




11/23/2018

# Feature-Based Image Stitching and Object Detection

Homework Assignment

Candidate: Mohsen Ghazel



Submitted to Zebra Technologies, in  
fulfilment of the Homework Assignment  
for my Computer Vision position  
Application

Submitted on: November 23<sup>rd</sup>, 2018

# TABLE OF CONTENTS

<b>LIST OF FIGURES.....</b>	<b>2</b>
<b>LIST OF TABLES.....</b>	<b>2</b>
<b>GLOSSARY.....</b>	<b>2</b>
<b>1. INTRODUCTION .....</b>	<b>3</b>
1.1 Objectives.....	3
<b>2 DATA ACQUISITION .....</b>	<b>3</b>
<b>3 IMAGE STITCHING.....</b>	<b>4</b>
3.1 Literature Review .....	4
3.1.1 Pixel-Based Image Stitching Approaches.....	4
3.1.2 Feature-Based Image Stitching Approaches.....	5
<b>4 OBJECT DETECTION .....</b>	<b>7</b>
4.1 Literature Review .....	7
4.1.1 Pixel-Based Object Detection Approaches.....	7
4.1.2 Feature-Based Object Detection Approaches.....	8
<b>5 APPROACH .....</b>	<b>9</b>
5.1 Selected Approach and Justification.....	9
5.2 Algorithms.....	9
5.2.1 Feature-Based Image Stitching.....	9
5.2.2 Feature-Based Object Detection.....	10
5.2.3 Combined Algorithm.....	10
<b>6 IMPLEMENTATION .....</b>	<b>11</b>
6.1 Software.....	11
6.2 Feature Detection and Extraction .....	12
<b>7 EXPERIMENTAL RESULTS AND DISCUSSION.....</b>	<b>12</b>
7.1 Data.....	12
7.2 Results and Discussion.....	13
7.2.1 Image Stitching .....	13
7.2.2 Object Detection.....	14
<b>8 CONCLUSIONS .....</b>	<b>18</b>
<b>9 REFERENCES .....</b>	<b>20</b>
<b>10 APPENDIX A .....</b>	<b>21</b>
<b>A.1 PROBLEM STATEMENT .....</b>	<b>21</b>
A.1.1 Objective .....	21
A.1.2 Concrete Example .....	21
A.1.3 Deliverables .....	21

## LIST OF FIGURES

Figure 1: A montage of the 9 acquired scene images .....	13
Figure 2: A montage of the 6 acquired template images .....	13
Figure 3: A panorama of the 12 scene images tsitched together .....	14
Figure 4: Template Object # 4 has too few matched features to be detected.....	18

## LIST OF TABLES

Table 1: Comparison Pixel-Based vs. Feature-based image stitching and Object techniques.....	9
Table 2: Object detection results for the individual scene images.....	16
Table 3: Object detection results for the stitched panorama image .....	17

## GLOSSARY

BRIEF	Binary Robust Independent Elementary Features
FAST	Features from Accelerated Segment Test
ORB	Oriented FAST and Rotated BRIEF
SIFT	Scale Invariant Feature Transform

# Feature-Based Image Stitching and Object Detection

By Mohsen Ghazel

## 1. INTRODUCTION

This report is submitted to fulfill the Homework Assignment requirement for my application for a Computer Vision position at Zebra Technologies, Mississauga. In particular, the report contains a high-level literature review of relevant image stitching and object detection approaches, justification of the selected and implemented image stitching and object detection methods and sample input data and their corresponding generated experimental image stitching and object detection results.

### 1.1 Objectives

Detailed specification of the Homework Assignment is presented in Appendix A. In summary, the objectives of the assignment can be outlined as follows:

1. Acquire a few overlapping images of a cluttered scene, which contains a set of reference objects of interest.
2. Acquire template images of these reference objects of interest and store them in a template library to be used for detecting these objects from the scene images.
3. Apply image stitching techniques to construct a panorama of the overlapping scene images, acquired in step 1.
4. Apply object detection techniques to detect the objects of interest in the acquired scene images as well as in the panorama of stitched images constructed in step 3.

Next, we describe the data acquisition process.

## 2 DATA ACQUISITION

Overlapping images of a staged cluttered scene of various objects of interests as well as template images of these objects are acquired using the same smart phone camera and under the following conditions:

- The scene and reference images are acquired from a relatively close range (1-2 meters).

- Different scene images are acquired by moving the camera almost parallel to the scene while minimizing any rotation effects on the camera.
- Images are acquired under approximately the same lighting conditions.
- Orientation and scale of reference objects are likely to be different in the scene images than in the reference images
- The objects of interest are stationary in the scene.
- In order to test the implemented image stitching and object detection techniques, some of the objects of interest in the reference images may be partially occluded in some of the scene images.

The acquired data is illustrated in section 7.

Next, we review some of the applicable image stitching and object detection techniques found in the literature and select the most suitable techniques for our specific application.

## 3 IMAGE STITCHING

Image stitching is the process of aligning and stitching multiple images with overlapping fields of view of a scene to construct a segmented panorama image from the different images. In this section, we shall review the main types of images stitching approaches.

### 3.1 Literature Review

Most image stitching algorithms adopt either a pixel-based or feature-based approach. A high-level literature review of these two approaches is presented next.

#### 3.1.1 Pixel-Based Image Stitching Approaches

Pixel-based image stitching techniques or sometimes referred to as direct-method image stitching approaches use information from all pixels of the images to be stitched together. Typically, these methods iteratively update an estimated homography to align overlapping images by minimizing the sum of absolute differences or maximizing the correlation between overlapping pixels. The updated homography is then applied to warp and align these images together.

Early image stitching techniques were based on patch-based translational alignment technique developed by Lucas and Kanade, which is still widely used today [8]. More recent work in this area has addressed the need to compute globally consistent alignments, the removal of parallax artifacts and object movement as well as dealing with varying exposures [13].

The main advantage of pixel-based image alignment and stitching techniques is their simple implementation. Despite the various research efforts to improve them, these techniques have several limitations, including [13]:

- They are computationally complex because they process every pixel in the images. This makes them unsuitable for real time applications.
- They are not invariant to changes in image scale or orientation. These methods are more suitable for identifying the overlapping region in images which vary only through translational shift transformation. If the images are subjected to complex geometric variations, direct methods likely fail to capture the overlapping part of the scene in the source and target images.
- Direct methods are also sensitive to changes in the illumination and exposure of the same object in different images to be stitched together.
- Another key disadvantage that makes direct stitching techniques undesirable is that they need some initialization, since they iteratively align overlapping images by estimating and updating the homography. This requires human interaction to make sure the stitching occurs correctly.
- Furthermore, these methods are not robust to image zoom, changes in illumination or to the presence of extra images which are not part of the scene sequence.

In summary, in spite of their appealing simple implementations, pixel-based image alignment methods and stitching appear to have too many limitations to be suitable to our application of interest.

Next, we discuss the feature-based image stitching approaches.

### **3.1.2 Feature-Based Image Stitching Approaches**

In feature-based image alignment and stitching methods, distinctive corresponding points, known as features, are selected from the source images and then matched to each other. If enough matching features has been detected, a homography is then estimated using these feature points only. The estimated homography is then applied to warp and align these images together.

The Scale Invariant Feature Transform (SIFT) features were originally proposed by Lowe et al. [6]. Since the introduction of SIFT features, there has been significant progress in the development and application of scale and orientation-invariant features for object recognition and matching over the past few decades [1][2][3] [7]. These features can be found more repeatably and matched more reliably than traditional methods such as cross-correlation of image patches, which is not invariant to changes in scale, orientation or illumination.

As the name suggests, the first step in feature-based image alignment and stitching methods is the extraction and matching of key features from different images and trying to match common features within overlapping subregions of different images. Over the past few decades, much work has been done in developing more distinctive features, which are less sensitive to variations in the illumination, scale or orientation of the image. Early image features such as Harris corners were distinctive enough, but they are not invariant to scaling of the image. Scale Invariant Feature Transform (SIFT) features are geometrically invariant under similarity transforms and invariant under affine changes in intensity [6]. Since the development of the SIFT features, several other features have been proposed in the literature, including Speeded Up Robust Features (SURF), Features from Accelerated Segment Test (FAST), Binary Robust Independent Elementary Features (BRIEF), Oriented FAST and Rotated BRIEF (ORB), where some of them were shown to be even better alternatives to the original SIFT features [11].

In general, feature-based image alignment and stitching techniques perform significantly better than pixel-based techniques. However, depending on the type of extracted features, the computational complexity and gain in performance may vary significantly. These techniques have several other advantages over their pixel-based counterparts as they overcome most if not all the limitations of the pixel-based methods, including:

- Extracted feature descriptors are scale invariant
- Extracted feature descriptors are rotation invariant
- Extracted feature descriptors are highly distinctive and less to noise and illumination changes.
- Feature-based methods are fast and efficient since they only process relatively low number of features and not all pixels in the images, as was the case for direct-methods.

Despite their clear advantages over their pixel-based counterparts, these techniques are not without limitations, such as:

- The extracted features are unique and, somewhat limited in number in many images and they are typically found in textured surfaces. On the other hand, flat, smooth and uniform-colored surfaces may have few or none of such features, which renders these techniques not suitable for aligning or stitching together images of such surfaces.
- Also, shiny objects and surfaces pose problems for most feature-point detectors and descriptors. Glare introduced by changes in brightness intensities around potential features points, caused by reflections on such an object's surface, may weaken or mask these points all together, rendering them undetectable by existing feature detection and extraction techniques [12].

This completes this high-level overview of the two types of image stitching methods. Next, we review object detection techniques.

## 4 OBJECT DETECTION

The aim of an object detector is to detect instances of semantic objects of a certain class, such as people, vehicles, buildings, etc., in digital images using image processing, computer vision or machine learning algorithms.

### 4.1 Literature Review

Object detection techniques may be classified into four general categories:

- Pixel-based
- Feature-based
- Motion-based
- Classification-based.

In view of our application of interest and acquired data, the last two techniques are not applicable since the objects of interest are stationary and no labelled data is available for training a classification-based object detector. Thus, we shall focus on the first two types of object detection approaches.

#### 4.1.1 Pixel-Based Object Detection Approaches

Most pixel-based object detection techniques are based on matching a known template within the target image. Thus, if a template describing a specific object is available, object detection becomes a



process of matching features between the template and the image sequence under analysis. There are two types of object template matching, fixed and deformable template matching. Given that our objects of interest are rigid, we shall focus on fixed template matching.

Fixed templates are useful when object shapes do not change with respect to the viewing angle of the camera. Most fixed-template matching approach are based on computing the cross-correlation between the template and target images and detecting any peaks in the correlation values. In particular, these techniques apply a sliding window over all the pixels in order to detect the position of the normalized cross-correlation peak between a template and a target image and locate the best match, if the template image is present in the target image. The absence of such cross-correlation peak indicates that the template image is likely not in the target image. Pixel-based template matching techniques are generally less sensitive to noise and illumination effects in the images [4].

The limitations of pixel-based object detection approaches are very similar to those of pixel-based image stitching techniques. As such they tend to be computationally expensive as they process all image pixels, they are also sensitive to changes in illumination, scale and orientation. Furthermore, pixel-based template matching is sensitive to occlusion, as the object needs to be fully visible in the scene image to be detected. These techniques are more suitable for restricted environments where imaging conditions, such as image intensity and viewing angles between the template and images containing this template are the same. However, recently pixel-based template-matching techniques, which are less sensitive to variations in orientation, scale, translation, brightness and contrast have been proposed with some reported success [5].

#### **4.1.2 Feature-Based Object Detection Approaches**

Feature-based object detection methods are based on very similar principles as the feature-based image stitching methods. As such features are first independently detected in the scene and reference images. An attempt is then made to match the features in the two images. If enough good matches (inliers) are obtained between the two images, then a geometrical transformation is then constructed from these matched features to estimate the position and orientation of the reference image in the target image [7].

The types of feature descriptors, the advantages and limitations of feature-based object detection techniques are similar to those of feature-based image stitching methods, as discussed in section 3.1.2.

## 5 APPROACH

In this section, select the implemented approach and outline the algorithms.

### 5.1 Selected Approach and Justification

Table 1 summarizes the desired requirements for the implemented approach and whether the pixel-based and feature-based method satisfy these requirements. In view of this table, it is evident that feature-based image stitching and object detection approach is the better choice.

**TABLE 1: COMPARISON PIXEL-BASED VS. FEATURE-BASED IMAGE STITCHING AND OBJECT TECHNIQUES**

#	Requirements	Pixel-Based	Feature-Based	Suitability for our Application
1	Less sensitive to illumination changes	✗	✓	Feature-Based
2	Less sensitive to scale changes	✗	✓	Feature-Based
3	Less sensitive to orientation changes	✗	✓	Feature-Based
4	Less sensitive to occlusion	✗	✓	Feature-Based
5	Has lower computational complexity	✗	✓	Feature-Based
6	Easier to implement	✓	✗	Pixel-Based
7	Available in MATLAB image process and computer vision toolboxes	✓	✓	Feature-Based

Next, we outline the implemented feature-based image stitching and object detection algorithms.

### 5.2 Algorithms

First, we outline the feature-based the image stitching algorithm.

#### 5.2.1 Feature-Based Image Stitching

The implemented algorithm for stitching  $n$  images can be outline as follows:

### Feature-Based Image Stitching Algorithm

---

1. Detect and extract the features of *image 1*
2. Estimate the geometric transformation between the first image and every other image
  - For image  $k = 2$  to  $n$ :
    - Detect and extract the features of *image k*
    - Match the detected features between *image 1* and *image k*
    - Remove any outliers in the matches and only keep inliers
    - In enough matched inliers are found between the 2 images, then compute the geometric transformation between the *image 1* and *image k*
3. Initialize the Panorama:
  - Create an initial, empty, panorama into which all the images are mapped.
  - Compute the minimum and maximum output limits over all transformations.
4. Create the panorama image:
  - Map images into the panorama-image and overlay the images together.

Next, we outline the feature-based object detection algorithm.

#### 5.2.2 Feature-Based Object Detection

Given, a template image  $T$  and a scene image  $S$ , the feature-based object detection algorithm may be outlined as follows:

### Feature-Based Image Object Detection

---

1. Detect and extract the features of the template image  $T$
2. Detect and extract the features of the scene image  $S$
3. Match the detected features between *image 1* and *image k*
4. Remove any outliers in the matches and only keep inliers
5. If enough inliers are found between the 2 images:
  - The template image  $T$  is detected in the scene image  $S$ :
  - Compute the geometric transformation between the template image  $T$  and the scene image  $S$  to estimate the position of the template image  $T$  in the scene image  $S$
  - Overlay the polygon of the template image  $T$  on the scene image  $S$ .
6. Otherwise, the template image  $T$  is likely not in the scene image  $S$ :

The combined feature-based image stitching and object detection algorithm is outlined next.

#### 5.2.3 Combined Algorithm

Given  $n$  scene images and  $m$  template images:

---

### Feature-Based Image Stitching and Object Detection Algorithm

---

1. Apply the feature-based stitching algorithm outlined above to stitch the  $n$  scene images to obtain the panorama image
2. For each *template* image  $i = 1$  to  $n$ :
  - For each *scene* image  $j = 1$  to  $m$ :
    - Apply the object detection algorithm to detect template *image i* in scene *image j*
  - Apply the feature-based object detection algorithm to detect template *image i* in the panorama image

This completes the description of our adopted approach. Next, we discuss how we implemented the selected approach.

## 6 IMPLEMENTATION

There are several software tools and open source libraries that perform out of the box feature-based processing such as feature detection and extraction, feature mating, removal of feature outliers, feature machining, image-to-image transformation and registration based on matched inliers. As mentioned earlier, the implemented feature-based image stitching and object detection algorithms make use of these functionalities. Therefore, there is no need to re-invent the wheel, instead we make use of these functionalities, as discussed next.

### 6.1 Software

Our feature-based image stitching and object detection approach was implemented in MATLAB and makes use of its Computer Vision System Toolboxes, which has various feature-based functionalities including:

- Feature detection and matching extraction based on different types of descriptors, such as SURF, BRIEF, FAST, etc.
- Feature matching
- Matched feature outlier detections
- Geometric transformation between images based on matched inliers
- Image registration
- etc.

Next, we discuss how to select the most suitable feature descriptors.

## 6.2 Feature Detection and Extraction

As mentioned earlier, the performance of feature-based approaches may vary significantly depending on the type of extracted features. Hence, we experimented with using a few different feature extractors including Harris Corners, BRIEF, FAST and SURF, since functionalities for detecting these features are available in MATLAB Computer Vision System Toolbox.

Next, we illustrate and discuss the experimental results.

## 7 EXPERIMENTAL RESULTS AND DISCUSSION

Sample results and assessment of the performance of the feature-based image stitching and object detection algorithms are presented and discussed in this section. We begin by illustrating the acquired data.

### 7.1 Data

As mentioned in section 2, the input data consists of overlapping scene images as well as template images of a set of reference objects of interest within the scene. The acquired scene images are illustrated in Figure 1 and the template images are illustrated in Figure 2. In order to test the implemented image stitching and objected detection techniques, the objects of interest have different scale and orientation in the template images than in the scene images. The illumination conditions of the scene images and the template images are also different. Furthermore, in order to test the sensitivity of the implemented object detection approach to occlusion, some of the objects appear only partially in some of the scene images.



**FIGURE 1: A MONTAGE OF THE 9 ACQUIRED SCENE IMAGES**



**FIGURE 2: A MONTAGE OF THE 6 ACQUIRED TEMPLATE IMAGES**

Next, we present and discuss the experimental results generated for this input imagery data.

## 7.2 Results and Discussion

Sample image stitching and object detection results generated from the acquired test data are illustrated in this section.

### 7.2.1 Image Stitching

The panorama of the scene images constructed using the implemented stitching algorithm is illustrated in Figure 3. In view of this result, we make the following observations:

- It appears that the scene images are aligned and stitched together reasonably well.
- We observe a slight mis-alignment in the panorama, which is more visible for objects that appear partially in 2 adjacent images. This is perhaps due to the relatively number of matched features (inliers) used to align and warp the images together. The background of the various images is the same and does not contain any significant features. Thus, most of the features come from the objects themselves. One way to remedy this mis-alignment is introduce more objects to the scene, even objects that are not of interest to us. These additional objects should have more strong features and should be scattered around the scene in order to increase the feature density and improve the slight mis-alignment problem.
- The slight difference in the brightness of the various images is also evident and the boundaries of the images are visible. In most cases neighbouring image edges show intensity discrepancies which is undesirable. One way to reduce these inevitable image stitching artifacts is to apply blending in order to make the panorama brightness more homogeneous and smooth-out the boundary artifacts of the stitched images [10].



**FIGURE 3: A PANORAMA OF THE 12 SCENE IMAGES TSITCHED TOGETHER**

This completes the presentation and discussion of the image stitching results. Next, we discuss the object detection results.

### **7.2.2 Object Detection**

Feature-based object detection results for detecting the template images in the scene and the panorama images are illustrated in this section.

The object detection results of detecting the 6 templates in the 9 scene images are illustrated in Table 2. Similarly, the object detection results of detecting the 6 templates in the panorama image are illustrated in Table 3.

In view of these results, we make the following observations:

- Most objects of interest are correctly detected in the scene images in spite of the significant differences in scale, orientation and illumination conditions between the template and the scene images.
- The implemented feature-based object detector was also able to detect most of the partially occluded objects, which illustrates its robustness to occlusion.
- Similarly, for the panorama stitched image, most of the objects are correctly detected.
- The bounding polygons of the detected objects are generally well located. However, in some cases, such as the first detection of Template Image #1, the polygon is poorly localized. This is likely due to low matched inliers.
- We note Template Image #4, was not detected in any of the scene images or the panorama image. Since this template appears in four different scene images, these missed detections have degraded the detection rate of the implemented object detector significantly. The failure of detecting this template was examined and it was likely attributed to the lack of sufficient number of matches extracted from this object. As illustrated in Figure 4, an insufficient number of good matches are generated between the template image and panorama image. This is also the case when attempting to detect this template image in the four scene images. One may wonder why too few features were extracted from this template image. A key difference between this template image and the rest of the template images is that, mostly, just the title of the book (text) is visible in this template image. However, for the other five template images, the background contains various objects with different colors, which adds more features. Thus, the failure to detect this template image is likely due to the faint, somewhat blurry and homogeneous background behind the foreground text. It may be concluded that text alone may not contain enough features to detect an object and additional background variations help with generating more distinctive features for the object to be detected.

It should be noted that, based on my understanding, the objective of this assignment was not to acquire the best possible test data in order to get the best stitched image and object detection results. If that is the case, one can easily select specific objects of interest, which are rich in features and acquire high-quality



data to ensure near perfect performance of the implemented feature-based stitching and object detection methods. However, from my perspective, it is best to test the implemented algorithms on real-world imperfect data in order to truly assess their performance and identify their advantages as well as limitations. Thus, this was my objective.

TABLE 2: OBJECT DETECTION RESULTS FOR THE INDIVIDUAL SCENE IMAGES














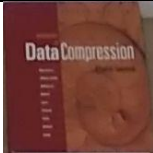




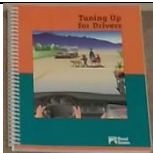




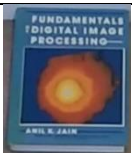








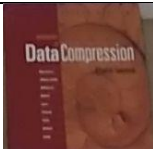

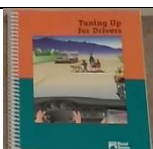

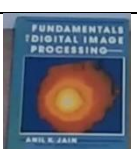
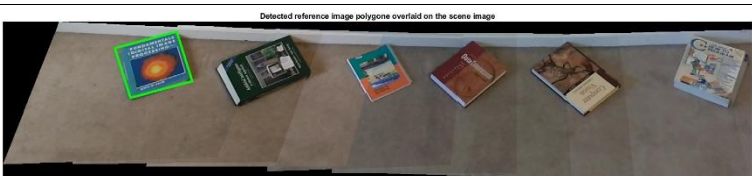
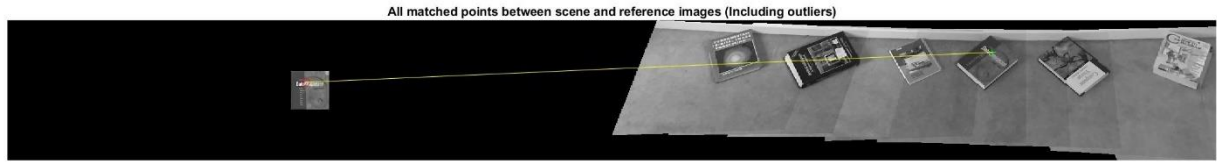
#	Template Image	Detection Results				Detection Rate
1						3/4
2						3/4
3						2/2
4						0/4
5						2/4
6						2/2
Overall						12/20 (60%)

TABLE 3: OBJECT DETECTION RESULTS FOR THE STITCHED PANORAMA IMAGE

#	Template Image	Detection Results	Detection Rate
1			0/1
2			1/1
3			1/1
4			0/1
5			1/1
6			1/1
Overall			4/6 (67%)



**FIGURE 4: TEMPLATE OBJECT # 4 HAS TOO FEW MATCHED FEATURES TO BE DETECTED.**

Next, we complete the report with concluding remarks.

## 8 CONCLUSIONS

This report summarizes the process of selecting and implementing applicable image stitching and object detection techniques in order to identify objects of interest in the scene images as well as the panorama of stitched images. This process involves the following steps:

- Conducting a high-level literature review in order to identify the types of suitable approaches. Relevant image stitching as well as the object detection techniques found in the literature were grouped into two categories: Pixel-based and feature-based.
- A high-level analysis of the advantages and limitations of these two types of techniques led us to conclude that feature-based techniques are the most suitable approaches for our image stitching and object detection tasks.
- Outlining the algorithms for the selected approaches and their implementation framework was selected.
- Acquiring suitable imagery data, consisting of scene images and template images of the objects of interest.
- Processing the acquired data using the implemented image stitching and object detection algorithms to generate the experimental results.
- Analyzing the experimental results in order to assess the performance of the applied algorithms.

Based on the limited experimental results, we conclude that feature-based image stitching and object detection methods are generally robust to changes in scale, orientation and illumination between the different images. The implemented feature-based object detector is also robust to partial occlusion of the objects of interest. Despite their advantages, the implemented feature-based methods are not without limitations. As observed in section 7, the stitched panorama of the scene images suffers from minor mis-

alignment and the some of the objects of interest were not correctly detected in the scene or the panorama image. These deficiencies are likely due to the lack of good matched features in the scene or template images. As discussed in section 3.1.2, not many good and distinctive feature can be found in images of smooth surfaces, such as book covers, but they are more abundant in images of textured surfaces.

Potential future work may include the following investigations:

- Apply blending in order to improve the quality of the panorama image by reducing the undesirable boundary artifacts of the stitched images.
- Explore algorithms that can handle low-textured or smooth surface, shiny or glared surfaces. Most existing feature-based approaches sometimes fail in finding meaningful features on such surfaces. For such surfaces, one may apply a feature detector to extract key points on edges and corners (less sensitive to glare) instead of a detector that typically extracts interest points on blob-like structures (more sensitive to glare), as explored in [12].

This concludes the report.

## 9 REFERENCES

- [1] M. Brown and D. Lowe, "Recognizing panoramas," in Ninth International Conference on Computer Vision (ICCV'03), (Nice, France), pp. 1218–1225, October 2003.
- [2] M. Brown, D. Lowe. Automatic panoramic image stitching using invariant features, *International Journal of Computer Vision*, 2007, vol. 74 (pg. 59-73)
- [3] M. Brown, D. Lowe, Automatic panoramic image stitching using invariant features, *International Journal of Computer Vision* 74 (1), 59-73
- [4] N. Gupta, R. Gupta, A. Singh, M. Wytock, "Object Recognition using Template Matching," December 12, 2008
- [5] H. Y. Kim and S. A. Araújo, "Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast," *IEEE Pacific-Rim Symposium on Image and Video Technology, Lecture Notes in Computer Science*, vol. 4872, pp. 100-113, 2007.
- [6] D. Lowe, Distinctive image features from scale-invariant keypoints, *International journal of computer vision* 60 (2), 91-110
- [7] D. Lowe, Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999.
- [8] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 674–679, Vancouver.
- [9] MATLAB, Image Processing Toolbox, and Computer Vision System Toolbox, 2018b. The MathWorks, Inc., Natick, Massachusetts, United States.
- [10] V. Rankov, R. Locke, R. Edens, P. Barber, B. Vojnovic, "An Algorithm for Image Stitching and Blending", *Proc. SPIE*, vol. 5701, pp. 190-199, 2005.H.-C.
- [11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conference on Computer Vision*, Barcelona, Spain, Nov. 2011, pp. 2564 – 2571.
- [12] H. Sperker, A. Henrich, "Feature-based object recognition: A case study for car model detection", *11th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 127-130, 2013.
- [13] R. Szeliski, *Image Alignment and Stitching: A Tutorial*, Technical Report MSR-TR- 2004-92, Microsoft Research (2004).

## 10 APPENDIX A

### A.1 PROBLEM STATEMENT

#### A.1.1 Objective

You are provided a set of unordered images of a cluttered scene. The images are all of the same scene and taken with the same camera. Therefore, for each image in the set there is at least one other image that shares some overlap with each other. Your goal is to:

- a. Identify what items are in the images. Example images of the items are available in a pre-existing database.
- b. Construct a larger image from the image set and identify where in the *larger* image the detected items are located.

#### A.1.2 Concrete Example

To provide a more concrete example, consider that you have been asked to develop a smart phone app that lets a user create a “virtual bookshelf”. They can take a series of photographs with their phone’s camera and the app will identify all of the books on their bookshelf. It will then present them with a composite image that they can then interact with; e.g. clicking on a book will bring up a search for related authors. To be clear, you are **not** required to create this interactive component. You only need to complete (a) and (b) in the above objectives.

#### A.1.3 Deliverables

##### A.1.3.1 LITERATURE REVIEW

We require that you perform a small literature review that discusses past and current methods on how to perform tasks (a) and (b) and discuss the advantages and disadvantages of each approach.

##### A.1.3.2 IMPLEMENTATION

You will then select which approach is the best out of the what you have found and you are to implement this solution in whichever programming environment you are most comfortable with (i.e. MATLAB, C++, Python, etc.). You are allowed to develop your code in either Windows or Linux. Please **refrain** from developing on Mac OSX and no one on the Computer Vision team uses Mac OSX.

1. If you decide to use MATLAB, you are strictly limited to using the Image Processing and Computer Vision Toolboxes.
2. If you decide to use C++, you may use OpenCV but you must use OpenCV 3.x.
3. If you decide to use Python, you are strictly limited to using the following packages:
  - a. NumPy
  - b. SciPy
  - c. OpenCV
  - d. Scikit-image

##### A.1.3.3 WORKING DEMO

Once you have completed implementation, you should ensure that you provide a working example of your implementation running and completing the objectives.

### A.1.3.4 SMALL REPORT

Once you have completed your literature review and implementation, you are required to write a small report that contains your literature review and the method that you selected to approach the problem. Please be sure to discuss **why** you chose your method of choice.

### A.1.3.5 SEND THE CODE!

Please send the completed report as well as your code, the images that you used as well as any additional dependencies required for your code to run. Please include a small README file that details how to run the code as part of your submission.