# FEATURE-BASED ALIGNMENT FOR GIF ANIMATION

*Ying-Ying Cheng* (鄭盈盈) *Chiou-Shann Fuh* (傅楸善)

Department of Civil Engineering,
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan,
E-mail: r06521806@ntu.edu.tw fuh@csie.ntu.edu.tw

**ABSTRACT**

This project discusses different kinds of methods of image matching for feature-based image alignment, including Scale Invariant Feature Transform (SIFT), Affine Scale Invariant Feature Transform (ASIFT), Gradient Location-Orientation Histogram (GLOH), and nearest neighbor matching with a nearest neighbor distance ratio test strategy. Flip-book animations needs taking a set of photos of an action scene or portrait (preferably in motor-drive—continuous shooting—mode) and aligning them with several steps of feature extraction, image matching and image rectification. Features extraction will be operated by Harris corner detector. Image matching and image rectification will mainly use Affine Scale Invariant Feature Transform (ASIFT) which performs well in practice. As all subsequent images are resampled onto the first image's coordinate frame, the final step is converting the resulting images into an animated GIF or optionally implementing cross-dissolves to turn them into a "slo-mo" video.

***Keywords:*** *Feature extraction, Image matching, Image rectification, Harris corner detector, ASIFT.*

## 1. INTRODUCTION

This project is for flip-book animations. Steps for an animated GIF contains feature extraction, image matching, and image rectification. Taking a set of photos and designing the experiment steps are both necessary for project preprocessing. In addition, together with writing program, it will fulfill what the project required.

This project uses Harris corner detector for feature extraction, and Affine Scale Invariant Feature Transform (ASIFT) which performs well in practice for image matching. As all subsequent images are successfully resampled onto the first image's coordinate frame, the final step is converting the resulting images into an animated GIF. Especially, different kinds of image matching methods for feature-based image alignment will be introduced in following sections.

## 2. METHOD

### 2.1. Images for experiment

Take 9 photos and use the first one as based image. The girl with red swimming suit is included in each photo for image matching. All subsequent images are through translation and rotation.

Fig. 1: Images for experiment.

## 2.2. Features extraction

Features extraction methods contain Harris corner detector, Gaussian Difference, and Derivatives of Gaussian. F¨orstner–Harris uses following mathematical formula for corner detection.

$$E_{\mathrm{AC}}(\Delta \boldsymbol{u}) = \sum w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2$$

$$\det(\boldsymbol{A}) - \alpha \,\mathrm{trace}(\boldsymbol{A})^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

Another kind of approach Gaussian Difference is suitable when the images being matched do not undergo large scale changes, e.g., when matching successive aerial images taken from an airplane or stitching panoramas taken with a fixed-focal-length camera. Early investigations into scale selection were performed by Lindeberg (1993; 1998b), who first proposed using extrema in the Laplacian of Gaussian (LoG) function as interest point locations. Based on this work, Lowe (2004) proposed computing a set of sub-octave Difference of Gaussian filters (Fig. 2a), looking for 3D (space+scale) maxima in the resulting structure (Fig. 2b), and then computing a sub-pixel space+scale location using a quadratic fit (Brown and Lowe 2002). The number of sub-octave levels was determined, after careful empirical investigation, to be three, which corresponds to a quarter-octave pyramid, which is the same as used by Triggs (2004).

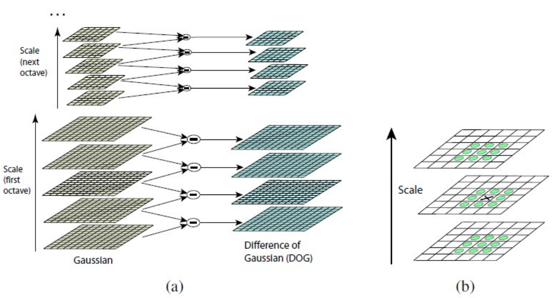

Fig. 2: Difference of Gaussian (DOG).

In the project, as with the Harris operator, pixels where there is strong asymmetry in the local curvature of the indicator function are rejected. This is implemented by first computing the local Hessian of the difference image.

First, transform the RGB images into gray-scale images. Then, use Harris corner detection method for features extraction. It is apparent that most of corners with symbol of green (+) are found in the edge of girl's hair and body. Therefore, the result indicates that it is appropriate to let the girl be the main feature included in each photo. That is, the girl included in each image is suitable for image matching. The Harris corner detection results are shown below. (Fig. 3)

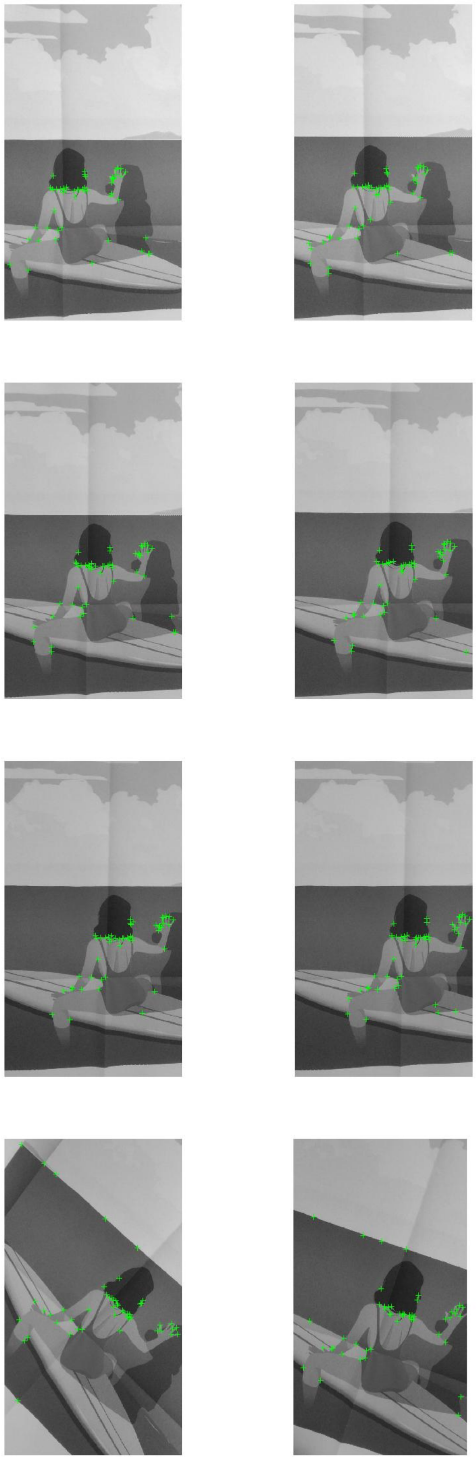

Fig. 3: Harris corner detection.

**2.3. Features descriptors**

While Lowe's Scale Invariant Feature Transform (SIFT) performs well in practice, it is not based on the same theoretical foundation of maximum spatial stability as the auto-correlation based detectors. In order to add a scale selection mechanism to the Harris corner detector, Mikolajczyk and Schmid (2004) evaluate the Laplacian of Gaussian function at each detected Harris point (in a multi-scale pyramid) and keep only those points for which the Laplacian is extremal (larger or smaller than both its coarser and finer-level values). An optional iterative refinement for both scale and position is also proposed and evaluated. Additional examples of scale invariant region detectors are discussed by Mikolajczyk, Tuytelaars, Schmid et al. (2005); Tuytelaars and Mikolajczyk (2007).

Instead of using SIFT for the image matching, in this work, in order to achieve robust and fast traffic sign detection, a rotation invariant binary pattern based feature in the affine space and Gaussian apace is proposed. This specific feature leverages the techniques from ASFIT to achieve robustness in scale, rotation and illumination, meanwhile improving the computation efficiency.

The SIFT feature generation mainly includes Gaussian pyramid, extrema localization and feature generation. Because of the Gaussian pyramid and local rotation invariant processing by accumulating the orientations and magnitude, the SIFT algorithm exhibits strong advantages on the rotation and scale invariant.

According to the ASIFT work, the ASIFT algorithm simulates the original image by rotation and tilt transformation, and then implements the SIFT algorithm on these simulated images. The ASIFT algorithm can perform equally to SIFT when implemented on the different affine simulations of the same image. Hence, it achieves high robustness in the affine space.

There are some difficulties in large changes in illumination due to SIFT feature is based on the local intensity of the pixels; though the normalization for the feature vector achieves the illumination invariant, the performance is worse at complex situations with large changes in illumination. Also, though ASIFT improves the computation efficiency, it really cost much time due to high computation complexity.

**2.3. Features matching**

In the project, as with ASIFT algorithm, the result of feature matching can be obtained.

First, choose one of 9 photos as based image, then the subsequent images can operate features matching with based image by ASIFT; 8 horizontal features matching results (Fig. 4-11) and 8 vertical features matching results (Fig. 12-15) are shown below.

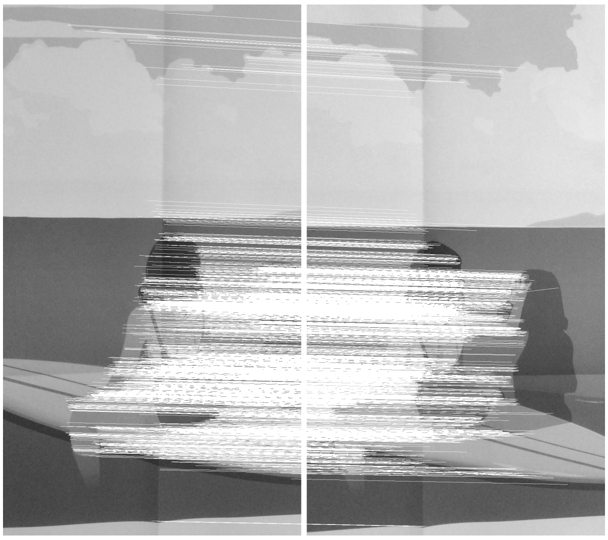

Fig. 4: Horizontal features matching (Images 1&2).

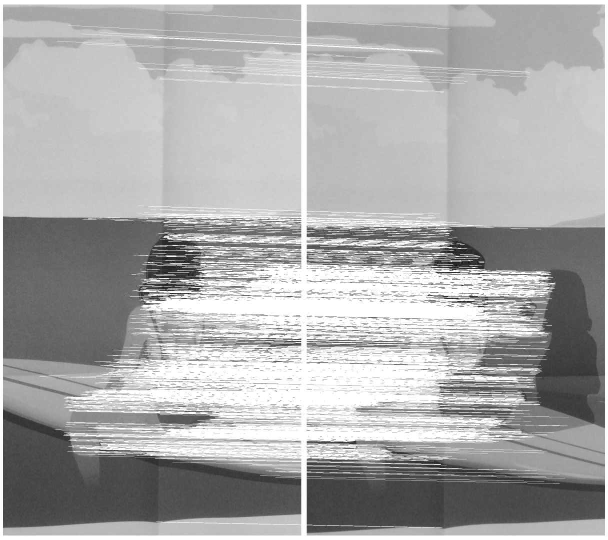

Fig. 5: Horizontal features matching (Images 1&3).

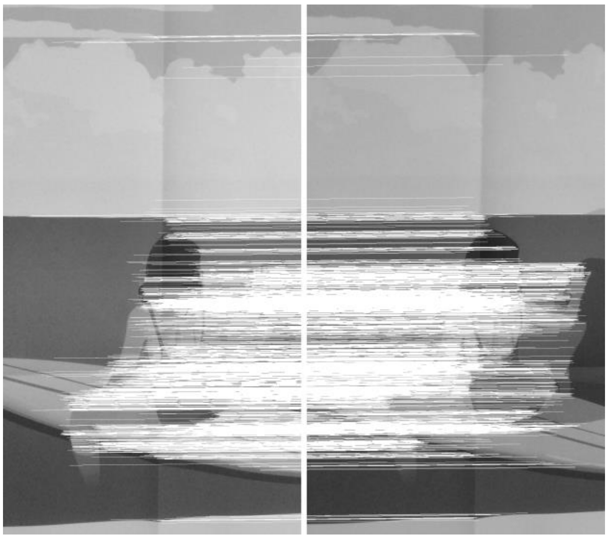

Fig. 6: Horizontal features matching (Images 1&4).

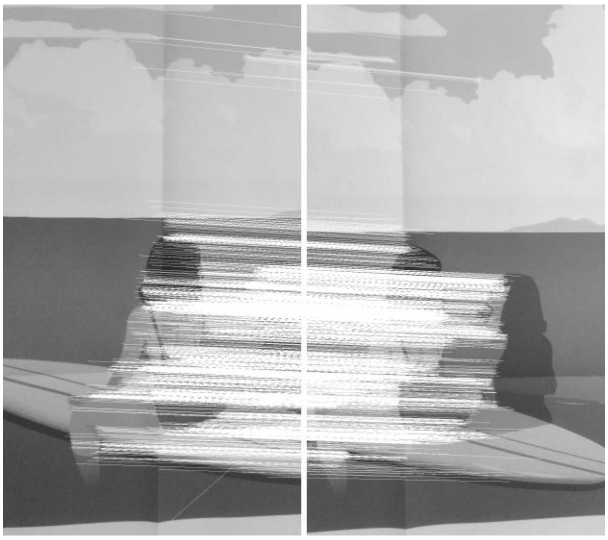

Fig. 7: Horizontal features matching (Images 1&5)

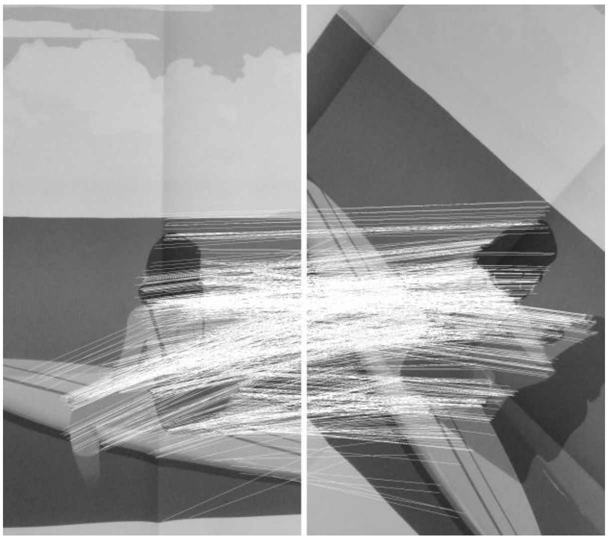

Fig. 8: Horizontal features matching (Images 1&6).

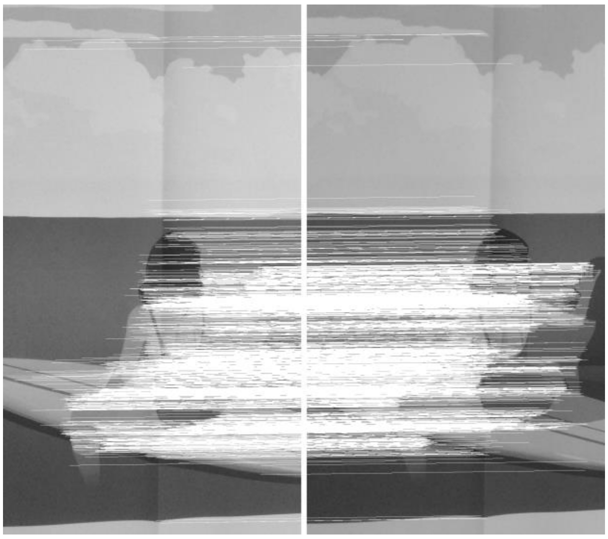

Fig. 9: Horizontal features matching (Images 1&7).

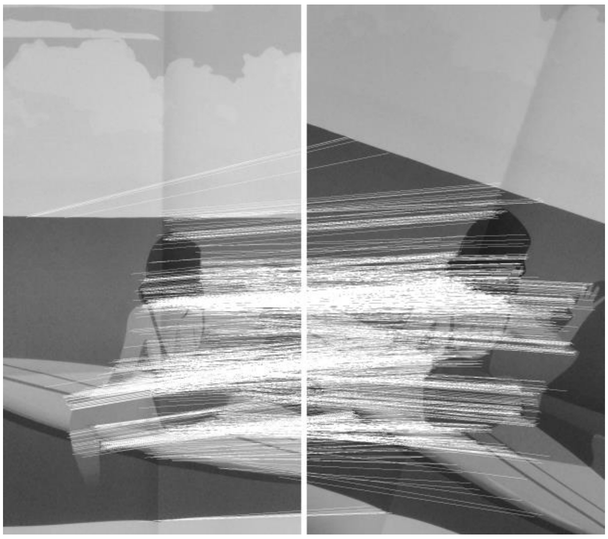

Fig. 10: Horizontal features matching (Images 1&8).

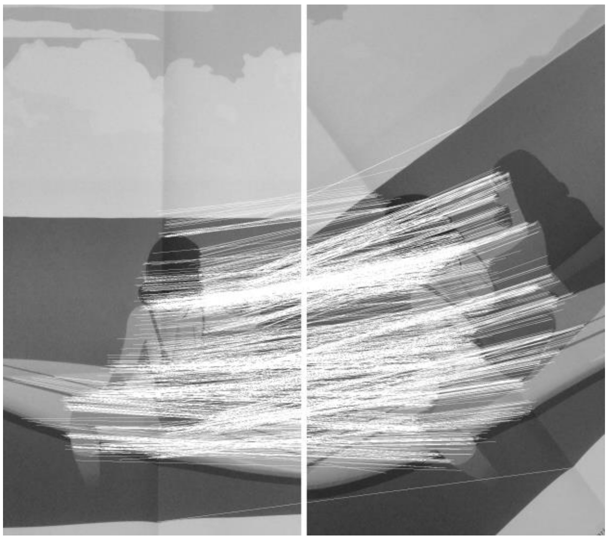

Fig. 11: Horizontal features matching (Images 1&9).

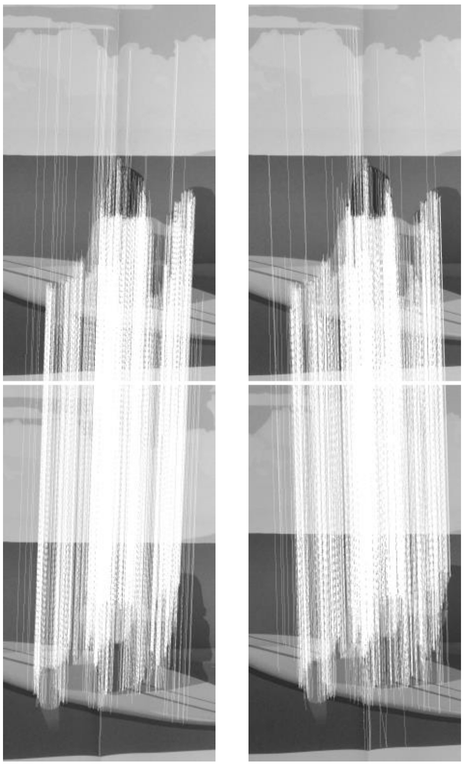

Fig. 12: Vertical features matching
(Left: Images 1&2, Right: Images 1&3).

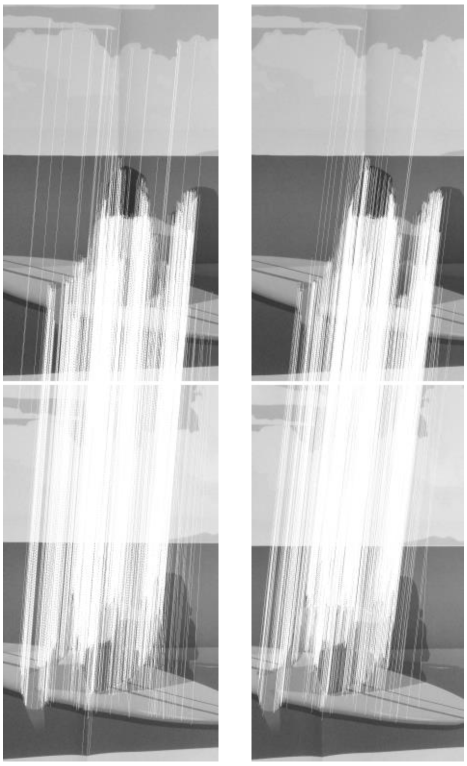

Fig. 13: Vertical features matching
(Left: Images 1&4, Right: Images 1&5).

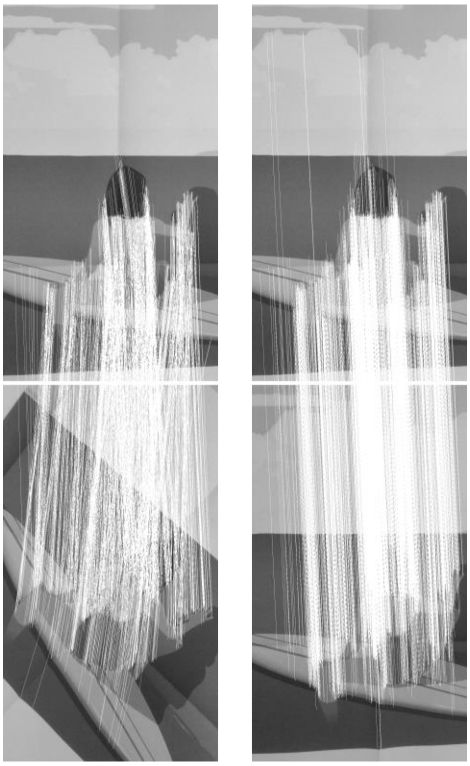

Fig. 14: Vertical features matching
(Left: Images 1&6, Right: Images 1&7).

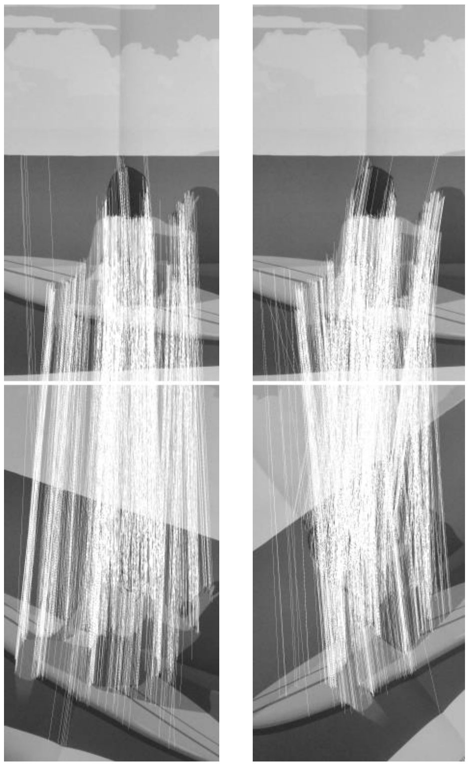

Fig. 15: Vertical features matching
(Left: Images 1&8, Right: Images 1&9).

### 2.4. Affine transformation

In the project, as with ASIFT algorithm, the result of affine transformation can be obtained. After features matching, all subsequent images are resampled onto the first image's (based image) coordinate frame. That is, image rectification is finished (Figs. 16-20).

Fig. 16: Affine transformation.

Fig. 17: Affine transformation.

Fig. 18: Affine transformation.

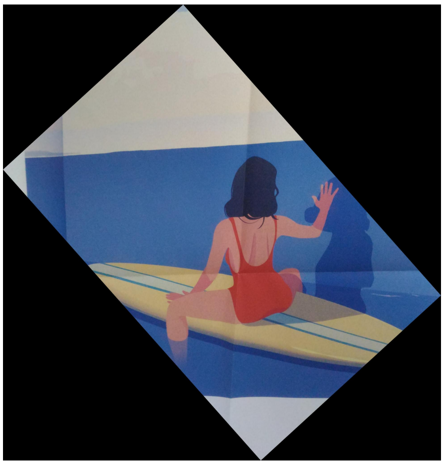

Fig. 19: Affine transformation.

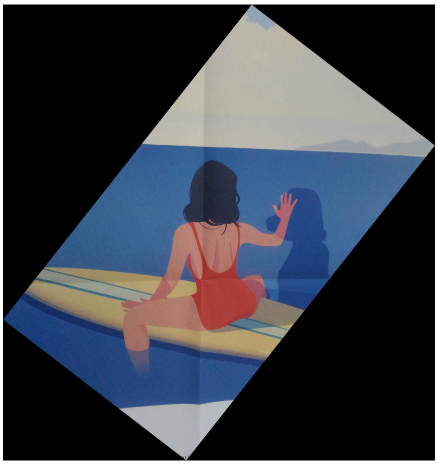

Fig. 20: Affine transformation.

### 2.5. Other image matching methods

Other image matching methods are Gradient location-orientation histogram and Nearest neighbor matching.

Gradient location-orientation histogram developed by Mikolajczyk and Schmid (2005), is a variant on SIFT that uses a log-polar binning structure instead of the four quadrants used by Lowe (2004). In their evaluation, Mikolajczyk and Schmid found that GLOH, which has the best performance overall, outperforms SIFT by a small margin (Fig. 21).

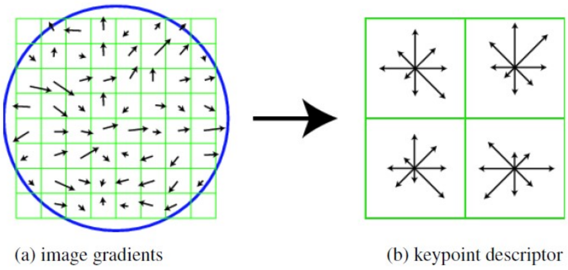


Fig. 21: Gradient location-orientation histogram.

Besides, a useful heuristic can be to compare the nearest neighbor distance to that of the second nearest neighbor, preferably taken from an image that is known not to match the target (e.g., a different object in the database). The nearest neighbor distance ratio in following formula, where d1 and d2 are the nearest and second nearest neighbor distances, $D_A$ is the target descriptor, and $D_B$ and $D_C$ are its closest two neighbors (Fig. 22).

$$\text{NNDR} = \frac{d_1}{d_2} = \frac{\|D_A - D_B|}{\|D_A - D_C|}$$

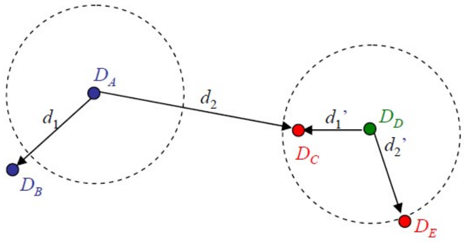


Fig. 22: Nearest neighbor matching.

### 2.6. GIF animation

After image matching and image rectification, as all subsequent images are resampled onto the first image's coordinate frame, the final step is converting the resulting images into an animated GIF or optionally implementing cross-dissolves to turn them into a "slo-mo" video.

Use the free software of website (http://gifmaker.me/) to finish the Gif animation.

## 3. RESULT

If using all 9 photos, the result would be not good due to 3 rotation images shoot in uneven plane. Also, it will result from not slowly continuous shooting. Because these 3 rotation images swayed too dramatically, the solution would be shot continuously and much slowly for better animation result.

In addition, if only use 6 of 9 photographs in GIF animation, it would get better result. These 6 photos are in even plane and are taken in continuous but slow motion. It indicates that it is necessary to take a set of photos in motor-drive-continuous shooting-mode and slow motion.

GIF animation result in this project can be got in following link: https://drive.google.com/drive/folders/1FWpoYWWJmzOWULzwTxEFAZnRDEBTCsU4?usp=sharing

From the link above, better_result.gif shows that the girl with yellow surfboard slide out from the beach.

## 4. DISCUSSION

From the previous result and introduction of different kinds of image matching above, we can infer that the following factors will affect the experiment. Besides, some suggestions will offer below.

### 4.1. Illumination

Using ASIFT algorithm would bring about some difficulties in large changes in illumination due to SIFT feature is based on the local intensity of the pixels; though the normalization for the feature vector achieves the illumination invariant, the performance is worse at complex situations with large changes in illumination.

Also, though ASIFT improves the computation efficiency, it really cost much time due to high computation complexity.

Two of nine photos in this experiment shows higher illumination and then affects the result apparently. Therefore, we should investigate all images after we took them. (Fig. 23-24).

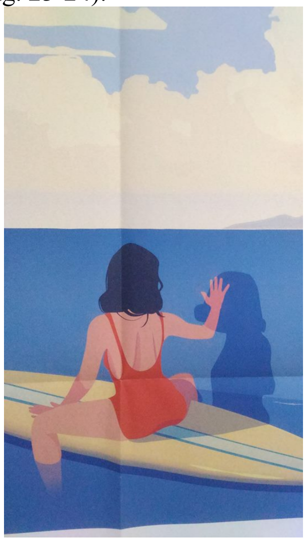

Fig. 23: Photo with higher illumination.

Fig. 24: Photo with lower illumination.

### 4.2. Shooting equipment

Use digital camera would get better result because of higher resolution and easier setting camera parameters.

This project only uses smart phone to take photos, and will cause worse result due to low resolution and fewer fixed camera parameters.

### 4.3. Image matching methods

Not only use SIFT and ASIFT algorithm for the project, other image matching methods like SURF, Gradient location-orientation histogram and Nearest neighbor matching can be considered.

If using variety of different image matching methods, we can compare the results and discuss their pros and cons.

## REFERENCES

[1] David.G. Lowe, "Distinctive image features from scale-invariant keypoints.", *International journal of computer vision*, Vol. 60, No. 2,pp.91-110, 2004.

[2] J.M. Morel, and G.Yu, "ASIFT: A New Framework For Fully Affine Invariant Image Comparison", *SIAM Journal on Imaging Sciences*, pp.438-469, 2009.