# Abstract

This project experiments with algorithms which could address problems in Time – lapse photography, where image processing for an image sequence might be required to obtain better visual effects. Specifically, image registration is essential when the images are not aligned decently, which could result in flickering at play back stage, a motion – estimation based algorithm is applied to align images after searching for the best reference image. In addition, when a large motion is presented between two frames, a new frame could be created and this indicates adjusting the play back timescale to be non - linear, as a consequence, the movements become relatively slower and more details are shown to the audiences.

# Acknowledgements

I would like to express my thanks to Mr. Colin Dalton, for his constant encouragement, invaluable advice and guidance throughout the project.

Thanks also to Fan Zhang, for his help on technical problems, and to Tao Shi, Bingqin Chen and Zonglin Du for providing photographic equipment.

Sincere thanks to my family, friends for their support, help and encouragement during this project.

# Contents

# Introduction

## Non – Linear Time – Lapse Photography

Time – lapse photography, by definition, is a process containing long – time image sequence capturing and then playing back at some rate (usually 24 frames per second), as a piece of animation, for example, a sunrise and sunset procedure can be recorded by digital camera every five minutes during an entire day, and software can be used for the short film construction.

Time appears to be moving fast and thus lapsing when the image sequence is played at a certain frequency, which is often 24 frames per second, to put another way, 24 images are played in sequence on screen within a second. If such frequency is modified, from the audiences’ perspective, objects in the scene would move faster (frequency increased) or slower (reduced), which, indicates a non – linear playing back rate.

In this project, non – linear time is implemented by image interpolation techniques, if new images containing intermediate positions between two neighboring frames, it would be tantamount to lessen the frame rate, for a smaller number of images will be presented per second and thus, the speed of moving parts in the scene would be felt slower.

## Aims and Objectives

This project explores algorithms that address problems in capture and playback phases of time – lapse photography. When capturing images, a tripod is indispensable to fix the position of a camera, however, usually it could be massively time – consuming for this process which could cover longer than weeks, thus a practical problem is that it is difficult to keep the camera exactly in the same position, and the first aim of this project is to find a proper method to implement registration for image sequence, including an algorithm to align two images as well as deciding the best image to act as the reference.

* Objectives for image registration
* Implementation of an algorithm to align two images.
* Apply the algorithm to image sequence.
* Experiment with the algorithm for acceleration and robustness.

In the playback stage, the main aim is the non – linear time, on the one hand, to increase the speed, removal could be easily applied in an certain software, thus the project primarily focus on the opposite aspect – to reduce the playing back frequency. Image interpolation is utilized to achieve this aim.

* Objectives for image interpolation
* Implement an algorithm which is able to create a new image between two neighboring frames.
* Apply the algorithm to image sequence.
* Experiment with the algorithm to enhance the plausibility of the new frame.
* Find methods to automatically decide where to create new frames.
* Algorithm acceleration

## Structure

This report consists of five main sections, firstly, background of relative methods and techniques in conjunction of previous research, are described in the following section, where a number of registration and interpolation methods are included. In section 3, there are methods designed for this project, including those are unsuccessful after experiments. Then the implementation and results section mainly explains thresholds selection for relatively better segmentation, along with data selection to achieve a plausible created frame. The next section is a discussion of the results from experiments, followed by an evaluation of the project to determine if it has obtained the objectives. Finally future work is discussed in different aspects, including software developing and extensions of present methods.

# Background

## Image Segmentation Methods

In this part I described two fundamental image segmentation techniques that are relative to this project, one is to divide an image into two parts by a threshold and the other is based on sub – divisions.

### Thresholding

Thresholding requires gray – level images and is the relatively simpler method segmentation process, it could works well when the image owns a decent contrast, which is suitable for a constant threshold to distinguish the background and objects in image. A basic thresholding algorithm can be described as follows:

**Algorithm 1 Basic Thresholding**

Search all pixels f(x, y) of the image f. An image element g(x, y) of the segmented image will be categorized as object if f(x, y) > T, where T denotes the threshold, otherwise, it belongs to the background.

There are a number of methods to determine T, usually detecting the mean or median gray value could be suitable for segmentation with brightness. Mean gray value is calculated from the mean value of all pixels, while median is often the gray value of the pixel at the median position after sorting all pixels.

### Region – Based Segmentation

A natural region – based method for segmentation is region growing. Beginning from the raw image data after sub – division, two adjacent regions will be merged if they satisfy some merging criterion, then try the next region until no pair matches the condition. On the opposite, segmentation could also be implemented by splitting to small regions from the entire image, while furthermore, splitting and merge techniques could be used simultaneously.

### Summary

Thresholding and region – based methods are basic but significant techniques for image segmentation, in this project, segmentation is utilized before processing images and some thoughts are adopted from these fundamental algorithms.

## Optical Flow and Motion Estimation

### Optical Flow

Optical flow indicates the motions in images during a small time interval, including movements from objects, edges, surfaces and even the entire scene (camera movement). Basic optical flow estimation is on the basis of the Optical Flow Equation (OFE)

(1)

Where is the image sequence, denote the 2 – D velocity in the image plane respectively, while are partial derivates of intensity within a frame and between frames. Equation (1) indicates that intensity along motion trajectory remains constant.

### Motion Estimation in H.264 / AVC

H. 264 / AVC is a commonly used compression, recording and distribution standard for high definition video, and motion estimation techniques form the core of such applications (Iain Richardson, 2010). A motion vector is used for representing motion information from one frame to a reference frame, and an image could be sub – divided into small motion blocks (usually square - shaped), each motion block contains a motion vector when compare to the reference.

Block Matching Algorithm (BMA) (I.E.G Richardson, 2003) is a popular method in motion estimation, as shown in Figure 1, the algorithm is described as follows:

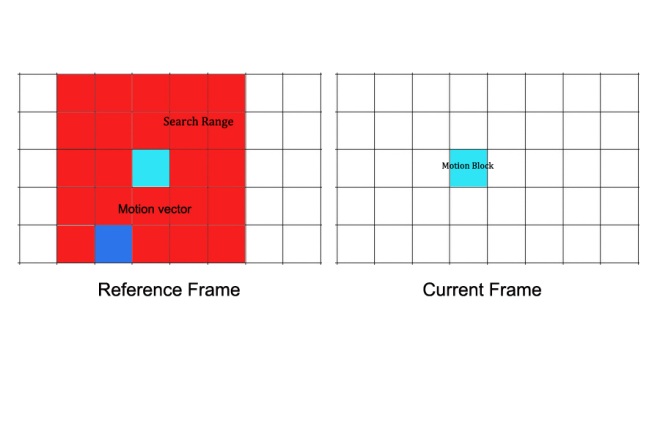


Figure 1 Basic concept of BMA

**Algorithm 2 Block Matching**

1. For one motion block in the current frame, we compare it within the search range (a larger region containing same size blocks around the corresponding block) in the reference frame, for each comparison.
2. A motion vector is recorded along with some figures that denote similarity.
3. Finally, the motion vector with the largest similarity will be selected, and all pixels in the block are regarded as having the same motion vector.
4. Repeat from 1) until all the motion blocks are checked in the current frame.

### Summary

As researching with motion problems, the basic concept is the principle of optical flow – constant intensity along motion trajectory, and the BMA in motion estimation could implement segmentation to distinguish the background and the moving objects between frames, this is fairly significant to create new images with the moving part in the correct position.

## Image Registration

### Introduction

Image registration is the process of transformation of coordination systems, from different ones into one. In this project, registration mainly is represented by image alignment operations, which could avoid flickering when the image sequence is played back, it is significant since the photographer cannot ensure that the tripod are installed at exactly the same position throughout the entire capturing phase, which could last longer than weeks or even months. In addition, as I experimented in practice, even if the tripod position are carefully marked, it is difficult to capture the a consistent sequence after moving away from original place, thus, registration is indispensable for generating a more reasonable sequence.

Another factor in time – lapse photography that requires consideration is the registration should only be applied to the background rather than the motion parts, which should actually be intentionally neglected when implementing the registration algorithm, therefore, segmentation is essential before alignment.

### Previous Work

Image registration has been applied to different regions, for example, virtual reality (A. Aseem, et al, 2006), analysis of medical images (K. Loewke, et al, 2008) or for military purposes (N. Heinze, et al, 2008). There are also a number of methods to implement alignment process, the method introduced in (Q. Zhang, et al, 2011) is a popular feature matching algorithm, with the combination of Fourier Transform and SIFT feature point extraction for efficiency and robustness purposes, while when applied to time – lapse photography, features from motion objects might also be detected and thus affects the results. (F. L. Seixas, et al 2008) proposed an Genetic Algorithm (GA) – image registration method, creating a new perspective for matching and mapping point sets, but its practicality is need to be tested. The algorithm preferred at early stage is (G. Ward, 2003), which is original designed for High Dynamic Range (HDR) images composition, during which capturing of several source images is similar time – lapse photography. This method thresholds images after transforming into grayscale color space, then applies an exclusive – or (XOR) operator to show misaligned coordinates, before adjusting x and y offsets. I implemented this in my project and results are discussed in section 4.

### Summary

In conclusion, as it is time – consuming for capturing images, and from practical experiments, registration is a first step in addressing problems in such image sequence, and some algorithms could be adopted and then applied to the entire sequence.

## Image Interpolation

### Introduction

Current films are usually projected at 24 frames per second, which means 24 images appear in each second (linear time), an interesting challenge in this project is to adjust the playing back speed when necessary.

Image interpolation is the method that is employed in this project to implement a non – linear timescale, by definition, image interpolation can also be regarded as image scaling and resampling, which will increase the image size and resampling to a newly – created image plane. Particularly, in time – lapse, the resampled image need a motion prediction algorithm, which could create a new image that depicting the scene time – related but not captured by a digital camera, but rescaling is not included in the scope.

Implementation of non – linear time results in a more expressive film because when something actually occurs between frames, the story could be displayed more in detail to the audience when new frames are created, objects appear to move slower than before while on the opposite, the process could be accelerated when nothing moves or only slightly movements take place.

### Previous Work

A path – based algorithm is introduced in (D. Mahajan, et al, 2009), they constructed a path between two images where new images need to be inserted, and along this path, pixel gradients could be easily copy and move to implement the interpolation, and the frequency contents of intermediate images could be reserved without ghosting and blurring, with temporal coherence maintained. (T. Stich, et al, 2011) proposed a method attempting to adapt human visual system for such image sequence, which included the indispensable motion prediction process. The two methods seem to perform in their own characters and I will describe them in detail respectively.

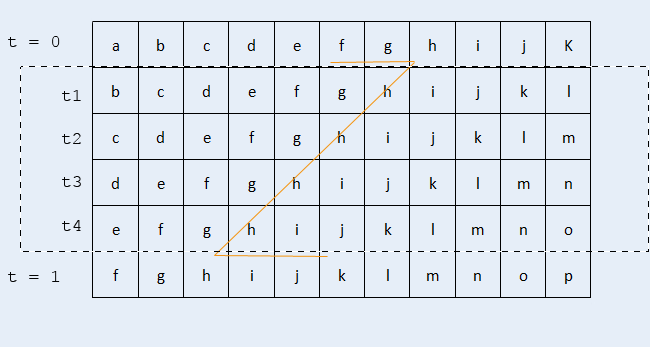
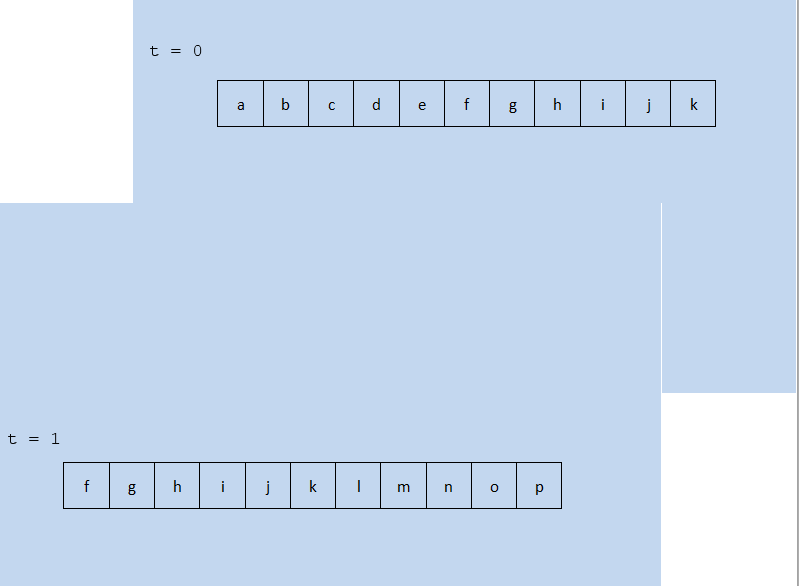


Figure 2 Basic notion of path

(a)

(b)

*p0*

The basic idea of (D. Mahajan, et al, 2009) is to create some ‘motion path’ to lead the variation of pixels between two input images, the basic notion of such path is described in Figure 2, (a) defined two simple images constructed by single lines (blue background not included), and at t = 1 the image is a left – shifted version of the one at t = 0, they are together regarded as the input images in this example, Figure 2(b) shows that for images could be interpolated between the inputs, and each time, the image is actually shift to the left at a step of 1 pixel, character ‘f’ is the information on pixel p0, which will be interpolated at pixel p0 – 1, in the next image, so the algorithm copies ‘f’, from the input t = 0 image, and then copies ‘g’, ‘h’, respectively, after that, the copy operation will be moved to t1 image, which is indicated by the orange line in Figure 2(b), and this process comprise the path. In the t = 0 image, the p0 + 2 pixel (recording ‘h’) is the point that the path changes direction, this is defined as transition point, where the intensity values of two images match, and hence, the interpolation could be plausibly implemented. In addition, the path could be represented by two general motion vectors mA and mB, thus path could be described as Equation (21).

Relative to interpolation, computing has been applied to optical flow as well as video segmentation (C. L. Zitnick, et al, 2005), but due to implicit occlusion reasoning, it is difficult to present some occlusions. Graph cuts are also employed for global optimization (V. Kolmogorov, et al, 2001) and (J. Sun, et al, 2005), however, they are difficult for explicitly occlusions considerations. With regard to image warping and morphing, an image – based rendering method is proposed by (A. W. Fitzgibbon, et al, 2003) utilizes manual correspondence based on stero and optical flow. In addition, interpolation is also applied to video processing, (H. Wang, et al, 2004) proposed an algorithm for video editing based on 3D Poisson equation, while similarly in the gradient domain, with Poisson equation, there are applications for image editing(A. Agarwala, et al. 2004) and (J. Jia, et al, 2008). However these methods are not compatible to the interpolation in time – lapse photography because their inputs and outputs are both images or videos, which could be regarded as static methods.

### Summary

The key factor for image interpolation is trying to create the most plausible images, from the previous work, it could be seen that the new frame data should be copy from the original ones, in which way the consistency in intensity and color could be kept to the largest extent, a robust algorithm for new position computation is the key element during this process, in addition, occlusion handling should be carried out as well for the completeness of the newly created frame.

## Background Summary

This section has explained the reason for the registration and interpolation in this project, in conjunction of relative methods and algorithms of this project, from basic concepts to previous research work, algorithms that might be adopted in the project is described in detail.

# Methods

In this section the all methods and algorithms that are employed in this project are described, they are primarily divided into two parts as per the operation of registration and interpolation, some are adopted from previous work while others are newly designed.

## Registration for Image Sequence

This part contains methods for image registration, firstly, the thresholding and XOR algorithm proposed by (G. Ward, 2003) is adopted and explained in 3.1.1, then, inspired by image interpolation, I experimented with motion estimation to align images, finally, in section 3.1.3, I describe the methods that benefits the processing speed, which is fairly significant in such image sequence solution.

### Extension of a previous Framework

**Thresholding and XOR method overview**

The fast and robust method proposed by G. Ward was taken into consideration in the early, the registration for HDR image construction might be relevant to some extent, since HDR images, similarly, requires several images to be overlapped with aligned perfectly before enhancement in color, intensity and contrast. A hardware solution will be a tripod to capture multi – images, however, this algorithm is designed for hand – held exposures, which could cause similar camera movement when creating time – lapse image sequence, hence, it should be worth experimenting with.

The input to the algorithm is two grayscale images, which could be converted by the following formula[[1]](#footnote-1).

(2)

Then one of the two is selected as the reference, both images will be segmented by a median threshold, which is selected from sorting all the gray values, the segmentation algorithm is displayed below.

**Algorithm 3 Median Thresholding**

1. Compute grey values for all pixels in the image and then accumulating frequencies to form a histogram.
2. Begin from 0, count pixel numbers located in each grey value until they are add up to half of the pixel amount (image height multiplied by image width).
3. At this point, the value is selected as the threshold .
4. Segment image, if pixel value , set pixel value as 255 (black), otherwise, set values to 0 (white).

This algorithm could segment image according to the exposure, brighter part will be completely set to white while darker part will be black, an example from images captured in Edinburgh, Scotland are shown in Figure 3 (with original image, grayscale image and segmented result), the result contains only two values and could be easy to implement the following XOR operation.

The identity of two images is measured by an exclusive – or (XOR) operator, which could get a value of 0 under the condition that two pixels own the same values, while non – zero if they are different (set to 255 for showing a recognizable result), an example of two unaligned images is shown in figure 4.



Figure 3 Results of grayscale and thresholding segmented images

Left: Original Image Middle: Greyscale image Right: Segmented image



Figure 4 Result of XOR operation

Left and Middle: Segmented images Right: XOR result

In spite of the cloud in the sky, the outline of buildings indicates the difference, and it is noticeable that computing one image with itself will result in a completely black scene, thus to find the offsets for a pair, we select one as the reference, then apply XOR in four directions respectively, and calculate the total non – zero pixel quantity, the smallest direction will be regarded as the correct movement.

**Acceleration**



Figure 5 Image pyramid

When applied in practices, the basic method could encounter difficulties, naturally images from time – lapse photographer may have a High Definition (HD) resolution at 1920 \* 1080, this could result in a time – consuming pixel comparison, or worse, a larger range to detect the displacement in contrast to smaller – sized images. A common solution is to create an image pyramid, which resizes images down by a quarter (halves height and width respectively), with such technique, computation cost will be massively reduced. Figure 6 gives a concept of image pyramid.

Now that the source images are shrunk to smaller sizes, we could limit a pixel range for XOR comparison in horizontal (x) and vertical (y) directions respectively. Finally, if the image pyramid is created to levels, returning consequence from the accelerated algorithm will be linearly expanded by a factor of 2 in each level, to put another way, the final shifting magnitude is times that of the bottom level.

In conclusion, to align images via threshold and XOR operation is less complicated comparing to others (SIFT, GA, etc.), the key factor is to compare differences in two images, in conjunction with acceleration method, the results are shown in section 4.

### A Motion Estimation – Based Method

Due to the block matching algorithm could segment background and moving parts in an image, with motion information (motion vector) in each motion block, it is deducible that the motion vectors of the background parts could indicate the displacements between two images, simultaneously, for aligned images, that of moving objects could be a reference for interpolation, I will explain the registration in this section, while the interpolation in section 3.2.

As described in section 2.2.2, if two images are input to an algorithm based on the BMA and one is selected as reference, blocks in the other image could find similarities by an search range in the reference, and thus obtaining a motion vector encoding as motion information, if images are divided into a number of motion vectors, after computation, motion information are recorded for registration and interpolation.

**Motion block and search range**

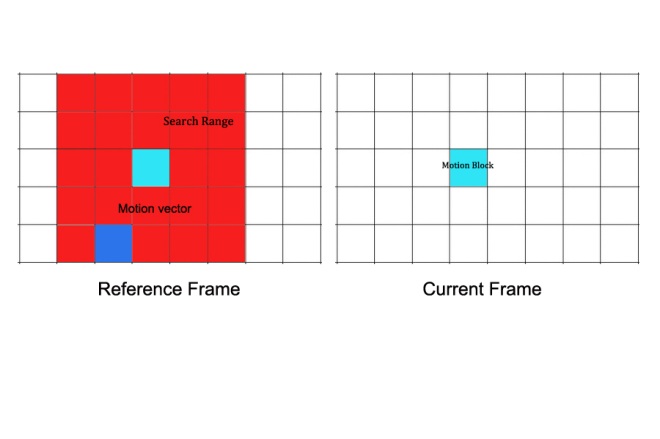


Figure 6 Sample Motion Block and search range demonstration

Block size: 8 \* 8, search range: 40 \* 40

The size of motion blocks is first set to 8 \* 8 pixels, with a - pixel distance search range, this is illustrated in Figure 6, which means that, each block will compare in the context up to two blocks shifting all directions, the comparison is to traversal in the search range with a step of one pixel, after that, we could get a reasonable motion vector indicates the smallest differences for this block. However, as experimented with HD images (1920 \* 1080), such size configurations are relatively small, which shoots up computation, simultaneously, another element to consider in interpolation is accuracy, which could be affected by size as well, so I left the size changeable, and results are discussed in section 4.

With respect to the measurement of block differences, in each block comparing pair, I compute the squared value of total pixel pair differences, this algorithm is demonstrated below.

**Algorithm 4 Block difference computation.**

1. For one motion block in the input image, start with the first pixel in the block, assume the input image has blocks, each block has pixels.
2. Subtract the pixel grey value with corresponding pixel in the reference image, square the result .
3. Add up to of all pixels in the block as an illustration of block difference, denoting by .
4. Traversal in the search range of this block from step 2) and 3), with step of 1 pixel, obtain to .
5. Find the smallest from to , record the motion vector for this block.
6. Repeat from 1) to 5) until all blocks in the input images are computed.

**Segmentation**

At this stage, motion vectors have been computed and then they should be categorized into two parts, background and foreground. In time – lapse photography, the foreground parts usually contain moving objects, and for photographer, camera will be carefully installed to avoid shifting, so an assumption here is that the background will move slower (own smaller vector norms) than the foreground, so that there will be a threshold for the motions.

It is noticeable that with algorithm 4, consider the background part, even if two images are aligned perfectly, the slight intensity difference might not result in a motion vector of (0, 0), hence, a key element for detecting the threshold is that it should be somehow large to avoid such incorrect election probability, a mean value is less reliable since zero vectors could still account for a certain proportion. As a consequence, there vectors are neglected when selecting the threshold, below listed the algorithm.

**Algorithm 5 Detection of motion vector threshold.**

1. Input all the motion vectors in algorithm 4, form a sequence , where is the quantity of motion blocks.
2. For , , compute , denotes the moving distance of a block.
3. Exclude all vectors when , and sum up other non – zero distances, with whose quantity accumulated, results in .
4. Threshold .

Now the threshold could be larger because zeros are intentionally disregarded, and then remaining data will result in a mean that could implement the segmentation, the vector norms of foreground blocks will be larger than , while the remaining belong to the background, an example result of segmentation is shown in Figure 7.It could be seen that the segmentation could roughly recognize the background, but via data selection before shift, such errors are under torlerrence.

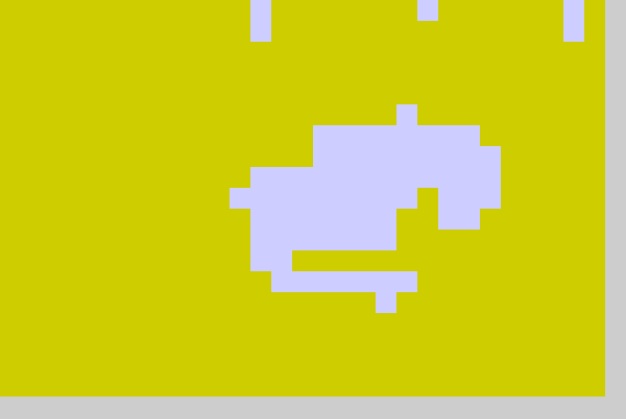


Figure 7 Sample segmentation by motion estimation

Left: Source image Right: Segmented result

**Shift the image**

In the practice of time – lapse photography, the moving parts should not be considered when applying alignment operation, and this is the reason I did the previous segmentation part, thus to obtain the shift vector for the input image, I select the motion vector that owns the largest frequency in the background part to avoid some error computation with intensity difference.

**Acceleration**

As demonstrated before, image pyramid is an effective method for acceleration, it equals to enlarge the block size in when using BMA algorithm, however, the level should be controlled to an extent for a relative reliable results, which is shown in section 4.

### Summary

In this part I employed two methods for image registration, the first one is done by an XOR operator and is simpler, but after experiments, it performed less robust, the second has a segmentation of background and foreground, which is an inherent character for time – lapse photography, the results turn out to be pleased.

## Interpolation for Image Sequence

### Motion Estimation for Interpolation

An exciting point from motion estimation method is the segmentation results, which could be extend to interpolation, but this should under the condition that images have been aligned, otherwise, it could gain computation for registration despite of a successful interpolation.

**Motion vector selection**

As input to the interpolation algorithm are aligned images, only the motion blocks from foreground (vector norms larger than threshold) is considered, similarly, I count frequencies for all vectors, the largest is regarded as the motion information of the object.

**Creation of intermediate frames**

Now that we have computed how much the objects moves from one frame to another, the intermediate position could be estimated, firstly, the motions are considered not large, so that the problem could be simplified as linear, if one image is need to be interpolated, the new position should at half way from the previous frame to the next one, then at these positions, pixels could be copied from either the previous or next frame.

Accounting for the background part, which are considered as static, I set the mean values of the neighboring two frames since they are calibrated and will not cause offsets.

In addition, there could be two methods for creating more images, first, repeat previous process, and the second could been implemented when new position calculation, if one more images is needed, for example, two positions should be computed by a step of one third from reference frame to current frame rather than half of that. However, this is out of project scope so I did not apply experiments

**Occlusion handling**

In image interpolation problems, there will be empty pixels in the newly created frames, which is displayed in Figure 8, as part of the edge is not being filled. Particularly, when previous method is applied, there could be holes at some edges because the input images are regarded as a 2 – D array, when processing the blocks near the edges, there will be out – of – bound problems, which have been purposely disregarded when comparing blocks, it is reliable because after that the motion vectors with largest frequency is forecast as the actual motion, therefore, empty pixels in the new frame is filled the same way as copying background.



Figure 8 Empty pixels after interpolation

**Algorithm overview**

As described before, Algorithm 6 gives the process from selected motion vectors.

**Algorithm 6 Novel frame creation (Considered for one image).**

1. Assume the foreground contains motion blocks with motion vectors, .
2. Select the vectors that owns largest frequency and estimated as motion vector, demoted by .
3. New position coordinates , where are the pixels positions in the current image, or , where denote the pixels positions in the reference image.
4. Copy pixels from the corresponding coordinates in current or reference frame.
5. For values of pixel in the background part of the new frame, , where denotes the pixel values of a pixel, note that the value here is in RGB colour channels.
6. Fill empty pixels with value calculation similar to 5).

### Summary

Motion estimation offers a linear and illustrative method for image interpolation, and if the correct positions of moving objects are able to achieve, plausible new frame could be obtained by copying pixels from source frames, in addition, it is indispensable to fill blank pixels caused by this method.

## Image Sequence Consideration

Previous methods still work for two images, it should be extend to the entire image sequence to create totally automatic program, only in this way, users are able to select image sequence and wait for their results. In this section, both registration and interpolation methods are developed for image sequence, but for the threshold and XOR algorithm, it is difficult to determine whether two images are aligned and hence less possible to find a correct reference point, as a consequence, it has been categorized as a unsuccessful attempt in this project.

### Find reference point for registration

For two images to be aligned, either could be selected as the reference image, but in an image sequence, it is completely different, a correct reference point could massively save computation, for example, consider an image sequence consisting of 100 images that have been captured via a tripod, if there were slightly movement after capturing the 5th image, the fastest registration should apply calibration to the first five images rather than the last 95, so any one from image 6 to 100 could be selected as a reference, if unfortunately image 1 to 5 is selected, the registration algorithm should be run for 95 times, thus it is significant to consider the speed issue by detecting a best reference image.

**Assumptions**

The assumption for reference point detection is formed from time – lapse practice, usually a photographer will be careful with the tripod to avoid movement, but it is almost impossible to re – install the tripod to exactly the same position after movement, this could be true when the process covers a long time, three days, for example, the photographer could mark the first position but it is still difficult to ensure perfectly aligned images, and that is the reason why I do registration. Under such consideration, it could be assumed that the input images are actually divided as groups, each group are perfectly aligned because of usage of tripod (it could also be possible that only one image is in the group). To write the assumption into mathematical form, for all images in image sequence , they are divided into groups, which could be denoted by a set , their corresponding image quantities is written as .

Group2

RP

Group1

Group3

Figure 9 An illustration of reference point detection

Start

**Main process**

Figure 9 gives an example of three image groups compromising an image sequence, the algorithm starts from the first image in the first group, selecting the first image as reference point, then time, check one image afterwards, if the image pair is aligned, an accumulator indicating the quantity of images that are aligned with the reference will increase by one, otherwise, record the quantity, move the reference point the first unaligned image, then repeat this process until all the images have been checked. At this stage, the images are grouped with each group has an aligned image numbers, the reference point will be the first image of the group containing the largest quantity.

**Condition of alignment**

At this point, I found the thresholding – XOR algorithm less reliable for time – lapse images because the motion could cause large errors, the smallest difference in entire image may not indicate the right offsets, thus this algorithm is abandoned and regarded as a failure in this project, in contrast, motion estimation could generate a correct result, when comparing only in the background part, if most blocks return vectors (0, 0), this background could be estimated as still, representing the alignment.

**Algorithm description**

The following box demonstrates the entire algorithm.

**Algorithm 7 Reference point detection**

1. For Groups in image sequence in total images, aligned image quantities for each group are written as (to be detected).Set the first image as the reference point, set up an image quantity accumulator .
2. Check one image each time after the reference point, if image satisfies the aligned condition, , then check the next image.
3. If two images are not aligned at image , store to , then move the reference point to , , repeat from 2) until the last one.
4. Find the largest , set the final reference point to the first image of .

### Determine the interpolation points

Similarly to registration, not all the gap between two frames requires interpolation, only these are sufficiently large will apply the algorithm. From the a motion vector , the distance of motion could be calculated as . For an image sequence, I first detect all the distances between frames, then select the points that are within a specified range (set to 0.8 in test, could be set by user in practice) of the largest distance. Algorithm 8 in the next table describes this process.

**Algorithm 8 Interpolation point detection**

1. For image sequence with images , start from the first pair (index = 1).
2. Compute the motion information and get the motion distance , then index = index + 1, repeat 2) until the last pair.
3. Find the largest , set as ..
4. Restart from the first pair, if , do interpolation between this pair.

### Summary

When extend the methods to the entire image sequence, for registration, it is significant to consider the speed issue, a best reference must be selected for the sequence since it could save enormous amount of computation, with regard to image interpolation, the magnitude for creating new frame could be selected by user, smaller range will limit the condition of interpolation and thus accelerating.

## Methods Summary

In section 3, all the methods and algorithms that are employed in this project have been demonstrated in detail with figures, firstly, registration and interpolation algorithms are applied for only two images for testing the effectiveness and robustness, among there is a failure which employs thresholding and then compares images with an XOR operator. Then, comparison is implemented before a background and foreground segmentation which is done by motion estimation, it is exciting when motion estimation is positive with both image operations. Finally, the methods are extended to image sequence and could completely automatically run.

# Implementation and results

## C and OpenCV

### Thresholds Selection

### Threshold for the XOR registration

### Motion Block Threshold

### Segmentation Threshold

### Summary

## Data Optimization

## Implementation Summary

# Results

## Failures

### XOR Registration

### Improper Motion Blocks

### Summary

## Registration by Motion Estimation

### Results

### Summary

## Interpolation

### Results

### Summary

## Results Summary

# Discussion

## Image Registration

## Image Interpolation

## Applied to Image sequence

## Evaluation

# References

1. Iain E.Richardson,The H.264 Advanced Video Compression Standard John Wiley & Sons ltd, 978-0-470-51692-8, 2010.
2. I.E.G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia, John Wiley & Sons, Ltd, 0-470-84837-5, 2003.
3. Aseem A, Maneesh A, Michael C, et al. Photographing Long Scenes with Multi-Viewpoint Panoramas. ACM Transactions on Graphics, 25(3), pp. 853-861, 2006.
4. Loewke K, Camarillo D, Piyawattanametha W, et al. Realtime Image Mosaicing with a Hand-held Dual-axes Confocal Microscope. Progress in Biomedical Optics and Imaging. Proc of SPIE, Endoscopic microscopy III, Proceeding of SPIE, 6851: 68510F1-9, 2008.
5. Heinze N, Esswein M, Krüger W, et al. Automatic Image Exploitation System for Small UAVs. Proc of SPIE on Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications, 6946: 69460G1-10, 2008.
6. Q. Zhang, Z. Zhang, D. Zeng, A fast, Automatic and Robust Image Registration Algorithm, International Conference on Virtual Reality and Visualization, 2011.
7. F. L. Seixas, L. S. Ochi, A. Conci, D. C. M. Saade, Image Registration Using Genetic Algorithms, Proceedings of the 10th annual conference on Genetic and evolutionary computation, 2008.
8. G. Ward. Fast, Robust Image Registration for Compositing High Dynamic Range Photographs From Hand-held Exposures. Journal of Graphics Tools: JGT, 2003.
9. D. Mahajan, F. Huang, W. Matusik, R. Ramamoorthi, P. Belhumeur Moving gradients: a path-based method for plausible image interpolation, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2009 TOG Homepage, Volume 28 Issue 3, 2009.
10. T. Stich, C. Linz, C. Wallraven, D. Cunningham, M. Magnor, Perception-motivated Interpolation of Image Sequences, ACM Transactions on Applied Perception: TAP, 2011.
11. C. L. ZITNICK, N. JOJIC, S. B. KANG, Consistent segmentation for optical flow estimation, Proc. IEEE Int. Conf. Computer Vision, pp. 1308–1315, 2005.
12. V. KOLMOGOROV, R. ZABIH, Computing visual correspondence with occlusions using graph cuts, Proc, IEEE Int. Conf. Computer Vision, pp. 508–515, 2001.
13. J. SUN, Y. LI, S. KANG, H. SHUM, Symmetric stereo matching for occlusion handling, Proc. IEEE Conf. Computer Vision and Pattern Recognition 2, pp. 399–406, 2005.
14. A. W. FITZGIBBON, Y. WEXLER, A. ZISSERMAN, Image-based rendering using image-based priors, Proc. IEEE Int. Conf. Computer Vision, pp. 1176–1183, 2003.
15. H. WANG, R. RASKAR, N. AHUJA, Seamless video editing, Int. Conf. on Pattern Recognition, vol. 3, pp. 858–861, 2004.
16. J. JIA, C. TANG, Image stitching using structure deformation, IEEE Trans. Pattern Analysis and Machine Intelligence 30, 4, pp. 617–631, 2008.
17. M. Sonka, V. Hlavac, R. Boyle, Image processing, analysis, and machine vision, second edition, Brooks/Cole Publishing Company, 1999.

# Appendix

1. This equation is from the OpenCV function that are used for color space transformation. [↑](#footnote-ref-1)