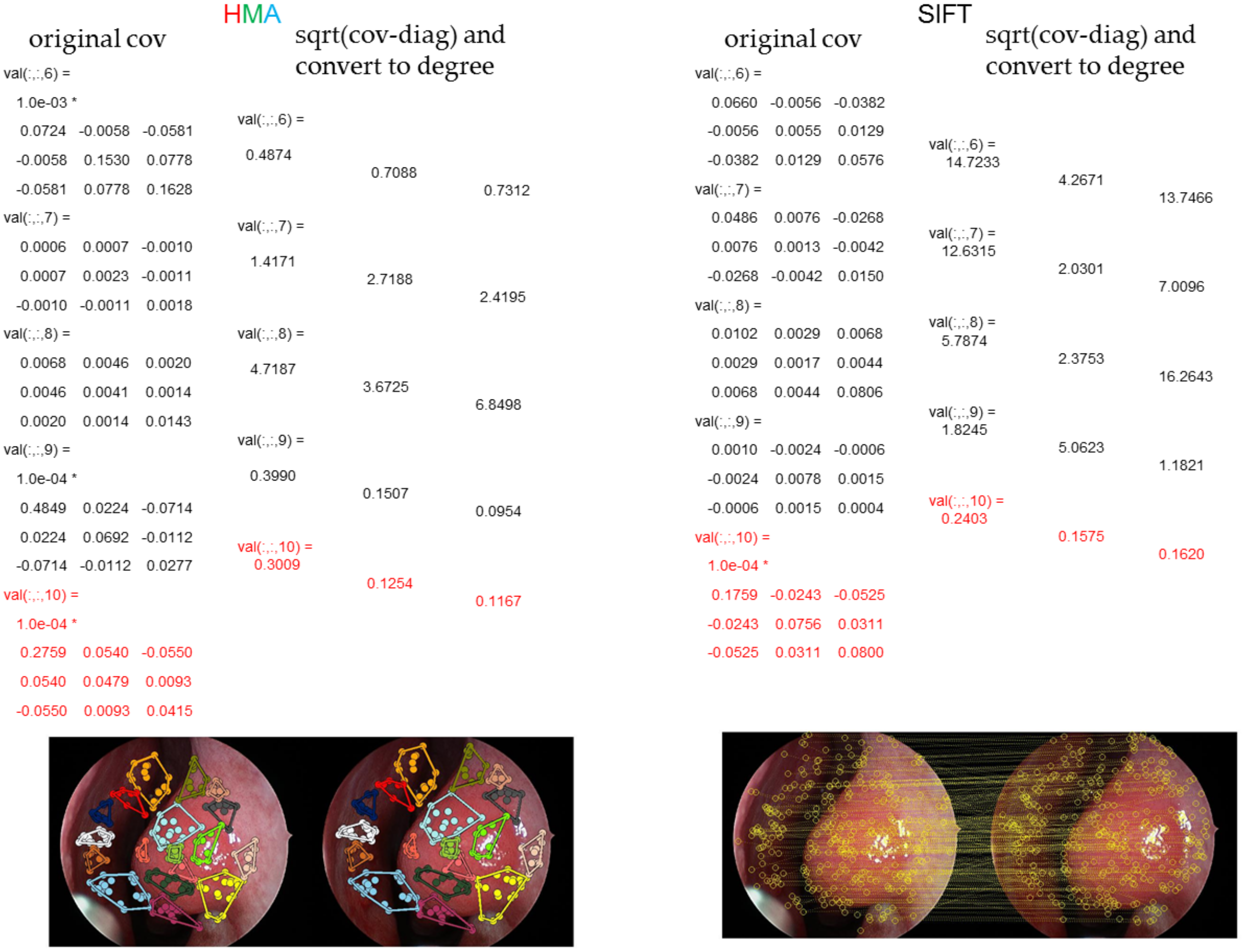
original cov

```
val(:,:,1) =

   0.0163   0.0024   0.0114
   0.0024   0.0028   0.0024
   0.0114   0.0024   0.0138

val(:,:,2) =

   0.0160   0.0021  -0.0002
   0.0021   0.0105  -0.0019
  -0.0002  -0.0019   0.0077

val(:,:,3) =

   0.0479   0.0067   0.0041
   0.0067   0.0084   0.0007
   0.0041   0.0007   0.0034

val(:,:,4) =

   0.0006   0.0001  -0.0007
   0.0001   0.0002  -0.0003
  -0.0007  -0.0003   0.0016

val(:,:,5) =

   0.1383  -0.0502  -0.0173
  -0.0502   0.0259  -0.0020
  -0.0173  -0.0020   0.1033
```

sqrt(cov-diag) and convert to degree

```
val(:,:,1) =
    7.3119
                    3.0409
                                     6.7259

val(:,:,2) =
    7.2586
                    5.8803
                                     5.0154

val(:,:,3) =
   12.5360
                    5.2589
                                     3.3432

val(:,:,4) =
    1.4090
                    0.7831
                                     2.3049

val(:,:,5) =
   21.3066
                    9.2278
                                    18.4181
```

Figure 15: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 5, frame 6).



HMA

original cov

```
val(:,:,6) =

  1.0e-03 *

   0.0724  -0.0058  -0.0581
  -0.0058   0.1530   0.0778
  -0.0581   0.0778   0.1628

val(:,:,7) =

   0.0006   0.0007  -0.0010
   0.0007   0.0023  -0.0011
  -0.0010  -0.0011   0.0018

val(:,:,8) =

   0.0068   0.0046   0.0020
   0.0046   0.0041   0.0014
   0.0020   0.0014   0.0143

val(:,:,9) =

  1.0e-04 *

   0.4849   0.0224  -0.0714
   0.0224   0.0692  -0.0112
  -0.0714  -0.0112   0.0277

val(:,:,10) =

  1.0e-04 *

   0.2759   0.0540  -0.0550
   0.0540   0.0479   0.0093
  -0.0550   0.0093   0.0415
```

sqrt(cov-diag) and convert to degree

```
val(:,:,6) =

  0.4874

                 0.7088

                              0.7312

val(:,:,7) =

  1.4171

                 2.7188

                              2.4195

val(:,:,8) =

  4.7187

                 3.6725

                              6.8498

val(:,:,9) =

  0.3990

                 0.1507

                              0.0954

val(:,:,10) =
  0.3009

                 0.1254

                              0.1167
```

SIFT

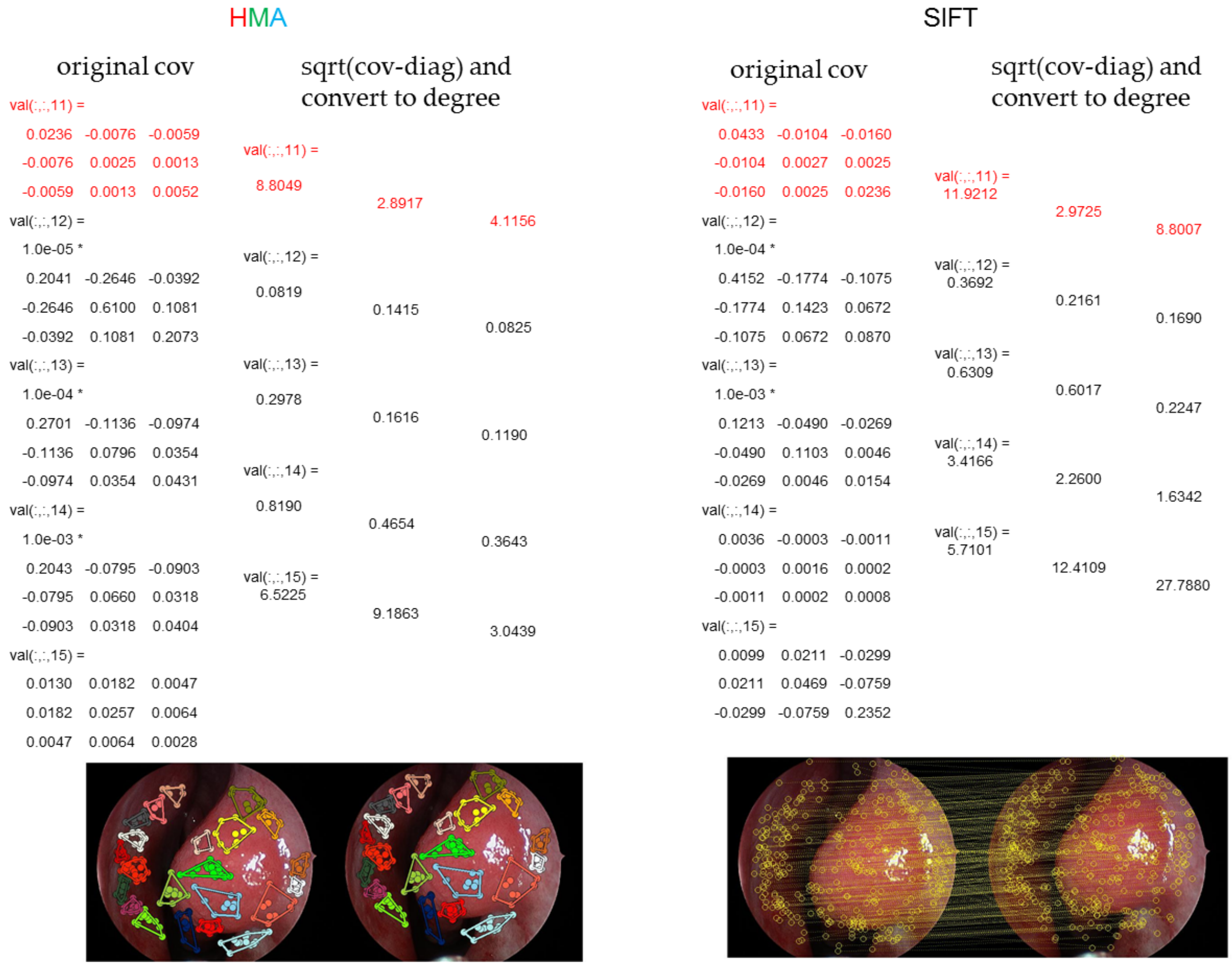
original cov

```
val(:,:,6) =

   0.0660  -0.0056  -0.0382
  -0.0056   0.0055   0.0129
  -0.0382   0.0129   0.0576

val(:,:,7) =

   0.0486   0.0076  -0.0268
   0.0076   0.0013  -0.0042
  -0.0268  -0.0042   0.0150

val(:,:,8) =

   0.0102   0.0029   0.0068
   0.0029   0.0017   0.0044
   0.0068   0.0044   0.0806

val(:,:,9) =

   0.0010  -0.0024  -0.0006
  -0.0024   0.0078   0.0015
  -0.0006   0.0015   0.0004

val(:,:,10) =

  1.0e-04 *

   0.1759  -0.0243  -0.0525
  -0.0243   0.0756   0.0311
  -0.0525   0.0311   0.0800
```

sqrt(cov-diag) and convert to degree

```
val(:,:,6) =
  14.7233
                 4.2671
                              13.7466

val(:,:,7) =
  12.6315
                 2.0301
                              7.0096

val(:,:,8) =
  5.7874
                 2.3753
                              16.2643

val(:,:,9) =
  1.8245
                 5.0623
                              1.1821

val(:,:,10) =
  0.2403
                 0.1575
                              0.1620
```

Figure 16: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 10, frame 11).


HMA
SIFT
original cov
sqrt(cov-diag) and convert to degree
original cov
sqrt(cov-diag) and convert to degree

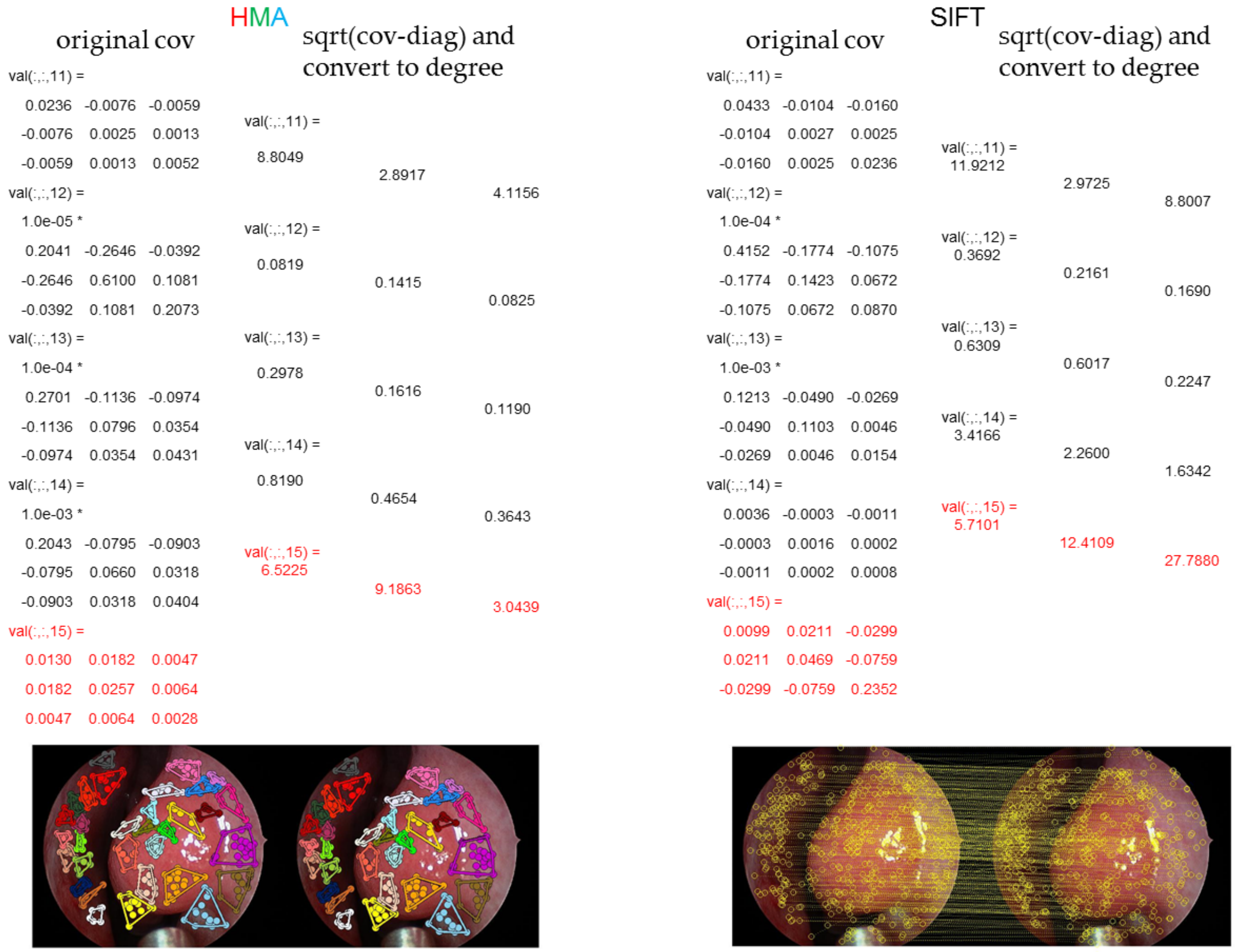
```
val(:,:,11) =

   0.0236  -0.0076  -0.0059
  -0.0076   0.0025   0.0013
  -0.0059   0.0013   0.0052
val(:,:,12) =
  1.0e-05 *
   0.2041  -0.2646  -0.0392
  -0.2646   0.6100   0.1081
  -0.0392   0.1081   0.2073
val(:,:,13) =
  1.0e-04 *
   0.2701  -0.1136  -0.0974
  -0.1136   0.0796   0.0354
  -0.0974   0.0354   0.0431
val(:,:,14) =
  1.0e-03 *
   0.2043  -0.0795  -0.0903
  -0.0795   0.0660   0.0318
  -0.0903   0.0318   0.0404
val(:,:,15) =
   0.0130   0.0182   0.0047
   0.0182   0.0257   0.0064
   0.0047   0.0064   0.0028
```

```
val(:,:,11) =
  8.8049
            2.8917
                      4.1156
val(:,:,12) =
  0.0819
            0.1415
                      0.0825
val(:,:,13) =
  0.2978
            0.1616
                      0.1190
val(:,:,14) =
  0.8190
            0.4654
                      0.3643
val(:,:,15) =
  6.5225
            9.1863
                      3.0439
```

```
val(:,:,11) =
   0.0433  -0.0104  -0.0160
  -0.0104   0.0027   0.0025
  -0.0160   0.0025   0.0236
val(:,:,12) =
  1.0e-04 *
   0.4152  -0.1774  -0.1075
  -0.1774   0.1423   0.0672
  -0.1075   0.0672   0.0870
val(:,:,13) =
  1.0e-03 *
   0.1213  -0.0490  -0.0269
  -0.0490   0.1103   0.0046
  -0.0269   0.0046   0.0154
val(:,:,14) =
   0.0036  -0.0003  -0.0011
  -0.0003   0.0016   0.0002
  -0.0011   0.0002   0.0008
val(:,:,15) =
   0.0099   0.0211  -0.0299
   0.0211   0.0469  -0.0759
  -0.0299  -0.0759   0.2352
```

```
val(:,:,11) =
  11.9212
            2.9725
                      8.8007
val(:,:,12) =
  0.3692
            0.2161
                      0.1690
val(:,:,13) =
  0.6309
            0.6017
                      0.2247
val(:,:,14) =
  3.4166
            2.2600
                      1.6342
val(:,:,15) =
  5.7101
            12.4109
                      27.7880
```

Figure 17: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 11, frame 12).



HMA

original cov

```
val(:,:,11) =
  0.0236  -0.0076  -0.0059
 -0.0076   0.0025   0.0013
 -0.0059   0.0013   0.0052
val(:,:,12) =
 1.0e-05 *
  0.2041  -0.2646  -0.0392
 -0.2646   0.6100   0.1081
 -0.0392   0.1081   0.2073
val(:,:,13) =
 1.0e-04 *
  0.2701  -0.1136  -0.0974
 -0.1136   0.0796   0.0354
 -0.0974   0.0354   0.0431
val(:,:,14) =
 1.0e-03 *
  0.2043  -0.0795  -0.0903
 -0.0795   0.0660   0.0318
 -0.0903   0.0318   0.0404
val(:,:,15) =
  0.0130   0.0182   0.0047
  0.0182   0.0257   0.0064
  0.0047   0.0064   0.0028
```

sqrt(cov-diag) and convert to degree

```
val(:,:,11) =
 8.8049    2.8917    4.1156
val(:,:,12) =
 0.0819    0.1415    0.0825
val(:,:,13) =
 0.2978    0.1616    0.1190
val(:,:,14) =
 0.8190    0.4654    0.3643
val(:,:,15) =
 6.5225    9.1863    3.0439
```

SIFT

original cov

```
val(:,:,11) =
  0.0433  -0.0104  -0.0160
 -0.0104   0.0027   0.0025
 -0.0160   0.0025   0.0236
val(:,:,12) =
 1.0e-04 *
  0.4152  -0.1774  -0.1075
 -0.1774   0.1423   0.0672
 -0.1075   0.0672   0.0870
val(:,:,13) =
 1.0e-03 *
  0.1213  -0.0490  -0.0269
 -0.0490   0.1103   0.0046
 -0.0269   0.0046   0.0154
val(:,:,14) =
  0.0036  -0.0003  -0.0011
 -0.0003   0.0016   0.0002
 -0.0011   0.0002   0.0008
val(:,:,15) =
  0.0099   0.0211  -0.0299
  0.0211   0.0469  -0.0759
 -0.0299  -0.0759   0.2352
```

sqrt(cov-diag) and convert to degree

```
val(:,:,11) =
 11.9212   2.9725    8.8007
val(:,:,12) =
 0.3692    0.2161    0.1690
val(:,:,13) =
 0.6309    0.6017    0.2247
val(:,:,14) =
 3.4166    2.2600    1.6342
val(:,:,15) =
 5.7101    12.4109   27.7880
```

Figure 18: Comparison of the estimated covariance matrix by HMA matching vs. SIFT matching. Images shown for the pair (frame 15, frame 16).