Air-Ground Image Matching for MAV Urban Localization

CPSC 515 Project, Dec 21, 2015

Minchen Li
Department of Computer Science
The University of British Columbia
minchenl@cs.ubc.ca

Jianhui Chen
Department of Computer Science
The University of British Columbia
jhchen14@cs.ubc.ca

Abstract

In this project, we use a vision based method to globally localize a Micro Aerial Vehicle (MAV) flying in GPS shadowed areas such as urban areas and university campuses. A large database consisted of Google Street View images with their location tags is used as the global map, where we match the MAV captured images to the map images by recognizing visually similar and geometrically consistent discrete places. By adding global geometric constraints to the air-ground matching method proposed by [12], our method achieves a higher recall rate at precision 1.0 *on their Zurich dataset. We also compared the two methods on a preliminary UBC dataset collected by ourselves. They performed similarly and both generated reasonable results.*

1. Introduction

Robot localization is a fundamental problem in computational robotics. It solves the problem of "where is the robot" by observing the environment and reasoning. Different application domains use a number of sensors such as cameras, GPS-based equipment, rangefinders and so on. Using GPS, the localization accuracy may fluctuate due to changes in signal strength, especially in urban areas. Cameras are complementary as they are cheap and light. As a result, vision-based methods have been broadly used in micro robots such MAVs.

In this project we globally localize a MAV in urban environments using images captured by an onboard camera. The global position of the MAV is recovered by recognizing visually-similar discrete places in the map. Specifically, the MAV image is searched in a database of ground-based geotagged images such as Google Street View images. Because of the large difference in viewpoint between the air-level and ground-level images, this problem is called air-ground matching [12].

The air-ground image matching is critical in a number of applications. For example, autonomous flying vehicles operate in urban environments where GPS signal is shadowed or completely unavailable. Moreover, the correctly matched image pairs can be further used to reconstruct 3D geometry of streets and buildings. Because the MAV images provide unique viewpoints that are very different from ground-level images, air-ground image pairs can provide more details for 3D reconstruction than general ground-ground image pairs.

The research in air-ground image matching has become popular in recent years as prices of MAVs drop. The air-ground image matching use large image databases with geotags as a global map. There are several publicly available such as Google Street View map, Bing Map and baidu map. Using geotagged images as potential locations, the localization problem becomes an image matching problem.

The challenging of air-ground image matching is on many facts such as large viewpoint difference, over-season variances, lens distortion and object occlusion. To overcome these challenges, an air-ground image matching method has been proposed by Majdik *et al*. [12]. In this project, our implement [12] and propose two global geometric constraints to improve the image matching correctness. We also collect our own dataset from Google Street View and testing images using a quadcopter on the UBC Vancouver campus. To summary, the contributions of this project are:

- Implement air-ground matching algorithm [12] as baseline
- Apply epipolar and sidedness constraints to the baseline to improve correctness
- Collect UBC dataset and conduct preliminary experiment

2. Related work

Our problem is related to city scale camera localization, and our proposed method is related to geometric consistency constraints.

City scale camera localization Badino *et al*. [1] described a method for long-term vehicle localization based on visual features alone. Their method utilizes a combination of topological and metric mapping to encode both the topology and metric of the route. First, a topometric map is created by driving the route once and recording a database of visual features. Then the vehicle is localized by matching features to this database at runtime. To make the localization more reliable, they employed a discrete Bayes filter to estimate the most likely vehicle position using evidence from a sequence of images along the route. Badino *et al*. [2] improved the speed of the system using a new global visual feature Whole Image SURF (WI-SURF) descriptor. With the new feature, they reduced the data storage requirements by a factor of 200 and increased the speed of the localization by more than 15 times, thereby enabling real-time (5 Hz) global localization. Their system achieved an average localization accuracy of 1 m over 8 km route. The limitation of the system is that users have to collect their own database in map creation stage.

Other researchers proposed systems that use publicly available data as the map. For example, Zamir *et al*. [21] proposed an accurate image localization method based on Google Maps Street View. First, they build a structured database using Google Maps Street View image with GPS-tags. Then, they use SIFT feature of the given image to query the database, pruning the matching using image group localization. The database size is 100, 000, and the number of query images is 521. The experiment showed that about 60% of the test set is localized to within less than 100 meters of the ground truth. Castano *et al*. [19] extended image localization method to estimate geo-spatial trajectory of a video camera. They use Bayesian tracking framework and Minimum Spanning Trees (MST) to reconstruct the trajectory. They test the method in 15 videos which covers downtown Pittsburgh and downtown Orlando. The average localization error value is 10.57 meters. Vision based image matching method is successful in above systems as the ground-ground RGB images provide sufficient number of distinctive point features.

Majdik *et al*. [12] tackled the problem of globally localizing a camera-equipped MAV flying within urban environments using a Google Street View ground image database. They first use affine SIFT (ASIFT) [16] to overcome the severe viewpoint changes between the MAV images and ground images. Then they use K Virtual Line Descriptor (KVLD) [10] to pair the MAV and ground images. The database is 113 images along a 2km trajectory. The query images are 405 MAV image. Their method achieved about 45% recall rate when the precision is 1. They also extended the method to MAV position tracking using textured 3D cadastral models [13, 14].

Geometric consistency constraints in image matching Given initial matching points between two images from appearance-based point feature matching method such as [11, 16, 4], geometric consistency constraint is an effective way to filter outliers.

For two images I and I', image matching is to find pixel-to-pixel correspondences between two images. Let $X = [x, y, 1]^T$ and $X' = [x', y', 1]^T$ are correctly matched pixels in I and I', respectively. X and X' must satisfy some constraints depends on the camera configurations, and the geometric property of projected objects.

The types of geometric consistency has been widely used in point feature based image matching. When planar objects only rotate and non-isotropic scale between two views, the matched points satisfy affine transformation:

$$X' = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} X \tag{1}$$

The constraint has six degrees of freedom and requires at least three points. In practice, local feature points on planer objects approximately satisfy affine transformation constraint. For example, Schaffalitzky *et al*. [18] used affine transformation to increase the feature point matching guided by affine transformation in local image areas.

When two cameras capture a planer object or the two cameras purely rotate, the matched feature points in two images satisfy homography transformation

$$X' = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} X \tag{2}$$

The constraint has eight degrees of freedom and requires at least four points. The homography constraint has been widely used in sports video calibration [11, 7].

When two camera capture general objects, two matched feature points satisfy the epipolar constraint:

$$X'^T F X = 0 \tag{3}$$

in which F is a 3×3 matrix with rank of two. In stead of point-to-point matching, fundamental matrix constraint is point-to-line matching. The constrained line is the epipolar line FX. As a result, this constraint is called epiploar constraint. The epipolar constraint requires at least eight points in general case [8].

RANSAC style outlier filtering method such as [6] and ORSA [15] can filter outliers using above geometric constraints [9]. The efficiency of outlier filtering methods mainly depends on the complexity of the model, the number of initial matchings and the ratio of inliers.

Algorithm 1 Vision based geometrically-consistent localization of MAVs

Input: A set $I = I_1, I_2, \cdots I_n$ of ground geotagged images
Input: An aerial image I_a taken by a drone in urban area
Output: The location of the drone in the discrete map, respectively the best match I_b^*

1: F = database of all the image features of I.
2: **for** `i ←1 to n` **do** // generate ASIFT
3: V_i = generate a set of virtual views $\{I_{i1}, \cdots, I_{ik}\}$
4: F_i = extract a set of image features $\{F_{i1}, \cdots, F_{ik}\}$
5: add F_i to F
6: **end for**
7: // up to this line the algorithm is computed off-line
8: V_a = generate a set of virtual views $\{I_{a1}, \cdot, I_{ak}\}$
9: F_a = extract a set of image features $\{F_{a1}, \cdot, F_{ak}\}$
10: $n_v \leftarrow$ number of virtual views
11: $M^{kvld} = \emptyset$
12: **for** `c ←1 to n` **do** // for each database image
13: initial point matching $M = \emptyset$
14: **for** `i ←1 to` n_v **do**
15: **for** `j ←1 to` n_v **do**
16: //inlier points:
17: $N_j = KVLD(F_{ai}, F_{cj}, I_a, I_c)$
18: $M \leftarrow M + N_j$
19: **end for**
20: $M^{kvld} \leftarrow M^{kvld} + M$
21: **end for**
22: **end for**
23: $I_b^{baseline} = max(M_i^{kvld})$ // baseline
24: select inlier points: M_e = $epipolar(M^{kvld})$
25: $I_b^e = max(M_i^e)$ // baseline + epipolar
26: select inlier points: $M_{sidedness}$ = $sidedness(M^{kvld})$
27: $I_b^s = max(M_i^s)$ // baseline + sidedness

3. Vision based geometrically-consistent global localization

Algorithm 1 is the psuedo-code descriptoin of our method. The input and output of the algorithm formulate our problem. For a given image database and a query image, our problem is to find one image in the basebase that has most similar visual appearance and obey the global geometric consistency.

The algorithm has three parts. The first part is from line 1 to line 22. It is a modified version of the air-ground matching method [12]. The second part is applying epipolar constraint to point matching result from KVLD (line 23). The third part is applying sidedness constraint to point matching result from KVLD (line 25). We briefly describe each steps here. Section 4 provides more details on each steps.

Baseline Because there is significant changes in viewpoint between the ground image and the MAV image, the baseline first uses ASIFT [16] to increase the number of initial feature matching. ASIFT assumes that the object in the scene is planar, and simulates the affine transformation of the object by the affine transformation of the image. Because ASIFT is based on SIFT, understanding the limitation of SIFT in large viewpoint changes is important. Lowe [11] pointed out that the repeatability of SIFT feature drops when the viewpoint difference between two views is larger than 40^o. As a complementary method, ASIFT simulates the tilt angle from 45^o to about 70^o. As a result, ASIFT is scale invariant and almost affine invariant at the cost of massive computation in feature extraction and initial matching.

In fact, ASIFT increases both inlier and outlier feature matches at the same time, where the inlier rate is still lower than 50%. Therefore, it is very important to filter out the outliers for the deduction of image matching to be reasonable. However, the traditional RANSAC method fails or requires extremely high computation since the inlier rate is too low in air-ground image pairs.

To overcome this challenge, the baseline paper adapted a semi-local constraint based filtering method KVLD [10], which can quickly eliminate most of the false feature matches under a near zero inlier rate. The key idea of KVLD is that on the two matched virtual lines formed by two feature matches, there should be similar photometric information. A feature match will be filtered out if it does not have enough neighbor matches where they form good similar virtual lines. However, KVLD contains many parameters that are set by its author without clear explanation and experiments. These parameters might make KVLD quite unstable, i.e. stricter parameters will probably increase the filtering precision while decrease the recall. Majdik *et al.* [12] did not specify how they exactly used KVLD. In our implementation, we use the default parameters as released code of [10]. However, we believe that further tuning the parameters can improve the performance.

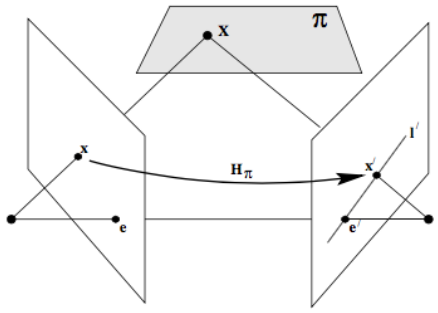

Figure 1. *Epipolar constraint. x and x' are the projection of X in two views. The epipolar line $l' = Fx$ where F is the foundamental matrix. x' lie on l' so that $x'l' = x'Fx = 0$. Figure from [9].*

Epipolar constraint Fig. 1 shows the epipolar constraint between two arbitrary views. In perspective projection, Fx describes a line (the epipolar line) on which the corresponding point x on the other image must lie. We add epipolar constraint to the baseline based on our observation of KVLD matching result. As shown in Fig. 4 (b), the KVLD achieves sufficient number of initial point feature matches, and the inlier ratio is roughly close to 50%. In that condition, applying epipolar constraint can effective remove false point feature matches and eventually improve the recall rate.

Sidedness constaint The sidedness constraint is inspired by the work of Bay *et al*. [3]. For a triplet of feature matches, the center of a feature should lie on the same side of the directed line going from the center of the second feature to the center of the third feature in both views. This constraint always holds if the three features are coplanar in the scene, and it also works in the vast majority of general 3D scenes.

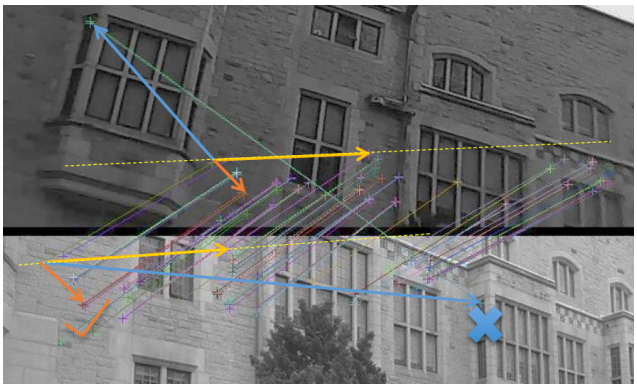

Figure 2. *Illustration of the sidedness constraint applied to feature matches. The little crosses and thin lines mark the feature points and their matches in the image pair. The thick arrows connect feature points, where the feature pointed by the orange arrow satisfies the sidedness constraint w.r.t. the yellow line, while the blue one does not satisfy.*

Fig. 2 explains how sidedness constraint works for feature matches. Given the triplet of feature matches, 2D vectors could be formed to define the direction of the lines. Then we just compare the cross products of the lines in the two images. If they are in the same direction, then the feature point is considered sidedness consistent.

4. Implementation

We implemented the ASIFT and KVLD based on the original papers.

4.1. ASIFT

Affine-SIFT (ASIFT) [16] is a modified version of SFIT feature detection method. It has two steps. In the first step, it simulates the affine transform using a combination of planar tilt and rotation angles. In the second step, it uses standard SIFT feature to detect point features. Because robust implementations of SIFT are publicly available (OpenCV [5] and VLFeat [20]), the implementation of ASIFT is mainly in the first step.

The affine transformation matrix has six degrees of freedoms (DOF):

$$H_A = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_x \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Let $A = [a_{11}, a_{12}; a_{21}, a_{22}]$ and decompose A as in [9]

$$A = R(\theta)R(-\phi)\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R(\phi) \tag{5}$$

Because the affine transformation in this problem is from one image (the original image) to another image (the warped image) and SIFT feature is scale-invariant, the simulation can fix one of the scales (λ_1 and λ_2) as 1. Without loss of generality, we set $\lambda_1 = 1$. As a result, the affine transformation only has five degrees of freedom in this problem. We decompose this transform to Euclidean transformation (3 DOF) and non-isotropic scalings (2 DOF):

$$H_A = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

We believe this decomposition is simpler than the one in the original paper [16]. In the experiment, we found that this decomposition achieves the same image warping result as the original paper.

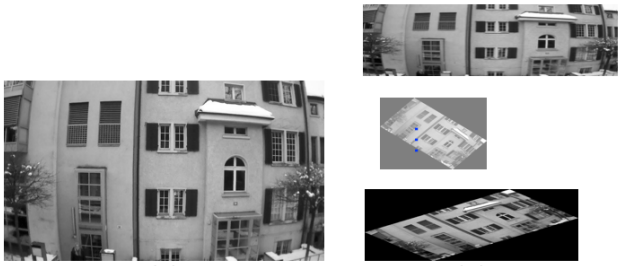

Figure 3. *(a) original image; (b) images after affine transformation (3 of total 9 images). The warped images are used to simulate the affine transformation.*

Fig. 3 shows the original image and affine transformed images (3 in total 9). Fig. 4 shows the SIFT matching result and ASIFT matching result for a positive MAV-ground image pair. ASIFT has more initial matches (61 vs 28) than SIFT.

(a) (b)

Figure 4. *(a) SIFT has 28 initial point matches; (b) ASIFT has 61 initial point matches. ASIFT increases the number of point feature matches.*

4.2. KVLD

The k-connected VLD-based matching method (KVLD) [10] is basically an image feature match filtering algorithm which applies a semi-local constraint based on virtual line descriptor (VLD) on the matches. The core idea is that for points P_i, P_j in image I and $P'_{i'}$, $P'_{j'}$ in image I', it is unlikely to find similar photometric information around lines (P_i, P_j) and $(P'_{i'}, P'_{j'})$ unless both $(P_i, P'_{i'})$ and $(P_j, P'_{j'})$ are correct matches (see Fig. 5). To measure the feature of

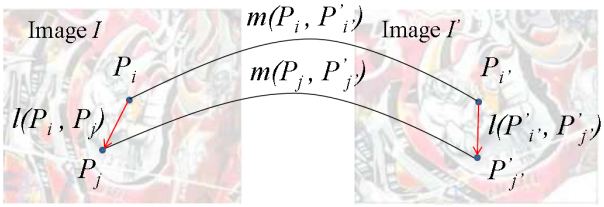

Figure 5. *Lines (P_i, P_j) and $(P'_{i'}, P'_{j'})$ are unlikely to be similar unless both matches $(P_i, P'_{i'})$ and $(P_j, P'_{j'})$ are correct.* Figure from [10].

the virtual line between two feature points in the same image, the virtual line descriptor (VLD) is defined to capture photometric information:

Line covering Between the two feature points in a same image, there are other u SIFT-like feature points covering the line between them. Each of these SIFT-like feature points has V SIFT-like gradient histograms, 1 orientation descriptor and 1 orientation normalizing factor. The gradient histogram is voted by each pixel in the disk:

$$(h_{u,v})_{v\in\{1,\dots,V\}} \tag{7}$$

The orientation descriptor is the angle between the most voted gradient direction and the direction of the virtual line:

$$w^*_u = \underset{w\in\{0,\dots,W-1\}}{\operatorname{argmax}} \hat{O}_{u,w} \tag{8}$$

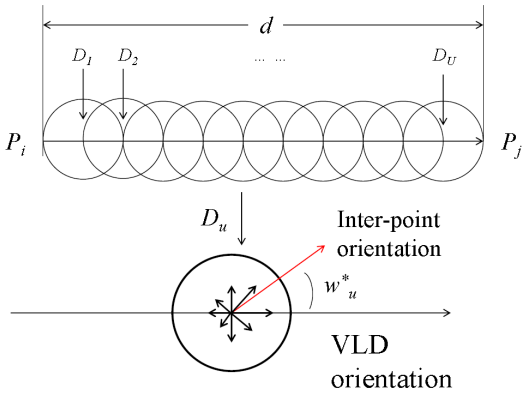

Figure 6. *Top: disk covering of line (P_i, P_j). Bottom: 8-bin histogram of gradient orientation for disk D_u, and main orientation w^*_u.* Figure from [10].

The orientation normalizing factor is calculated by the histogram score of each disk orientation:

$$\gamma_u = \frac{\hat{O}_{u,w^*_u}}{\sum_{u=1}^{U} \hat{O}_{u,w^*_u}} \geqslant 0 \tag{9}$$

Distance measurement The distance measurement between two VLDs considers each disk's gradient histogram difference and main orientation difference, where the latter one is scaled by the orientation factor. The considered two differences are linearly combined:

$$\tau(l,l') = \beta \sum_{u=1}^{U}\sum_{v=1}^{V} |h_{u,v} - h'_{u,v}| + (1-\beta)\sum_{u=1}^{U}(D_{orient}) \tag{10}$$

where

$$D_{orient} = \frac{\gamma_u + \gamma'_u}{2} \cdot \frac{min(|w^*_u - w'^*_u|, W - |w^*_u - w'^*_u|)}{W/2} \tag{11}$$

Two matches are said to be VLD-consistent if the distance between two matched virtual lines are smaller than a threshold $\tau_{max} = 0.35$.

Geometric consistency With rigid transformation, a correspondence of feature point P_j could be computed from the other feature correspondence $(P_i, P'_{i'})$ on the matched virtual lines (see Fig. 7). The scale s and orientation a of the other feature correspondence and the vector P_iP_j are used to compute the transformation:

$$Q'_j = P'_{i'} + \frac{s(P'_{i'})}{s(P_i)} R(a(P'_{i'}) - a(P_i))\vec{P_iP_j} \tag{12}$$

The error is defined to be the distance between $Q'_{j'}$ and $P'_{j'}$. The geometric consistency criterion is constructed by con-

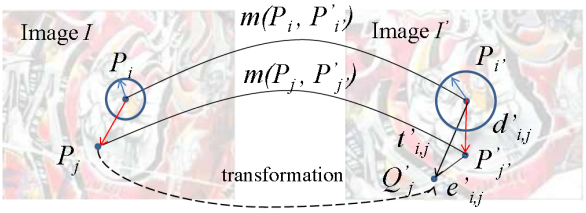

Figure 7. *Distances used for the computation of the transformation error* $\eta_{i,i',j,j'}$. Figure from [10].

straining the error relative to the distance between the feature points:

$$\chi(m_{i,i'}, m_{j,j'}) = min(\eta_{i,i',j,j'}, \eta_{j,j',i,i'}) < \chi_{max} = 0.5 \tag{13}$$

where

$$\eta_{i,i',j,j'} = \frac{e_{i,j}}{min(dist(P_i, P_j), dist(P_i, Q_j))} \tag{14}$$

Two matches are said to be gVLD-Consistent iff they are both geometry- and VLD-consistent.

Algorithm The algorithm is described in Algorithm 2, where we maintain the main idea while alter the program structure to make it faster and more concise. It is based on the KVLD open source library[1] shared by the author Zhe Liu. We also applied Intel TBB[2] to parallelize the tasks.

5. Experiments

In this section, we compare three methods on two dataset.

Baseline We implemented [12] as our baseline. The baseline has three steps. The first step is using ASIFT to increase the initial matching numbers. The second step is putative match selection using 1-point-ransac method [17]. The third step is using kVLD to pair and accept good matches as the final matching result. We implemented the first and the third step. We have tried to implement the second step. However, we found that our implementation of the 1-point-ransac method did not give satisfying result. As a result, we did not put the 1-point-ransac method in the Baseline. Lacking the 1-poing-ransac makes the program much slower than the author's implementation. However, it does not effect the matching correctness as we found in the experiment.

Baseline + epipolar constraint This method is based on the baseline method. At the end of the baseline method, each candidate pair has matched points from kVLD. We use epipolar constraint (Eq. 3) with RANSAC to filter outliers. If the number of inliers is larger than a threshold, the image pair is accepted as true positive.

Algorithm 2 KVLD feature match filtering

Input: Image I and I' and their feature points F_I and $F_{I'}$
Input: Initial matches $M_{initial}$ between image I and I'
Output: Filtered matches M_{KVLD} between I and I'

```
Initialize variables
//calculate gVLD-consistency scores:
for m_{i,i'} ← M_{initial} do
    for m_{j,j'} ← M_{neightbor,i} do
        if gVLDConsistent(m_{i,i'}, m_{j,j'}) then
            score_i += gVLDScore(m_{i,i'}, m_{j,j'})
            score_j += gVLDScore(m_{i,i'}, m_{j,j'})
            gVLDCount_i++
            gVLDCount_j++
        end if
    end for
end for
//remove matches according to gVLD-consistency criterion:
for m_{i,i'} ← M_{initial} do
    if gVLDCount_i < thresh_{count} then
        remove m_{i,i'}
        continue
    end if
    if score_i < thresh_{score} then
        remove m_{i,i'}
    end if
end for
//remove redundant matches by only keeping the one with best gVLD-consistency score:
for f_i ← F_I do
    for f'_{j'} ← M_i do
        Find m(i,i') with gVLDScore_{max}
    end for
    Add m(i,i') into M_{KVLD}
end for
```

Baseline + sidedness constraint This method is also based on the baseline method. Instead of using epipolar constraint, we use sidedness constraint (Sec. 3) to filter outliers.

Please note that, all the three methods are tested on the same environment. Because our implementation of the baseline method is not exactly the same as the author's, we are unable to directly compare our result with author's result. However, we roughly compared the recall ratio at precision 1 with author's result. The two results are very sim-

[1] https://github.com/Zhe-LIU-Imagine/KVLD
[2] https://www.threadingbuildingblocks.org/

ilar: ours is 44.6% and author's is about 45%. It means that the accuracy of our implementation is very close to the original method.

The experiment is taken on two dataset.

Zurich dataset This dataset is publicly available from Majdik *et al*. [12]. It is composed of 405 MAV images and 113 geotagged Google Street View images, spanning 2km wide area in Zurich. The images have a lot of challenging characteristics. Besides the big difference in view point, there are strong illumination and over-season variations since the MAV images don not contain snows while most of the Google Street View images do. Moreover, the lens distortion also differs from the MAV camera and the ground camera.

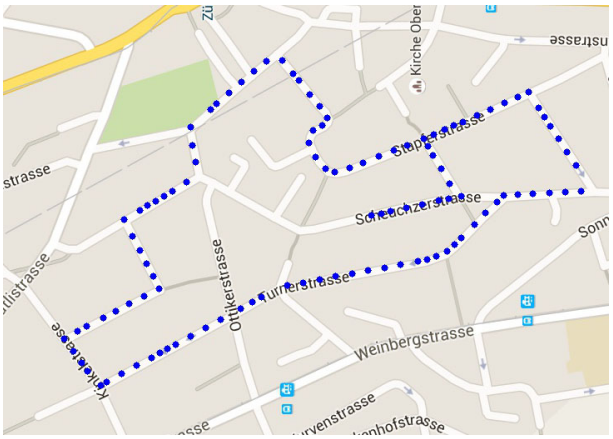

Figure 8. *Birds-eye view of the area that images in Zurich dataset spans. The blue dots mark all the locations of the ground Google Street View images.*

UBC dataset The UBC dataset is collected by ourselves. We used an entry level MAV to capture videos for several buildings on UBC Vancouver Campus. Then we extracted frames without severe motion blur for four locations: ICICS building, Kaiser, biology building and library building. For the ground image database, we developed a Google Street View image crawler using Java to automatically download images on one side of a straight path with their geographic information each time by specifying latitude and longitude coordinates of the path start and end. The MAV is Hubsan X4 H107C which costs about 100 USD. It has a narrow field of view (FOV) 720p camera.

Our UBC dataset currently contains 20 MAV images and 288 geotagged Google Street View images where their spanning area is also approximately 2km wide. They also have some challenging characteristics, especially that the Google Street View images are somehow out of date. Those street images were taken around 2 to 3 years ago, when many of the present buildings and lawns did not exist.

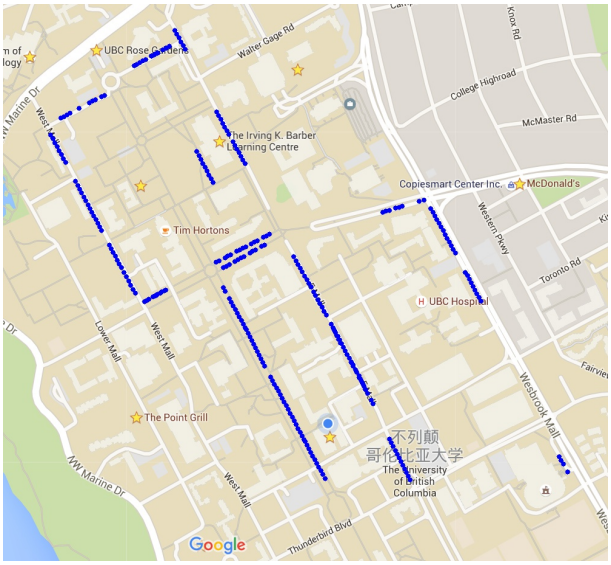

Figure 9. *Birds-eye view of the area that images in our UBC dataset spans. The blue dots mark all the locations of the ground Google Street View images.*

Method	Dataset
	Zurich
Baseline	44.6
Baseline + epipolar	**54.1**
Baseline + sidedness	47.0

Table 1. **Recall rate at precision 1**. Our method (baseline + epipolar) outperforms the baseline method [12] about 10% at Zurich dataset. The best performance in each dataset is highlighted by bold.

We tried to avoid collecting data from those completely changed spots, while there are still variations of plants and building appearances somewhere.

5.1. Result

We adopt recall rate at precision 1 as our localization performance measure. Table 1 shows the recall rate for three methods at the Zurich dataset. The baseline + epipolar constraint achieves the best performance at 54.1%, outperforming the baseline method about 10%. The baseline + sidedness method is slightly better than the baseline method. The result confirms our expectation that the global geometric constraint can effectively remove some of the false positives so that improve the recall rate.

We also visualized the correctly matched street view image locations and compared with author's result in Fig. 11. Our method matches more street view locations than author's result (77 vs 65).

We also test the three methods on the UBC dataset. However, we do not report the recall rates at the current stage,

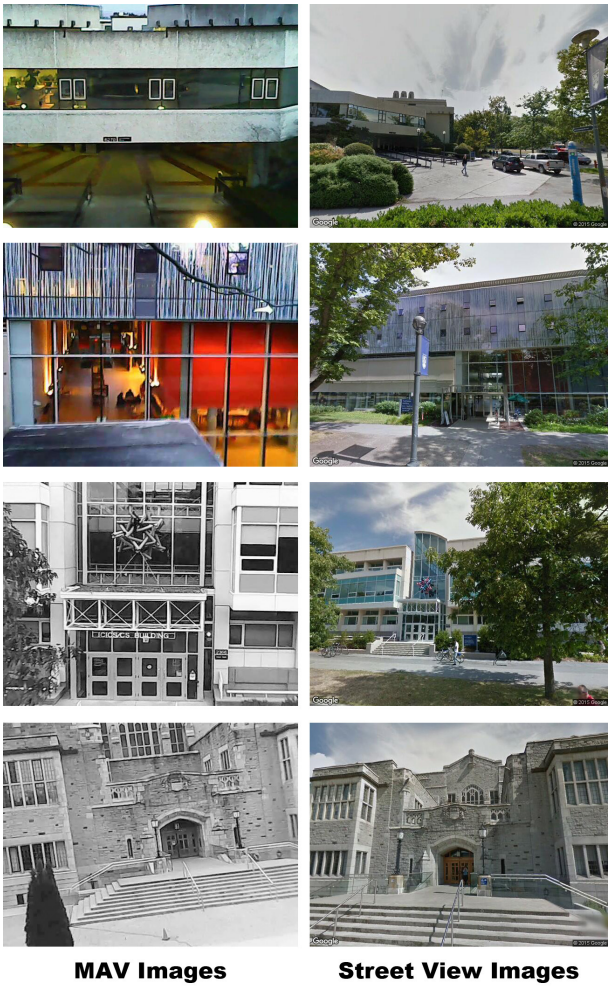

Figure 10. *Comparison between airborne MAV (left) and ground-level Google Street View images (right) in our current UBC dataset. Note the significant changes - in terms of viewpoint, illumination, lens distortions, and scene between the query (left) and the database images (right) - that obstruct their visual recognition and matching.*

because:

1. The number of testing images is 20 which is too small to get a reasonable precision-recall curve.
2. The image matching difficulty is not evenly distributed. The difficulty is either too easy so that all methods match them correctly or too difficult so that all methods fail.

For the UBC dataset, all methods achieve 40% correctness rate on the 20 MAV images. If we qualitatively analyze the correctly matched (see Fig. 12 for example) and mis-matched (see Fig. 13 for example) image pairs, we can found that the mis-matched pairs either depict a window of which the appearance totally depends on the scenes inside and changes dramatically among different view points, or only contain line featured textures which are considered unstable by SIFT feature detector, while the correct matches obviously are just consisted of common types of walls and windows.

From Fig. 13 we can also see that the building features in the air images are mainly mis-matched to the floor features in the ground images, which suggest that it might be better for us to cut the floor part of the ground images or lift up the camera a little bit while collecting the street views, which are quite reasonable because distinctive features are located on buildings . Actually, we neither want the floor nor the sky, so narrowing down the vertical view angle might also be a necessary improvement. Moreover, there should be considerable distance between the MAV and the building so that the air images could contain enough scenes and features for matching.

5.2. Qualitative comparison

KVLD v.s. KVLD + epipolar Fig. 14 and Fig. 15 show the effectiveness of epipolar constraint in removing false positives. Fig. 14 compares the point feature matching result of KVLD and KVLD + epipolar constraint on a true positive pair. Because most of the initial KVLD matches obeys the epipolar constraint, the number of inliers after applying the epipolar constraint is still high (30) with the inlier ratio of 50%. It is a strong indication of true positive. On the contrary, Fig. 15 shows a false positive example. Because most of the initial KVLD matches does not obey the epipolar constraint, the number of inliers is only 15 with the inlier ratio of 33%, which indicates a false positive. In this way, epipolar decreases the number of false positives and keeps the true positives at the same time.

KVLD v.s. KVLD + sidedness Now we compare our sidedness constrained result with the baseline result in a different point of view. Because sidedness constraint is also a strong geometric constraint, it is able to filter out many incorrect feature matches provided that the inlier rate is high enough, which is supported by KVLD. Fig. 16 shows an example of this phenomenon, where even the photometrically reasonable false feature matches are successfully eliminated. Since sidedness constraint does not apply to all the situations, it may also filter out some correct matches sometimes (see Fig. 17 for example). However, sidedness constraint is heuristic because it does help to improve the recall rate of the baseline result (shown in Fig. 5). The reason is that it is satisfied by the correct matches in most of the situations. Besides, sidedness constraint is very fast compared to the epipolar constraint since sidedness computation does not need to iteratively calculate the transformation of feature points.

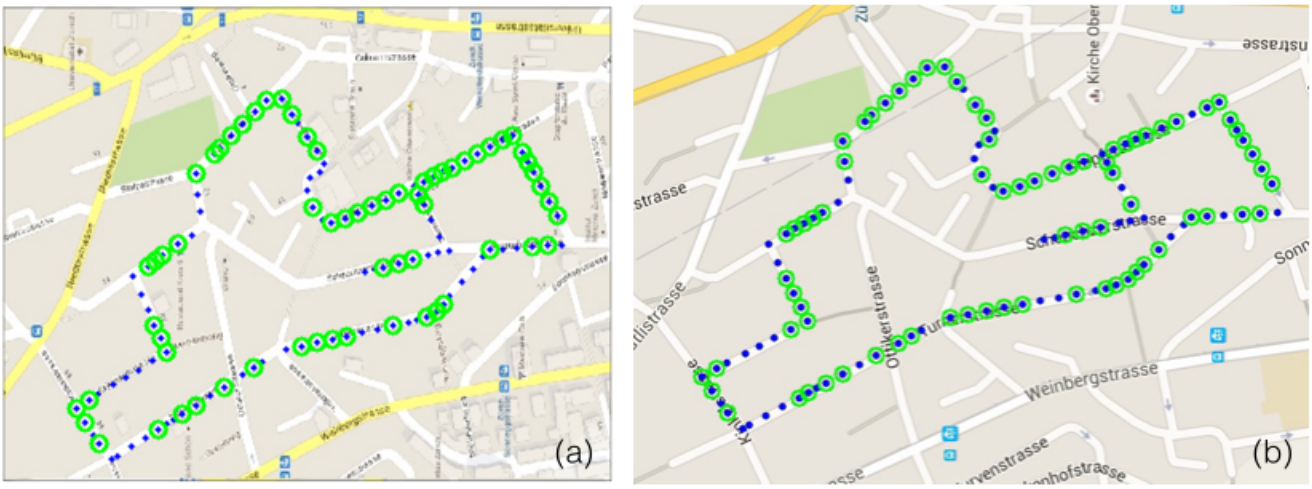

Figure 11. *Birds-eye view of the test area. (a) from [12]; (b) our result. The blue dots mark the locations of the ground Google Street View images. The green circles represent those places where the aerial images taken by the urban MAV were successfully matched. Our method matches more street view locations than author's result (77 vs 65).*

Figure 12. *Examples of good baseline results on our UBC dataset.*

Figure 13. *Examples of bad baseline results on our UBC dataset. These two pairs are just the 1st and 2nd row in Fig. 10, where those two rows show the correct matches.*

6. Discussion

We proposed a vision based geometrically-consistent global localization method for air-ground image matching. The result is tested on a public dataset and our own dataset. The preliminary result shows that our method improves the correctness of the image matching in challenging test set.

Due to lack of time, there are a number of noticeable limitations in our project. For example, we did not successfully implement the 1-point-ransac so that the system has to search a large number of candidate images. Also we did not collect enough MAV images for the UBC dataset.

Lessons Learned This project provides a valuable learning experience. First, data collection is time consuming and less controllable because of the hardware, weather and field conditions. Collecting enough data at the beginning of the project would benefit the whole project.

Second, speed is important for robotic system. At first, we ignored this problem as the 1-point-ransac method does not work well. Eventually, the speed becomes the bottle neck of the system. It not only decreases the usability of the system but also makes the testing very inefficient.

Third, working in a team, we could discuss and refine a lot of initial ideas. we also could anticipate problems that could become critical of we were working alone.

Table 2 shows the division of work for this project.

7. Conclusion and future work

In this project, we implemented the air-ground matching algorithm from [12] as the baseline. Based on the baseline, we added two global geometric constraints namely epipolar

(a) (b)

Figure 14. *True positive example of KVLD matching. (a) KVLD matching result, (b) KVLD + epipolar constraint result. The number of inliers is 30 (60 total matches) after epipolar constraint. The image matching is accepted.*

(a) (b)

Figure 15. *False positive example of KVLD matching. (a) KVLD matching result, (b) KVLD + epipolar constraint result. The number of inliers is 15 (45 total matches). The image matching is rejected.*

Figure 16. *A comparison between a baseline result (left) and a sidedness constrained result (right) on Zurich dataset. Our constraint is able to eliminate those photometrically reasonable false feature matches while keeping the correct ones.*

constraint and sidedness constraint. The extra global geometric constraints improve the recall rate. We also collected Goolge Street View image for UBC dataset and part of MAV testing images.

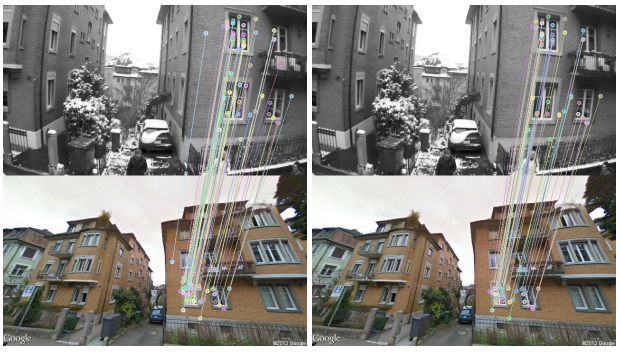

Figure 17. *A comparison between a baseline result (left) and a sidedness constrained result (right) on Zurich dataset. Some correct matches are actually filtered out by our sidedness constraint because this heuristic constraint does not hold in all the situations.*

Task	Minchen	Jianhui
ASIFT	0%	100%
KVLD	100%	0%
Epipolar	0%	100%
Sidedness	100%	0%
UBC data collection	70%	30%
Writing	50%	50%

Table 2. Division of work

At the current stage, the localization result is discrete GPS data. The accuracy is only about 15 meters. In the future, we would like to estimate the continuous trajectory of MAVs using camera position tracking. We also would like to explore applications such as 3D geometry reconstruction using multiple MAVs.

References

[1] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium (IV), IEEE*, pages 794–799, 2011.

[2] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *Robotics and Automation (ICRA), IEEE International Conf.*, pages 1635–1642, 2012.

[3] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conf.*, volume 1, pages 329–336, 2005.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV*, pages 404–417. 2006.

[5] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] A. Gupta, J. J. Little, and R. J. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Computer and Robot Vision (CRV), Canadian Conf.*, pages 32–39, 2011.

[8] R. Hartley et al. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Trans.*, 19(6):580–593, 1997.

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[10] Z. Liu and R. Marlet. Virtual line descriptor and semi-local matching method for reliable feature correspondence. In *British Machine Vision Conf.*, 2012.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[12] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza. Mav urban localization from google street view data. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conf.*, pages 3979–3986, 2013.

[13] A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza. Micro air vehicle localization and position tracking from textured 3d cadastral models. In *Robotics and Automation (ICRA), IEEE International Conf.*, pages 920–927, 2014.

[14] A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza. Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles. *Journal of Field Robotics*, 2015.

[15] L. Moisan, P. Moulon, and P. Monasse. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line*, 2:56–73, 2012.

[16] J.-M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[17] D. Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95(1):74–85, 2011.

[18] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or how do i organize my holiday snaps?. In *Computer Vision–ECCV*, pages 414–431. 2002.

[19] G. Vaca-Castano, A. R. Zamir, and M. Shah. City scale geospatial trajectory estimation of a moving camera. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf.*, pages 1186–1193, 2012.

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

[21] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *Computer Vision–ECCV*, pages 255–268. 2010.