Illumination Invariant Camera Localization Using Synthetic Images

Sota Shoman†  Tomohiro Mashita†  Alexander Plopski‡
Photchara Ratsamee‡  Yuki Uranishi†  Haruo Takemura†

†Osaka University  ‡Nara Institute of Science and Technology

ABSTRACT

Accurate camera localization is an essential part of tracking systems. However, localization results are greatly affected by illumination. Including data collected under various lighting conditions can improve the robustness of the localization algorithm to lighting variation. However, this is very tedious and time consuming. By using synthetic images, it is possible to easily accumulate a large variety of views under varying illumination and weather conditions. Despite continuously improving processing power and rendering algorithms, synthetic images do not perfectly match real images of the same scene, i.e., there exists a gap between real and synthetic images that also affects the accuracy of camera localization. To reduce the impact of this gap, we introduce “REal-to-Synthetic Transform (REST).” REST is an autoencoder-like network that converts real features to their synthetic counterpart. The converted features can then be matched against the accumulated database for robust camera localization. Our results show that REST improves matching accuracy by approximately 30%.

Index Terms: Camera Localization—Variable Lighting—Synthetic View—Autoencoder.

1 INTRODUCTION

Camera localization is an important requirement for displaying a virtual object spatially correct in augmented reality and a variety of other fields like robotics. To solve this, a variety of approaches that use features extracted from the scene [1] or neural networks [2] have been developed over the years. Both approaches require a large number of images that cover the expected scene appearance to recover a representation of the scene. However, as the appearance over the course of a day, under different weather conditions, and time of the year, it is difficult and time-consuming to acquire a database that covers sufficient variety of appearance. Typically, a cost-efficient solution is to generate synthetic views of the scene instead. Although synthetic image databases offer much flexibility in deciding scene parameters, a synthetic scene will rarely match the appearance of the real scene when captured by a camera. This is, in part, because it is difficult to obtain accurate geometric and optical properties for all objects in the scene. This difference inevitably leads to feature matching failure and decreased localization accuracy [3].

In this paper, we introduce an autoencoder-like network called “REal-to-Synthetic Transform (REST)” to overcome the gap between real and synthetic images for feature-based localization. REST transforms a feature extracted from an input image (we refer to this as a real feature) into a corresponding feature if it was extracted from a synthetic image that was generated under the same conditions, i.e., 3D points and lighting conditions, as the real image (we refer to this as a synthetic feature). After REST transforms a real feature, the transformed feature closely resembles the corresponding synthetic feature, thus increasing the accuracy of matching for it with the feature database.

REST can be trained with a small number of real images and corresponding synthetic views. That is because we do not learn how to accurately estimate the pose of the camera under varying lighting conditions, but the relationship between an extracted real feature and its corresponding synthetic feature.

Our main contributions are:

- We introduce REST to improve matching of real features and a database of synthetic features.
- We show that REST is robust to illumination changes and performs well with different feature descriptors even when trained on a small number of real images.

2 REAL-TO-SYNTHETIC FEATURE TRANSFORM

Figure 1 shows an overview of our camera localization system. Our goal is to estimate the camera pose $[R \mid t]$, where $R$ is a rotation matrix and $t$ is a translation vector, from a single RGB image.

We generate synthetic images that simulate a large variety of lighting conditions. Afterward, we extract features from these synthetic views to compute the corresponding 3D location from known camera parameters and the scene model. For each feature, we store the computed 3D location and descriptor in a database.

To estimate the pose of an input image, we detect real features $x_r$ and match them with synthetic features $x_s$ in our database using a random forest. From matched 2D-3D points, we compute the camera pose with the perspective n-point (PnP) algorithm. If the number of correct matches is too small, due to the gap between descriptors acquired from real and synthetic images, the pose estimation fails to recover the correct pose (see Fig. 3).

To overcome the gap, we need a feature transform between real and synthetic features. Two possible directions for transforming features include real-to-synthetic and synthetic-to-real [5]. An obvious difference between the real and synthetic scenes is their complexity. Generally, a real scene is very complex and varies due to various factors, e.g., cars, pedestrians, or weather. Moreover, some factors are difficult to measure, estimate and/or synthesize. On the other hand, a synthetic scene is a simplification of the real scene, i.e., it
is a subset of real scene. Thus, synthetic-to-real transform needs latent variables to supplement the differences in complexity, while real-to-synthetic transforms can be achieved with a simpler network because it is a simplification of the real scene and does not require latent variables. We therefore adopt the real-to-synthetic transform for our camera localization. We then execute the real-to-synthetic feature transform with an autoencoder-like network called REST to overcome the gap. We assume that the gap between real and synthetic features is primarily caused by noise, so we train REST to minimize the loss using the following:

\[
L(x^s) = \|x^s - x^r\|
\]

where \(x^s\) is the real feature \(x^r\) transformed with REST, and \(x^r\) is the corresponding ground truth synthetic feature. To train REST efficiently, we pre-train REST to transform a synthetic feature into itself before training with correspondences between real and synthetic features.

To train REST, correspondences between real and synthetic features are required, so we first compute 3D points for both types of features. Given a scene model, 3D points of synthetic features can be obtained because we know the camera parameters used in the simulation. Given projection matrices of real images, we can then compute the 3D points of real features. Finally, we match the groups by computing the nearest 3D points between both feature sets.

Then, we compute correspondences between real and synthetic features in both groups. However, the ideal correspondences between real and synthetic features in both groups are unobtainable because the perfect simulation is impossible. If we directly search for correspondences between real and synthetic features using position, like in Fig. 2(a), most real features will correspond with synthetic features located on the boundary side of the group of synthetic features, which can cause a reduction in feature matching accuracy. To find better correspondences, we apply whitening to both groups, as shown in Fig. 2(b). Since whitening changes the mean to zero and the variance to one, the two groups tend to overlap in this whitened space. We then deploy nearest-neighbor searching further improved the matching results for both descriptors, and performance of REST without whitening for SIFT was worse than naive matching. As discussed in Sec. 2, directly corresponding causes to transform a real feature into a synthetic feature in another feature group. Therefore, omitting the whitening step decreases feature matching accuracy for REST.

Table 1: Result of evaluation for REST. From left column, median of matching accuracy (MA), median of position error (PE), median of orientation error (OE), and median of localization time are indicated.

<table>
<thead>
<tr>
<th>FeatureDesc.</th>
<th>naive method</th>
<th>REST w/o whitening</th>
<th>REST w/ whitening</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>57.58</td>
<td>74.54</td>
<td>85.88</td>
<td>343.00</td>
</tr>
<tr>
<td>SURF</td>
<td>69.93</td>
<td>80.52</td>
<td>86.95</td>
<td>300.80</td>
</tr>
</tbody>
</table>

Table 2: Breakdown of computational times for REST with whitening.

<table>
<thead>
<tr>
<th>FeatureDesc.</th>
<th>REST</th>
<th>Matching</th>
<th>RANSAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>95.64 ms</td>
<td>4.78 ms</td>
<td>218.39 ms</td>
</tr>
<tr>
<td>SURF</td>
<td>54.39 ms</td>
<td>3.42 ms</td>
<td>218.93 ms</td>
</tr>
</tbody>
</table>

Figure 2: Corresponding between real and synthetic features. Red and blue dots represent real and synthetic features respectively. Red and blue ellipses represent the groups of real and synthetic features respectively, and the 3D points corresponding to each of the groups are the same.

Figure 3: Localization results. Green and blue lines represent the ground truth and the projection of the model given the estimated camera pose respectively.

to accurately obtain correspondences between real and synthetic features.

3 Evaluation

We use the DTU robot image dataset [4] in a simulation. To match the illumination conditions in the dataset, we use two parallel light sources: one illuminates the scene from above and the other moves around the scene and is always oriented towards the model. We set the latitude angle to \([10, 20, \cdots, 80]\) and the longitude angle to \([0, 30, \cdots, 180]\) degrees. We generate views of the scene from 4 \(\times 4\) positions in front of the model and rotate the camera from \(-10^\circ\) to \(10^\circ\) in \(5^\circ\) steps. Overall, we generated 80 images per lighting condition, i.e., 4480 images in total.

To evaluate what effect REST as well as whitening has on the feature matching and localization accuracy, we compare REST with whitening, REST without whitening, and naive feature matching (directly matching a real feature with the synthetic feature database). We also compare two feature descriptors: SIFT and SURF.

We evaluate all methods on 60 images from the DTU dataset [4]. The images are selected to match a pedestrian’s view at a building. We split them into 5 groups for cross validation by training REST with 4 groups, i.e., 48 images, and evaluating with the rest.

We judge whether a real feature is correctly matched to its synthetic counterpart if the projection of the 3D location of the synthetic feature into the camera image is less than three pixels away from the detected real image. We show the results in Table 1 and Fig. 3, and indicate computational times in Table 2.

REST improves matching accuracy compared to the naive method for all feature descriptors, which indicates that REST can successfully convert real features into synthetic features. Applying whitening further improved the matching results for both descriptors, and performance of REST without whitening for SIFT was worse than naive matching. As discussed in Sec. 2, directly corresponding causes to transform a real feature into a synthetic feature in another feature group. Therefore, omitting the whitening step decreases feature matching accuracy for REST.

4 Conclusion

In this paper we presented a new approach for robust camera localization under varying lighting conditions. We introduce REST, an autoencoder-like network that converts real features into synthetic features. We also utilize a whitening process to improve the match ratio between real and synthetic features. Our result shows that REST with whitening improves matching and localization accuracy.

References


