2007-02-01

Feature-based Mini Unmanned Air Vehicle Video Euclidean Stabilization with Local Mosaics

Damon Dyck Gerhardt
Brigham Young University - Provo

FEATURE-BASED MINI UNMANNED AIR VEHICLE VIDEO EUCLIDEAN STABILIZATION WITH LOCAL MOSAICS

by

Damon Gerhardt

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

April 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Damon Gerhardt

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date Bryan S. Morse, Chair

Date Michael A. Goodrich

Date Parris K. Egbert

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Damon Gerhardt in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

__________________ Date	____________________________________ Bryan S. Morse Chair, Graduate Committee
Accepted for the Department	____________________________________ Parris K. Egbert Graduate Coordinator
Accepted for the College	____________________________________ Thomas W. Sederberg Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

FEATURE-BASED MINI UNMANNED AIR VEHICLE VIDEO EUCLIDEAN STABILIZATION WITH LOCAL MOSAICS

Damon Gerhardt
Department of Computer Science
Master of Science

Video acquired using a camera mounted on a mini Unmanned Air Vehicle (mUAV) may be very helpful in Wilderness Search and Rescue and many other applications but is commonly plagued with limited spatial and temporal field of views, distractive jittery motions, disorienting rotations, and noisy and distorted images. These problems collectively make it very difficult for human viewers to identify objects of interest as well as infer correct orientations throughout the video.

In order to expand the temporal and spatial field of view, stabilize, and better orient users of noisy and distorted mUAV video, a method is proposed of estimating in software and in real time the relative motions of each frame to the next by tracking a small subset of features within each frame to the next. Using these relative motions, a local Euclidean mosaic of the video can be created and a curve can be fit to the video's accumulative motion path to stabilize the presentations of both the video and the local Euclidean mosaic.

The increase in users' abilities to perform common search-and-rescue tasks of identifying objects of interest throughout the stabilized and locally mosaiced mUAV video is then evaluated. Finally, a discussion of remaining limitations is presented along with some possibilities for future work.

ACKNOWLEDGMENTS

In behalf of my wife, my children, and my whole family, I am sincerely grateful to Dr. Bryan Morse for his unwavering support, understanding, time, suggestions, direction, and encouragement that he has dedicated to me to enter and complete graduate school. Our lives are forever changed and improved because of him and his many sacrifices for us, and to him we owe an unpayable debt.

In addition, I owe many thanks to the following: God for everything, my parents for their lifelong nurturing, my sisters for their cheerleading efforts, BYU and its Computer Science Department for its leadership, challenges, and funding, my father-in-law and family-in-law for their support, Dr. Douglas Campbell and Dr. Mark Clement for their graduate school inspirations, Dr. Bryan Morse and Dr. Michael Goodrich for fathering this mUAV vision effort as well as Dr. Parris Egbert and Dr. Bill Barrett for their advice, Cameron Engh for working with me from the beginning and suggesting the novel idea of using curves to stabilize video, Dr. Kevin Steele and Dave Cline for their many helpful discussions and suggestions, Nathan Rasmussen for introducing me to applicable and useful OpenCV functionality, Josh Keeler for good lab laughs, Doug Kennard for being interested, the BYU Magicc lab for making the mUAVs fly, Joe Jackson, Morgan Quigley, and Stephen Cluff for providing useful mUAV videos, Joseph Cooper for trusting my code to work with his, Chris Monson for a great LaTeX template, and Dr. Dennis Eggett for his invaluable statistical analysis of our results. I also thank all of these for their invaluable friendships and support.

Most of all, under God, I dedicate my work and my life to my wonderful wife, Christine, and to my two wonderful boys, Jarom and Talon, who have all been so amazingly supportive and understanding to sacrifice their time with me for this degree. Now I can finally be a real part of the family.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Using video transmissions from inexpensive camera-equipped Mini Unmanned Air Vehicles (mUAVs) is becoming popular in a variety of applications, including search and rescue, military reconnaissance and target acquisition, counterterrorism, and border patrol. The small sizes of mUAVs make them very portable, and their ease of deployment enables them to more quickly gather up-to-date and high resolution aerial surveillance that could be much more difficult to obtain otherwise. In addition, advancements in the hardware used to mount the camera on a gimbal platform are becoming more popular, making possible the ability to increase the persistence of objects within the viewing frustum of the mUAV.

In particular, wilderness search and rescue (WSAR) operations may be improved using mUAV-acquired aerial video. Traditionally, WSAR operations usually entail searching for humans who are lost or injured in mountain, desert, lake, river, or other remote settings. Common problems associated with these operations include contaminated search areas caused by human search teams, slow searching conditions due to vast search areas and difficult terrain, and high related costs in money and man-hours; Utah alone spends hundreds of thousands of dollars and thousands of man-hours per year in WSAR related operations. Moreover, timeliness of WSAR operations is critical; for every hour that passes, the search radius must increase by approximately 3km, and the probability of finding and successfully aiding the victim(s) greatly decreases [1].

The use of camera-equipped mUAVs in WSAR operations may help diminish the negative impact of these problems. mUAVs can be relatively cheap and very easy to transport, enabling quick response times. They can also very quickly provide a broad aerial perspective of the search area without introducing ground search contamination to the search area—such as disturbing useful tracks or scents possibly left by the victim(s). This work focuses on presenting mUAV-acquired video to users in a way that greatly increases their abilities to more quickly, more precisely, and more accurately detect, identify, and select victim sightings within the video.

1.2 Problem Description

Unfortunately, mUAV-acquired video suffers from four major problems that make it very difficult for humans to identify features or objects of interest within the mUAV video: (1) limited spatial and temporal fields of view, (2) jitter, (3) quick motion, and (4) noise.

First, objects seen within the video tend to move quickly through the mUAV's viewing frustum. The ***viewing frustum*** is composed of the spatial and temporal fields of view of the camera mounted on the mUAV. The ***spatial field of view*** is a combination of the camera's focal length and the mUAV's height above ground, *e.g.*, if the plane is flying relatively low with a smaller camera focal length (*i.e.*, the camera is zoomed out), this has a similar spatial field of view as if the plane was flying relatively high with a larger camera focal length (*i.e.*, the camera is zoomed in). The ***temporal field of view*** is a combination of both the spatial field of view as well as a combination of the velocities of the mUAV in each of its six degrees of freedom 6-DOF, *e.g.*, a plane that has some spatial field of view while flying high above the ground has a much more sensitive temporal field of view to a slight rolling motion than if the plane had that same spatial field of view while flying closer to the ground. Because objects seen within the video tend to move quickly through this viewing frustum, the time a user can really evaluate the scene or “look back” at objects that may have been interesting can dramatically be shortened, often making it very challenging for users to identify objects of interest actually captured within the mUAV video.

Second, due to their relatively small size, mUAVs are inherently unstable platforms and highly susceptible to atmospheric turbulence. Such turbulence coupled with the inherent instabilities introduced with the possibility of the camera being mounted on an additionally unstable gimbal platform on the aircraft can greatly contribute to high-frequency jitter throughout mUAV video.

Third, and also due to their relatively small size, mUAVs are also highly maneuverable aircraft. This enables a mUAV to maneuver sharply and frequently in all 6-DOF. In search situations it is also common to put the mUAV in some kind of loitering phase, *e.g.*, circle about a point in the world. Such sharp, frequent, or constant motions can very easily cause a user to become quickly disoriented while watching mUAV video.

Fourth, because the mUAV is too small to carry the payload for the hardware currently required to record the large amounts of video data, it must transmit its video to a ground station. This transmission can introduce significant amounts of noise into the mUAV video. Also, the quality of the small video cameras currently being used on mUAVs lag behind their larger counterparts commonly used in the consumer market as well as in related research areas. This difference in image quality also contributes to a larger amount of noise as well as distortion in the mUAV video, which further isolates mUAV video stabilization and mosaicing research from existing work.

Each of these problems can individually make it very difficult as well as highly strenuous for users to detect, identify, select, and track interesting features throughout the mUAV video. Our experience in field trials has shown that the combination of these problems can very easily render the mUAV video entirely useless for human-user interaction in the context of aerial search and identification tasks.

1.3 Related Work

In order to address these four individual problems, there has been a significant amount of research done to use image analysis to assist in both stabilizing and mosaicing not only the more noisy aerial acquired-video but also commonly acquired video. However, as discussed in Chapter 2, most of these lines of research are either not tailored for real-time applications or involve equipment that is not suitable in the context of mUAVs. The collection of problems associated with the mUAV video has only recently begun to be addressed, and we have found that stabilizing and mosaicing mUAV video in real time is still relatively new and unexplored.

1.4 Contributions

1.4.1 Stabilization and Euclidean Local Mosaic

The scope of this work entails stabilizing and mosaicing de-interlaced and calibrated frames of video from a predominantly forward-velocity mUAV in real time, *i.e.*, at least 30 fps (frames per second), using only software-based vision techniques and curve fitting without the aid of attitude estimation equipment.

To do this we find semi-pose-invariant features within frame f_{t-1} and establish their correspondences to similar features in f_t. We then filter correspondence outliers using a RANSAC [2] homography filter [3] with a novel short-circuit step to both estimate the Euclidean transformation between each relevant frame using the correspondence inliers as well as estimate a goodness measure for the correspondences. These relationships are then exploited to register each current valid frame to its previous valid frame, which can be used to build a local Euclidean mosaic, which we will refer to as an E-mosaic, of the mUAV video's scene.

We then present a novel method of fitting curves to an accumulating history of these frame motions, which are used to smooth out the high-frequency motions in the presentation of the video. A scrolling local Euclidean mosaic (E-mosaic) view, stable Euclidean (stable-E) view, or a stable Euclidean mosaic (stable-E-mosaic) view which is a novel combination of the previous two views can then be presented to the user.

1.4.2 User Interface and User Study

The main focus of this thesis is to show that presenting a user with a stable-E, E-mosaic, or stable-E-mosaic view of the mUAV video will respectively increasingly improve the user's ability to detect and more precisely and more accurately identify victims—or more generally, objects of interest—seen throughout the video as well as improve the user's sense of orientation and attention throughout the presentation of the mUAV video.

In order to quantify these improvements, we present a user study performed on several non-biased subjects in which each subject was presented with a controlled random ordering of 16 different mUAV-acquired short video clips, each clip presented using one of the four possible views—original, stable-E, E-mosaic, or stable-E-mosaic. Each clip-view combination was presented using an interface that allows the subjects to easily and intuitively select objects of interest seen throughout the clip-view while being presented with an additional realistic cognitive load. The resulting relative objective performances among the four different views as well as the subjective preferences among the test subjects are discussed in Chapter 4.

1.5 Thesis Outline

Chapter 2 begins by presenting a definition of terms in Section 2.1 that will be used throughout the rest of this thesis. Then, in Section 2.2, we present some background material that will be needed for the following discussion of related work in Section 2.3. Afterwards, in Section 2.4, we present some foundational material on which we build this work presented in the following Chapter 3.

Chapter 3 describes the processes that we use to build and present the three presentation views. All three views depend on adequate estimations of the spatial relationships among frames; which requires first deinterlacing and calibrating the images (Section 3.1), finding and establishing good correspondences among the common features between contiguous frames (Sections 3.2 and 3.3), and then using those feature correspondences to estimate these spatial relationships (Section 3.4). As these spatial relationships are estimated, they can be used to create the E-mosaic presentation view which follows the image aggregation path of the building E-mosaic (Section 3.6), the stable-E presentation view which follows this path using a smoothed view path (Section 3.7), and the stable-E-mosaic presentation view which combines the previous two views into one (Section 3.8). This chapter then concludes by proposing a user interface in Section 3.9.3 that can further decouple the eye-hand coordination skills needed to identify and select objects of interest within search situations involving mUAV-acquired video.

The format and composition of the user study performed is presented in the beginning of the results chapter, Chapter 4. Then we present a discussion and our analyses of the objective (Section 4.2) and subjective (Section 4.3) results of the user study.

Finally, we begin Chapter 5 with a summary of this work in Section 5.1. Then, in Section 5.2, we discuss possible solutions to remaining limitations followed by some of our ideas for possible improvements in Section 5.3.

Chapter 2

Background and Related Work

Current research related to this work addresses using mosaicing or stabilization techniques to improve the presentation of visual information acquired using either image or video capture devices from both non-aerial as well as aerial perspectives. This work primarily focuses on using mosaicing and stabilization techniques to improve the presentation of aerial video acquired using small and lightweight consumer grade video capture devices mounted on mUAV platforms [1].

After defining some commonly used terms throughout this thesis in Section 2.1, this chapter provides some of the common background material in Section 2.2 that will be needed for our discussion of related mosaicing and stabilization work presented in Section 2.3. Then, Section 2.4 provides some of the foundational material that this work builds upon.

2.1 Definition of Terms

Before we begin, we need to clarify some terms that will be commonly used throughout this document—image, scene, frame, canvas, view, and presentation.

An ***image***, I, is the common static 2-D collection of intensity information about a part of a 3-D ***scene***, or the 3-D real world, captured by any visual acquisition system or camera. A ***frame*** is an image within a sequence of multiple images. This means that each frame consists of an image with some corresponding identifier to express its temporal relationship to the other images within the sequence, or other frames. This identifier is usually a time-stamp or an integer together expressed with its frame as f_t and is relative to the first frame whose integer usually starts at zero, f_0. The image corresponding to f_t is written as I_t.

A ***canvas*** is the medium used to place the images in some spatial relation to one another. Common canvases related to this work usually exist in either the 2-D or 3-D domains, and may also contain a time component. The canvas we use in this work will always be the spatial 2-D canvas that will also usually include the time component. This is similar to watching a painting being painted onto an artist's canvas. The ***view*** can be defined by which part of the canvas is visible to the viewer at any given time, *i.e.*, defined by the bounding box of the intersection of the virtual camera's viewing frustum with the plane of the canvas which is the region of the canvas that is presented to the user at any given time. This is synonymous to how close a person is to the canvas being painted—the closer one is, the greater the detail or resolution but the less of the overall picture or spatial information that can be seen; and the further one is away from the canvas, the more one can see of the overall picture but less of the resolution can be seen. Also, the ***viewpoint*** is the point from which the virtual camera is viewing the canvas.

The ***presentation*** refers to how the images are being painted onto the canvas with relation to each other and time as well as how the pose of the view changes in time, *i.e.*, how the view is moved around the canvas in relation to the images that are being painted onto the canvas.

2.2 Frame Registration

In order to achieve video stabilization as well as video mosaicing, the current frame's image has to first be registered to its previous frame's image (see Figure 2.1). ***Registering*** two spatially and temporally adjacent frames to each other means to define

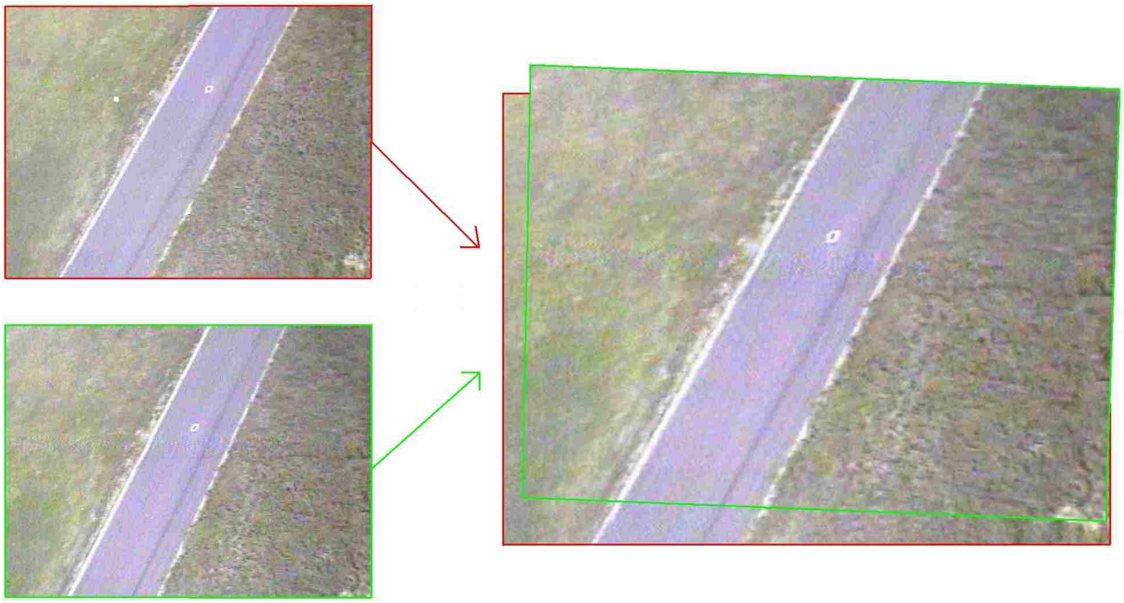


Figure 2.1: An example of two frames (left) that have been spatially aligned into one image (right) after being registered to each other. The image on the right is the aggregation or mosaic of the two frames on the left.

or to estimate the spatial relationship between their images such that the overlapping regions within their images can be closely spatially aligned using this relationship. This can be done using the projective geometric, intrinsic, and extrinsic relationships between the two frames, described in Sections 2.2.1, 2.2.2, and 2.2.3, respectively.

2.2.1 Projective Geometry

In order to understand the information that a frame presents and to register that information to another frame taken of the same scene at a slightly different time (temporally adjacent) and from a slightly different viewpoint (spatially adjacent), we need to first understand how a point in the real world relates to a point within a frame's image. Projective geometry is useful in describing this projective relationship of a point in the real world to its corresponding point on the capturing device, which is then represented by a pixel intensity estimation within the image.

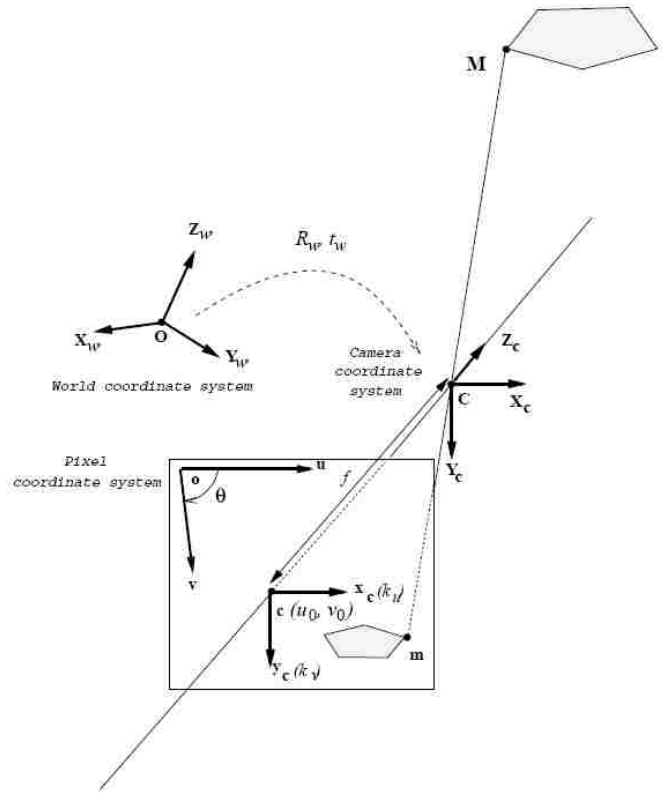


Figure 2.2: The General Projective Camera Model (from [4])

All visible points in the 3-D real world with coordinates X, Y, and Z with respect to the camera's coordinate system (C, X_c, Y_c, Z_c), where C is its origin (see Figure 2.2), project to an ideal 2-D viewing or retinal plane with coordinates u and v as follows:

$$u = -f\frac{X}{Z},\ v = -f\frac{Y}{Z} \tag{2.1}$$

where f is the focal length of the visual acquisition system (see Figure 2.2). Writing this in homogeneous coordinates, we get

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} -fX \\ -fY \\ Z \end{bmatrix} \tag{2.2}$$

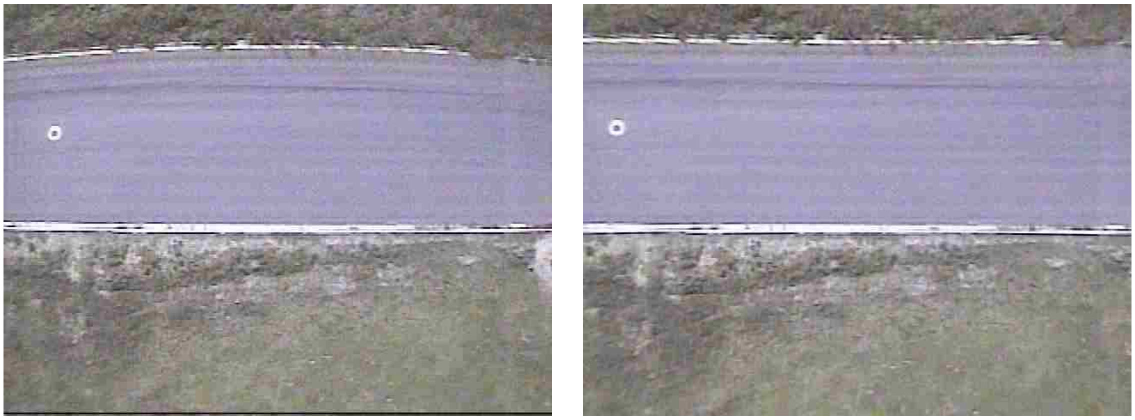
The usual retinal coordinates u and v are related to the projective coordinates, U, V, and W by

$$u = \frac{U}{W}, v = \frac{V}{W} \tag{2.3}$$

Thus, the relationship between the world coordinates and the retinal coordinates on the viewing plane is linear projective [4, 5], *i.e.*, straight lines are preserved through the projection [6].

2.2.2 Intrinsic Parameters

However, as can be seen in Figure 2.3, the projection of world coordinates to a camera's viewing plane is usually not so ideal, *e.g.*, straight lines do not always project to straight lines, and needs to take into account the usually imperfect physical properties of the camera. These physical properties can be described by a 3×3 matrix $\mathbf{A}$ and are called the ***intrinsic parameters*** of a camera. Intrinsic parameters are used to help define the relationship between real world objects seen and the pixels that respectively represent them in an image of a scene using an imperfect camera model.



(a) An Uncalibrated Image

(b) The Image Calibrated

Figure 2.3: Image Calibration Example. In the uncalibrated image (a) the intended straight lines of the road can be seen to bend across the image compared to its more ideal calibrated counterpart (b).

Rolling the focal length of the camera f into $\mathbf{A}$ gives us $\mathbf{A}$ expressed in terms of f and the intrinsic parameters:

$$\mathbf{A} = \begin{bmatrix} -fk_u & fk_u \cot\theta & u_0 \\ 0 & -\frac{fk_v}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

where u_0 and v_0 are the offset of the center of the camera's retinal plane in relation to the center of the image's pixel plane as shown in Figure 2.2. k_u and k_v are skew measurements along the axes of u and v, respectively, and take into account the possible non-squareness of the acquisition elements, which have aspect ratios depending on the actual size of the photosensitive cells of the camera as well as on the idiosyncrasies of the acquisition system. θ is a measurement of the physical angle between the possibly non-orthogonal u and v axes [4, 5].

These intrinsic parameters are independent of the camera's *extrinsic parameters*, *i.e.*, the pose of the camera, and can also be used to relate two different retinal coordinate systems of possibly two different cameras to each other.

$\mathbf{A}$ is useful in calibrating an image so that the linear projection property is preserved; otherwise, the extrinsic parameters or spatial relationship between two spatially adjacent images may be impossible to adequately estimate and would prevent us from properly registering the images.

Calibrating an image will appropriately warp it to estimate the image as if it was captured using an ideal camera. This can be done using the relationship

$$\mathbf{p}' = \mathbf{A}\mathbf{p} \tag{2.5}$$

where $\mathbf{p} = [u, v, 1]^T$ is any point in the original image and $\mathbf{p}' = [u', v', 1]^T$ is its corresponding point in its respective calibrated image.

We have integrated this functionality into our system using OpenCV's calibration functionalities [7]. This has provided us the means to estimate fk_u, fk_v, u_0, and v_0 as well as the radial distortion parameters k_1, k_2, p_1, and p_2. The cameras that our mUAVs currently use suffer mostly from these radial distortions. We can calibrate the image by compensating for these distortions by applying the following:

$$r = \sqrt{u^2 + v^2} \tag{2.6}$$

$$\begin{aligned} u'' &= u(1 + k_1 r^2 + k_2 r^4) + 2p_1 uv + p_2(r^2 + 2u^2) \\ v'' &= v(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2v^2) + 2p_2 uv \end{aligned} \tag{2.7}$$

$$\begin{aligned} u' &= fk_u u'' u + u_0 \\ y' &= fk_v v'' v + v_0 \end{aligned} \tag{2.8}$$

where (u, v) is each pixel's location in the original image and (u', v') is its respective location in the calibrated image.

2.2.3 Extrinsic Parameters

Extrinsic parameters define the physical relationship between the world's coordinate system and the camera's coordinate system (see Figure 2.2). They can be used to relate the poses of the camera(s), or viewpoints, used to capture two separate images.

A viewpoint's physical displacement in the real world, or its change in pose in the world coordinate system, can be described by using these extrinsic parameters. Matrix **D** is a 4 × 4 matrix describing this change of world coordinate system. **D** depends on six extrinsic parameters: three within a standard rotation 3 × 3 matrix

$\mathbf{R}_w$ to describe the viewpoint's 3-D rotation, and the other three within the vector $\mathbf{t}_w$ to describe the viewpoint's 3-D translation:

$$\mathbf{D} = \begin{bmatrix} \mathbf{R}_w & \mathbf{t}_w \\ \mathbf{0}_3^T & 1 \end{bmatrix} = [\mathbf{R}_w | \mathbf{t}_w] \tag{2.9}$$

2.2.4 The Projection

Altogether, the relationship between the world's 3-D coordinates (relative to the camera's coordinate system), the projective coordinates, and the image coordinates can then be described by the following:

$$\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \mathbf{A} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{D} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.10}$$

$$u' = \frac{U'}{W'},\ v' = \frac{V'}{W'} \tag{2.11}$$

The composite 3×4 matrix $\mathbf{P}$ is called the ***perspective projection matrix***, which relates 3-D world projective coordinates and 2-D projective coordinates [4, 5].

2.3 Related Work

Once the intrinsic and extrinsic relationships are established, they can then be used to estimate the spatial relationships among the images (Section 2.3.1) to then enhance the presentation of the visual information gathered either by (1) mosaicing a sequence of images together by merging them into a larger mosaic to be presented to the user, or by (2) stabilizing each image by warping it with respect to its temporally and spatially adjacent previous image in such a way that the user is presented with a stabilized sequence of images. Both approaches will effectually stabilize a sequence of images, and a review of the current strengths and weaknesses of mosaicing and stabilization is presented here as related to the current literature in Sections 2.3.2 and 2.3.3, respectively.

2.3.1 Image Alignment

Image and video stabilization and mosaicing could primarily use **D** to help define the spatial relationship between two spatially adjacent images. However, it is also common to not estimate **D** at all, but rather to directly estimate the spatial relationship needed to align the two images using a translation [8], rigid or Euclidean transformation [9, 10], similarity transformation [11], affine transformation, or projective transformation known as a homography [6, 12, 13, 14, 3, 15]. The relationships among these transformations are shown left to right in Figure 2.4—the transformations to the left are special cases of and less expressive than their respective transforms to the right.

In particular, we are interested mainly in the homography and the Euclidean transformation. The homography **H** is a model used to describe the spatial relationships among images taken of a 2-D scene (*i.e.*, planar surface) in 3-D space using a camera that may undergo motion in all 6-DOF$(x, y, z, \beta, \alpha, \gamma)$. **H** can be used to estimate the spatial relationships among images of a 3-D scene if the scene is relatively

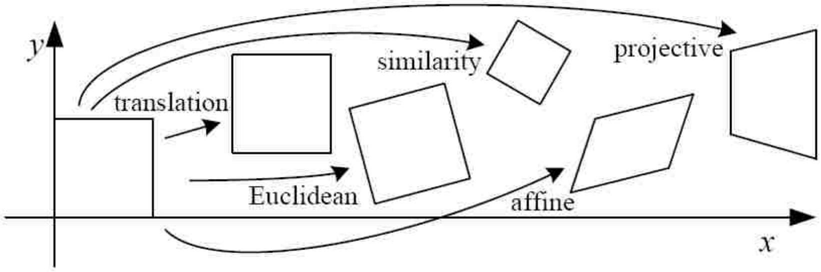


Figure 2.4: The basic 2-D planar transformations (from [6])

planar. We can assume that the scene we are capturing is relatively planar due to the capture device's relative distance from the scene, explained in Section 3.3. On the other hand, the Euclidean transform $\mathbf{Q}$ is a model used to describe the spatial relationships among images taken of a planar surface in 3-D space using a camera that is assumed to always be pointed straight down at the same relative distance from the 2-D scene and allowed only the 3-DOF motions (x, y, γ)).

In order to define or estimate the relative extrinsic parameters or the spatial relationship between any two images, we can use the relationships between the images' (1) respective camera poses, (2) overlapping image regions (area-based approaches), (3) common image features (feature-based approaches), (4) profiles, or (5) a combination of any of the four previous relationships.

Camera pose estimations need to be very accurate to be independently useful. If we had adequately accurate and synchronized pose estimate updates at least as frequent as our frame-rates, then stabilization and mosaicing could be much more easily achieved [13, 16, 17]. However, pose estimation equipment currently able to be carried on mUAVs is not yet accurate enough, nor are the current pose estimate updates as frequent as our current frame rate, nor are the pose estimates synchronized well enough to the corresponding frames [10, 1]. Therefore, this work assumes the complete absence of pose estimates.

Area-based approaches commonly estimate the extrinsic parameters by employing gradient-descent methods based on cumulative error differences between the overlapping regions of two images [18, 19, 13]. They tend to give much more accurate estimates of the extrinsic parameters than feature-based approaches will, but area-based approaches are still much too slow to be used in a real-time system.

Many area-based approaches employ pyramidal schemes to aide in speeding up the fitting process as well as to avoid possible local minima in the gradient-descent path [20, 21, 16, 9, 11, 13]. Others may employ correlating only subsets of the images' areas that contain a strong probability of having high information content relative to small adjustments made to each of the extrinsic parameters [15].

On the other hand, ***feature-based*** approaches [14, 3, 10] try to establish a correspondence between a sparse set of common features within two adjacent images. They are usually much faster than area-based approaches, but they also tend to suffer more from inherent noise in the correspondences and are usually not as accurate as area-based approaches can be. However, our experience supports that feature-based approaches can provide fast and accurate enough results needed to adequately improve the presentation of mUAV video in a real-time system. We address feature-based techniques in more detail in Section 2.4.

In addition to area-based and feature-based approaches, one of the simplest methods developed to register two images together employs a ***profile*** matching algorithm where the sums of the rows and columns of each image creates a profile of each image that is then used to align the temporally adjacent previous image [8]. Though fast, this method is limited to only roughly describe 2-D translational spatial relationships between frames. Also, it breaks down as soon as any disorienting rotations are introduced into the video.

2.3.2 Related Mosaicing Research

Mosaicing spatially aggregates a series of images together to expand the spatial views of their respective scene, essentially providing the user with a global history of what has been seen. It also removes the temporal component from the sequence of images and effectually stabilizes the presentation of the images' information. Mosaics are traditionally large static images that, if done well, eliminate much of the jitter related artifacts from the presentation.

However, the benefits gained in using mosaic presentations of the video come with related costs. Increasing the spatial view implies decreasing the viewable resolution of the presentation. Also, merging the images together into one static image eliminates many of the benefits of having the temporal and spatial information that is inherently within the video. Even though mosaicing may be facilitated by using the temporal relationships among images when creating the mosaic, once constructed, this temporal component is then removed from the presentation of the sequence of images. Mosaics then only represent a very small subset of the information provided within the video and discard the rest as redundant data [14], *e.g.*, a mosaic may only represent one side of a static object as well as only represent moving objects as stationary.

Global aerial mosaic construction can be greatly enhanced when facilitated with corresponding geo-reference images and terrain models [17, 13, 16]. However, current academic literature only addresses aerial video that is acquired using much larger and more stable aircraft that can be equipped with very accurate camera pose estimation and flight control equipment on board the aircraft. Such equipment is still too heavy and expensive for practical use on mUAVs.

Global error minimization techniques, also known as bundle adjustments, are vital to creating a convincing global mosaic [14, 3, 12]. Not performing some kind of bundle adjustment when creating a global mosaic can very quickly lead to a large amount of accumulated error, which will cause devastating misalignments in the mosaic. However, bundle adjustments are still too computationally expensive and not yet suitable to use in the context of real-time mUAV applications. In order to avoid having to perform these costly bundle adjustments, we will present only temporal-spatial local mosaics, essentially forgetting whatever information goes out of the presentation view.

2.3.3 Related Stabilization Research

In order to directly stabilize the presentation of a video sequence, the high frequency motions among the extrinsic relationships of the sequence of images must first be suppressed while still allowing the sequence to follow the intended motion of the scene, essentially smoothing the spatial relationships among the sequence. Using these new smoothed spatial relationships, the images can then be warped appropriately to present the user with a more stabilized presentation of the image sequence—preserving the temporal benefits of the video.

A common method used to smooth these high frequencies is to apply a low-pass filter. [8] uses a uniform kernel and [22] uses a gaussian kernel to convolve over a history of these spatial relationships to estimate a smoothed motion sequence. [10] uses parabolic curves to weight each motion within a neighborhood of frames relative to the current frame in order to compute a stabilized motion path. [11] computes a smooth motion path by performing a minimization of a cost function created to balance the motion of objects within the frame and the motion parameters computed to align each frame to its previous frame.

A restriction that is common among the current video stabilization literature is to display only a subset of the captured data in order to achieve stabilization. This includes limiting the viewing size of the stabilized sequence to the original frame size, which introduces blank regions at the edges of the stabilized view while other regions in the images are slightly shifted out of view to achieve stabilization [10, 9, 11, 13, 8]. In order to avoid revealing these blank regions, the data within them may be estimated and displayed. [22] uses an inpainting technique to estimate and display this missing data.

Another technique used to avoid displaying these blank regions is to further limit the display of the captured data. Many modern hand-held cameras apply this technique by using measurements from physical displacement instruments built into the camera to choose and display only a subregion of the samples of the scene acquired by the camera's CCD array. This stabilization technique can also be applied to images within a video sequence by using image analysis to measure the physical displacements between images to stabilize the image sequence as well as present the user with only a subregion of each image to avoid displaying these blank regions. This technique essentially performs some stabilizing transformation on each image and then zooms the view in close enough to the image or canvas to avoid displaying these blank edge regions.

However, in the context of using aerial video to perform searching tasks, it is important to present as much visual data to the user as possible. In order to avoid the need to throw away acquired data or to estimate the real data by restricting our presentation view to the original frame size, we will focus on adequately expanding the size of the presentation view to achieve stabilization.

2.3.4 Image Acquisition Platforms

Images may be acquired using capture devices restricted to platforms that are free to be displaced in any combination of the six degrees of freedom, 6-DOF—$(x, y, z, \beta, \alpha, \gamma)$, where x, y, and z are the capture device's displacements along its horizontal, vertical, and looking-at axes, and β, α, and γ are the pitch, roll, and yaw angles of rotation about the x, y, and z axes, respectively.

[15] creates mosaics of images acquired using a capture device whose focal point is fixed but is otherwise free to rotate about its α and γ axes. [9] stabilizes video using a platform with predictable physical motion properties, whose dominate motion is fixed in either x, y, or z, and whose high-frequency motion, or jitter, is assumed to be mainly interpreted small motions in x, y, or β. Because mUAV platforms are free to move spontaneously in any of the 6-DOF, like [10], we cannot assume a dominant

motion in any one axis and we must also be prepared to handle unpredictable motions in all 6-DOF.

However, we can assume that most of the objects in the scene are relatively far away from a capture device mounted on a mUAV, making the parallax effects between I_t and I_{t-1} negligible. It then follows that the image registration of a mUAV video approaches a degenerative planar case, allowing us to approximate the spatial relationships between frames with a homography (Section 2.4.3) rather than with the more complex fundamental constraint [23]. Similar to [9], this assumption also allows us to approximate small motions in β and α as displacements along the x and y axes, respectively, and we can also approximate small displacements along the z axis with a scale factor, s.

Another useful observation used by [9] is that the human visual system is mostly sensitive and distracted by high-frequency motions in x and y, or horizontal and vertical jitter, making these motions of greater concern.

2.3.5 Related UAV Vision Research

Much research has been done to improve the presentation of aerial information acquired using large manned aircraft and UAVs [13, 16, 17]. These larger manned aircraft and UAV airframes commonly use consumer- and professional-grade capture devices and equipment that offer high-quality video and images as well as very accurate pose estimations, respectively. However, such equipment is also still much heavier and more expensive than their smaller, lightweight, and lower quality counterparts that can currently be used on mUAV airframes. This gives mUAV-acquired images and video the disadvantages of having lower resolution, lower signal to noise ratios, and greater distortion caused by a lack of imperfect lenses and imperfect camera calibration.

2.3.6 Related mUAV Vision Research

Except for [10], whose work was developed in tandem to this work, we have found very little in the academic research that addresses the problems commonly associated with mUAV-acquired video. One line of recent research addressing mUAV-acquired video has been done in the context of smaller helicopter mUAVs to assist in landing the aircraft autonomously by visually locating its physical relationships to a landing site, visually estimating the aircraft's current height above ground, and visually estimating its current velocities [24]. However, we have found that the topic of analyzing video acquired using fast-moving forward-velocity mUAVs for the purpose of enhancing a user's ability to identify objects of interest is quite new and unexplored.

2.4 Feature-Based Methods

Since bundle adjustments as well as area-based methods are still too computationally expensive for real-time applications, we will concentrate on feature-based methods without bundle adjustments. In order to mosaic or stabilize calibrated images or video using feature-based methods, we first need to establish a sufficient set of feature correspondences between images of neighboring frames, I_t and I_{t-1}.

This is usually done using principles from optical flow and structure from motion to help describe the relationships between the pixels seen in both I_t and I_{t-1} by finding interesting features in I_t (Section 2.4.1), like corners, and matching those features to corresponding interesting features found in I_{t-1} (Section 2.4.2). Using these feature correspondences, under certain conditions, the spatial relationship between the two images can be approximated by a homography (Section 2.4.3), which can be estimated using a RANSAC algorithm (Section 2.4.4).

2.4.1 Identifying Good Features to Track

Assuming that each image of the video is calibrated, the next step is to identify and correspond common features between adjacent images. One common method used to do this is to calculate each pixel's probability of being an interesting point in its own image as well as in its spatially adjacent images. Points of the same scene that usually remain of high interest from multiple displaced viewpoints are edges and corners.

One of the most popular algorithms used to find these highly probable corner pixels is described in [7]. Sobel first-derivative operators are used to take the derivatives in the horizontal (D_x) and vertical (D_y) directions of an image. The following 2×2 matrix $\mathbf{c}$ is then created from the sums of the derivatives D_x and D_y over a small region of interest to detect corners:

$$\mathbf{c} = \begin{bmatrix} \sum D_x^2 & \sum D_x D_y \\ \sum D_x D_y & \sum D_y^2 \end{bmatrix}$$

If λ_1 and λ_2 are the eigenvalues of $\mathbf{c}$, and ξ is some predetermined threshold, then all of the regions that satisfy $\lambda_1, \lambda_2 > \xi$ can be considered highly probable corner pixel regions. This set of pixels we will call $\mathbf{P}_t$.

2.4.2 Matching Good Features

By identifying this relatively small set of pixels in $\mathbf{P}_t$—small in comparison to the total number of pixels in each image—that have high probabilities of being good features from one frame of video to the next, we can use it to establish an initial feature correspondence set $\mathbf{C}_t$ between I_t and I_{t-1}.

This can be done by applying a Lucas-Kanade pyramidal algorithm that is based on the sum of squared differences of areas local to each feature in $\mathbf{P}_t$ and similar corresponding feature areas in I_{t-1}. So, for each feature element b in $\mathbf{P}_t$, it will find a likely corresponding or matching feature element a, and add the corresponding features b, a to $\mathbf{C}_t$ [7]. It is important to note that this process does not consider the good features $\mathbf{P}_{t-1}$ found for I_{t-1} when composing $\mathbf{C}_t$—which avoids the negatives of introducing a bias based on $\mathbf{P}_{t-1}$, but on the other hand does not take advantage of its possibly beneficial prior.

2.4.3 The Homography

Assuming that the images are of a relatively planar scene (Section 2.3.4), these feature correspondences can now be used to infer an approximate spatial relationship between the two adjacent images.

Also known as a perspective transform, a homography $\mathbf{H}$ can be used to describe the spatial relationship between each point within an image of a planar scene to a point within a spatially adjacent image, or rather an image taken of the same scene from a slightly displaced viewpoint. $\mathbf{H}$ therefore constrains the mapping of each feature point in I_t to an estimated corresponding feature point in I_{t-1}. It is important to note that $\mathbf{H}$ is homogeneous, *i.e.*, it is defined only up to a scale. This means that each image point actually corresponds to some point that lies anywhere on a corresponding ray in the real world [6, 5].

To precisely solve for $\mathbf{H}$, we need at least four exact non-co-linear feature correspondences. This provides a system of eight equations and eight unknowns, which can then be solved for directly. However, since we can expect a certain measure of error due to the inherent discretization problems that can prevent us from establishing exact feature correspondences, it is good practice to overconstrain this problem by using several more feature correspondences to set up a system of equations and estimate $\mathbf{H}$ using a linear least-squares minimization algorithm.

2.4.4 RANSAC

A common problem in estimating $\mathbf{H}$ using an overconstrained system of equations is that several of the feature correspondences may be severe outliers, which can very negatively impact the estimation of $\mathbf{H}$. Following is a brief explanation about how a RANSAC algorithm can be used to improve the probability of adequately estimating $\mathbf{H}$ given an initial feature correspondences set that may contain several outliers.

Once we have a set $\mathbf{C}_t$ that contains a set of best-guess feature correspondences between two frames, we can then feed $\mathbf{C}_t$ into a RANSAC (Random Sampling with Consensus) algorithm [2, 3]. This algorithm assumes that there exists a matrix $\mathbf{H}$ that can constrain each feature feature correspondence to contain a point in I_t and another point in I_{t-1}. Given that $\mathbf{C}_t$ is a large enough set, RANSAC will be able to estimate this constraining matrix, $\mathbf{H}$.

It begins by randomly selecting a subset of $\mathbf{C}_t$, then computes a temporary constraining matrix $\mathbf{H}_t$ that minimizes a total distance error measure (ϵ_t) incurred when that constraint is applied to all of the feature points in the original $\mathbf{C}_t$ that are in I_{t-1} and compared to the locations of their corresponding points that are in I_t. Only four non-linear feature correspondences are needed to estimate this matrix [5]. RANSAC will then randomly select another subset of $\mathbf{C}_t$, compute a new $\mathbf{H}_{t+1}$, and set $\mathbf{H}$ to the matrix $\mathbf{H}_m$ with the lowest ϵ. It will therefore return the constraining matrix, $\mathbf{H}_m$, with the lowest ϵ, as well as which subset of $\mathbf{C}_t$ is most consistent with respect to $\mathbf{H}$ within some predetermined error threshold [3].

2.5 Our Approach

Since area-based approaches are currently too slow to use in a real-time search situation, we use a feature-based approach. Using the projective geometric and intrinsic relationships among the images of mUAV-acquired video, we can calibrate the images and identify and correspond common features between neighboring frames.

Because the scene is relatively planar with respect to the altitude of the mUAV's camera, **H** can then be useful in approximating the spatial relationships among the images as well as refining our feature correspondence set by identifying and omitting outliers, many of which are caused by noisy and lower quality images commonly associated with mUAV-acquired video. With a good inlier set of feature correspondences, we can better estimate these spatial relationships using either a refined **H** or a Euclidean transform **Q**, also known as a rigid body transform (see Section 3.4.1).

Wanting the benefits that a mosaic can provide, but unable to perform the costly bundle adjustments required for a global mosaic, we concentrate on building and maintaining only a local mosaic using a stabilized presentation of the video. So once we have a sufficient number of frames with corresponding **Q**'s in our history, we can then aggregate the images together to estimate a local mosaic. We can also use curve fitting to compute a smoothed sequence of spatial relationships among the frames, *i.e.*, a smoothed viewing path of the image aggregation path. These spatial relationships along with the novel smoothed path can then be used to transform each frame to provide both a localized mosaic as well as a stabilized presentation of the video. Further details of our approach are described in greater detail in the following chapter, Chapter 3.

Chapter 3

Methods

In order to improve the presentation of mUAV-acquired video, we have devised three separate but related presentations: the E-mosaic view, the stable-E view, and the stable-E-mosaic view. As outlined in Algorithm 1, several common steps are involved in creating these three different view presentations. This chapter presents the logical progression of these steps, which include capturing and preprocessing the images (Section 3.1), identifying good features shared among adjacent images (Section 3.2.1), establishing feature correspondences between these images (Section 3.2.2), identifying and discarding correspondence outliers (Section 3.3), and then estimating the spatial relationships between images to establish the image aggregation path (Section 3.4). Once these spatial relationships are established, we can then create and display any of the three different presentation views.

The first and simplest of these presentation views is the Euclidean mosaic view, or ***E-mosaic view***. It involves expanding the viewing size and aggregating each image onto the canvas to create a larger local mosaic. These image aggregations work well until the images begin to be aggregated onto the canvas outside of the viewing frustum. At this point, an obvious solution would be to translate the viewpoint when necessary so as to follow this image aggregation path and always keep the current frame within the presentation's viewing frustum. These viewpoint translations compose what we call the ***view path***. This solution is employed by

For each frame_t of mUAV-acquired video:

1. Sample and deinterlace frame_t into I_t (Section 3.1.1).
2. Calibrate I_t (Section 3.1.2).
3. Find a good set of features $\mathbf{P}_t$ in I_t (Section 3.2.1).
4. Fill the set of feature correspondences, $\mathbf{C}_t$, by corresponding each element of $\mathbf{P}_t$ to similar features in I_{t-1}, $\mathbf{P}'_t$, such that $\{\{b, a\} \in \mathbf{C}_t : a_i \in \mathbf{P}'_t \ \& \ b_i \in \mathbf{P}_t\}$ (Section 3.2.2).
5. Apply a homography filter to $\mathbf{C}_t$, to make a filtered set of feature correspondences $\mathbf{Y}_t$, such that $\mathbf{Y}_t \subset \mathbf{C}_t$ (Section 3.3.1).
6. Apply the homography RANSAC filter using $\mathbf{Y}_t$ and $\mathbf{C}_t$ to make a feature correspondence inlier set $\hat{\mathbf{C}}_t$ (Section 3.3.2).
7. Estimate $\mathbf{t}_t$, which is the average of the current to previous disparity vector of each element within $\hat{\mathbf{C}}_t$ (Section 3.4.2).
8. Compute the residual vector set $\mathbf{V}_t = \{\{\mathbf{t}_t + b, a\} : \{b, a\} \in \hat{\mathbf{C}}_t\}$ (Section 3.4.3).
9. Estimate a center of rotation $\mathbf{o}_t$ among $\mathbf{V}_t$ (Section 3.4.3).
10. Estimate an angle of rotation θ_t among $\mathbf{V}_t$ around $\mathbf{o}_t$ (Section 3.4.3).
11. Compose a spatial relationship $\mathbf{Q}_t$ from I_t to I_{t-1} by combining $\mathbf{t}_t$ and θ_t. (Section 3.4.4)
12. Accumulate the cumulative Euclidean transform, $\mathbf{Q}'_t = \mathbf{Q}_t \mathbf{Q}'_{t-1}$.
13. Use $\mathbf{Q}'_t$ to spatially align I_t relative to I_{t-1} and I_0 onto the canvas I'_t using one of the three presentation views—the E-mosaic, stable-E, or stable-E-mosaic views (Sections 3.6, 3.7, and 3.8, respectively).
14. Display the view of I'_t to the user using a mUAV video presentation user interface (Section 3.9).

Algorithm 1: The general algorithm we use to enhance the presentation of mUAV-acquired video to the user.

our E-mosaic presentation method and works quite well at expanding the viewing frustum, removing video content jitter, and improving users' orientation.

However, in the context of fast forward-velocity mUAV video, these necessary viewpoint translations become commonplace and effectually reintroduce some of the original distracting jitter back into the presentation. In order to further remove this jitter from the presentation, we present another solution in Section 3.7.1 that smooths the view path to create a novel ***smoothed view path***. In Section 3.7 we address our presentation of a stabilized Euclidean view, or ***stable-E view***, using this smoothed view path independent of any mosaic, which improves users' orientation and balances the removal of content jitter from the original presentation and the presentation jitter of the E-mosaic view.

We then combine the complementary strengths of the E-mosaic and stable-E views into our stabilized Euclidean mosaic view, or ***stable-E-mosaic view***, in Section 3.8. We commonly refer to the view of each presentation as the view or presentation view of the original, E-mosaic, stable-E, or stable-E-mosaic.

Finally, in Section 3.9 we propose some user interface approaches that we use to address some of the remaining issues of these three presentation views.

3.1 Image Capture and Preprocessing

Due to the payload limitations of mUAVs and the weight of the current hardware required to store the large amounts of video data needed, it is necessary to transmit mUAV-acquired video to a ground station for storage and processing. Both this transmission as well as the miniature cameras used introduce some artifacts that need to be addressed before we can register the images of the video.

First, the transmission of the video introduces noisy and invalid regions on the edges of the images that our system needs to be adequately robust to. Second, the video camera currently being used transmits the data in an interlaced fashion.

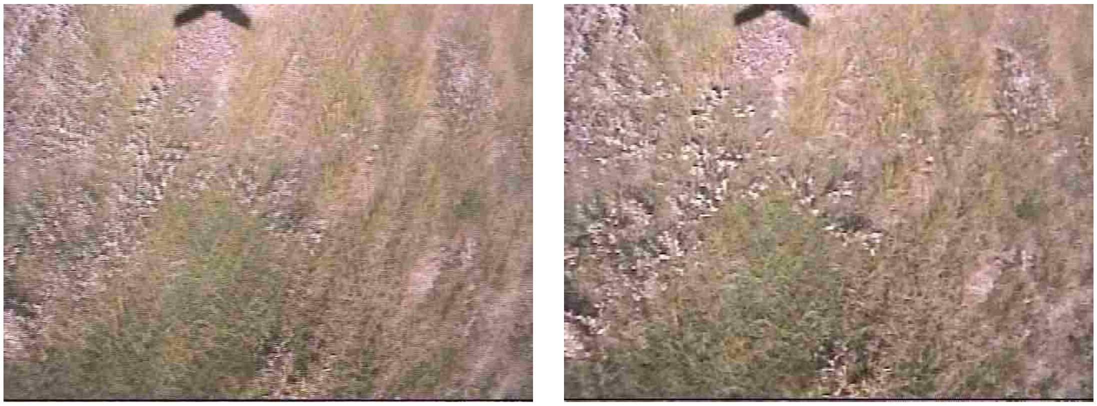
This interlacing can introduce many harmful artifacts such as false good features in the image as well as blurry ghosting effects when the video is viewed on a progressive scan or higher-resolution monitor. Image sampling and deinterlacing can be used to address these issues as described in Section 3.1.1.

Third, the inherent physical misalignments within the components of the camera will introduce misalignments within the images that may need to be corrected. Also, the lenses used can also introduce radial distortions that will need to be addressed. These misalignments and distortions can be mostly corrected by calibrating the image, which we discuss in Section 3.1.2.

3.1.1 Sampling and Deinterlacing the Image

It is useful to note that before we can calibrate each image, we should first perform any image sampling and deinterlacing that needs to take place; otherwise, the calibration may warp the image, *e.g.*, bend each row across multiple rows, in such a way that would make the sampling and deinterlacing process much more difficult to perform.

Sampling and deinterlacing mUAV-acquired images need to be considered within two different contexts: finding good features and displaying the images to the user. Finding good features within an image is key in the frame registration process, but it can also be a computationally expensive process; so, we would like to find good features as fast as possible and accurately enough to be used to describe well the spatial relationships between two adjacent images by using a representation of each image. Using a down-sampled representation of the image can help us speed up the process of finding good features. However, the images that we display to the user need to contain as much detail as possible with the least amount of distracting artifacts. So we do not want to down-sample too much, nor in a way that will introduce harmful artifacts.



(a) Original interlaced image

(b) Same image, but deinterlaced

Figure 3.1: Examples of an interlaced and deinterlaced image

The transmitted video that we presently receive from mUAVs consists of 640×480-sized interlaced images at 30 fps. Interlacing basically transmits every even row at $t = i/60$ and every odd row at $t = (i + 1)/60$ and then combines the odd and even rows into one image received every 1/30 seconds. This introduces into the interlaced video blurry ghosting effects as seen in the image of Figure 3.1(a) compared to its deinterlaced counterpart seen in Figure 3.1(b). This ghosting can introduce false good features and can have a very negative impact on our finding good features process as well as decrease the detectability of the presentation. In order to remove these ghosting effects, we need to deinterlace the image.

Since we are essentially receiving 640×240 of new image data at 60 fps, *i.e.*, every 640×480-sized interlaced image that we get at 30 fps can be split into two 640×480-sized temporally adjacent interlaced images at 60 fps, each having its odd rows blank and even rows filled in, or alternately visa versa. These split images can then be deinterlaced and displayed to the user at up to 60 fps.

We deinterlace each image by convolving each blank row using one of the following four convolution kernels:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \frac{1}{6}\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \text{or } \frac{1}{8}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{3.1}$$

The first of these is the fastest and basically just copies each filled-in row into the blank row directly below it; however, it will also cause the most aliasing effects.

The fourth of these is the slowest and basically estimates each pixel by performing a center-weighted averaging of the six neighboring pixels directly above and below it. This deinterlacing technique is also the best of the four at estimating each pixel value and causes the least amount of aliasing; however, the human visual system has a hard time distinguishing the difference between duplicating the rows and the more accurate means of filling in interlaced intensity information. These deinterlaced images can be used to identify good features as well as used in the presentation of the video.

Because the speed of the process of finding good features in an image is dramatically affected by the size of that image, and since we get only 640 × 240 new data at 60 fps, we can use smaller images sampled from the original 640 × 480-sized interlaced images to help us decrease the related computational overhead of finding good features in the full-sized images. However, the coarser our image sampling is, the less accurate the estimated good features will be.

One useful sampling method is to create a half-height image by copying into its rows every other row of the original image. Doing this will provide us with a deinterlaced version of the image as well as cut the time taken to find good features in half, implying that we can essentially process 640 × 240-sized deinterlaced images at 60 fps. Also, using half-height images requires an additional two things: that our estimated good features will need to be horizontally scaled by two, and that our calibration parameters need to be relative to the size of the half-height image.

One drawback to this approach is that it will introduce some aliasing artifacts that can possibly degrade the good features contained in the image due to the original data having been sampled below the Nyquist sampling rate; however, we have not experienced any noticeable resultant degradation in the overall accuracy of good features found using half-height images versus a combination of the other options previously described.

Another sampling issue that needs to be addressed is the noisy and invalid rows and columns commonly introduced along the borders of mUAV transmitted video. For example, looking closely at the images in Figure 3.1, the bottom four rows of pixels are either black rows or invalid rows. This becomes a problem when locating good features (addressed further in Section 3.2.1) as well as when aggregating the images together into a local mosaic as we do in our E-mosaic and stable-E-mosaic views. These artifacts are usually particular to the capture device and usually remain constant, so the invalid regions of the transmitted images may be analyzed preflight so that the sampling process may also exclude these regions. In our data, we have observed that the number of rows or columns along each border that needs to to be excluded varies between zero and ten.

In practice, we commonly use the first kernel previously listed to deinterlace the display and the half-height images to find good features; however, the exact combination of deinterlacing and sampling methods vary depending on each situation's requirements for speed versus accurate results.

3.1.2 Calibrating the Image

After the image is sampled and deinterlaced, we can calibrate the image if needed. Estimating the calibration parameters needed to calibrate the images of a mUAV video should occur before each flight unless there is enough confidence that the camera's calibration parameters are known and have not changed since the camera's last flight. Small changes in calibration parameters may occur if the camera on the mUAV is jarred sufficiently enough to affect the physical properties of the camera. So, our system has to be—and is capable of—performing a camera calibration sequence previous to each flight on the field.

Calibrating (pre-warping) each image introduces quite a lot of computational overhead into the system. Even though calibrating only half-height images helps, the calibration process still introduces a significant amount of overhead. Other than the several multiplications per pixel required, each pixel's new location in the calibrated image rarely falls onto an exact pixel location in the calibrated image. Similar to the issues related to rotating an image, in order to avoid the holes as well as adequately estimate the correct pixel intensities in the calibrated image, a costly backwards warp using bilinear interpolation is usually performed.

Another issue that needs to be handled is that the calibrated images are no longer rectangular images. We address this by clipping enough of the edges so that our calibrated images appear rectangular. Another method would be to use an alpha channel or image mask to flag valid and invalid regions of the calibrated images.

Since this whole calibration process can be relatively computationally expensive for a real-time system to perform, rather than having to calibrate the whole image, we can calibrate only the x and y location of each good feature found. Doing this allows us to find good features using an image whose data is less estimated than a calibrated image would present. It also allows us to forego the complications related to non-rectangular calibrated images. Another approach would be to not calibrate the images at all and to relax the feature correspondence requirements imposed on valid feature correspondences in the filtering processes—which we address further in Section 3.3.

If calibrated images are to be used in the display, then either the full-height or half-height images will each need to be calibrated. Calibrating half-height images is obviously less expensive than calibrating full-height images but requires that the camera's calibration parameters be estimated based on half-height image sizes. The resulting calibrated half-images can then be used to establish good features which will need to be scaled appropriately. Also, before the calibrated half-height images can be used in the display, they too need to first be adjusted back to appropriately scaled images—and in the case of displaying full-height images, this can be done by using any of the deinterlacing methods previously described.

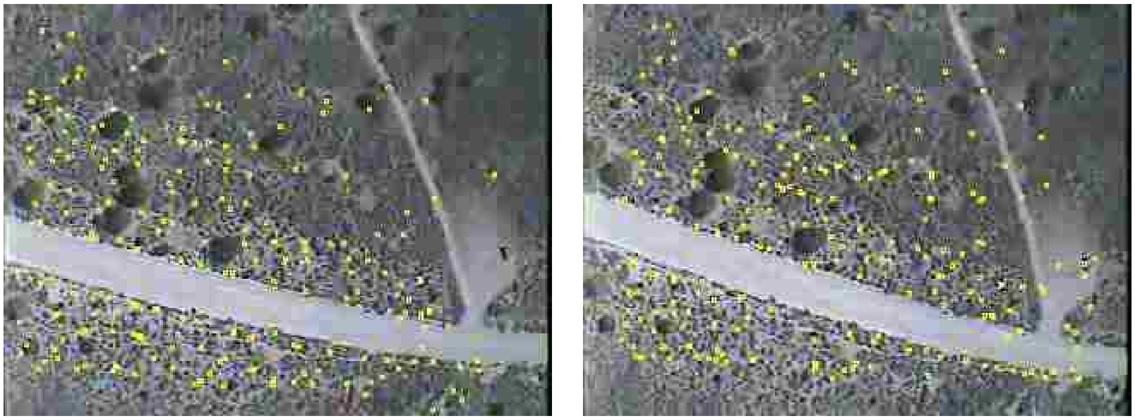
3.2 Establishing Point Correspondences

At this point, we will assume that I_t now represents an appropriately scaled version of the image of frame_t—*e.g.*, it may be a full-height or half-height half-width image. I_t may also be deinterlaced or deinterlaced and calibrated.

The next step in our process is to establish enough valid feature correspondences between the images of adjacent frames so that we can describe the spatial relationships among the images. This involves two steps. First, we need to find good features in each image that have a high probability of being good features found in their respective temporally and spatially adjacent images. Once we have a good feature set for I_t, for each feature in that set we need to then establish our best guess as to which pixel it corresponds to in I_{t-1}, giving us our correspondence set $\mathbf{C}_t$.

3.2.1 Finding Good Features

We find good features in each image using the methods described in Section 2.4. In our context, it is important that the features in the feature set of I_t, $\mathbf{P}_t$, be distributed well throughout the image as well as be adequately distanced from one another and the least co-linear as possible so as to avoid degenerative cases similar to those mentioned



(a) Good features $\mathbf{P}_{t-1}$ found in I_{t-1}

(b) Good features $\mathbf{P}_t$ found in I_t

Figure 3.2: Examples of unfiltered similar good features highlighted in yellow found within adjacent images

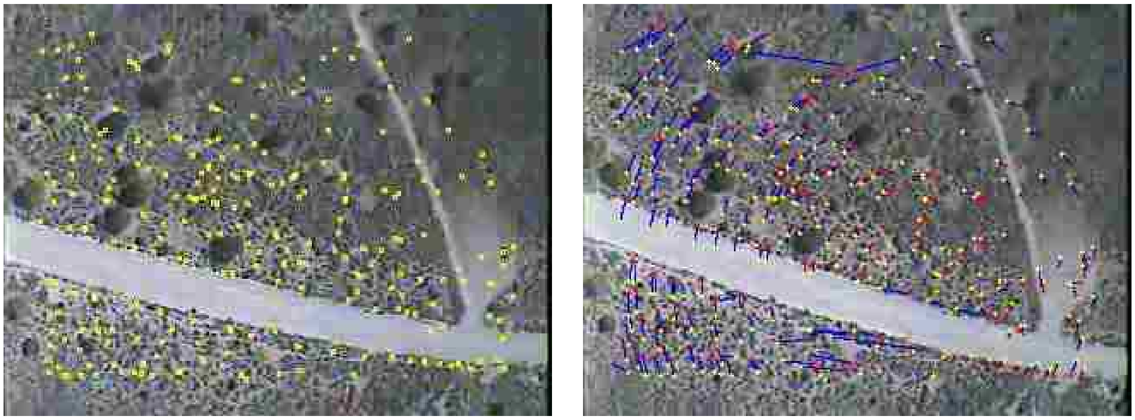
in [23]. In addition, features near the borders of I_t need to be excluded since their content is usually not within the unpredictable overlap of adjacent images.

An example of similar good features found between I_{t-1} and I_t with healthy distributions can be seen in Figure 3.2. Note that there are many similar features that could be correctly matched in both $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$, and that there are also many different features between them that could easily cause false matches.

It is also possible to perform a sub-pixel accuracy refinement on $\mathbf{P}_t$. Doing this may help improve our homography estimate processes described in Section 3.3 as well as compensate for any simplification steps that may have been taken in Section 3.1. However, as explained in Section 3.3.2, if we reasonably relax the accuracy requirements of estimating $\mathbf{H}$, then performing sub-pixel accuracy refinements on $\mathbf{P}_t$ would turn into wasted cycles.

3.2.2 Matching Good Features

Once $\mathbf{P}_t$ is established, we can establish a likely feature correspondence set $\mathbf{C}_t$ between I_t and I_{t-1}. As shown in Figure 3.2, making a correspondence set between $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$ would be limited to identifying only the similar features between them, which



(a) The feature set $\mathbf{P}'_t$

(b) The correspondence set $\mathbf{C}_t$

Figure 3.3: Unfiltered correspondence set between the images of two adjacent frames, I_{t-1} and I_t. The red highlights in (b) indicate the feature set $\mathbf{P}_t$ of I_t. The yellow highlights in both (a) and (b) indicate $\mathbf{P}'_t$, the set of features of I_{t-1} found relative to $\mathbf{P}_t$. The blue lines show $\mathbf{C}_t$, the set of correspondences between $\mathbf{P}_t$ and $\mathbf{P}'_t$.

could end up being a relatively small set. However, if we try to find a corresponding feature in I_{t-1} for each feature in $\mathbf{P}_t$, we are no longer limited to only the similar features between $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$. This is basically how the Lucas-Kanade pyramidal algorithm works, as described in Section 2.4.2. It is what we employ to establish our initial feature correspondence set $\mathbf{C}_t$, as shown in Figure 3.3.

3.3 Filtering Point Correspondences

The correspondences shown in Figure 3.3 are obviously not all consistent with each other, and adequate frame registration is not possible with noisy correspondences like these. The sets of $\mathbf{C}$ of mUAV-acquired video are commonly cluttered with many similar extreme outliers. Such outliers will cause poor estimations of the spatial relationships among the images which will very quickly accumulate a significant amount of error in the image aggregation path. Therefore, it is imperative to identify and disregard these bad feature correspondences, *i.e.*, the correspondence outliers.

To do this, we apply two filtering processes. The first filter that we apply is an optional filter on the correspondences (Section 3.3.1). This filter can basically seed the following homography RANSAC filter (Section 3.3.2) with a better set of correspondences to shorten the average time needed for RANSAC to converge and settle on a good set of correspondences. Then, we apply the homography RANSAC filter, which has thus far given us the most accurate set of correspondences compared to the many other correspondence filter possibilities that we have implemented and experimented with.

We define a good set of inliers to be a consistent set of correspondences between two images spatially related by a reasonable spatial transformation. The evaluation function used in both filters to estimate this spatial transformation will be a homography, previously described in Section 2.4.3. We use a homography based on the assumption that we are capturing a scene that is relatively planar [23]. This means that the relative distances from the focal point of the camera to any point within the overlap between two adjacent images of the scene captured can be approximated by a plane, and that the distance from any of these points to that approximating plane will be very small compared to the distance of the camera d to that point, *i.e.*, $d_{approximatingPlane}(p) << d_{camera}(p)$.

One of the problems with trying to determine this set of inliers is that the relative sizes, orientations, or the locations of the correspondence vectors cannot be used independently to determine a sufficient inlier set. The only way to really determine a good inlier set is to use the composite relative relationships among the correspondence vectors' relative sizes, orientations, and locations. Assuming a near-planar scene, a homography can provide such a composite relative relationship. We can therefore use an estimation of the homography given $\mathbf{C}_t$ to help us define an inlier subset of $\mathbf{C}_t$.

3.3.1 Homography Filter

As previously explained, this homography filter's main purpose is to seed the homography RANSAC filter with a better set of correspondences so that the homography RANSAC filter has a greater probability of converging slightly faster than it would otherwise. Accordingly, we have not seen any noticeable differences in the accuracy of inlier correspondences after applying the homography RANSAC filter after having been seeded with filtered correspondences $\mathbf{Y}$ using this filter. The reason for this will be explained further in Section 3.3.2.

We implemented the homography filter as outlined in Algorithm 2. The values of d are the error distances between feature points in I_{t-1} and the estimated transformed point of the corresponding feature points in I_t given $\mathbf{H}'_t$. The $\mathbf{H}'_t$ used for this filter is the homography computed using all of the points in $\mathbf{C}_t$, which over-constrains the problem. $\mathbf{H}'_t$ is thus estimated by a least squares minimization solution—estimating the true homography relationship $\hat{\mathbf{H}}_t$ between I_{t-1} and I_t.

The logical premise for the usefulness of applying this filter is that the feature correspondences within $\mathbf{C}_t$ that correspond to the lowest residual vectors' magnitudes will have a higher probability of being within the true inlier subset of $\mathbf{C}_t$ than the contrary—implying that $\mathbf{Y}_t$ arguably represents a more probable percentage of

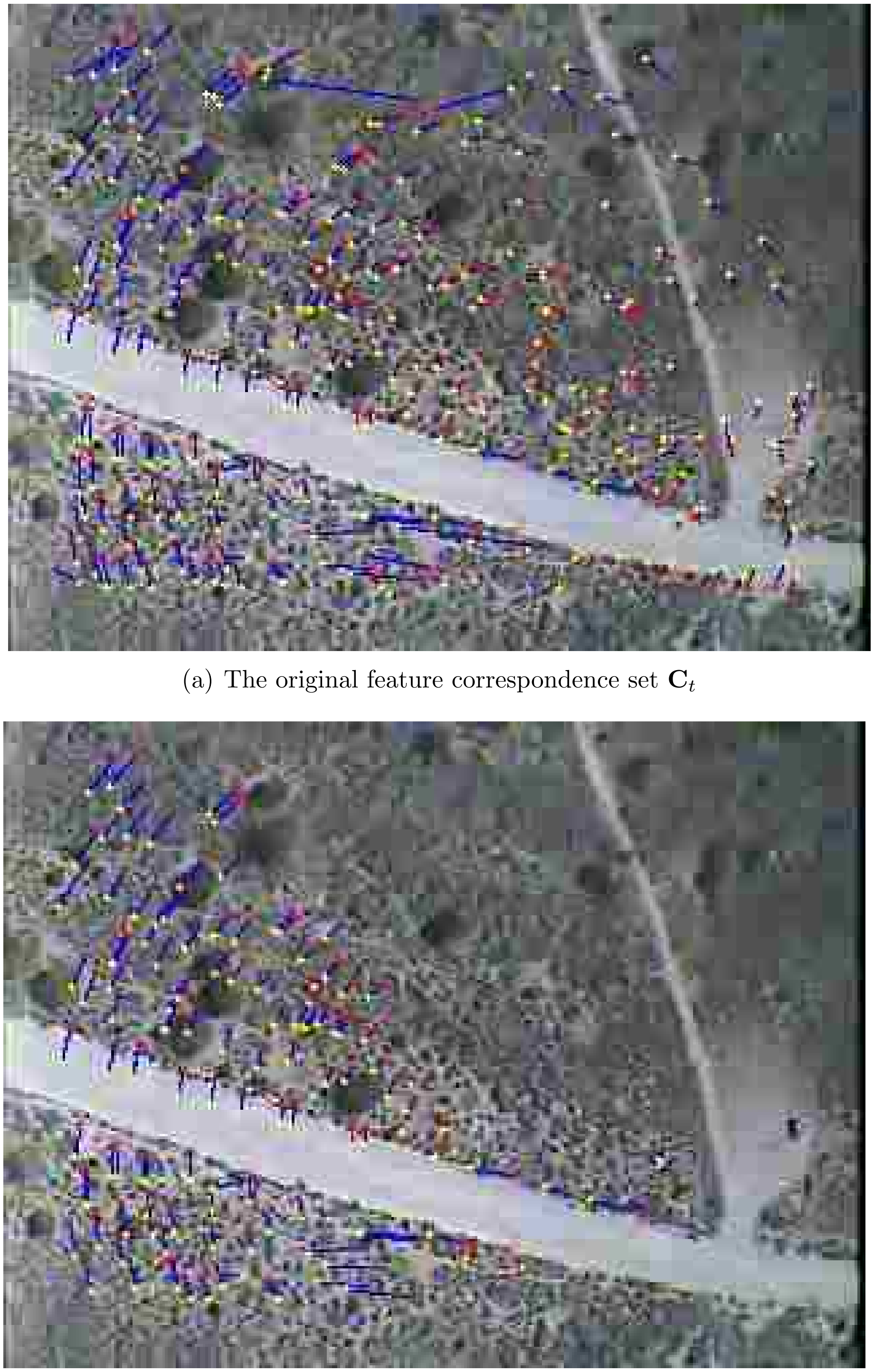
```
1. Estimate $\mathbf{H}'_t$ using $\mathbf{C}_t$.
2. Compute an array $d$ such that $d_i = ||\mathbf{H}'b_i - a_i||$ for each $\{b_i, a_i\} \in \mathbf{C}_t$.
3. Sort $d$.
4. Set $d_{low}$ to the lowest value of $d$.
5. Set $d_{high}$ to the highest value of the lowest $\mu\%$ values of $d$.
6. For each $\{b_i, a_i\}$ where $(d_{low} \leq ||\mathbf{H}'b_i - a_i|| \leq d_{high})$
    (a) Insert $\{b_i, a_i\}$ into the filtered correspondence set $\mathbf{Y}_t$.
This implies $\mathbf{Y}_t \subset \mathbf{C}_t$.
```

Algorithm 2: The homography filter algorithm

inliers than its superset $\mathbf{C}_t$. However, even in the cases that it might not, the homography RANSAC filter can still remain logically robust to producing a highly probable close estimate of the true $\mathbf{H}$, $\hat{\mathbf{H}}$, given that $\mathbf{C}_t$ contains an adequate sample of good correspondences. Furthermore, if $\mathbf{C}_t$ does contain an adequate sample of good correspondences, then it is highly probable that this subset will too; and the converse is also true, that if $\mathbf{C}_t$ does not contain an adequate sample of good correspondences, then it is highly probably that this subset will also not contain an adequate sample of good correspondences.

Because this filter's evaluation function uses $\mathbf{H}'_t$, which is a noisy estimation of $\hat{\mathbf{H}}_t$ and is commonly estimated using noisy feature correspondences, it is possible that applying this filter could filter out many inliers as well as leave many outliers within the filtered set. However, given that the original $\mathbf{C}_t$ contains a sufficient number of inliers, it may be safe to assume that $\mathbf{H}'_t$ is approximated well enough to preserve enough inliers within $\mathbf{Y}_t$ to provide the same or better percentage of inliers than $\mathbf{C}_t$ contains, mainly filtering out the most flagrant outliers from $\mathbf{C}_t$.

Figure 3.4 shows an example of the filter applied to the $\mathbf{C}_t$ of Figure 3.3(b), with $\mu = 60\%$ (see Algorithm 2). In comparing $\mathbf{C}_t$ (of Figure 3.4(a)) to $\mathbf{Y}_t$ (of Figure 3.4(b)), the most flagrant outliers as well as many of the less significant outliers of $\mathbf{C}_t$ are not included in the set $\mathbf{Y}_t$. In addition, 6.3% of the correspondences in $\mathbf{C}_t$ are outliers, whereas 5.6% are outliers in $\mathbf{Y}_t$. However, this also means that 57% of the inliers were filtered out as well. Therefore, we need an additional filter that can preserve more inliers while disregarding as many or more outliers, and the homography RANSAC filter can do this while benefitting from $\mathbf{Y}_t$—mainly in the average convergence time required.



(a) The original feature correspondence set $\mathbf{C}_t$

(b) The homography filtered correspondence set $\mathbf{Y}_t$

Figure 3.4: Homography filtered correspondence set after being applied to the $\mathbf{C}_t$ of Figure 3.3(b), which (a) is a copy of, with $\mu = 60\%$. The blue lines of (b) show $\mathbf{Y}_t$, the filtered set of the initial correspondences $\mathbf{C}_t$.

3.3.2 Homography RANSAC Filter

Assuming that $\mathbf{Y}_t$ will most probably still be cluttered with outliers, we need a filter that can be robust to the common noise and outliers within $\mathbf{Y}_t$ as well as $\mathbf{C}_t$. Because the homography filter can optionally be pre-applied, the homography RANSAC filter's initial correspondence set $\mathbf{Z}_t$ may be initialized to either $\mathbf{Y}_t$ or $\mathbf{C}_t$. RANSAC can provide us with the needed robustness and still identify a relevant spatial relationship among the correspondence inliers within $\mathbf{Z}_t$ based on the homography and our assumption that the captured scene is relatively near planar with respect to the pose of the camera. This homography RANSAC filter can also lessen the amount of frames dropped due to too few remaining correspondence inliers (Section 3.3.2).

If we had the ideal $\mathbf{H}$ or the ideal $\mathbf{Z}'_t$ then we could use it to easily define the other, respectively. This then becomes a kind of "chicken-and-the-egg" problem—we are trying to define a set of inliers within $\mathbf{Z}_t$, $\mathbf{Z}'_t$, so that we can use $\mathbf{Z}'_t$ to best estimate the spatial relationship, or $\mathbf{H}$, between two adjacent frames; however, we are wanting to use $\mathbf{H}$ to help us define the set $\mathbf{Z}'_t$.

In cases like this, one popular and effective algorithm that can be used is RANSAC [3, 14]. RANSAC essentially estimates $\mathbf{H}_i$ using a small random subset of $\mathbf{Z}_t$, $\mathbf{Z}''_i$, and then computes the consensus set $\mathbf{Z}'_i$ by evaluating how many elements in $\mathbf{C}_t$ are consistent with $\mathbf{H}_i$. It continues this until some termination criteria is met, *e.g.*, the size of $\mathbf{Z}'_i$ is large enough. Upon termination, RANSAC will produce the best homography $\mathbf{H}'_t$ computed until termination that describes the largest consensus set. $\mathbf{Z}'_i$ can then be considered an approximate best correspondence set $\hat{\mathbf{C}}_t$—most likely containing fewer outliers than $\mathbf{Z}_t$. $\hat{\mathbf{C}}_t$ can then be used to compute a best homography estimate $\hat{\mathbf{H}}_t$ that is less sensitive to discretization and a better representation of $\hat{\mathbf{C}}_t$ than $\mathbf{H}'_t$ would be.

RANSAC works quite well in cases like ours because it can successfully identify a best subset of inliers $\hat{\mathbf{C}}_t$ within $\mathbf{Z}_t$ that share a relationship that can be explained

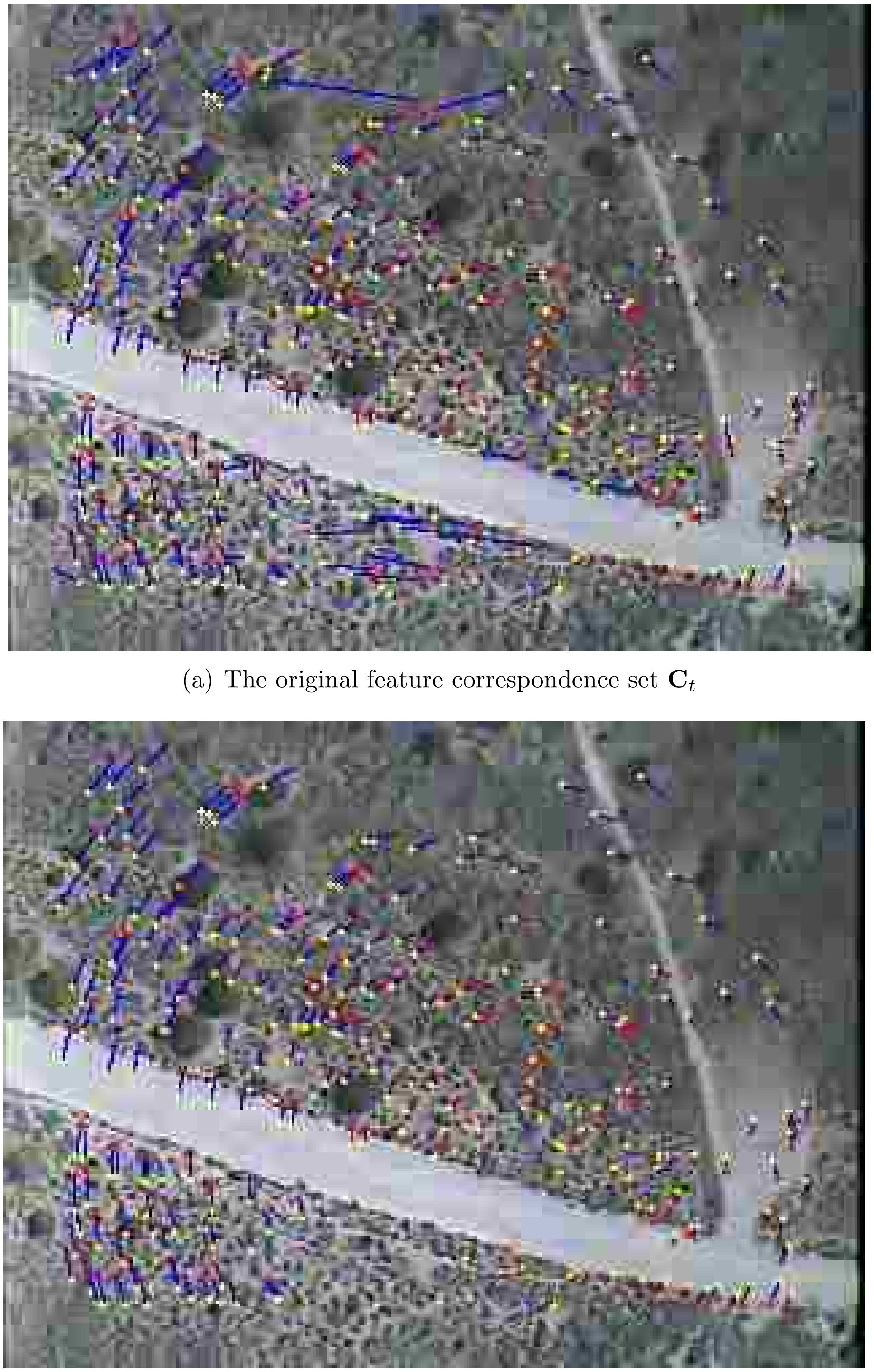
```
1. Initialize values $\hat{c} = 0$, $i = 0$, $\hat{\mathbf{H}}_t = 0$, and $\hat{\mathbf{C}}_t = 0$.
2. Do until $\mathbf{Z}'_i$ is large enough or $i > maxIterations$:
    (a) $\mathbf{Z}'_i = \emptyset$, $\mathbf{Z}''_i = \emptyset$, $c' = 0$.
    (b) Randomly insert $n$ elements from $\mathbf{Z}_t$ into $\mathbf{Z}''_i$.
    (c) Using an overconstrained system, estimate $\mathbf{H}_i$ using $\mathbf{Z}''_i$.
    (d) If each element of $\mathbf{Z}''_i$ is described by $\mathbf{H}_i$ within $\xi$ distance (short-circuit):
        i. Add each element of $\mathbf{C}_t$ into $\mathbf{Z}'_i$ that is described by $\mathbf{H}_i$ within $\varepsilon$ distance, $c = c + 1$.
        ii. If $c' < \hat{c}$, then $\mathbf{H}'_t = \mathbf{H}_i$ and $\hat{c} = c'$.
    (e) $i = i + 1$.
3. Add each element of $\mathbf{C}_t$ into $\hat{\mathbf{C}}_t$ that is described by $\mathbf{H}'_t$ within $\epsilon$ distance.
4. Using an overconstrained system, estimate $\hat{\mathbf{H}}_t$ using $\hat{\mathbf{C}}_t$.

This implies $\mathbf{Z}_t \subseteq \mathbf{C}_t$, $\mathbf{Z}'_i \subseteq \mathbf{Z}_t$, $\hat{\mathbf{C}}_t \subseteq \mathbf{C}_t$, and $\hat{\mathbf{C}}_t \nsubseteq \mathbf{Z}_t$.
```

Algorithm 3: The homography RANSAC filter algorithm with a short-circuit step.

by a homography, $\mathbf{H}$—this of course assumes that the minimum subset of inliers do exist within $\mathbf{Z}_t$ and that enough iterations are performed by RANSAC to identify such a subset. RANSAC is also effective because it is very robust to outliers; only the percentage of outliers in $\mathbf{Z}_t$ and not the magnitude of the errors of these outliers will have a negative influence on the performance of RANSAC. Specifically, we have implemented our homography RANSAC filter as outlined in Algorithm 3.

We made n greater than the minimum four feature correspondences required to directly compute $\mathbf{H}$ for two reasons. First, this serves as a short-circuit condition (see Algorithm 3 Step 2d) for each $\mathbf{H}_i$. This step can help improve the speed of the algorithm because if at least one correspondence that was used to estimate $\mathbf{H}_i$ cannot be explained well by $\mathbf{H}_i$, then it signals that $\mathbf{H}_i$ is contaminated by at least one outlier and allows us to short-circuit the algorithm. Of course, this short circuit step assumes that n is chosen so that it is faster to compute the least squares minimization solution of $\mathbf{H}_i$ using n correspondences and evaluate the integrity of those n correspondences than it is to compute the integrity of $\mathbf{H}_i$ using all of the elements within $\mathbf{Z}_t$ based



(a) The original feature correspondence set $\mathbf{C}_t$

(b) The correspondence set $\hat{\mathbf{C}}_t$

Figure 3.5: This is an example of a homography RANSAC filtered correspondence set $\hat{\mathbf{C}}_t$ after being applied to the filtered correspondences $\mathbf{Y}_t$ of Figure 3.4(b) and the initial correspondences $\mathbf{C}_t$ of Figure 3.3(b), of which (a) is a copy. The blue lines of (b) show $\hat{\mathbf{C}}_t$.

on the exact solution of $\mathbf{H}_i$ solved using the minimum $n = 4$. Second, using $n > 4$ helps to average out the negative aliasing effects caused by the discretization of the domain that inherently exist in the feature locations within $\mathbf{C}_t$. This makes each $\mathbf{H}_i$ a better representation of the true image function with which to better compute the consensus set $\mathbf{Z}_i'$.

It is interesting to note the relationships and implications among the chosen threshold values of ξ, ε, and ϵ. The smaller ξ is, the more strict $\mathbf{H}_i$ has to describe each element within $\mathbf{Z}_i''$ in order to pass this short-circuit condition. Similarly, the smaller ε and ϵ are, the more strict $\mathbf{H}_i$ has to describe each element allowed into $\mathbf{Z}_i'$ and $\hat{\mathbf{C}}_t$, respectively. We prefer to have ξ be a stricter error distance than ε and ϵ because it is more probable to get a very agreeable small $\mathbf{Z}_i''$ subset of $\mathbf{C}_t$ than it will be to get similarly agreeable larger $\mathbf{Z}_i'$ and $\hat{\mathbf{C}}_t$ subsets of $\mathbf{C}_t$, respectively. On the other hand, the larger we allow the values of ξ, ε, and ϵ to be, the larger the acceptable errors will be. This implies that each of our assumptions of a non-planar surface, discretized domain, as well as an imperfectly calibrated image will collectively be more acceptable.

Figure 3.5 shows an example of how effective the homography RANSAC filter can be. In this case, using $\xi = 2$, $\varepsilon = 6$, and $\epsilon = 6$ pixel distances, the homography RANSAC filter disregards 90% of the original outliers and preserves 89% of the original inliers, resulting in more than 99% of the feature correspondences in $\hat{\mathbf{C}}_t$ as inliers and less than 1% as minor outliers.

Determining Inadequate Frame Registrations

Depending on the chosen value of $maxIteration$, if RANSAC does happen to meet the $maxIteration$ termination criteria, then it may be possible to assume that $\mathbf{C}_t$ does not contain a sufficient percentage of inliers given the termination criteria. Another case that may lead to an invalid tag is an insufficient number of elements in $\hat{\mathbf{C}}_t$. Both cases with the appropriate requirements can imply that either $\mathbf{C}_t$ or $\hat{\mathbf{C}}_t$ is insufficient to describe the spatial relationship between I_t and I_{t-1}; therefore, I_t can be tagged and handled appropriately as an invalid frame (Section 3.9.2).

3.4 Estimating the Spatial Relationships

Once we have $\hat{\mathbf{C}}_t$ we can estimate the spatial relationship between I_t and I_{t-1}. To do this, we have chosen to use the Euclidean transformation rather than $\hat{\mathbf{H}}_t$ for reasons explained in Section 3.4.1. To compute the Euclidean transformation, we first estimate the translation $\mathbf{t}$ between I_t and I_{t-1} (Section 3.4.2), and then we estimate the rotation $\mathbf{R}$ needed to more closely align I_t to I_{t-1} given $\mathbf{t}$ (Section 3.4.3). In Section 3.4.4 we describe the process of combining $\mathbf{t}$ and $\mathbf{R}$ into the Euclidean transformation matrix $\mathbf{Q}$, which will be used to create the three different presentation views.

3.4.1 Motivations for Using the Euclidean Transformation

After having computed $\mathbf{H}$, we currently do not use it in the generation of our presentation views in this work for a few reasons. One reason is that we do not want to distort the images coming from the mUAV. Because $\mathbf{H}$ is a perspective projection, using $\mathbf{H}$ alone to align images will quickly distort a sequence of image aggregations and degrade the presentation given a common sequence of mUAV motions like significant changes in the mUAV's roll α, large gradual changes in the mUAV's altitude z, or no changes in z but a rapid change in height-above-ground caused by the mUAV flying over steep terrain.

Another reason is that by assuming the camera is fixed on a fast forward-velocity vehicle, alignment using a homography seems to be a bit of overkill due to the fact that the viewing frustum moves over the scene so quickly that the slightly better alignment a homography can buy us does not empirically show a noticeable increase in the detectability of the presentations.

Third, our experience in using the homography to align and aggregate the images together without using bundle adjustments shows evidence that there is a rapid accumulation of small errors that can quickly have a dramatic negative impact on the aggregate image. The homography can be used effectively if these cascading errors are first addressed. This is discussed in more detail in Section 5.2.4.

Instead of using $\mathbf{H}$ to register images together, we use a rigid body or Euclidean transformation for several reasons. First, we postulate that compensating for rotational γ motions in the mUAV video can provide the user with a better sense of orientation throughout the improved video presentation. Second, as described in Section 2.3.4, we can approximate small motions in β and α as displacements along the x and y axes, respectively. Third, because compensations made in the altitude of the plane or the plane's distance to the objects of the scene z using a scale factor s may introduce distortions in the video presentation—which could decrease a user's ability to detect objects of interest in the video, *i.e.*, decrease the detectability of the presentation—we do not address compensations in z in this work.

Instead, we preserve the original size and aspect ratio of each image in the video presentation so as to not introduce misleading artifacts by distorting the images. Thus, similar to [9], the model we will use to estimate the spatial relationships among adjacent images of a 3-D scene is thus simplified and will directly compensate only for motions detected in (x, y, γ), which can be described by a Euclidean transformation, $\mathbf{Q}$ (Section 2.4).

We will be spatially aligning adjacent frames together by estimating a $\mathbf{Q}_t$ that will transform I_t with respect to I_{t-1}, such that the features within $\hat{\mathbf{C}}_t$ that are also within $\mathbf{P}_t$ align as closely as possible to their corresponding features within $\mathbf{P}'_t$.

In order to estimate $\mathbf{Q}_t$, we need to first compute the 2-D translation $\mathbf{t}_t$ (Section 3.4.2) and the 2-D rotation $\mathbf{R}_t$ (Section 3.4.3) relationships among their respective $\hat{\mathbf{C}}_t$. We map point $\mathbf{p}$ to point $\mathbf{p}'$ by rotation $\mathbf{R}_t$ and translation $\mathbf{t}_t$:

$$\mathbf{p}' = \mathbf{R}_t\mathbf{p} + \mathbf{t}_t \tag{3.2}$$

If we define the transformation $\mathbf{Q}_t$ in terms of $\mathbf{R}_t$ and $\mathbf{t}_t$,

$$\mathbf{Q}_t = [\mathbf{R}_t|\mathbf{t}_t] \tag{3.3}$$

this allows us to simplify the mapping to the simple transformation $\mathbf{Q}_t$:

$$\mathbf{p}' = \mathbf{Q}_t\mathbf{p} \tag{3.4}$$

3.4.2 Estimating the Global Translation

The relative global translation $\mathbf{t}'_t$ of features in I_{t-1} to I_t can be easily defined as the average of the corresponding feature motion vectors, or the differences among the matching feature correspondence points, in $\hat{\mathbf{C}}_t$ from I_{t-1} to I_t:

$$\mathbf{t}'_t = \frac{1}{N} \sum_{\{\mathbf{p}_t, \mathbf{p}_{t-1}\} \in \hat{\mathbf{C}}_t} (\mathbf{p}_t - \mathbf{p}_{t-1}) \tag{3.5}$$

The translation needed to align I_t to I_{t-1} is then

$$\mathbf{t}_t = -\mathbf{t}'_t \tag{3.6}$$

$\mathbf{t}$ then will compensate for the x and y translation motions in our 3-DOF model, (x, y, γ).

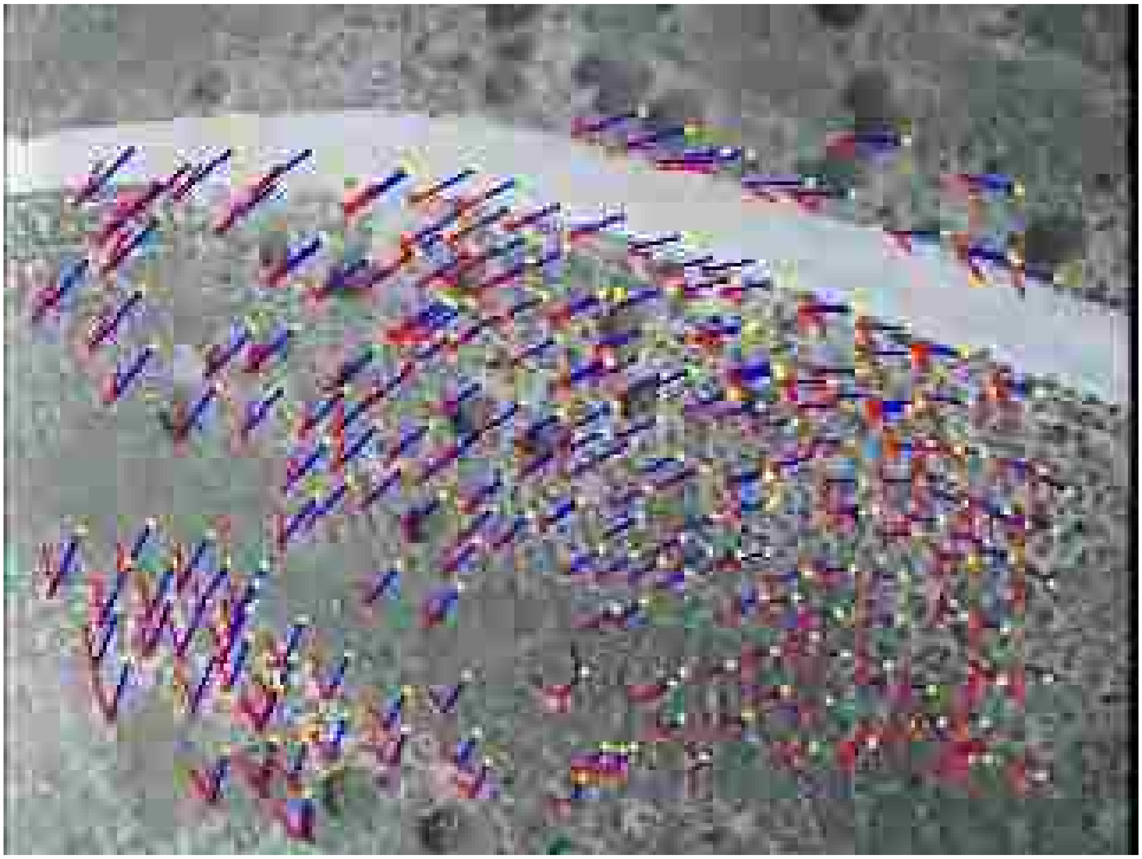


Figure 3.6: This is an example of the set of rotational vectors $\mathbf{V}_t$. The blue lines are the correspondences within $\hat{\mathbf{C}}_t$ and the red lines are the vectors within the set $\mathbf{V}_t$. Note how the red lines are all circling about the same general arbitrary point within the I_t.

To visualize this, in a most simple case, there would be only one correspondence element in $\hat{\mathbf{C}}_t$ whose value would be {(1,1),(2,2)}. This would mean that a feature in I_{t-1} at pixel location (2,2) moved to the corresponding pixel location (1,1) in I_t. According to Equation 3.5, the average feature motion vector is defined by $\mathbf{t}' = (-1, -1)$, and the translation needed to be applied to I_t to align the feature of I_t to its corresponding feature in I_{t-1} would be $\mathbf{t}_t = (1, 1)$.

However, in the usual case where $\hat{\mathbf{C}}$ has many elements, translating I_t by the negative average of the feature motion vectors with respect to I_{t-1} will minimize the sum of the magnitudes of the residual vectors $\mathbf{V}_t$,

$$\mathbf{V}_t = \{\mathbf{p}_{t-1} - (\mathbf{p}_t + \mathbf{t}_t) : \{\mathbf{p}_t, \mathbf{p}_{t-1}\} \in \hat{\mathbf{C}}_t\} \tag{3.7}$$

with respect to translational motion compensation, as shown as red lines in Figure 3.6. However, many, if not all, of the features in I_t will still not be well aligned to their matching features in I_{t-1}. This is mostly due to small changes in the pose of the camera from I_{t-1} to I_t with respect to the rotational γ and can be mostly compensated for by estimating a compensating global rotation angle, θ.

3.4.3 Estimating the Global Rotation

It can be observed in Figure 3.6 that the residual vectors $\mathbf{V}_t$ of adjacent mUAV acquired images after compensating for the translation $\mathbf{t}_t$ have a circular pattern centered about a single point. This center of rotation is commonly located within I_t after being translated by $\mathbf{t}$. We will call this point the center of residual rotation or the point $\mathbf{o}$.

Each I_t's residual vector pattern and the location of their respective center of rotation $\mathbf{o}$ depend on the patterns of the respective motion vectors of I_t that are used to determine the compensating $\mathbf{t}$. It can then be observed that $\mathbf{t}$ essentially determines which single feature in I_t will be approximately aligned to its respective corresponding feature in I_{t-1}, and that the average intersection of the perpendicular bisectors (shown in Figure 3.7) of the residual vectors can approximate the location of this point $\mathbf{o}$, which is not restricted to be a member of $\mathbf{P}_t$ and can end up being any feature within I_t or a non-visible feature outside of I_t.

Given these observations along with the residual vectors of I_t, we can then estimate the location of this point $\mathbf{o}$ as well as the residual vectors' average angle of rotation around the point $\mathbf{o}$.

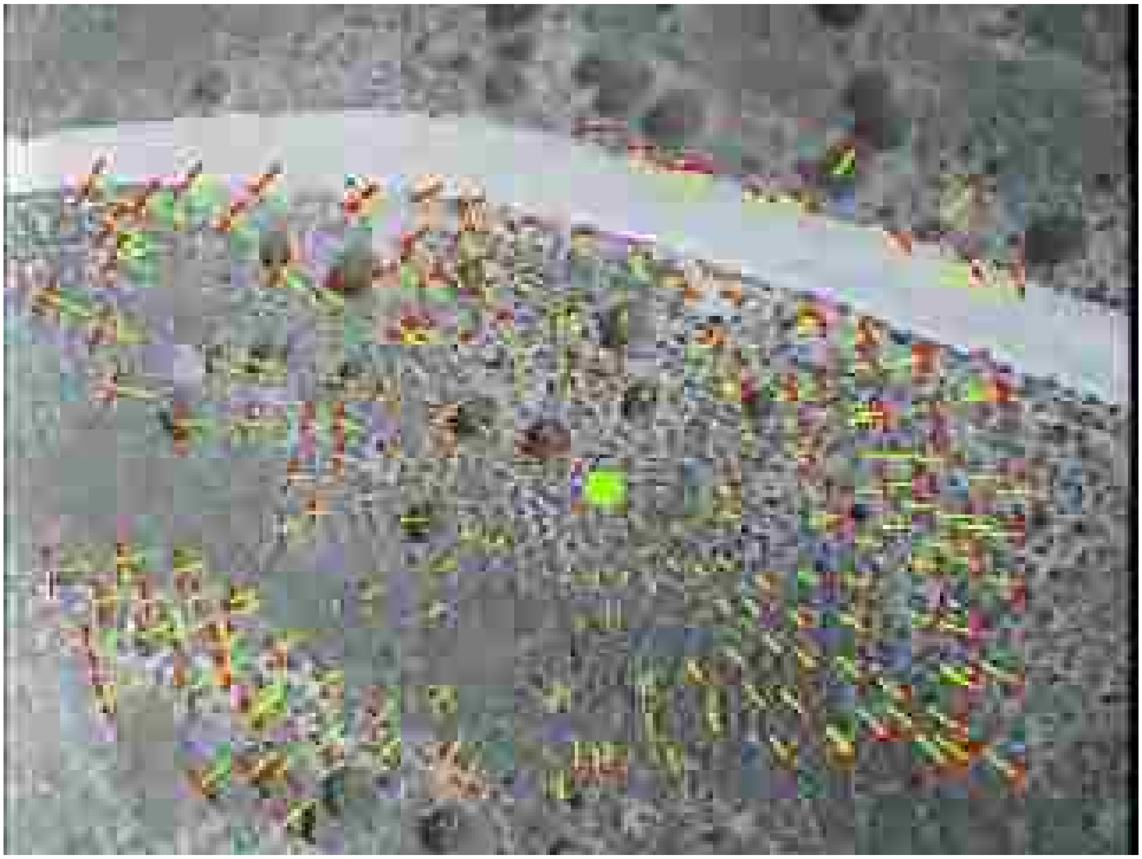


Figure 3.7: Example of the set of perpendicular bisectors of the rotational vectors $\mathbf{V}_t$ that are also shown in Figure 3.6. The red lines are the vectors within the set $\mathbf{V}_t$, and the green lines are their respective perpendicular bisectors. Note how the green lines are all generally pointing to the green spot, which is the estimated center of rotation $\mathbf{o}$ of the residual vectors of $\mathbf{V}_t$ within I_t.

Estimating the Center of Rotation

There are many ways to estimate the location of the point $\mathbf{o}$. One is to set up a least squares minimization problem to minimize the distances among the intersections between each residual vector perpendicular bisector with all of the other residual vectors' perpendicular bisectors [25]. We set the point $\mathbf{o}$ to the average location of the intersections of each residual vector's perpendicular bisector with all or a subset of all of the other residual vectors' perpendicular bisectors, respectively.

Estimating θ

Once the location $\mathbf{o}$ has been estimated, we can now estimate the average angle θ' of the angles made by the $\mathbf{o}$, the mid-point of each residual vector, and each residual vector's corresponding $\mathbf{p}_t$. $2\theta'$ and $\mathbf{o}$ then estimate the rotational difference left between the features of I_{t-1} and I_t. In order to compensate for this residual rotation and more closely align the feature within I_t to their corresponding features within I_{t-1}, we need to compute θ, which is simply $\theta = -2\theta'$. For example, the compensating rotation angle θ to align I_t in Figure 3.7 to its respective I_{t-1} is $-4.38°$ about the point $\mathbf{o}$.

3.4.4 Estimating the Euclidean Transformation

With the estimations of the relative $\mathbf{t}$, $\mathbf{o}$, and θ, $\mathbf{Q}_t$ can now be constructed and I_t can now be registered to I_{t-1}. The key to understanding the construction of $\mathbf{Q}_t$ is that the $\mathbf{o}$ feature of I_t is the only feature in I_t that can be registered to I_{t-1} using $\mathbf{t}_t$

after undergoing a rotation about the feature point $\mathbf{o}$.

$$\mathbf{p}' = \mathbf{T_t T_o R_\theta T_{-o} T_{-t} T_t p} \tag{3.8}$$

$$\mathbf{p}' = \mathbf{T_t T_o R_\theta T_{-o} p} \tag{3.9}$$

$$\mathbf{p}' = \mathbf{Qp} \tag{3.10}$$

where $\mathbf{T_t}$ and $\mathbf{T_o}$ are the translation matrices of $\mathbf{t}$ and $\mathbf{o}$, respectively, and $\mathbf{R}_\theta$ is the 2-D rotation matrix of angle θ.

$$\mathbf{Q} = \begin{bmatrix} \cos\theta & -\sin\theta & \mathbf{o}_x + \mathbf{t}_x - \mathbf{o}_x\cos\theta + \mathbf{o}_y\sin\theta \\ \sin\theta & \cos\theta & \mathbf{o}_y + \mathbf{t}_y - \mathbf{o}_x\sin\theta - \mathbf{o}_y\cos\theta \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Now that we have registered I_t to I_{t-1} using $\mathbf{Q}$, we can now use $\mathbf{Q}$ to create our three separate presentation views.

3.5 The Presentation Views

As mentioned in Section 1.2, since mUAV's travel very quickly, (1) objects within the mUAV-acquired video move quickly through the camera's viewing frustum making users less able to "look back". Also, mUAVs are relatively small and unstable but highly maneuverable platforms. Because they are unstable, (2) mUAV-acquired video is commonly plagued with distracting jitter causing objects within the video to appear very shaky when within the viewing frustum; and, because mUAVs are highly maneuverable, (3) the viewing frustum may frequently rotate, introducing disorienting motions into the video. These three problems associated with mUAV-acquired video make it very difficult for the human visual system to identify or focus on objects of interest when within the viewing frustum.

In our context, an ideal presentation view of mUAV-acquired video would be to present video to users in such a way that would provide a full-resolution global birds-eye-view mosaic of the information captured onto an infinitely large canvas. The alignment and aggregation of each image onto this mosaic would also completely remove the jitter as well as the disorienting rotations of the view from the presentation of the content within the video, *i.e.*, the captured information of the scene.

However, our canvas size and viewing resolution are limited by our screen size, our computations are bound to perform in a real-time mobile environment, and our computational resources do not scale well to an infinitely large canvas.

So, in order to improve the presentation of mUAV video so that users have a greater probability of visually identifying these objects of interest in our context, we devised three different but related presentations. The simplest and first logical building block of the three presentation views is the E-mosaic view. The E-mosaic view addresses all three of these problems but still suffers from some jitter, the stable-E view addresses improving orientation and removing the jitter, and the stable-E-mosaic view addresses all three of these problems by combining the strengths of both the E-mosaic and stable-E views.

3.6 Creating the E-mosaic Presentation

The process of building a mosaic using temporally and spatially adjacent images involves aligning and aggregating these images together onto a common viewing canvas. In the context of improving the detectability of interesting objects within mUAV-acquired video, as previously discussed in Section 2.3.2, we will only address building a local mosaic so as to avoid the costly bundle adjustments required to build a global mosaic. Using a local mosaic will also allow us to forget the areas of the mosaic that are not currently in view. Also, due to the fast-paced real-time presentation of the aggregation of images acquired using a fast forward-velocity platform, we will not concern ourselves with aligning the images in finer detail than what our estimations of **Q** will afford us.

3.6.1 E-mosaic Motivations

The first step in building any mosaic presentation view is to first expand the viewing size of the canvas, kind of like stepping back from the video to get a larger picture of what is being seen. Then, we can proceed to aggregate images together onto the canvas to build and display a local mosaic that will increase a user's spatial view as well as the user's understanding of what is being and has been seen.

Two approaches to building and viewing this E-mosaic view are to fix the viewpoint relative to the canvas (*i.e.*, fixed viewpoint-canvas E-mosaic view), or to allow the viewpoint to move freely over the canvas but remain fixed above the current image being aggregated onto the canvas (*i.e.*, fixed viewpoint-image E-mosaic view). Using the fixed viewpoint-canvas approach will completely stabilize the sequence of images; however, in the context of fast forward-velocity capture platforms, the aggregation of each image onto the canvas also very quickly moves out of view. On the other hand, using the fixed viewpoint-image approach both always keeps the current image centered within the view and causes the E-mosaic to appear to grow away from the current image. This approach provides a local history of what has been seen, but it does not provide a stabilized presentation of the video's sequence of images.

So the problem now becomes one of trying to keep the viewpoint as fixed as possible over the canvas of image alignments and aggregations in order to remove as much of the jitter from the presentation of the video as possible while still allowing the viewpoint to follow the image aggregation path, *i.e.*, the placement of the current image onto the canvas. This leads to a logical compromise of moving the viewpoint only when the current image is being aggregated out of the view in such a way that all of the current image and as much of the mosaic remains within the view. This is equivalent to panning or translating the mosaic enough to keep the current image within view only when the placement of the current image would be otherwise out of view.

This method maximizes the time that objects remain within the user's view, *i.e.*, maximizes the persistence of the video's content, and also greatly reduces this presentation view's rotation and improves user orientation (see Section 4.3.4).

3.6.2 E-mosaic View Methods

Once we have a good estimate of $\mathbf{Q}$, it can be used to create the E-mosaic presentation view. To do this, we currently treat our canvas as an infinite plane. The first image of the video I_0 is placed onto and centered on the canvas's origin. Each successive image I_t is then aggregated or copied onto the canvas with respect to its spatial relationship to I_0, as described by the cumulative Euclidean transform $\mathbf{Q}'_t$, which is initially $\mathbf{Q}'_1 = \mathbf{Q}_1$ with each successive $\mathbf{Q}'_t = \mathbf{Q}_t\mathbf{Q}'_{t-1}$.

To keep I_t always within the view, we compare the bounding box of I_t to see if any of its corners are outside of the view. If they are, then we compute the respective horizontal x'_t and vertical y'_t disparities that can be used to then translate the viewpoint with respect to the canvas by x'_t and y'_t, essentially appearing to translate the E-mosaic by $-x'_t$ and $-y'_t$ so that I_t correctly aggregates onto the canvas and remains wholly in view.

In practice, because our viewpoint is only allowed x and y translational freedom a constant distance from the canvas, this can translate into keeping an image that is the size of the view image I'. I_0 is then copied onto the center of I', which is then displayed to the user. The content of each successive I' is then translated by $-x'_t$ and $-y'_t$, any of which can be 0. I_t is then copied onto I' with respect to its spatial

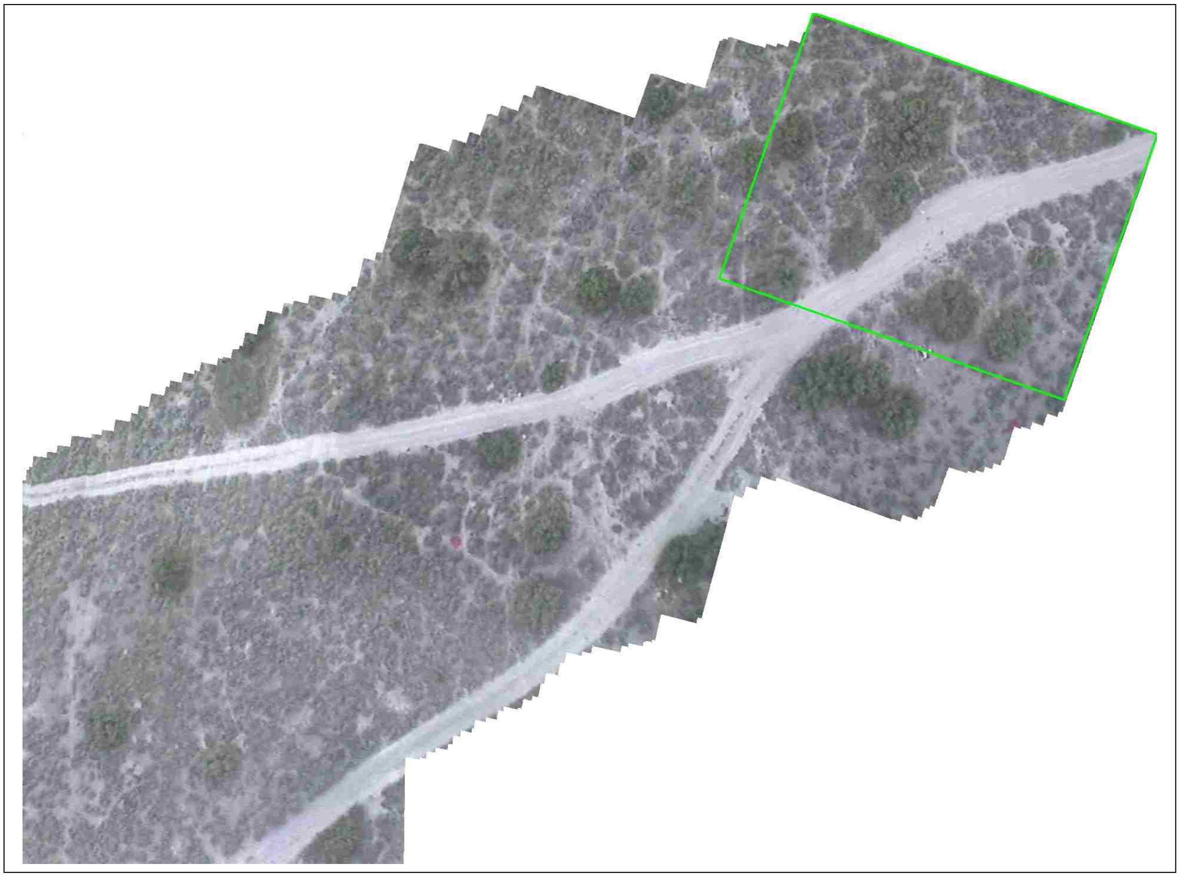


Figure 3.8: Example of the E-mosaic view presentation with a view three times the size of the original capture frame size. Compare to Figures 3.10 and 3.11.

relationship to I_0 as described by $\mathbf{M}_t = \mathbf{Q}'_t\mathbf{T}''_t$, where

$$\mathbf{T}''_t = \begin{bmatrix} 1 & 0 & -x'_t \\ 0 & 1 & -y'_t \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

An example of the resulting E-mosaic view is shown in Figure 3.8.

3.6.3 E-mosaic View's Strengths and Weaknesses

Although the E-mosaic view does maximize the persistence of information in the presentation of mUAV-acquired video as well as compensate for much of the associated jitter and rotational γ problems, it still suffers from a few shortcomings—imperfect compensations for motions in γ, introduced artifacts in the history area of the mosaic, and residual jitters in the view translations or the panning motions of the mosaic.

This E-mosaic view does compensate for much of the disorienting motions in rotation γ, but it does not remove them completely from the presentation due to a slow accumulation of error. Since performing costly bundle adjustments is not yet possible within our framework as mentioned in 2.3.2, we present other possible solutions to this problem in Chapter 5.

In addition, mosaics in general are susceptible to producing distracting artifacts in the history, *i.e.*, the part of the mosaic that is not within the bounds of the current image. These artifacts can be caused by noise in the video as well as imperfect alignments in the aggregate image and can cause possible false positives in the presentation. We discuss this more in Section 4.2.6.

In the context of fast forward-velocity mUAVs, it is the common case that the mosaic will constantly be panning in order to always keep the current frame within the view. This case commonly regresses the presentation of the video back to a jittery presentation, albeit one with a reduced amount of jitter. The E-mosaic view will always reduce the jitter in at least three of the eight possible 2-D translational directions—worst case being image aggregations beyond a corner of the view. For example, in the case of the images being aggregated to the upper right-hand corner outside of the current view, the NW, N, NE, E, and SE jitters would remain (N, W, S, and E being synonymous to the $+y$, $+x$, $-y$, and $-x$ directions of the view, respectively), but the S, SW, and W jitters would have no effect. This is the state of the E-mosaic presentation shown in Figure 3.8. However, even though the jitter is still reduced, the remaining jitter affects a greater amount of viewable content than the original view which can make it just as distracting as the original jitter was.

Believing that this remaining jitter could degrade the detectability of interesting objects within the E-mosaic view as well as cause significant visual and attentive fatigue on the user, we pursue a stabilization technique in Section 3.7 to address both the jitter as well as the rotational γ compensations and then combine this stabilization technique with the E-mosaic view in Section 3.8.

3.7 Creating the Stable-E Presentation

This local E-mosaic presentation comes closer to the optimal solution to stabilizing mUAV-acquired video; however, the stabilization problem again arises when this breakdown at the borders of the view occurs, *i.e.*, when the images begin to aggregate onto the canvas outside of the view and the jitters are again reintroduced. Our stabilization problem now becomes one of stabilizing not the alignment of the images but rather the viewing path used to follow the image aggregation path.

We hypothesize that providing a user with a stable-E view may increase the user's ability to detect and focus on objects within view as well as increase the user's attentive endurance to the video while decreasing possible fatigue. We also suspect that the general orientation of the user will be increased. We discuss the actual results in Chapter 4.

3.7.1 Computing the Smoothed View Path

Since the alignment of each I_t to I_{t-1} within a sequence of images effectually stabilizes the presentation of the content within the sequence, we want to preserve this stable content behavior as much as we can while preventing I_t from ever being aggregated outside of the view. We will call the balancing of these processes *stabilizing the view.*

Because the history of $\mathbf{Q}'$ from $\mathbf{Q}'_0$ to $\mathbf{Q}'_t$, $[\mathbf{Q}'_0, \mathbf{Q}'_t]$, essentially describes the image aggregation path, and because our view needs to follow this path in a smooth fashion, we can use this history to create our smoothed view path. This can be done by fitting a smoothed curve, B_t to an n-sized history of $\mathbf{Q}'$, $[\mathbf{Q}'_{t-n}, \mathbf{Q}'_t]$. This curve may contain k number of control points, where $k \leq n$. If the center point of each I_t is $\mathbf{c}_t$, then its transformed center point is $\mathbf{c}'_t = \mathbf{Q}'_t \mathbf{c}_t$. Each control point b_i of B_t can then be defined by a corresponding progression of $\mathbf{c}'$s:

$$b_i = \mathbf{c}'_j, \text{ where } \ (0 \leq i \leq k) \ \text{ and } \ (t - n \leq j \leq t) \tag{3.13}$$

This implies that B_t's set of b's may consist of either every corresponding $\mathbf{c}'_{t-n}$ to $\mathbf{c}'_{t-n+k}$, or an evenly spaced sparse set of $\mathbf{c}'$'s from $\mathbf{c}'_{t-n}$ to $\mathbf{c}'_t$.

We can now define each smoothed viewpoint $\mathbf{q}_t$ that corresponds to I_t by using B_t. We define $\mathbf{q}_{t,m}$ to be the point on the curve B_t evaluated at m where $m = [0, 1]$. It is important to observe that if $\mathbf{q}_{t,0} = \mathbf{c}'_{t-n}$ and $\mathbf{q}_{t,1} = \mathbf{c}'_t$, then the closer that we allow $\mathbf{q}_{t,m}$ to be to $\mathbf{c}'_t$—*i.e.*, as m approaches the value of 1—the stronger that the placements of each I_t will be forced to the center of I', effectually preserving more of the original jitter in the presentation. On the other hand, the closer we define $\mathbf{q}_{t,m}$ to be to $\mathbf{c}'_{t-n}$ with a large value of n relative to the size of the view, the more the stable-E view will behave like the E-mosaic view, allowing I_t to commonly "hug" the edges of the view which would preserve the E-mosaic view related jitters in the presentation. Therefore, the balance between these two extremes depend on both the chosen value of m as well as the chosen size of n.

In practice, similarly described in Section 3.6.2, I_0 is copied onto the center of I', which is then displayed to the user; however, the content of each I' is cleared after each display. Each successive I_t is then copied onto I' with respect to its stabilized spatial relationship to I_0 as described by $\mathbf{A}_t = \mathbf{Q}'_t \mathbf{T}''_t \mathbf{S}_t$, where $\mathbf{c}''_t = \mathbf{c}'_t - \mathbf{q}_{t,m}$ and

$$\mathbf{S}_t = \begin{bmatrix} 1 & 0 & \mathbf{c}''_t[x] \\ 0 & 1 & \mathbf{c}''_t[y] \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

Since the relationship between $\mathbf{A}_t$ and $\mathbf{M}_t$ is just a translation, this then implies that the stable-E view effectually compensates for the same amount of disorienting motions in γ as does the E-mosaic view.

3.7.2 Understanding the Stable-E Translation

This process stabilizes the content of the video by allowing each I_t to transform in a way that will compensate for enough of the motion detected within the video from I_{t-1} to I_t to stabilize the content while keeping I_t always within the view. This behavior can be observed in Figure 3.9. The progression of $\mathbf{q}_{t,m}$ (the red path) is constantly trailing behind its corresponding progression of $\mathbf{c}'_t$ (the green path) in a much smoother fashion. This trailing distance is influenced both by the separation of the control points used to define B as well as the value of m.

It is also helpful to note that this trailing relationship between the red spot ($\mathbf{q}_{t,m}$) and the green spot ($\mathbf{c}'_t$) is exactly the same as the trailing relationship between the black spot (the center of the view) and the blue spot ($\mathbf{c}''_t$). Essentially, we are forcing the center of the view to always be directly above the red spot, $\mathbf{q}_{t,m}$, which shows the relationship between the green and the blue boxes, *i.e.*, the E-mosaic and the stable-E views, respectively.

This then allows the progression of frames themselves to jitter quite freely back-and-forth in order to compensate for the high frequency motions that caused the objects seen in the video to appear jittery. The content jitter is therefore suppressed, and the presentation is stabilized by transposing each I_t over I_{t-1} in a manner that the objects seen in the video tend to remain in the same general area relative to the view—general enough to allow the view to follow the general motion of the image aggregation path and create a smoothed view path. This is evident in the blue path of Figure 3.9. The jitter seen in this blue path indicates that the frames' motion is

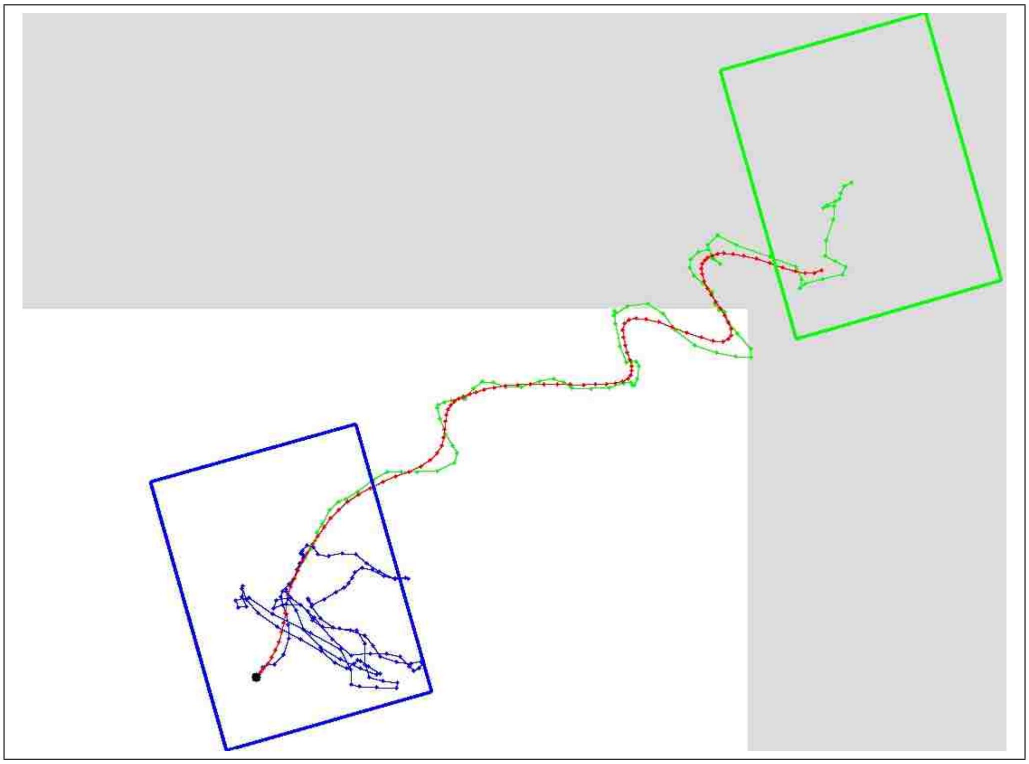


Figure 3.9: Stabilization path. The white area represents the stable-E view area, part of which is out of view for illustrative purposes only. The black spot represents the origin of the view (as well as the origin of the canvas this case). The green represents $\mathbf{c}'_t$, *i.e.*, the cumulative translations of each I_t's center. The green box represents the E-mosaic transform of I_t. The red represents $\mathbf{q}_{t,m}$, *i.e.*, the point on the B evaluated at m. The blue represents $\mathbf{c}''_t$, *i.e.*, the red points subtracted from their corresponding green points. The blue box represents the stable-E transform of I_t.

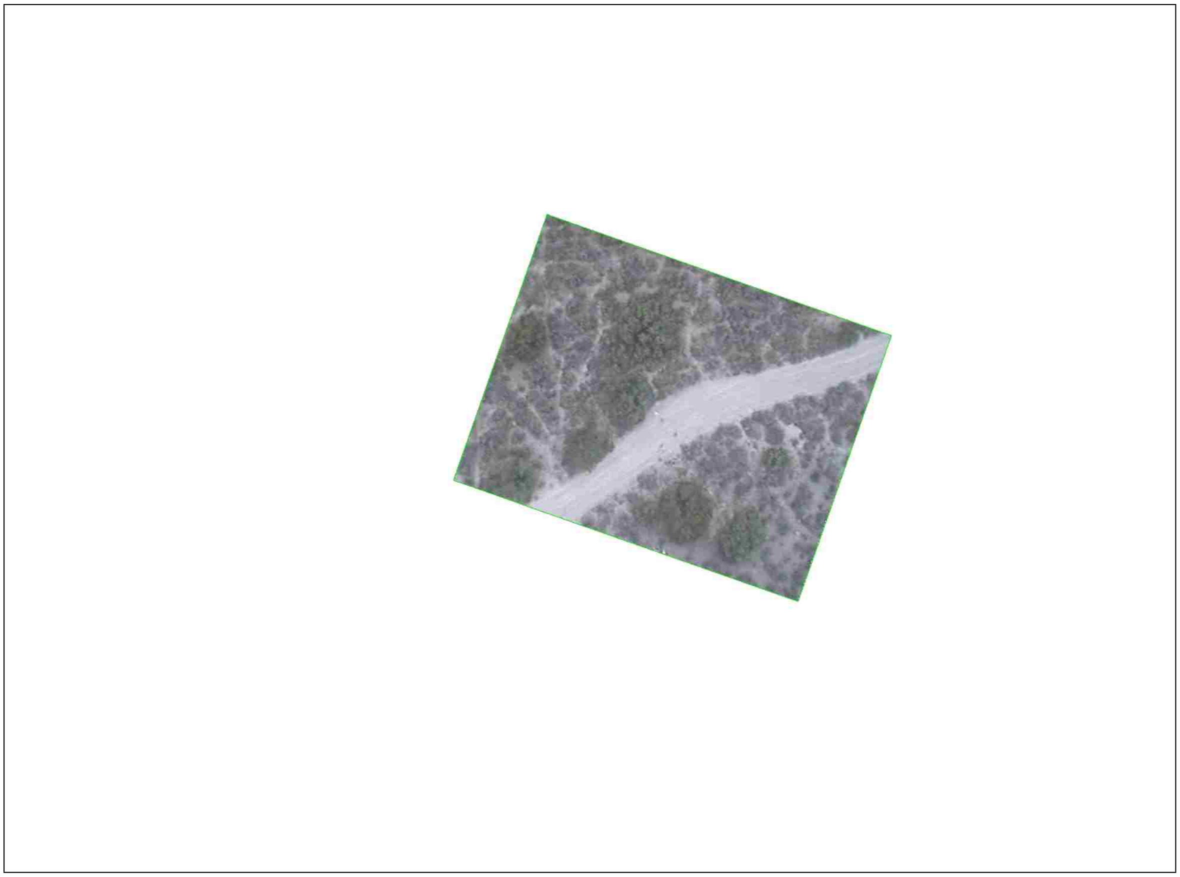


Figure 3.10: Example of the stable-E view presentation with a view three times the size of the original capture frame size. Compare to Figures 3.8 and 3.11.

rapidly moving back-and-forth in order to stabilize the content of the video in the presentation.

In particular, using a view size twice the size of the original frame size (640 × 480), we fit a k-degree Bezier curve, where $k = n = 30$, to the image aggregate path using every $\mathbf{c}'_{t-n}$ to $\mathbf{c}'_t$ as the control points of B_t. Then, we evaluate each $\mathbf{q}_{t,m}$ with $m = 0.5$ using de Casteljau's mid-point algorithm, and copy each I_t onto the canvas after transforming it by $\mathbf{A}_t$. This process is relatively computationally negligible and has produced favorable stable-E view paths based on several samples of mUAV-acquired video. Figure 3.10 shows an example—the same frame shown in Figure 3.8—of what such a stable-E view frame would look like. Note that the frame is rotated to compensate for cumulative rotational γ motions and shifted slightly from the center of the view to compensate for the detected high frequency motions of the video.

3.7.3 Stable-E View's Strengths and Weaknesses

Compared to an E-mosaic view with the same view size, the stable-E view can remove almost all reasonably high-frequency content jitter while at the same time almost completely avoiding the reintroduction of the border hugging jitter particular to the E-mosaic view. In the stable-E view, objects move through the viewing frustum in a smoother and more predictive fashion, one that the human visual system is more adept to following and searching—almost as if the stable-E view path is enhanced with an element of momentum. The stable-E view also removes the exact amount of disorienting motions in γ as does the E-mosaic view.

Another advantage to this stabilization algorithm over all of the other software-based stabilization algorithms in the current literature mentioned in Section 2.3.3 is that just like our E-mosaic presentation, the stable-E presentation does not introduce any lag in the presentation due to a required history of frames. This allows the real-time stabilized presentation of each I_t of the video at time t and without any required ramp-up time.

However, unlike the E-mosaic view, the stable-E view does not build a mosaic and allows objects to again quickly move through the viewing frustum, dramatically and negatively affecting the detectability of the stable-E view—similarly to the original presentation view. In the next section, Section 3.8, we present the combination of the E-mosaic and the stable-E views.

3.8 Creating the Stable-E-mosaic Presentation

The strengths and weaknesses of the E-mosaic and stable-E views are near compliments of one another. Therefore, by combining the E-mosaic view's localized mosaic and the stable-E view's stable following of the image aggregation path into one stable-E-mosaic view, we can benefit from the strengths of both presentations' views combined and thus compliment and eliminate much of their combined weakness. This means that within the stable-E-mosaic view, we can expect objects to both move through this viewing frustum in a smoother and more predictive fashion than the E-mosaic view allows as well as persist longer within the view that contains a larger viewing frustum than the stable-E view provides.

3.8.1 Stable-E-mosaic View Methods

Given information needed to create the E-mosaic and stable-E views, creating the stable-E-mosaic view basically turns into aggregating the images as described in Section 3.6.2 and translating the $\mathbf{q}_{t,m}$ to be directly above $\mathbf{c}''_t$. Doing this gives us the stable-E-mosaic view as shown in Figure 3.11, which is basically the E-mosaic view's Figure 3.8 translated so that the current frame's position in the view is the same as the current frame's position of the stable-E view as shown in Figure 3.10. Note that the current frame is not near the edge of the view as it was in the E-mosaic view's Figure 3.8, which allows it to avoid reintroducing some of the original jitter of the mUAV-acquired video back into the presentation.

In practice, this E-mosaic and stable-E view combination can be accomplished in one of two ways. The first way to render the stable-E-mosaic view is to recognize that the relationship between $\mathbf{A}_t$ and $\mathbf{M}_t$ is just a translation. This allows us to compute the E-mosaic view and then easily translate it by $\mathbf{w}_t$, the difference of their transformed image centers,

$$\mathbf{w}_t = \mathbf{M}_t\mathbf{c}_t - \mathbf{A}_t\mathbf{c}_t \tag{3.15}$$

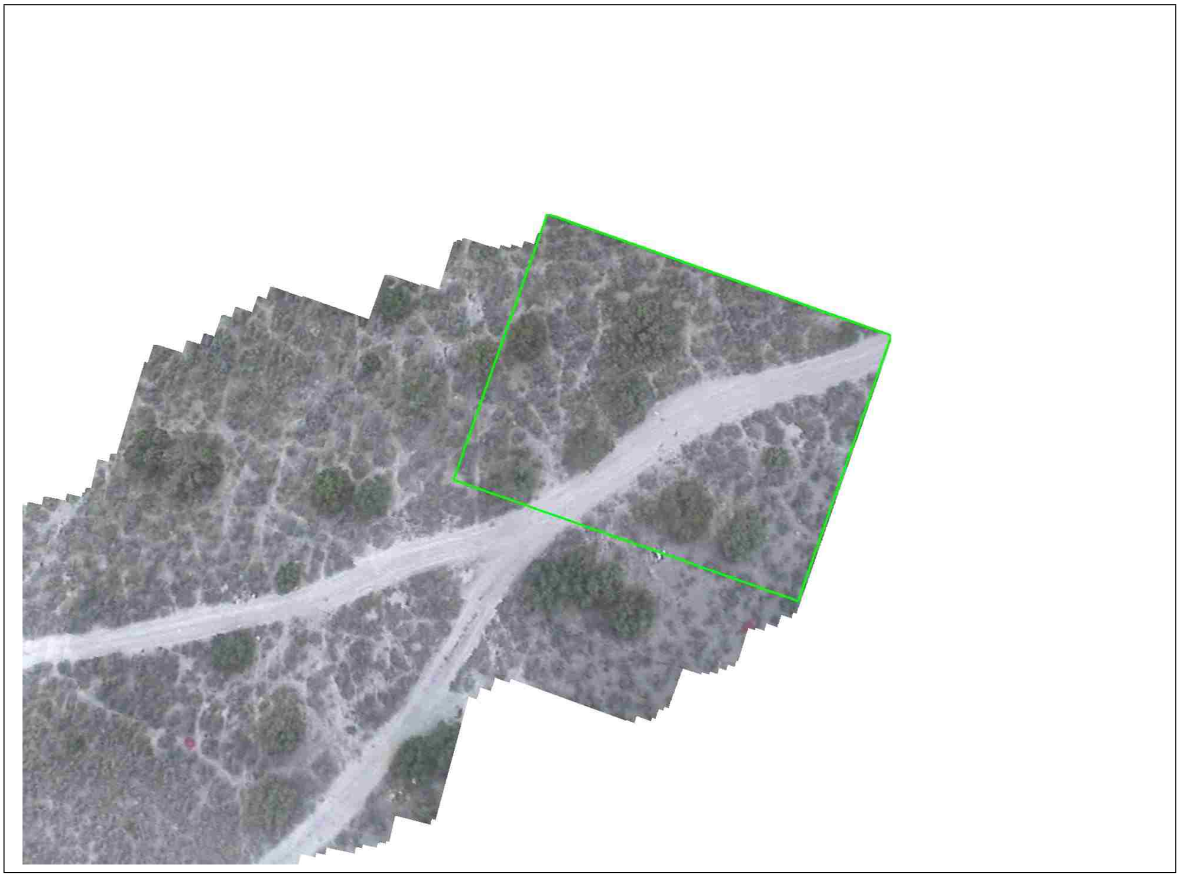


Figure 3.11: Example of the stable-E-mosaic view presentation with a view three times the size of the original capture frame size. Compare to Figures 3.8 and 3.10.

Since we are computing the E-mosaic view image and then translating the E-mosaic view to present it as the stable-E-mosaic view, then the stable-E-mosaic view provides an additional benefit of remembering information as long as that information is still within the E-mosaic view—even after it goes out of the stable-E-mosaic view. This slightly expands the temporal field of view of the stable-E-mosaic presentation.

The second way is to not clear the canvas after every processing of I_t, but rather to first translate the canvas by $\mathbf{z}_t = \mathbf{c}''_{t-1} - \mathbf{c}''_t$, and then to copy I_t appropriately using $\mathbf{A}_t$. This second method is a little faster than the first one described since we do not have to perform an extra image copy; however, it does permanently forget the information of the mosaic, or the history, that moves out of view.

3.8.2 Stable-E-mosaic View's Strengths and Weaknesses

With similar view sizes, just like the stable-E view, the stable-E-mosaic view can also remove almost all reasonably high frequency content jitter while at the same time almost completely avoiding the reintroduction of the border hugging jitter particular to the E-mosaic view. In addition, objects can now move through the viewing frustum in a smoother and more predictive fashion, similar to the stable-E view, but with a larger viewing frustum similar to the E-mosaic view. The stable-E-mosaic view also removes the exact amount of disorienting motions in γ as does the E-mosaic view.

Combining the E-mosaic and stable-E views, however, still suffers from some remaining shortcomings. First, the stable-E-mosaic presents less of a history than the E-mosaic presents, but this may be an acceptable tradeoff for a more stable presentation of the video as discussed further in Chapter 4. Second, just as the E-mosaic and stable-E views, the stable-E-mosaic view suffers from a gradual accumulation of error in $\mathbf{Q}'$, and in particular in γ which may possibly mislead the user to infer an incorrect orientation into the video presentation, discussed more in Section 5.2.3. Third, the E-mosaic and stable-E-mosaic views equally may suffer from noisy, blank, or adjacent frames with too little overlap, which can completely invalidate the estimated $\mathbf{M}_t$ transformation of I_t as it relates to the current mosaic, which we address in Section 3.9.2.

In the next section, Section 3.9, we propose some approaches to address some of the remaining weaknesses of these three presentation views. Also, in Chapter 5, we discuss ways in which future innovations may address them and help significantly reduce their negative effects in using and understanding mUAV-acquired video for searching tasks.

3.9 User Interface

Some of the remaining problems that the E-mosaic, stable-E, and stable-E-mosaic presentation views still suffer from can be addressed in the interface used to present each view to the user. In particular, we can highlight the current frame I_t (Section 3.9.1), communicate to the user a possible invalid transformations or mosaic (Section 3.9.2), as well as decouple the eye-hand coordination required to accurately select detected objects-of-interest throughout the presentation of mUAV-acquired video (Section 3.9.3).

3.9.1 Highlighting the Current Frame

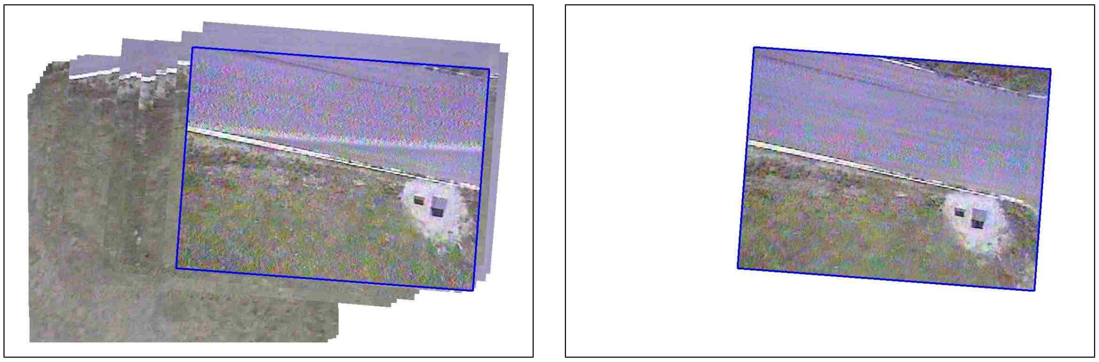
When watching the E-mosaic or stable-E-mosaic views, we have found it helpful to present to the user the location of each I_t as it relates to its corresponding mosaic. The importance in doing this is that all of the information in the mosaic that is outside of I_t (*i.e.*, in the history of the mosaic) can be considered a frozen-in-time representation of information that has been seen. When using mUAV-acquired video to search for interesting objects, the image aggregations can progress very quickly, making it important for the searcher to stay focused on the current information while being provided with a history of what has been seen that allows the searcher the ability to "look back". In order to contrast I_t to the history of the mosaic, we draw a bright green border around each I_t, as shown in Figures 3.8 and 3.10.

3.9.2 Presenting Invalid Frame Registrations

As mentioned previously in Section 3.8.2, the E-mosaic and stable-E-mosaic views may equally suffer from noisy, blank, or adjacent frames with too little overlap which can each completely invalidate the estimated $\mathbf{M}_t$ transformation of I_t as it relates to the current mosaic. These cases can be detected as described in Section 3.3.2 and the user notified of the invalid registration of I_t.

To do this, we draw a blue border around each possibly invalid I_t without applying $\mathbf{Q}_t$ to indicate to the user that I_t may not align well to the mosaic as shown in Figure 3.12. In addition, we keep count of the number of contiguous invalid frames that have been displayed relative to I_t. If this tally surpasses a predetermined threshold ψ, then we reset the history of the mosaic by clearing the canvas as shown in Figure 3.12(b).

Doing this can be helpful in avoiding displaying to the users distracting invalid information in the history of the mosaic in relation to I_t while both continuing to display I_t to the user as well as preserving the positions of both the current aggregation of images and I_t relative to the view. This also ensures that all of the frames will be displayed even though bad frame registrations are detected so that the fall-back worst case scenario is presenting to the user what would be displayed using the original video view—though perhaps at a rotated and translated position within the view. Furthermore, if a valid frame registration is detected before the contiguous invalid frame registration tally exceeds ψ, then I_{t-1} will be in the position of the last good $\mathbf{Q}'$—having been copied directly over the last frame causing the invalid transformation—and I_t will be aggregated correctly using a valid frame registration



(a) Indication of an invalid frame registration

(b) Mosaic being reset

Figure 3.12: Two examples of how we indicate to the user that the current frame was not adequately registered. Figure (a) shows a blue border around the current frame, and (b) shows the following frame resetting the mosaic's history due to too many contiguous invalid frame registrations ($\psi = 5$). Note the high frequency noise that is present in (a) and most likely causing the invalid frame registrations.

with respect to I_{t-1}. This prevents frames that had invalid frame registrations from being accumulated into the history of the mosaic.

Misalignments in the history of the mosaic due to invalid frame registrations depend on the value of ψ. We have used a ψ where $1 < \psi \leq 5$ because users are able to make general compensations for slight misalignments within the history of the mosaic and still benefit from the information presented. However, if there are too many misalignments in a row, then the history of the mosaic can become very distracting in the presentation if it is not reset.

3.9.3 Decoupling the Eye-hand Coordination

[10] mainly uses stabilization to assist users in more accurately and more precisely selecting the positions of objects-of-interest seen within the presentation view in order to provide several position estimates to help improve object localization and better estimate the geo-location of objects-of-interest. His experimental results support that using a stabilized presentation of the video can improve a user's ability to follow an object-of-interest with a mouse-pointer; however, the results also support the

conclusion that even using a stabilized presentation of mUAV-acquired video to select and follow objects-of-interest with a mouse-pointer remains difficult.

In contrast, our presentation views concentrate mainly on increasing the detectability of interesting objects within the presentation of mUAV-acquired video. To adequately measure effects in detectability among the views, we need a system that is more robust to eye-hand coordination variability among users. Since video content is still quite difficult to follow with a mouse-pointer, even if the user is using a stabilized presentation view, we suggest a pause-select control sequence that can increase a user's ability to accurately select the positions of objects-of-interest—further decoupling the eye-hand coordination required to detect and select objects-of-interest.

Also, using a short selection sequence that many users are already very familiar with, like mouse-double-left-clicks, can help increase the selection speed of an interesting object once identified. This may help provide more precise and higher percentage of more accurate position estimates per object-of-interest seen within the video, but may provide fewer estimates than [10] will provide—however, having fewer but more precise and accurate estimates may arguably help improve localization and geo-location estimations.

Therefore, in order to pause the presentation view once an interesting object is suspected to have been detected, the user can left-mouse click once anywhere within the view to freeze the frame. At this point, the user may refine the position of the selection by moving the mouse-pointer over the object-of-interest and mouse-left-click again to select the object-of-interest. However, if it is determined that an interesting object was not detected, the user may right-click to cancel the freeze frame. Once the presentation is in a freeze frame state, either the mouse-left-click to select or the mouse-right-click to cancel will also unfreeze the frame and resume the video presentation from the "live" feed (helpful in a real-time search situation), or from where it was frozen (helpful in an off-line search situation).

Chapter 4

Results

We can currently compute the calibrated spatial relationships among the images of the video using half-height frames at about 60 fps on a Dell Precision 380. However, the main focus of this thesis is to show that presenting a user with a stable-E, E-mosaic, or stable-E-mosaic view of mUAV-acquired video will respectively increasingly improve the user's ability to detect objects of interest seen throughout the video as well as improve the user's sense of orientation and attentiveness throughout the presentation of the mUAV video. In addition, using our proposed user interface, an average computer user can more precisely and more accurately identify (select) these objects of interest.

In order to quantify these improvements, we performed a user study that was administered as explained further in Section 4.1. The resulting relative objective performances among the four different views (Section 4.2) as well as the subjective preferences (Section 4.3) among the test subjects provides some very useful insights into how these three presentation views compare to the original video presentation (or the original view).

We categorize and present a discussion of the results of this user study as objective results in Section 4.2, and as subjective results in Section 4.3.

4.1 User Study Format

We designed this user study to quantify the comparative effectiveness of each presentation view—original, stable-E, E-mosaic, and stable-E-mosaic—at improving users' abilities to detect and identify objects of interest seen within mUAV-acquired video with a secondary visual cognitive load reasonably similar to a real search situation. It is also designed to measure the preferences of each user among these four presentation views [1].

4.1.1 The Sample Population

Experimental sessions were designed to last approximately 60 minutes and were scheduled daily at each subject's convenience, Monday through Saturday. Each subject completed an IRB-approved consent form as well as a pre-study questionnaire (Appendix B.2) that was used to judge the subject's bias.

This user study was performed on 14 näive and 12 biased volunteer subjects. Subject bias was determined based on the subject's familiarity with this work combined with their familiarity with others' preferences of this work's four different presentation views, as shown in Table 4.1. The näive subjects were recruited from the general population and compensated $15.00 for their participation. The biased subjects were recruited among those affiliated or familiar with this work and were not

	Näive		Biased				
This Study	1	2	2	4	4	4	4
Others' Preferences	1	1	2	1	2	3	4
User Count	11	3	3	3	3	1	2
Total Users	14		12				

Table 4.1: The tally of subjects' familiarity with this study combined with their familiarity with others' preferences. Rankings are 1–4, 4 being the most familiar. Those whose respective familiarity combination was a 1-1 or a 2-1 are considered näive subjects, whereas all others are considered biased subjects.

	Novice	Average	Expert
Computers	0	11	15
Search and Rescue	17	9	0
Aerial Search	17	9	0

Table 4.2: Subjects' experience with tasks related to this work

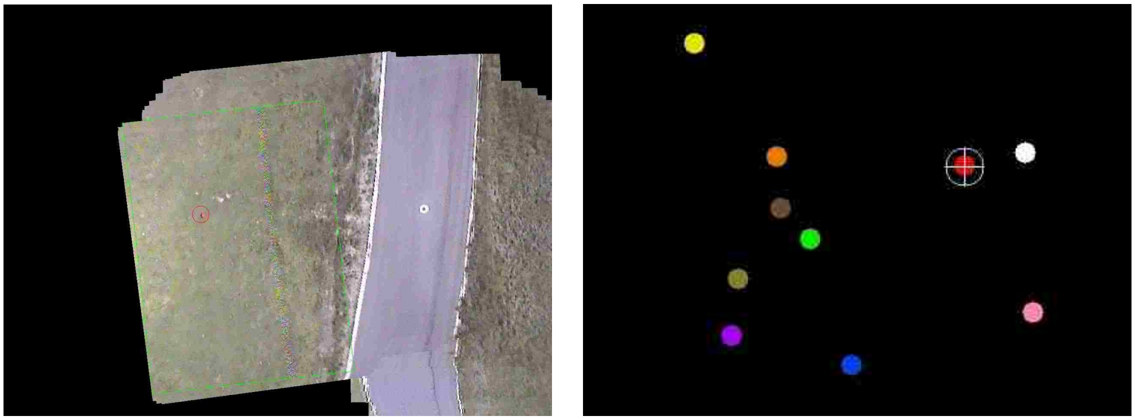
compensated. In addition, our target user population consists of people who have at least an average experience with using a computer but not necessarily much experience with aerial search nor general search and rescue tasks, as indicated in Table 4.2.

Only one subject reported having a physical limitation that may affect the subject's effectiveness at performing these tasks; however, the subject's results pattern the general consensus and are therefore included. This can be justified because the physical limitation would arguably affect the results of each presentation view similarly. Also, we would hope that the improved presentation views would similarly improve the performance of those with physical limitations.

4.1.2 User Study Design

Each user was asked to perform two tasks simultaneously in a controlled scenario over 16 different trials: (1) the primary task was to detect and identify pre-described objects of interest in the video display shown on a monitor in front of them positioned to their left (Figure 4.1(a)), and (2) the secondary task was to detect and identify pre-described objects of interest in this secondary or additional visual cognitive task shown on a monitor of the same size positioned to the right of the video display (Figure 4.1(b)). Both tasks were designed to mimic common search components needed to be performed in scenarios in which these mUAVs have been used.

Prior to completing the trials, each user was required to complete training. The script used for the training is included in Appendix B.1. The training introduced the user to samples of the described objects of interest (red umbrellas) that would be seen



(a) Video Display

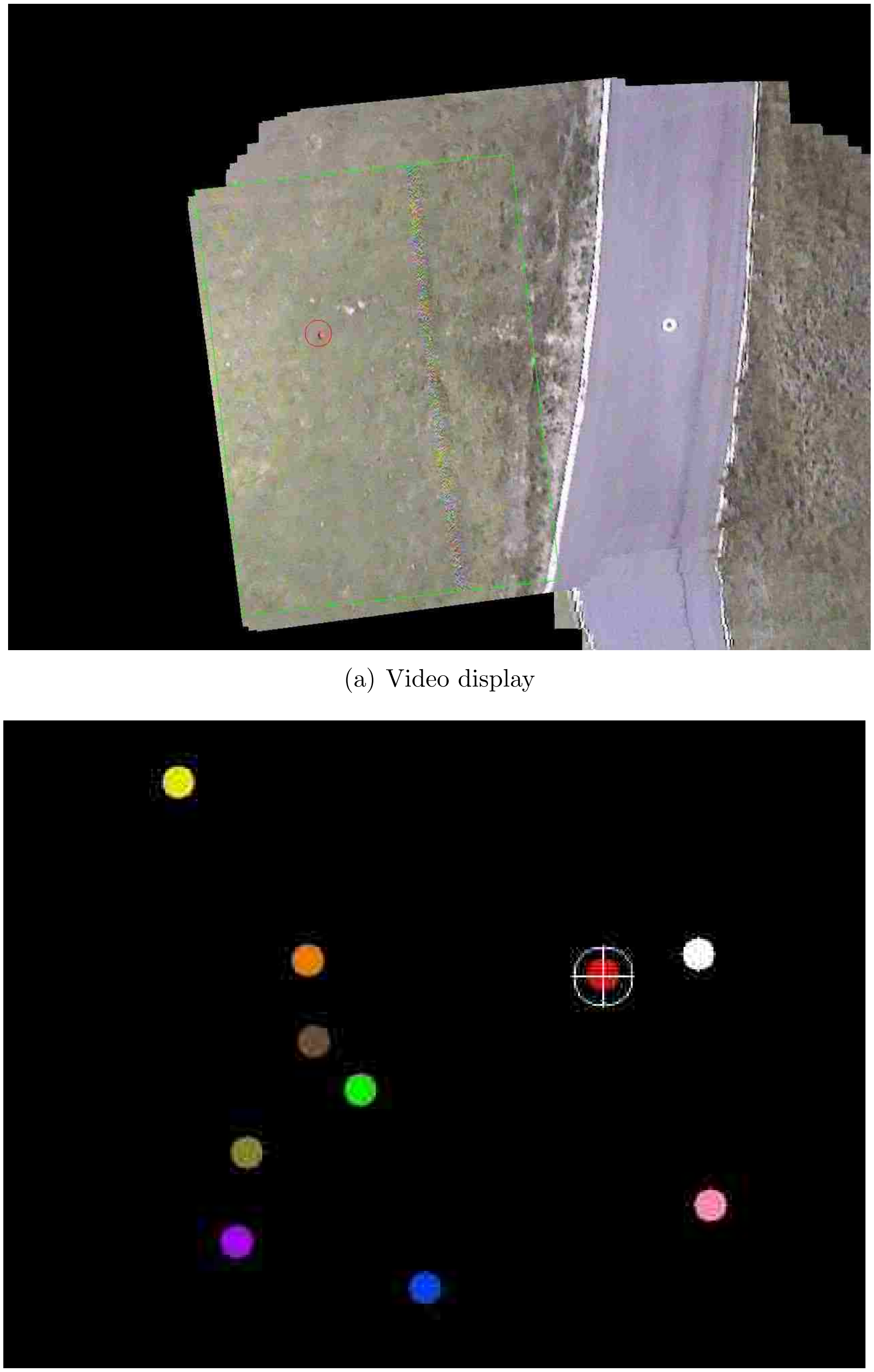
(b) Secondary Display

Figure 4.1: An example of the user study displays as they would be seen side by side using two separate monitors

in the mUAV video, samples of the secondary display, and samples of the video that would be used. Each user was also trained on the user interface using four different 30-second examples of what they would experience in the trials—which consisted of clips of the video that would be used on the video display using a random ordering of each of the four different views respectively, as well as with samples of the secondary sequences on the secondary display.

4.1.3 The Video Display

On the video display, each user was presented with a controlled random ordering (Section 4.1.5) of 16 different short video clips acquired using a mUAV that was engaged in common search patterns. Each clip lasted about 1.5 minutes and was presented to the user using one of the four possible views: original, stable-E, E-mosaic, or stable-E-mosaic. Within each clip, there was a random number of objects of interest (red umbrellas) of familiar shape and color to the user placed randomly throughout each video sequence (see Figure 4.2(a)). The subjects were asked to detect and identify as many of these objects as possible within the video presentation throughout each trial. This task was used to measure the user's ability to detect



(a) Video display

(b) Secondary display

Figure 4.2: Examples of the user study displays. The video display (a) has a red umbrella selected with a red circle around it. The secondary display (b) has a red spot selected with the white circled cross selection marker placed on it.

and identify correct objects of interest using that clip-view combination. The video searching task presented to the users was exactly what one could expect to be using in a real search situation involving a mUAV.

4.1.4 The Secondary Display

In addition to the objects of interest, the subjects were also asked to detect and identify as many red spots in the secondary window as they felt possible without jeopardizing their ability to detect and identify as many objects of interest in the video display as possible. This secondary searching task was designed to provide a measure of the user's ability to simultaneously perform a task similar to that traditionally required for simultaneously piloting the aircraft while performing a video search.

On the secondary display, the user was shown a controlled random sequence of uniquely colored spots dependant on the corresponding clip (see Figure 4.2(b)). Using a number q of unique colors, one of which was red, we colored and displayed p number of uniquely colored spots against a black background using a controlled random sequence that consisted of the time intervals, spot positions, and coloring of each spot. Each clip corresponded to its own particular randomized spot generation sequence.

This secondary searching task was adjusted so as to make the video searching task uniformly difficult enough across all 16 clips and four presentation views to make the comparative results among the four views more distinct. We regenerated the spots every 2–5 seconds, and used values of $q = 12$ and $p = 10$, implying that the red spot has an 82% chance of being displayed per regeneration.

4.1.5 Clip-View Ordering

In order to facilitate within and between subject, clip, and view comparisons, each user was assigned a controlled random ordering of the 16 clips presented using one of the four different views. For these controlled random orderings to comply, we needed to ensure that every clip and every view is seen an equal number of times per user as well as seen a progressively equal number of times by all users. We also needed to ensure that each clip-view was seen a progressively equal number of times across all users.

First, to maximize the between-view comparisons, we create all of the possible permutations of the four different views, which we will call *view-blocks* (see Appendix A.1). We then restrict each user to seeing only four of these view-blocks. In creating a user view-blocks table (see Appendix A.2), we also enforce that each randomly chosen view-block be used an equal number of times by the collective previous users before allowing it to be used again. This ensures that the view-blocks are used an equal number of times in a progressive fashion relative to the user number.

Now that the ordering of the views each user will see is determined using the user view-blocks table in conjunction with the view permutations table, we can now create the final user clip-view schedule table (see Appendix A.3) by ordering the pairing of the 16 clips with the views. In doing this, it is important that each user see each clip once, that the clip orderings are random, and that each clip-view combination is seen a progressively even number of times relative to the collective previous users. To accomplish this, we created the clip-view count table and ensured that each random ordering of views per user preserves this clip-view evenness over the orderings of all previous users' orderings before moving on to determine the next user's clip ordering.

As shown in the clip-view tallies table in Appendix A.4, for 26 users, each view is seen 104 times, each clip is seen 26 times, each view-block is seen 4–5 times, and each clip-view is seen 6–7 times. This controlled random ordering within the clip-view schedule can provide the desired within and between comparisons among the subjects, clips, and views.

4.1.6 User Study Interface

Each clip-view combination was presented using an interface that allowed the subjects to easily and intuitively select objects of interest seen throughout the video and secondary displays. The user training script included in Appendix B.1 describes these processes in more detail. This user study interface was designed to require the least amount of training as possible so as to minimize the performance differences between näive and biased subjects caused by this training overhead. The user interacted with the system using only the mouse, and the controls for the video display and the secondary display were very similar.

As described in Section 3.9.3, anytime an object of interest—a red umbrella—was thought to be seen on the video display, the user could freeze the frame by mouse-left-clicking anywhere in the video display window. Freezing the frame caused the display of the video to freeze, but did not pause the video—the video continued to play in the background. So the longer the frame was frozen, the more of the video content the user would miss seeing.

In this freeze-frame state, the user had two options: to select the object of interest by mouse-left-clicking on it—at which point a red ring would very briefly highlight the selected area (see Figure 4.2(a))—or make no selection and cancel the freeze frame by mouse-right-clicking anywhere within the video display window. Any part of the object of interest (red umbrella) could be within the ring in order to be counted as a precise hit (see Section 4.2.7). Once either of these two options was selected, the video presentation would be unfrozen and again display the video. This method of freezing and unfreezing the video was designed to imitate a live video search situation.

The control sequence in the secondary display was similarly simple. When an object of interest, a red spot, was detected, the user could place a marker over it by mouse-left-clicking on the red spot (see Figure 4.2(b)). This marker was a large white-circled cross. The center of the white cross needed to be on the red spot to count as a hit. This marker's position could be adjusted as many times as needed by mouse-left-clicking. To cancel the selection and remove the marker, the user could mouse-right-click anywhere within the secondary display.

After each trial, the subject was asked to use the mouse to answer three post-trial questions that were shown on the secondary display. These questions related to their relative preference between the previous two trials' presentation views and their perceived performance relative to the previous trial's red umbrella and red spot misses (see Appendix B.2).

After each subject completed the 16 trials, they were asked to complete a brief questionnaire about their overall impressions and preferences among the four different presentation views (see Appendix B.2). Their responses are presented and discussed in Sections 4.2 and 4.3.

4.2 Objective Results

We gathered results about (1) the primary task's umbrella hit rates, (2) secondary task's spot hit rates, (3) hit rates given whether the subject is either biased or näive and hit rates (4a) within the current frame, (4b) within the history of the mosaic (applicable only to the E-mosaic and stable-E-mosaic presentation views), or (4c) within an invalid region (mostly applicable to the original and stable-E presentation views and considered misses). We also gathered results about (5) false-positive rates and types. It is important to keep in mind that these results reflect the differences among the four different presentation views, the differences among the 16 different clips used, and the differences in user bias.

First, however, one very important observation to mention is that there is no statistical difference between the objective results of this study for näive and biased subjects, who had a 73% and 72% probability of hitting the red umbrellas, respectively (see Table 4.4). This is very important because it suggests that biased users who are familiar with the video or who already have certain preference among the presentation views perform no differently from näive users who are completely unfamiliar with the video and who may have no preferences among the views. This observation also allows us to combine the objective results of the bias and näive users together for analysis that will represent a larger sample of the general population. Furthermore, the subjective results between the näive and biased subjects also contain obviously similar patterns suggesting little significant differences between them, allowing us to combine them as well for analysis.

4.2.1 Spot Hit Rates

One important observation to make early about the objective results is that, as seen in Table 4.3, the success rates of detecting and identifying the red spots in the secondary display are very high and consistent across all of the users as well as across all of the views at about 94%. This suggests that any influence that the additional cognitive load may have had in the results will be expressed mainly in the differences among the red umbrella hit rates.

We also performed a between-clip analysis to quantify the differences in success probabilities among the different clips. This identified that one particular clip was an outlier wherein all subjects identified all of the umbrellas regardless of the accompanying view—so it has been thrown out of the results analysis.

	Spot Hit Rate
E-mosaic	94.88%
stable-E-mosaic	93.24%
stable-E	93.67%
original	94.99%

Table 4.3: Subjects' overall performance at the secondary task per presentation view

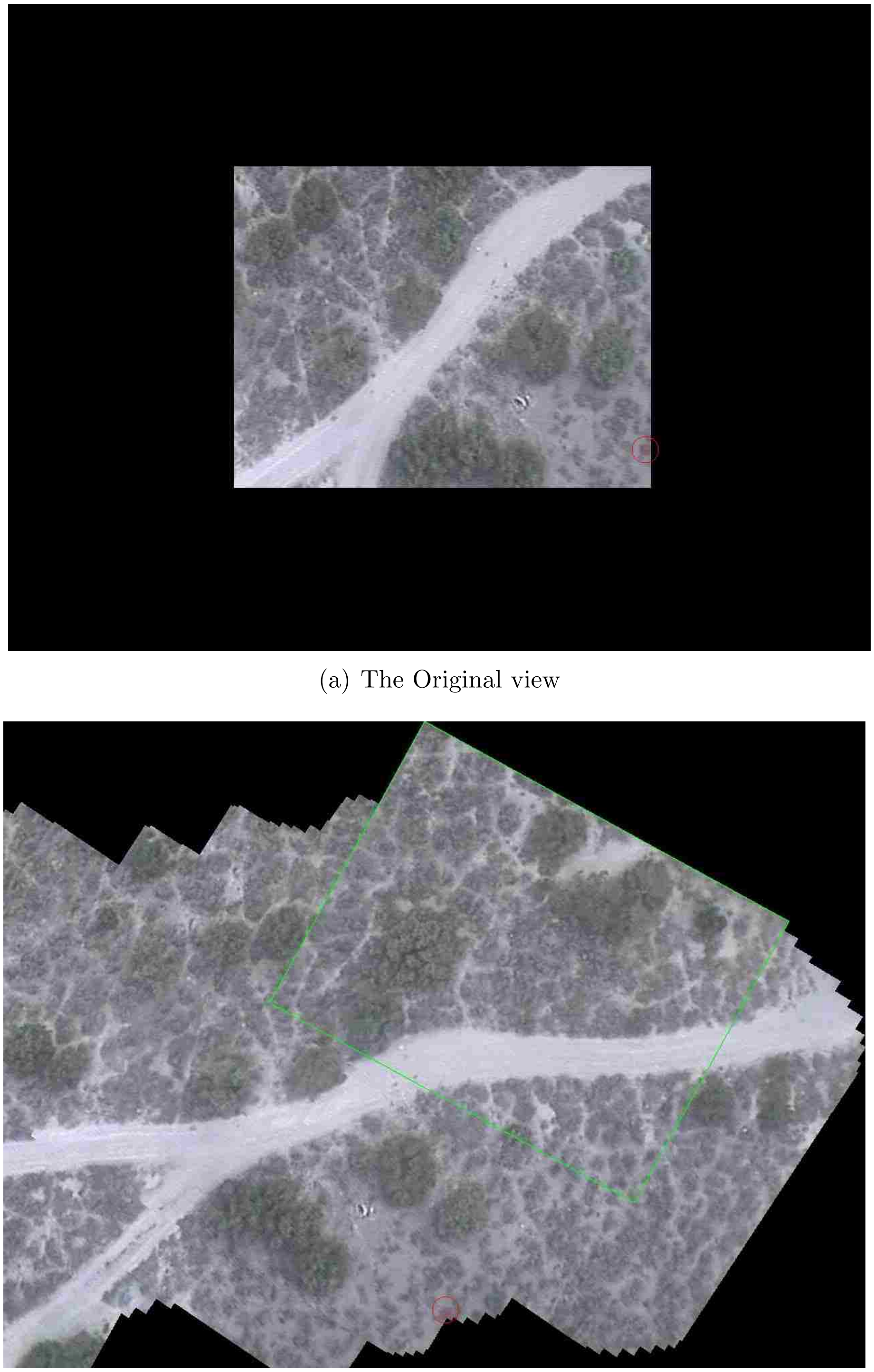
4.2.2 Hit Probabilities

The results shown in Table 4.4 support our hypotheses that providing the user with an increased viewing frustum and stabilized view will increase the probability that objects of interest will be detected throughout mUAV-acquired video. The E-mosaic view gave the largest increased percentage at 45.33% in hit probability over the original view. Also, there is a strong ($\sim 43\%$) improvement from the non-mosaiced to mosaiced views. However, there seems to be almost no statistical difference in hit probabilities between the two non-mosaiced views, nor between the mosaiced views.

This improvement is largely explained by referring to Figures 4.3(a) and 4.3(b). In Figure 4.3(a), you can see that the object of interest, the red umbrella, is visible only for a couple of frames (or $1/15^{th}$ of a second) in the lower-right corner of the original view—which would appear very similar to the stable-E view. However, in the corresponding mosaiced view, as seen in Figure 4.3(b), we can see this red umbrella for a much longer time frame over possibly hundreds of frames, or several seconds, before it moves out of our viewing frustum.

	ω	P	% improvement over P_{low}
E-mosaic	1.6610	84.04%	45.33%
stable-E-mosaic	1.5486	82.47%	42.62%
stable-E	0.3935	59.71%	3.26%
original	0.3156	57.83%	0.00%
biased	1.0051	73.21%	-
näive	0.9543	72.20%	-

Table 4.4: Hit probability comparisons among the different presentation views as well as between the näive and biased subjects, where ω is the least-squares means estimate and P is $(e^{\omega})/(1+e^{\omega})$, *i.e.*, the probability that the object of interest will be detected given the corresponding presentation view or subject. Also, the improvement over the lowest P_{low}, which happens to corresponds to the original view, was computed by $(P_{view} - P_{low})/P_{low}$.



(a) The Original view

(b) The E-mosaic view

Figure 4.3: These two images illustrate how the history of a mosaic can increase the hit rate. The original view's (a) red umbrella, seen at the lower-right corner of the view and circled here with red, is visible in only a couple of frames. On the other hand, the E-mosaic view (b) has that same red umbrella, seen in the lower-middle of the view and circled here with red, is visible over hundreds of frames.

4.2.3 Similarity Measure among View Presentations

These observations are further supported in Table 4.5, where it can be seen that there is a very high similarity between results of the E-mosaic and stable-E-mosaic views (the mosaiced views), the stable-E and original views (the non-mosaiced views), as well as between the biased and näive subjects. Furthermore, it can be seen that there are no statistical similarities expressed in the results of the non-mosaiced views as compared to the mosaiced views. These results suggest that the biggest improvement in hit probability is due to the presence of the mosaic in the presentation, which is supported by the results in Table 4.6.

		ψ
E-mosaic	stable-E-mosaic	0.9674
E-mosaic	stable-E	<0.0001
E-mosaic	original	<0.0001
stable-E-mosaic	stable-E	<0.0001
stable-E-mosaic	original	<0.0001
stable-E	original	0.9804
biased	näive	0.8247

Table 4.5: A comparison of the similarity measures of the results among the four different views as well as between the näive and biased users. ψ are the differences of least squares means respectively, where values closer to 1.0 indicate high similarity, and values closer to 0.0 indicate low similarity.

4.2.4 Current Hits versus History Hits

Table 4.6 shows how many of the hits were made in the current frame versus in the history of the mosaic. Since neither of the non-mosaiced views (original and stable-E) provide a mosaic in their presentation, only the mosaiced views will show a percentage of hits in the history.

It is interesting to note the strong correlation of the increases in hit probabilities between the mosaiced and non-mosaiced presentation views shown in both Table 4.4 and the "In the History" column of Table 4.6. In Table 4.4 the mosaiced

	In the Current Frame		In the History		Total
E-mosaic	128	62.44%	77	37.56%	205
stable-E-mosaic	137	68.66%	62	31.34%	199
stable-E	147	100.00%	0	0.00%	147
original	144	100.00%	0	0.00%	144

Table 4.6: A comparison of the percentage of hits that were made in the current frame versus the hits that were made in the history of the mosaic among the four presentation views. Note that neither the stable-E nor the original views allow a hit within the history. Each view had 254 total possible hits.

views show about a 43% hit rate increase over the non-mosaiced views, and Table 4.6 shows that about 34% of the mosaiced views' hits occurred in the history of the mosaic. We believe that this correlation empirically shows that the increase in hit probability is largely due to the provision of a history in the mosaiced presentation views.

Also, we believe that the main difference between hit probabilities between the stable-E-mosaic and E-mosaic views is because the stable-E-mosaic view presents less of a history than the E-mosaic view presents. This may be an acceptable tradeoff. One advantage that the stabilized view path views (stable-E-mosaic and stable-E views) may have over the non-stabilized view path views (E-mosaic and original views) is an effective decrease in user fatigue. This study was not designed to provide a fatigue measure associated with each view, discussed more in Chapter 5. Regardless, the stabilized view path views do show small improvements—either the 3.26% improvement of the stable-E view over the original view, or the seemingly corresponding ∼3.51% compensation for the lack of the longer history that the E-mosaic view provides.

4.2.5 Miss Categorizations and Probabilities

We also gathered results to categorize and analyze the misses among the four presentation views as presented in Table 4.7; but before we discuss these results, we first need to explain the first three columns of the table. “In the Black” refers to red

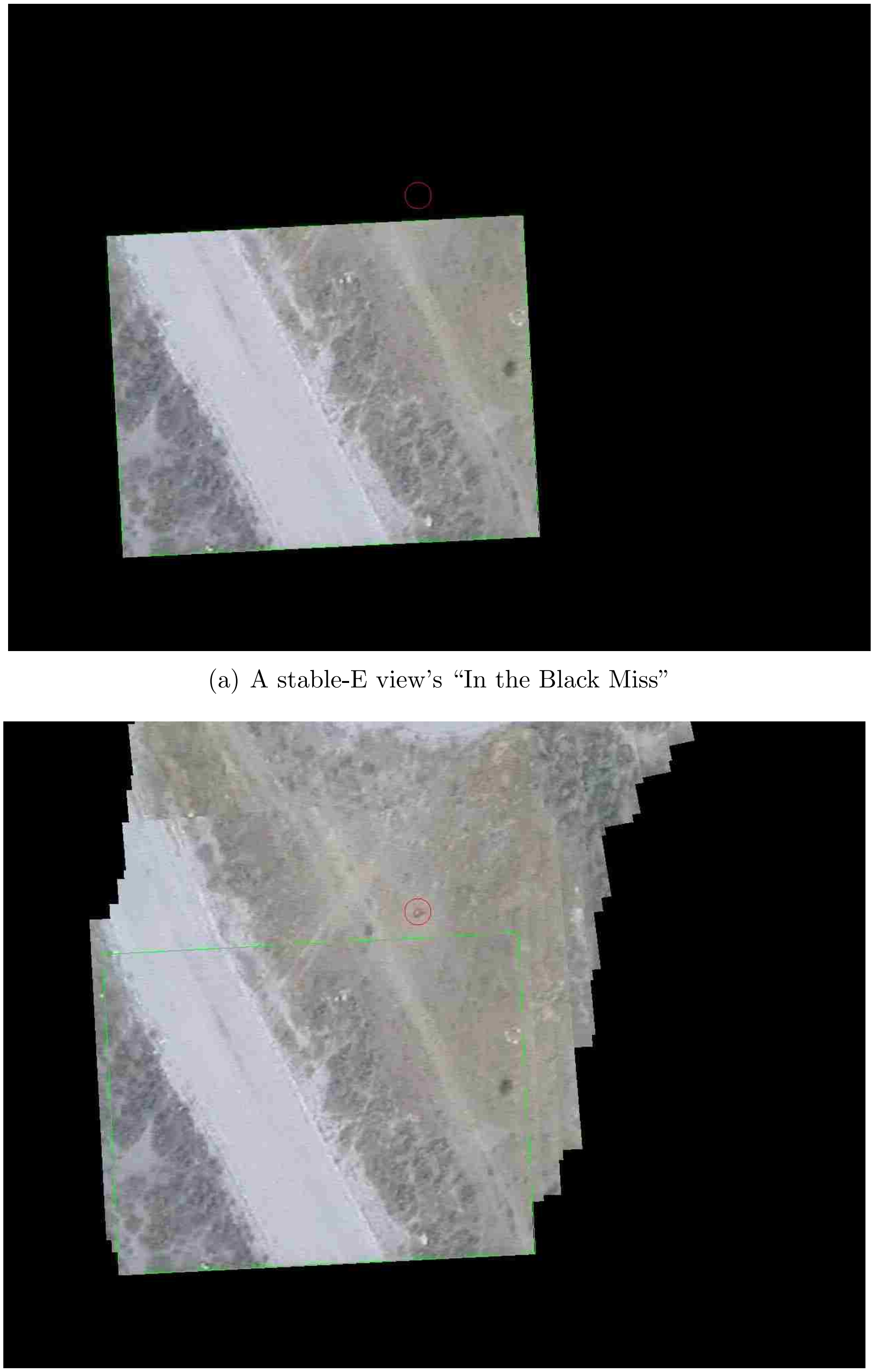
	In the Black		Late Hit		Not Detected		Total	T-B	D2M
E-mosaic	0	0.00%	2	4.08%	47	95.92%	49	49	0
stable-E-mosaic	2	3.77%	7	13.21%	46	86.79%	53	51	2
stable-E	31	29.25%	33	31.13%	73	68.87%	106	75	26
original	31	27.93%	31	27.93%	80	72.07%	111	80	31

Table 4.7: Classification of the misses among the different presentation views into the three categories. Note that "In the Black" < "Late Hit", and that "Late Hit" + "Not Detected" = 100%. The corresponding number of misses each view had is also reported. There were a total of 254 possible misses per view across all subjects.

umbrellas that were detected but selected after the red umbrella had already passed outside of the viewing frustum, as shown in Figure 4.4. This suggests that the user would most likely have hit the red umbrella if a history or local mosaic had been available.

"Late Hit" refers to red umbrellas that were detected, but a selection was made after the red umbrella had passed outside of the viewing frustum. Late hits indicate either a delayed reaction by the user, or that the user believes that something had passed through the viewing frustum that may have been an object of interest. In a real search situation, this kind of selection would merit another search of that area; but without good knowledge that an interesting object was seen, the location estimate would most likely be inaccurate which could waste time having to circle the mUAV back to re-evaluate.

An "In the Black" miss is always also considered a "Late Hit"; however, a "Late Hit" miss is not always an "In the Black" miss. A "Late Hit" miss does not always occur in an invalid region, indicating that the user thinks he or she saw something, but cannot make a guess as to where it would be given the information currently being presented to him or her. In other words, "In the Black" can be interpreted as spatial guesses of where the detected object of interest may be, and "Late Hit" misses can be interpreted as temporal guesses.



(a) A stable-E view's "In the Black Miss"

(b) A stable-E-mosaic view's hit of the same frame

Figure 4.4: Example of an "In the Black Miss". Image (a) shows the stable-E view with a red circle in the black region indicating an "In the Black Miss". The same frame using the stable-E-mosaic view, however, shows the red umbrella hit with the same red circle around it.

"Not Detected" indicates those red umbrellas that were most likely missed because they were not detected by the subjects, *i.e.*, the user gave no indication of having detected anything interesting temporally or spatially nearby the interesting object. Therefore, if 100% of the misses of a view were late hits, then it would imply that all of the missed objects of interest were detected—the converse is also true.

Therefore, Table 4.7 indicates that almost all (96%) of the E-mosaic view misses were simply missed because the users did not detect the objects of interest. The results of the stable-E-mosaic view are similar; however, about 4% of misses occurred in the black. This increase is most likely because the stable-E-mosaic view floats around the screen more freely than the E-mosaic view, causing the user to be less sure of the detected object's position when looking back from the secondary view. Also, 13% of the misses of the stable-E-mosaic view were late hits. This increase is likely due to the fact that the stable-E-mosaic view presents less of a history than the E-mosaic view.

On the other hand, the non-mosaiced views share very similar results: ∼30% of their respective misses were in the black and late hits. This suggests that almost all of their misses were accompanied by a best-guess selection in the black that would most likely have resulted in a hit had a corresponding history been presented. The stable-E view does show mild signs of improving detectability in that it had a slightly lower percentage of misses that were not detected than the original view had.

So the question arises, "How many misses would have occurred in each view had the 'In the Black' misses that were close to the object of interest been considered hits?" This question is answered by the "T-B" column which is the difference of the "In the Black" misses from the total number of misses per view. The "D2M" column then shows the relative difference of the values in "T-B" to the fewest number of misses corresponding to the E-mosaic view. These numbers then still show a significant difference in detectability between the mosaiced and non-mosaiced views and suggest that the primary difference in detectability is in the expanding of the viewing frustum, *i.e.*, the length of the history displayed, which expands the opportunity a subject has to detect the objects of interest.

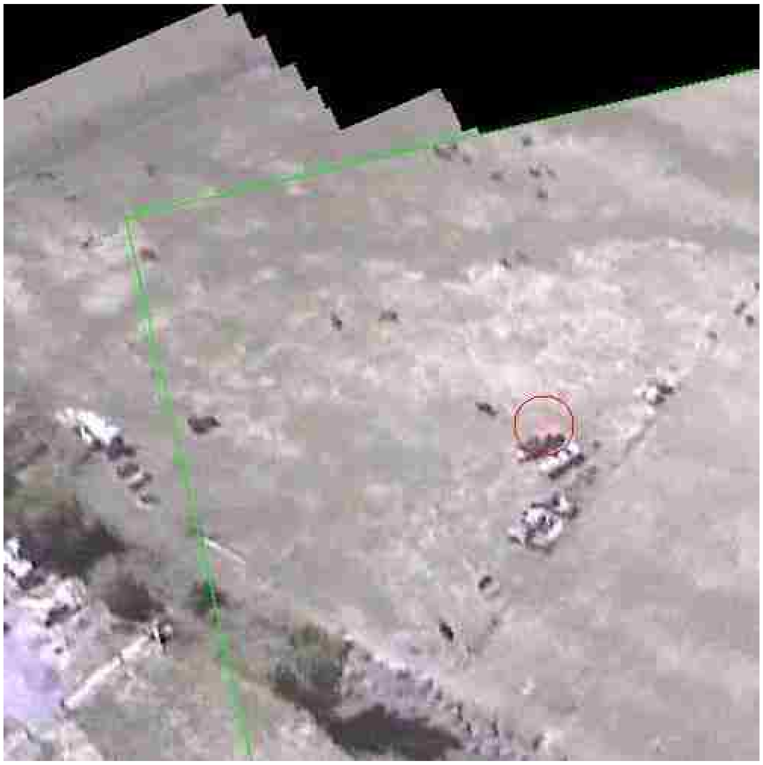
4.2.6 False Positives

False positives that occur in the current frame are likely to occur regardless of the view presentation. For example, Figure 4.5(a) is an example of a believable false positive circled in red and occurring in the current frame. At first it appears like a red umbrella, but on a closer look it is about the size of the nearby vehicle and not a red umbrella. On the other hand, Figure 4.5(b) shows another example of a believable false positive circled in red but occurring in the history of the mosaic. It too appears like a red umbrella, but it is actually a red artifact caused by noise in the images aggregated together.

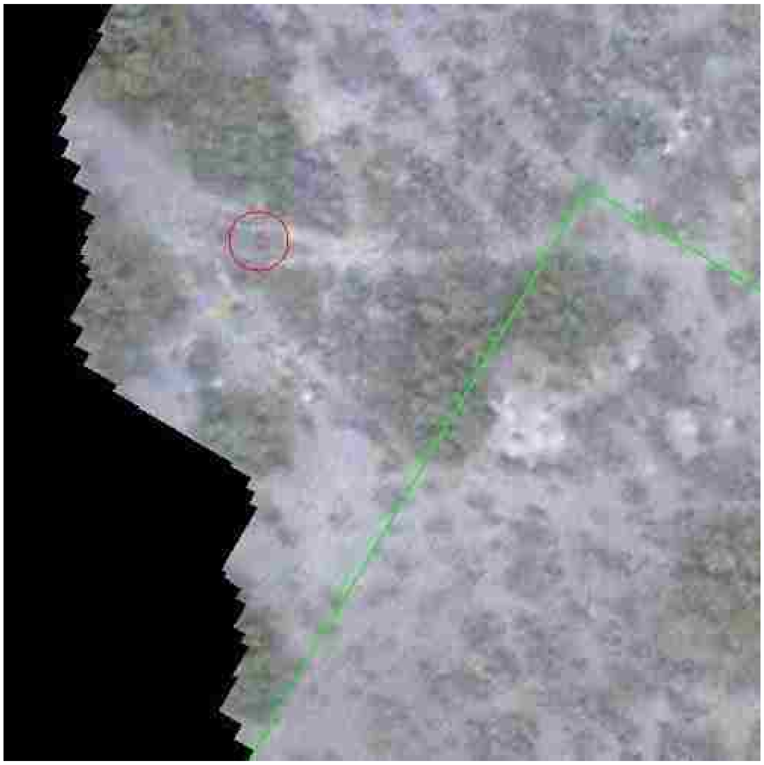
This illustrates that one of the down-sides to providing a mosaiced view is the arguably inevitable increase in the probability of false positives. False positives can occur to the fault of a mosaic presentation mainly due to possible noise caused by the video transmission or capture device, or due to possible misalignments in the mosaic. Such FP's would be manifest as false positives made in the history; and according to Table 4.8, our results show a significant increase in FP's in the history of the E-mosaic view over those made in the black of the original view. They also show a 4% chance

	FP total		FP in Current		FP in History/Black	
E-mosaic	19	18.27%	7	6.73%	12	11.54%
stable-E-mosaic	11	10.58%	7	6.73%	4	3.85%
stable-E	6	5.77%	4	3.85%	2	1.92%
original	9	8.65%	7	6.73%	2	1.92%

Table 4.8: False positives (FP). Across all subjects, each view was seen a total of 104 times and contained a total of 254 red umbrellas.



(a) False positive occurring in the current frame



(b) False positive occurring in the history of the mosaic

Figure 4.5: These images are two examples of believable false positives

of having a false positive occurrence in the history given the stable-E-mosaic view, and a 12% chance given the more lengthy history presented in the E-mosaic view.

Similar to late hits, false positives can also cause unnecessary repeat searches; however, our results indicate that these unnecessary repeat searches due to false positives caused by a mosaiced view would occur fewer than the number of repeat searches caused by late hits when using a non-mosaiced view. In addition, the increase in detectability that the mosaiced view can provide far outweighs the cost of the associated risks of possible unnecessary repeat searches.

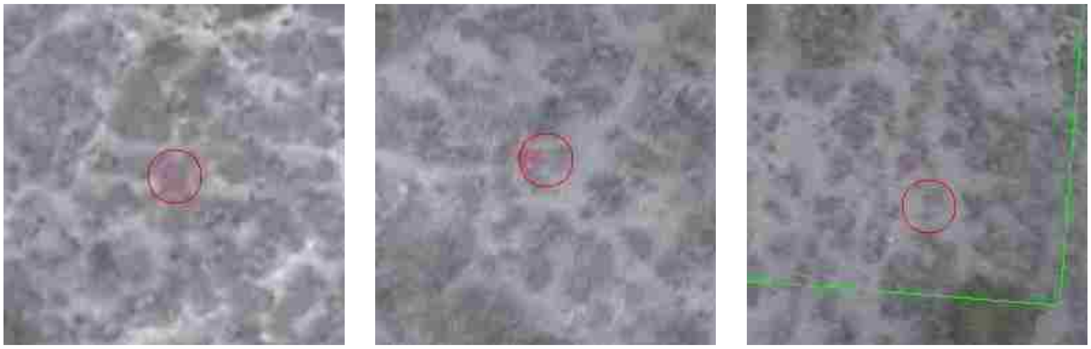
4.2.7 Hit Repeats and Precise Hits

Table 4.9 reports on the comparative number of repeat hits. Traditionally, these can be useful when wanting to place a confidence measure on the detected object of interest as well as providing more samples to better estimate the relative geo-location of the object of interest. However, we instructed the subjects that they needed to only select each unique object of interest once but selecting them more than once would not count against them. So it is a "better safe than sorry" repeat hit rate.

Accordingly, the stabilized view path views (the stable-E-mosaic and stable-E views) have the higher percentages. This is most likely because of the added secondary task that required the subjects to have to repeatedly divert their visual attention to another screen; and when they look back to the video screen, the current frame is

	% of Hit Repeats	% of Precise Hits	Total # of Hits
E-mosaic	10.73%	97.56%	205
stable-E-mosaic	6.03%	95.98%	199
stable-E	10.88%	96.60%	147
original	8.33%	97.22%	144

Table 4.9: Hit repeats and precise-hit percentages. Across all subjects, each view was seen a total of 104 times and contained a total of 254 red umbrellas.



(a) Most precise hit (b) Less precise hit (c) Imprecise hit

Figure 4.6: Examples of acceptably "Precise Hits", and an imprecise hit.

usually not in as predictable a position as the non-stabilized view path views (the E-mosaic and stable-E views) would be.

Also, the "% of Precise Hits" give a positive result in regards to the user interface. The corresponding high percentages across all four presentation views imply that given our user interface—regardless of the view presentation used—a user with moderate computer experience has a 96–97% probability of precisely identifying (selecting) the detected object of interest. These results directly support our initial hypotheses that the ability to easily and briefly pause our user interface allows the user to more precisely and more accurately identify objects of interest.

This also suggests that the precise and accurate estimates of an object's geolocation would then rely more heavily on the precision of the pose estimation and frame-time synchronization processes, and rely less on the user's eye-hand coordination skills to precisely select the objects of interest—given our user interface.

4.3 Subjective Results

The subjective results of this study are composed solely of the responses to the questions asked of the 26 subjects between each trial and after they completed all 16 trials. These questions can be found in Appendix B.2. As in the objective results, it is important to keep in mind that these results also reflect the differences among the four different presentation views, the differences among the 16 different clips used, and the differences in user bias.

The first results that we will discuss are the subjects' initial impressions of the views (Section 4.3.1). We will then discuss their responses to the questions asked after each trial about their between-view comparisons (Section 4.3.2) and their estimates of their umbrella and spot misses (Section 4.3.3). Finally, we will present and discuss their overall impressions and preferences among the four different presentation views (Section 4.3.4).

4.3.1 Initial Impressions of the Views

After each subject completed his first trial, he was asked to rate his overall impression of that view presentation since there was no previous view to compare it to. The tally of these initial impressions are listed in Table 4.10. The sum of the values in this table sum to 26, the number of subjects tested. These results of first impressions reflect a combination of the subject's prior knowledge about the presentation views, the clip used in the first trial, as well as the type of view presented. These results are significant because they suggest that regardless of the subjects bias and clip difficulty level, initial impressions among the view presentations tend to favor the mosaiced views over the non-mosaiced views. More interesting is that without prior knowledge

	Easy	Medium	Hard
E-mosaic	6	1	0
stable-E-mosaic	2	6	0
stable-E	0	4	1
original	0	5	1

Table 4.10: Initial presentation view impressions

of the other presentation views, none of the näive subjects considered either of the mosaiced views hard nor either of the non-mosaiced views easy.

4.3.2 Between-View Comparisons

After each trial, the subject was asked to compare the presentation view of that trial to the one previous to it. They were given three choices: harder (<), about the same (∼), or easier (>). The collective results are shown in Table 4.11. For example, 22 subjects thought that the E-mosaic view was easier than the original view (E-mosaic > original = 22).

	E-mosaic	stable-E-mosaic	stable-E	original	E-mosaic	stable-E-mosaic	stable-E	original
Row ∼ Column								
E-mosaic	3	9	3	6	3	26	7	16
stable-E-mosaic	17	6	5	5	-	6	9	13
stable-E	4	4	5	11	-	-	5	23
original	10	8	12	7	-	-	-	7
Row > Column								
E-mosaic	2	17	28	22	2	26	**53**	**41**
stable-E-mosaic	4	1	22	18	8	2	**44**	**32**
stable-E	1	10	4	0	1	14	8	10
original	2	7	7	1	5	11	15	2
Row < Column								
E-mosaic	0	4	0	3	2	8	**1**	**5**
stable-E-mosaic	9	1	4	4	26	2	**14**	**11**
stable-E	25	22	4	8	53	44	8	15
original	19	14	10	1	41	32	10	2
	Original Data				Combined Data			

Table 4.11: The comparisons between the couplings of the four different presentation views. The “Combined Data” columns combine the similar comparisons within the “Original Data” columns, *i.e.*, (E-mosaic > original) ∼ (original < E-mosaic).

Also, since there is an approximate symmetry in saying that option A was harder than option B, and saying that option B was easier than option A, we also provide a combination of these symmetric results of the "Original Data" columns into the "Combined Data" columns. For example, a combination of 41 subjects felt that the E-mosaic view was easier than the original view or that the original view was harder than the E-mosaic view. Note the symmetry of the Combined Data's "Row > Column" table to the "Column < Row" table along the diagonal. The two 4×4 tables are thus transposes of each other.

These results show an obvious heavy leaning towards the easiness of the mosaiced views over the non-mosaiced views in both the "Original Data" and "Combined Data" tables. This can be seen in the distant relationship between the collective bold values of the two "Combined Data" preference tables.

Furthermore, the values along the E-mosaic view's "easier" rows are by far the strongest values in the tables, and the values along the E-mosaic view's "harder" rows are by far the smallest values. These results strongly suggest that the presentation of the E-mosaic view mUAV-acquired video is the easiest among these four presentation views for users to use. Similarly, the stable-E-mosaic's presentation is easier than both of the non-mosaiced views.

One unexpected result is that the stable-E view seems to have been perceived as more difficult than the original view. Several subjects afterwards commented to suggest two reasons for the relative difficulties associated with the stabilized view path views.

First, we believe that this difficulty was heavily influenced by the visual secondary task that required the subject to be visually engaged on another screen. When the user was forced to look away from the video display for a moment and then look back to again focus on the video display, the current frame was in a much less predictable position on the video display. Even though the content of the video would be in a more predictable position on the video display, the human visual system mainly uses the strong gradients of the borders of the frame to refocus, cancelling the benefits of having more stabilized content within the video. This effect may be negatively influencing both the stable-E-mosaic and the stable-E view (the stabilized view path views).

Another possible reason for this relative difficulty is that the human visual system is very sensitive to movement along high gradients in the video; and since the edges of the current frame of the stable-E video have a very high gradient and can move rapidly back and forth in order to stabilize the content of the video, the human visual system is constantly battling trying to remain focused on the content and trying to not focus on the movement. This problem would not effect the original view at all, but it would only slightly negatively effect the E-mosaic view, more negatively effect the stable-E-mosaic view, and most negatively effect the stable-E view. We propose some approaches to minimize these problems to further improve the stable-E-mosaic presentation in Chapter 5.

4.3.3 Performance Confidence Measures

In addition to asking each subject to compare difficulty levels after each trial, we also asked them to report the number of spots and umbrellas that they believe they may have missed during each trial. These collective results are reported in Table 4.12 and provide a measure of confidence in their performances relative to each presentation view—serving as a hybrid of objective and subjective results.

This table presents two groupings of columns under both “Umbrellas” and “Spots”: “Occurrences” and “Total”. Each grouping contains a “-”, a “0”, and a “+” column. The “Occurrences” grouping are the collective occurrences of the negative, zero, or positive values of the number of umbrellas or spots that the subjects thought they missed subtracted from the number of umbrellas or spots that they actually

	Umbrellas						Spots					
	Occurrences			Total			Occurrences			Total		
	-	0	+	-	0	+	-	0	+	-	0	+
E-mosaic	10	64	30	-11	0	38	21	49	34	-24	0	66
stable-E-mosaic	14	61	29	-17	0	36	23	44	37	-24	0	80
stable-E	11	44	40	-11	0	61	20	44	40	-23	0	77
original	9	44	49	-11	0	74	19	53	32	-19	0	46

Table 4.12: Hit confidence measures, where ((Values) = (Actual Missed) - (Thought Missed)).

missed for each presentation view across all subjects. Similarly, the "Total" are the corresponding sums of these negative, zero, or positive values. "-", "0", and "+" "Occurrences" values indicate how many times the subjects collectively felt under-confident, accurately confident, or overconfident in the number of umbrellas or spots that they missed, respectively. Similarly, the "Total" values provide a measure of how under-confident, accurately confident, or overconfident they were, respectively.

Accordingly, the "Total" "-" and "+" values will always be equal to or more negative and positive, respectively, than their corresponding "Occurrences" values. Gross differences will indicate severe respective under-confidence or over-confidence. Also, the "Total" "0" values will always be 0, *i.e.*, the sums of 0's will always be 0. Values in the "0" column of the "Occurrences" group indicate how many times umbrellas or spots were missed and that the subject was accurately confident that he missed.

Therefore, we can observe a large consistent difference between the "-" and "+" values of both groupings, suggesting that the subjects tend to consistently be overconfident, *i.e.*, when they missed an umbrella, they usually were confident that they did not miss one. This can have a very drastic effect in a real search situation because it suggests that once an area is searched, if the object of interest was in fact in that area but not seen, we may be too mistakenly confident that the area covered does not contain an object of interest and mistakenly discourage searching those areas again. This stresses even more the high importance of increasing the detectability of objects of interest in the presentation of the video.

With respect to the "Umbrella" results, we can also observe by the similar difference of ∼20 between the "0 Occurrences" values of the mosaiced and non-mosaiced views—and knowing that the non-mosaiced views caused more misses as shown in Table 4.7—that when subjects missed an umbrella using a mosaiced view, they were more likely to know that they missed one and were more accurately confident about it than they were when they missed an umbrella using a non-mosaiced view. Furthermore, subjects were more frequently and more grossly overconfident that they had not missed umbrellas when they used a non-mosaiced view, as seen in the "+" values of the "Total" group as compared to the "+" values of the "Occurrences" group. This empirically suggests that using a mosaiced view can significantly decrease the possibility of grave false negatives in a real search situation involving mUAV-acquired video.

However, the results under the "Spots" group surprisingly suggest that the subjects are more accurately confident of spot misses when using the original view than they are with the other four views. In addition, they tend to be less under- or overconfident about spot misses when using the original view compared to the others. In fact, the views that caused the most frequent and gross overconfidence related to missed spots were the two stabilized view path views (stable-E-mosaic and stable-E views). We believe that these results further support our reasonings presented at the end of Section 4.3.2, and we will address this topic further in Chapter 5.

4.3.4 Overall Impressions and Preference Orderings

After each subject completed the 16 trials, they answered some follow-up questions (Appendix B.2). Their collective responses are shown in Table 4.13. It can be seen

	Most Comfortable	Least Straining	Most Straining	Least Oriented	Most Oriented	Best Stamina	#1 Preference	#2 Preference	#3 Preference	#4 Preference
E-mosaic	17	13	0	2	18	17	17	8	1	0
stable-E-mosaic	7	6	0	4	5	4	6	11	8	1
stable-E	1	1	18	10	2	2	2	1	9	14
original	1	6	8	10	1	3	1	6	8	11
Desired	H	H	L	L	H	H	H	H	L	L

Table 4.13: Follow-up subjective questionnaire results. The 'Desired' row indicates the desired values H (for high values) and L (for low values).

that the E-mosaic view has the strongest desirable values across the table and is overwhelmingly the preferred view. The stable-E-mosaic view is a distant second.

One of the most significant results shown in this table is the subjects' overall impression about how the four presentation views compare in improving orientation. As we hypothesized, by suppressing the content motions in the rotational γ, the stable-E, the stable-E-mosaic, and the E-mosaic views all increased user orientation over the original view; and, given the overall strengths that the E-mosaic view is shown to have over the other three views, it is no surprise that the E-mosaic view had the most impressive impact on user orientation in this study. However, our results slightly differ from our hypothesis in that the E-mosaic view gave the subjects the greatest sense of orientation, followed by the stable-E-mosaic then the stable-E views, respectively.

Another result applicable to our hypothesis is the subjects' overall impression about how the four presentation views compare in improving attentiveness or viewing stamina. Similar to the orientation results, the E-mosaic view was perceived by the subjects as the view that they could watch the longest. This result also varies from our hypothesis in that the E-mosaic view was preferred over the stable-E-mosaic view, but also that the original view was preferred over the stable-E view.

Even more surprising is that the original view has a more desirable collective row than the stable-E view. Even though the subjects performed arguably slightly better using the stable-E view over the original view as shown throughout Section 4.2, subjects found it to be overwhelmingly the most straining view—causing the stable-E view to be the least preferred of the four presentation views. We again believe the reasons for this are the same reasons that are presented at the end of Section 4.3.2, and we will address this topic further in Chapter 5.

Preference Orderings

Table 4.14 shows the preference orderings in another light. It further supports how overwhelmingly preferred the mosaiced views (A and B) were: 88% of the subjects preferring a mosaiced view the most, and 62% of the subjects preferring both mosaiced views over the non-mosaiced views.

ABCD	6
ADBC	5
ABDC	4
BADC	4
BACD	2
CABD	2
ACBD	1
ADCB	1
DBAC	1

Table 4.14: The permutation preferences among users where A=E-mosaic, B=stable-E-mosaic, C=stable-E, D=original views.

4.4 The Bottom Line

These objective results show that the two mosaiced presentation views' results (E-mosaic and stable-E-mosaic views) are about 97% similar to each other but very different from the other two non-mosaiced presentation views' results (original and stable-E views), which are also shown to be about 98% similar. Subjects were found to generally have a 45% increased probability of correctly detecting, identifying, and selecting objects of interest throughout mUAV-acquired video using our mosaiced views over both the original view as well as our stable-E view. This increase in hit probability is shown to be closely related to the presentation of the history of the local mosaic, where about 34% of the hits in the mosaiced views occurred.

Also, the subjects were more accurately confident about their misses using the mosaiced views. They overwhelmingly (92% of the subjects) preferred the mosaiced views over the non-mosaiced views and found the mosaiced views to be more orienting (88%) and less straining (73%) than the non-mosaiced views.

Given the entirety of these results, we have made several observations to empirically support our hypotheses that the stable-E, E-mosaic, and stable-E-mosaic presentation views of mUAV-acquired video improve a user's ability to detect objects of interest seen throughout the video as well as improve the user's sense of orientation and attentiveness throughout the presentation of the mUAV video. In addition, we have shown evidence to support that by using our proposed user interface, the average computer user can more precisely and more accurately identify (select) these objects of interest.

However, we also found that the E-mosaic view, rather than the stable-E-mosaic view, consistently gave the most positive and overwhelming results. In addition, the stable-E view did not outperform the original view quite as well as we had expected; and to the contrary of our expectations, it was the least preferred presentation view of the four.

In the next chapter, Chapter 5, we present some adjustments that may be made to possibly improve the results of the stable-E-mosaic view above those of the E-mosaic view. We also present some suggestions for another user study that may be able to better compare the stabilized and non-stabilized view path presentations.

Chapter 5

Conclusions and Future Work

In this chapter, we review in summary the whole of this work as well as present our conclusions (Section 5.1). Also, we present a discussion of the limitations of this work as well as possible solutions to them (Section 5.2). Finally, we present some additional ideas that may be explored to possibly further enhance this work (Section 5.3).

5.1 Where We have Been

5.1.1 The problem

This work focuses on presenting fast forward-velocity mUAV-acquired video to users in a way that will greatly increase their ability to more quickly, more precisely, and more accurately detect and identify victim sightings within the video.

Four problems commonly plague mUAV-acquired video that have traditionally made these tasks difficult: (1) a limited viewing frustum shortens users' detection and reaction times, (2) high-frequency jitters in the video content make it difficult to focus on, follow, and select objects of interest, (3) unpredictable 6-DOF motions and rotational motions in γ can very quickly disorient the user, and (4) noisy and distorted images make it quite difficult to visually interpret the video content.

5.1.2 Our Hypotheses

We hypothesized that using a presentation of mUAV-acquired video that collectively addresses and diminishes the effects from these four problems would improve the user's ability to more precisely and more accurately detect and identify objects of interest—or more specifically, victims in a search and rescue scenario—seen throughout the video as well as improve the user's sense of orientation and attention throughout the presentation of the mUAV video.

5.1.3 Our Solutions

In order to collectively reduce the negative effects of these four problems, we have developed three presentation views of mUAV-acquired video: the Euclidean mosaic (E-mosaic) view, the stable Euclidean (stable-E) view, and the stable Euclidean mosaic (stable-E-mosaic) view. All three views reduce distortions in the video by first performing deinterlacing and undistortion routines. They then compute the spatial relationships among the frames by finding common features, matching these features, and then filtering them using our proposed short-circuited homography RANSAC filter to establish good feature-correspondence sets. Finally, these sets are used to construct the Euclidean or rigid body transformation relationships among the frames. This process has been shown to be significantly robust to noisy video.

The E-mosaic presentation view uses this Euclidean transformation to stabilize the content of the video by aggregating the sequence of images by spatially aligning them to a local mosaic, and then followed this aggregation of images by translating its viewpoint—or the canvas—so as to always keep the current frame within the view. The E-mosaic view successfully expanded the viewing frustum, but only partially removed the high-frequency content jitter from the presentation of the video content—reintroducing much of the original jitter back into the presentation as the current frames try to aggregate outside of the view.

The stable-E presentation view was introduced to further suppress these remaining high-frequencies in the content presentation by causing the view to follow the image aggregation path in smooth fashion, keeping a cushion between the border of the view and the current frame. By fitting a smoothed curve path to the cumulative image aggregation path, and following the cumulative path using this smoothed path, we are able to provide immediate real-time software-based stabilization.

However, because the stable-E presentation lacks the benefits of having the local mosaic, we created the stable-E-mosaic presentation view, which combines the complimentary benefits of both the E-mosaic and stable-E views.

In addition, we introduced a user interface that allows the users to easily pause, evaluate, and then either cancel or select objects of interest seen throughout the video. This helps to further decouple the eye-hand coordination required to select objects of interest that are detected throughout the video as well as facilitate more accurate and precise hits.

5.1.4 Our Results

In order to quantify the comparative improvements of these three presentation views over the conventional original view, we performed and presented a discussion of a user study on several biased and näive subjects. The results of this user study empirically show that the mosaiced views (the E-mosaic and stable-E-mosaic views) greatly improve detectability mainly due to the presentation of a history of frames in the local mosaic. It was also suggested that the stabilized view path presentation views (the stable-E and stable-E-mosaic views) present some unexpected visual difficulties to the users when presented with an additionally separate visual cognitive task. Overwhelmingly, however, the E-mosaic and the stable-E-mosaic presentation views were most preferred by the subjects, which were also reported to provide the greatest sense of orientation and least amount of fatigue—further supporting our hypotheses.

5.1.5 Conclusion

In conclusion, these methods provide significant contributions that enhance the real-time presentation of mUAV-acquired video and increase a user's abilities to more confidently and precisely detect, identify, and select objects of interest seen throughout the video by presenting the user with a local mosaic and stabilized content of the video in such a way that serve to minimize the collective negative effects of the four problems previous mentioned. Even with these very positive results, there are still some improvements that can be made which we discuss in the next two sections.

5.2 Remaining Limitations

Although the methods for mUAV video display presented here do enhance users' abilities to detect, identify, and select objects of interest seen throughout the video, there are still some limitations and areas for improvement.

5.2.1 User Study Adjustments

To our surprise, this user study had two very unexpected results. The first was that the stable-E view did not show the expected improvements over the original view and it was even preferred less than the original view was. Also, the stable-E-mosaic view did not perform quite as well as we had initially thought that it would, and it too was less preferred than its non-stabilized view path and mosaiced counterpart, the E-mosaic view.

As hypothesized at the end of Section 4.3.2, we believe that because the stable-E view's current frame constantly jitters back and forth without a mosaic, a lot of strong edge (high gradient) motions are introduced at the edges of the frames. Since the human visual system is sensitive to this kind of motion, subjects were constantly strained while trying to stay focused on the content of the video rather than the motion along the edges. This seems to somewhat negate the stable-E view's improved presentation of stabilized content.

Also, the stabilized view path views were perceived as being more difficult than their counterparts due to the study's secondary task constantly requiring the subjects to look away from the video presentation. Since the stabilized view path views both unpredictably moved the positioning of the current frame, even though doing this allowed the content of the video to move in a more predictable fashion, it seemed to cause the subjects difficulty in finding and refocusing on the current frame when looking back to the video.

Removing the Secondary Visual Task

It seems appropriate, therefore, to perform the user study again but without the additional visual cognitive task. It is possible that by removing the visual secondary task, the results of the stabilized view paths may improve due to the fact that the user would be given more uninterrupted time to concentrate on the stabilized content and to train their visual system to be less distracted by the jitter at the frames edges as well as the overall stabilized motion of the view path.

We do not expect that this will dramatically increase the detectability of objects between the mosaiced and non-mosaiced views or polarize the results; rather, we expect that doing this would likely increase the detectability among all of the views about the same since the users would be equally devoting their concentration more on each of them rather than dividing their attention between the secondary and video displays. However, we would anticipate that the users will be less fatigued and may actually prefer the stable-E and stable-E-mosaic views over the original and E-mosaic views, respectively.

Another commonly reported problem with our user study was that the red spots from the secondary display caused residual ghosting of spots in their visual system when they looked back to the video display. Subjects reported being distracted by seeing spots in the video display that really were not there. However, this effect was equally present during each of the four presentation views throughout the user study and would have not biased any of the results of one view over another.

Measuring a Fatigue Factor

Because this user study only presented users with 16 clips lasting 1.5 minute each, between which the user was allowed to break as long as needed before proceeding to the next trial's clip, it was not able to objectively measure a fatigue factor associated with each view. In a more realistic search situation, the user may be required to search the video for a much longer duration of time. Also, there seems to be a paradigm shift in searches involving mUAVs such that the video searchers' only task will be to watch the video presentation.

With users concentrating continually on the content of the video for much longer durations of time, we believe that the jitter within the original and E-mosaic views will have a much more negative effect on users than the results of this study suggest. Therefore, we suggest performing a similar user study without an additional visual cognitive task—*i.e.*, could possibly use an additional non-visual cognitive task or no secondary task at all—and that still measures and compares the detectability of objects among the four views, but that would use much longer video clips in order to concentrate more directly and objectively on the fatigue factor associated with each view.

Accordingly, we suspect that providing users with the stable-E or stable-E-mosaic views would respectively decrease the fatigue incurred by the presentation view, increasing their attentive endurance and abilities to detect and focus on objects within the video.

5.2.2 Combining the E-mosaic and Stable-E-mosaic Views

Because the stable-E-mosaic view neither performed quite as well as nor was preferred more than the E-mosaic view as had hypothesized that it would, we propose a hybrid of the two views that may be an improvement in comparison to both of them.

Even without an additional visual cognitive task, we suspect that the stable-E-mosaic view can have a negative effect on users because of its tendency to keep the viewpoint unpredictably floating in constant motion above the canvas while smoothly following the image aggregation path. This may have a dizzying effect on a user that is either sensitive or not accustomed to this kind of motion. In order to combine the benefits of both the E-mosaic and the stable-E-mosaic views into a single presentation view, we must avoid the residual content jitter that the E-mosaic view is prone to as well as this floating effect that the stable-E-mosaic view is prone to.

To do this we suggest starting the initial frame in the center of the view, and then performing the E-mosaic algorithm until the current frame aggregates too closely to the edge of the view. At this point, we suggest performing the stable-E-mosaic algorithm to smoothly "pull" the position of the current frame closer to the center of the view—*i.e.*, to smoothly translate the view to follow the image aggregation path. Once the current frame is close enough to the origin, the presentation would then return to performing the E-mosaic algorithm.

This new hybrid avoids reintroducing the jitter back into the presentation while disallowing the current frame to aggregate outside of the view. This would also minimize the floating motions that could cause a dizzy effect on users. Altogether, it could be more effective than both the E-mosaic and stable-E-mosaic presentation views both increasing detectability while decreasing fatigue.

5.2.3 Integrate Telemetry Pose Estimations

One of the strengths of this work is that it relies solely on the basic visual information content present in the mUAV-acquired video. However, a limitation among all three of our presentation views is that they all suffer from a gradual accumulation of error in their cumulative transform $\mathbf{Q}'$, causing our compensating rotations in γ and image aggregations to gradually drift. Without any means to compensate for this error, and being unable to perform any bundle adjustments in real time, we are unable to infer into the presentation any true constant direction marker which would be very useful in a real search situation.

One presently emerging technology that will allow us to compensate for this cumulative drift is frame synchronized pose estimations that are already being transmitted from the mUAVs. By integrating these mUAV pose estimations we can further enhance the presentation of the video we could display a compass or keep the North direction fixed in the video search view as well as distribute into the local mosaic a global compensation.

5.2.4 Using the Homography Directly to Aggregate Images

The Euclidean or rigid-body transformation $\mathbf{Q}$ has provided us with adequate estimations of the true spatial relationships among the images of the video. Using $\mathbf{Q}$ we have performed very useful local mosaicing and stabilization of the view path to smoothly follow the aggregation of images assuming a fast forward-velocity mUAV. However, gimbal-mounted cameras can keep objects within the view for long periods of time while the mUAV follows a circular path. Similar to trying to align the surface of a cone on plane, this scenario can cause very distracting misalignments in an E-mosaic, making the $\mathbf{Q}$ an inappropriate model for aggregating the images together.

In this case, using the homography $\mathbf{H}$ would more closely estimate the true spatial relationships among the images of the video and would create a better representation of the captured scene. However, as mentioned in Section 3.4.1, because of possible degenerative cases and the gradual accumulation of error and distortions, using the **H** to spatially align images together can be very unstable. To be useful, the degenerative cases must be detected and the accumulation of error and of image distortions must be reasonably constrained so as to not unrecognizably warp the images. One possible way to constrain the **H** is to restrict the cumulative homography from performing too drastic a warp in any of its degrees of freedom as well as give it the tendency to always drift back to warping images to the original frame size, but not the original placement or rotational orientation of the first frame.

5.2.5 Addressing Bottlenecks

We currently have two bottlenecks in our system that still need to be addressed before any of these three presentation views will be preferred over viewing the raw video transmissions from the mUAV on a CRT monitor. Using a Dell Precision 380 with a dual-core Intel P4 3.8 GHz processor and 2 GB of RAM, we can currently compute the calibrated spatial relationships among the images of the video using half-height frames at about 60 fps, which is in real time. However, we can still only display the uncalibrated image aggregations at about 13 fps using half-height half-width images. If we calibrated the images for display, that frame rate drops to about 7 fps.

There are two bottlenecks that we fault for these dramatic drops in frame rate: calibrating each image, and generating the larger view images. Calibrating the images is costly because it has to perform a backwards bilinear interpolation to avoid distracting holes in the calibrated image. Generating the images for display is a bottleneck because the size of the images of the view are much larger than the original frame size, and because we again have to perform the costly backwards bilinear interpolations to avoid holes in the result due to rotations. Much of this image processing is performed on the CPU, and we believe that moving as much as possible to the GPU would significantly speed up the processing of the large images needed by these presentation views.

5.3 Other Possible Future Enhancements

In addition to addressing some of the current limitations of this work, additional enhancements could also be made.

5.3.1 Other Smoothing Possibilities

Instead of using Bezier curves, we would like to consider and experiment with using other curves, like the B-Spline curve, to the cumulative path to achieve view path stabilization.

Other possibilities of path smoothing using curve fitting would be to smooth the 6-DOF pose estimations path of the mUAV in order to recreate a more continuous estimation of the true path of the mUAV. Also, curve fitting could be used to smooth the homography scale and non-z rotation values so that the cumulative homography is always wanting to warp the current image to the same size as it was originally, as suggested in Section 5.2.4.

5.3.2 Implement Feature Tracking

Once an object of interest is detected, identified, and selected by a user, it would be helpful to keep it highlighted until it moves out of the view. This could help reduce unnecessary repeat hits and allow the searcher to more easily further scrutinize the highlighted object of interest as well as more easily communicate the object of interest to other users.

5.3.3 Integrate Terrain Information

Another useful emerging technology that could prove very useful in search situations is to implement into the presentation of mUAV video both the corresponding reference imagery and the corresponding DEM (Digital Elevation Model) data.

For example, to estimate an object's geo-location, [10] use the selected pixel's location with corresponding DEM data as well as pose estimates from the aircraft that loosely correspond to a user selected frame. Merging this technology into our presentation views could allow the users to select objects of interest which would then mark on reference imagery the geo-locations of each object of interest. This marked reference imagery could then be used by the incident command of the search to help separate searching tasks and allow different teams to concentrate on their own respective separate tasks.

By using reference images and DEM data, each frame can also be projected onto the terrain, however, the presentation of this projection would still need to be stabilized or mosaiced in order to benefit from the respective increases in detectability. The combination of reference imagery and terrain, however, can be used to further refine the spatial alignments of frames. This could open the door for refining pose estimations using epipolar geometry.

5.3.4 Implement a Scrub Feature

A scrub or rewind feature with the ability to pause could also be implemented into the presentation of the video. Doing so could increase the spatial search but inhibit the temporal search—users could see any of the previously seen video at any time for review, but doing so could obviously incur longer searches through the video.

The scrub feature may lend itself well to an off-line search situation when there are additional resources available to review previously obtained flight video or in situations when the search is not as time sensitive. This feature may also work for a multi-teamed solution where one team watches for times in the continuous video where interesting objects were seen and mark the frame or the object of interest that would then signal another team to scrub back in the video to search more in-depth around the temporal spatial area of the cue within the video.

Bibliography

[1] M. A. Goodrich, B. Morse, T. McLain, and J. Dan R. Olsen, "UAV-enabled wilderness search and rescue: A human-centered approach," May 2005, nSF Proposal.

[2] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography." *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. [Online]. Available: http://www.ai.sri.com/pubs/files/836.pdf

[3] M. Brown and D. G. Lowe, "Recognising panoramas," in *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision.* Washington, DC, USA: IEEE Computer Society, 2003, p. 1218. [Online]. Available: http://www.cs.ubc.ca/~mbrown/papers/iccv2003.pdf

[4] Q. Luong and O. Faugeras, "Self-calibration of a moving camera from point correspondences and fundamental matrices," 1997. [Online]. Available: citeseer.ist.psu.edu/luong97selfcalibration.html

[5] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision.* Springer, November 2003. [Online]. Available: http://www.amazon.fr/exec/obidos/ASIN/0387008934/citeulike04-21

[6] R. Szeliski, "Image alignment and stitching: A tutorial," Microsoft Research, Tech. Rep., 2005. [Online]. Available: http://www.cs.huji.ac.il/course/2005/impr/articles/MSR-TR-2004-92.pdf

[7] I. Corporation, "Open source computer vision library." [Online]. Available: http://www.cs.unc.edu/Research/stc/FAQs/OpenCV/OpenCVReferenceManual.pdf

[8] K. Ratakonda, "Real-time digital video stabilization for multi-media applications," 1998. [Online]. Available: http://ieeexplore.ieee.org/iel4/5627/15081/00698760.pdf

[9] J. S. Jin, Z. Zhu, and G. Xu, "Digital video sequence stabilization based on 2.5d motion estimation and inertial motion filtering." *Real-Time Imaging*, vol. 7, no. 4, pp. 357–365, 2001. [Online]. Available: http://dx.doi.org/10.1006/rtim.2000.0243

[10] D. L. Johansen, "Video stabilization and object localization using feature tracking with small uav video," December 2006. [Online]. Available: http://www.ee.byu.edu/faculty/beard/papers/thesis/DaveJohanson.pdf

[11] H.-C. Chang, S.-H. Lai, and K.-R. Lu, "A robust and efficient video stabilization algorithm." in *ICME*. IEEE, 2004, pp. 29–32. [Online]. Available: http://ieeexplore.ieee.org/iel5/9605/30344/01394117.pdf

[12] M. G. Gonzlez, "Improved video mosaic construction by accumulated alignment error distribution." [Online]. Available: citeseer.ist.psu.edu/502026.html

[13] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pose, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt, "Aerial video surveillance and exploitation," *Proceedings of the IEEE: Special Issue on Third Generation Surveillance Systems*, vol. 89, no. 10, pp. 1518–1539, October 2001. [Online]. Available: http://ieeexplore.ieee.org/iel5/5/20732/00959344.pdf?arnumber=959344

[14] D. Steedly, C. Pal, and R. Szeliski, "Efficiently registering video into panoramic mosaics," in *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1300–1307. [Online]. Available: http://ieeexplore.ieee.org/iel5/10347/32976/01544870.pdf?arnumber=1544870

[15] F. Dellaert and R. Collins, "Fast image-based tracking by selective pixel integration," September 1999. [Online]. Available: citeseer.ist.psu.edu/dellaert99fast.html

[16] H. Schultz, A. Hanson, E. Riseman, F. Stolle, Z. Zhu, C. Hayward, and D. Slaymaker, "A system for real-time generation of geo-referenced terrain models," in *SPIE Symposium on Enabling Technolgies for Law Enforcement, Boston, MA*, November 2000. [Online]. Available: ftp://vis-ftp.cs.umass.edu/Papers/schultz/spie2000.pdf

[17] R. Kumar, H. S. Sawhney, J. C. Asmuth, A. Pope, and S. Hsu, "Registration of video to geo-referenced imagery," *Proceedings of IEEE CVPR*, pp. 54–62, August 1998. [Online]. Available: http://ieeexplore.ieee.org/iel4/5726/15322/00711963.pdf?arnumber=711963

[18] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework: Part 1," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-02-16, July 2002. [Online]. Available: http://cs.gmu.edu/~kosecka/cs803/baker_simon_2004_1.pdf

[19] R. Szeliski, "Image mosaicing for tele-reality applications," in *WACV94*, 1994, pp. 44–53. [Online]. Available: http://citeseer.ist.psu.edu/szeliski94image.html

[20] Z. Zhu, E. Riseman, and A. Hanson, "Parallel-perspective stereo mosaics," in *International Conference on Computer Vision, Vancouver, Canada*, July 2001. [Online]. Available: ftp://vis-ftp.cs.umass.edu/Papers/zhu/ppsm-iccv01.pdf

[21] Z. Zhu, A. Hanson, H. Schultz, and E. Riseman, "Error characterization of parallel perspective stereo mosaics," in *ICCV Workshop on Video Registration, Vancouver, Cananda*, July 2001. [Online]. Available: http://www.cs.umass.edu/~zhu/zhuVideoReg2001.pdf

[22] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum, "Full-frame video stabilization." in *CVPR (1)*. IEEE Computer Society, 2005, pp. 50–57.

[23] J.-M. Frahm and M. Pollefeys, "RANSAC for (Quasi-)degenerate data (QDEGSAC)," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 453–460. [Online]. Available: http://ieeexplore.ieee.org/iel5/10924/34373/01640792.pdf

[24] S. Saripalli and G. S. Sukhatme, "Landing on a moving target using an autonomous helicopter," in *Proceedings of the International Conference on Field and Service Robotics*, July 2003. [Online]. Available: http://cres.usc.edu/pubdb_html/files_upload/335.pdf

[25] C. N. Taylor, "How to compute the closest point to the intersection of two or more lines in 3-d space," December 2005.

Appendices

Appendix A

User Study Schedule Composition

This appendix presents the tables used to structure the clip-view schedule for each user. As described in Section 4.1.5, we have structured this schedule in a way that will facilitate within and between subject, clip, and view comparisons, each user was assigned a controlled random ordering of the 16 clips presented using one of the four different views. For these controlled random orderings to comply, we needed to ensure that every clip and every view is seen an equal number of times per user as well as seen a progressively equal number of times by all users. We also needed to ensure that each clip-view was seen a progressively equal number of times across all users.

We enumerate all of the possible permutations of the four presentation views in the View Permutations table (Table A.1). Also, the number of times each permutation block is used in the User View-Blocks table (Table A.2) is tallied and listed in the "Times Used" relative to each permutation in Table A.1. The User View-Blocks Table lists the controlled random ordering of the four view permutation blocks that each user will see, which correspond to Table A.1. This ordering was controlled so as to ensure that no view-block was used twice by a single user, and that each view-block was used a progressively equal number of times, as indicated by the "Times Used" column of Table A.1.

Using these two tables, we ordered the pairing of the 16 clips that will be viewed per user using the view permutation and view-block orderings in Tables A.1 and A.2. The clips also needed to be in a controlled random order so that each view-clip combination was seen a progressively equal number of times across all of the users and also so that each user saw each clip once and in the random ordering. To accomplish this, we created the User View-Clip Schedule table (Table A.3) in conjunction with the View-Clip tallies table (Table A.4). The random ordering of each view-clip combination that each user saw had to preserve the progressively even tally shown in Table A.4.

Block#	1	2	3	4	Times Used
0	B	D	C	A	4
1	B	A	C	D	4
2	D	A	C	B	4
3	A	C	B	D	5
4	D	C	B	A	5
5	B	D	A	C	5
6	D	C	A	B	5
7	C	B	A	D	4
8	A	C	D	B	4
9	C	B	D	A	5
10	C	D	A	B	4
11	A	D	B	C	4
12	B	C	A	D	5
13	D	A	B	C	4
14	D	B	C	A	4
15	A	D	C	B	4
16	D	B	A	C	4
17	C	D	B	A	4
18	A	B	C	D	4
19	B	C	D	A	4
20	C	A	D	B	4
21	A	B	D	C	5
22	C	A	B	D	5
23	B	A	D	C	4
					104

Table A.1: View Permutations. A=Mosaic, B=Original, C=StableMosaic, D=Stable. There are a total number of 104 view permutations that will be seen by all users collectively.

Building the User View-Clip Schedule table, which outlines the view-clip ordering that each user will see, allowed the within and between subject, clip, and view comparisons initially desired.

User#	1	2	3	4
0	9	1	12	10
1	21	2	11	15
2	14	3	0	19
3	13	8	17	5
4	18	4	23	20
5	22	6	7	16
6	15	16	21	5
7	18	0	4	6
8	10	23	9	20
9	22	17	12	13
10	7	3	11	19
11	14	1	2	8
12	3	8	19	18
13	23	14	2	16
14	11	20	21	15
15	22	6	4	17
16	5	13	10	12
17	0	1	9	7
18	5	7	9	17
19	20	8	10	0
20	16	13	12	18
21	4	14	3	19
22	21	1	22	15
23	6	23	11	2
24	21	4	6	9
25	3	22	5	12

Table A.2: User View-Blocks. The row values are indices into the view permutation table.

	Block # 1				Block # 2				Block # 3				Block # 4			
User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	C.13	B.8	D.5	A.16	B.3	A.10	C.4	D.1	B.2	C.15	A.14	D.12	C.6	D.9	A.11	B.7
1	A.3	B.16	D.7	C.1	D.4	A.5	C.8	B.9	A.2	D.15	B.11	C.10	A.6	D.14	C.12	B.13
2	D.6	B.14	C.16	A.7	A.15	C.11	B.12	D.3	B.5	D.10	C.2	A.13	B.1	C.9	D.8	A.4
3	D.13	A.1	B.15	C.7	A.12	C.14	D.11	B.6	C.3	D.2	B.10	A.8	B.4	D.16	A.9	C.5
4	A.1	B.3	C.6	D.15	D.7	C.10	B.14	A.4	B.2	A.5	D.8	C.11	C.13	A.16	D.9	B.12
5	C.15	A.11	B.1	D.6	D.14	C.12	A.13	B.10	C.9	B.5	A.7	D.2	D.4	B.8	A.3	C.16
6	A.12	D.11	C.7	B.16	D.3	B.15	A.14	C.5	A.9	B.4	D.1	C.8	B.13	D.10	A.6	C.2
7	A.15	B.6	C.1	D.16	B.7	D.13	C.4	A.8	D.12	C.3	B.9	A.2	D.5	C.14	A.10	B.11
8	C.3	D.4	A.10	B.7	B.1	A.15	D.13	C.14	C.11	B.5	D.2	A.9	C.12	A.16	D.6	B.8
9	C.8	A.5	B.2	D.9	C.6	D.3	B.11	A.12	B.14	C.1	A.13	D.10	D.16	A.4	B.15	C.7
10	C.13	B.6	A.7	D.8	A.1	C.15	B.9	D.12	A.2	D.5	B.16	C.4	B.3	C.10	D.11	A.14
11	D.15	B.10	C.2	A.11	B.4	A.3	C.16	D.14	D.7	A.6	C.5	B.13	A.8	C.9	D.1	B.12
12	A.10	C.15	B.5	D.13	A.8	C.1	D.2	B.6	B.11	C.16	D.7	A.3	A.14	B.4	C.12	D.9
13	B.13	A.6	D.14	C.3	D.11	B.8	C.4	A.16	D.10	A.1	C.5	B.7	D.15	B.12	A.9	C.2
14	A.2	D.6	B.10	C.14	C.13	A.11	D.4	B.16	A.5	B.1	D.8	C.7	A.12	D.3	C.9	B.15
15	C.10	A.4	B.9	D.5	D.12	C.11	A.13	B.14	D.16	C.8	B.2	A.15	C.6	D.1	B.3	A.7
16	B.7	D.13	A.12	C.1	D.6	A.14	B.8	C.15	C.4	D.2	A.5	B.3	B.16	C.9	A.10	D.11
17	B.6	D.4	C.10	A.7	B.14	A.13	C.16	D.15	C.11	B.2	D.3	A.1	C.8	B.5	A.9	D.12
18	B.12	D.9	A.6	C.13	C.7	B.4	A.15	D.1	C.5	B.11	D.14	A.8	C.2	D.16	B.10	A.3
19	C.6	A.16	D.7	B.13	A.11	C.12	D.10	B.9	C.3	D.8	A.2	B.1	B.15	D.5	C.14	A.4
20	D.8	B.5	A.1	C.10	D.16	A.11	B.7	C.4	B.13	C.6	A.15	D.3	A.12	B.2	C.14	D.9
21	D.6	C.15	B.16	A.13	D.1	B.12	C.9	A.10	A.5	C.8	B.3	D.4	B.14	C.11	D.7	A.2
22	A.7	B.10	D.12	C.3	B.9	A.6	C.16	D.2	C.13	A.8	B.11	D.14	A.4	D.15	C.5	B.1
23	D.13	C.7	A.9	B.6	B.8	A.3	D.11	C.2	A.16	D.5	B.4	C.1	D.10	A.14	C.12	B.15
24	A.13	B.16	D.4	C.12	D.7	C.15	B.5	A.9	D.3	C.2	A.10	B.14	C.6	B.8	D.11	A.1
25	A.14	C.5	B.1	D.16	C.10	A.11	B.13	D.12	B.9	D.6	A.8	C.3	B.2	C.4	A.7	D.15

Table A.3: The view-clip schedule listed relative to the user number. Each view-clip is listed as ViewType.ClipNumber within the Blocks.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A:	7	6	6	6	6	6	7	7	7	7	7	6	7	7	6	6	104
B:	7	7	6	6	7	6	6	7	7	6	6	6	7	7	6	7	104
C:	6	7	7	7	7	7	6	6	6	7	6	7	6	6	7	6	104
D:	6	6	7	7	6	7	7	6	6	6	7	7	6	6	7	7	104
	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	

Table A.4: The tally of how many of each view-clip combination is seen.

Appendix B

User Study Material

This chapter contains the instructions (Section B.1) and questions (Section B.2) used in the user study.

B.1 User Study Instructions

Purpose of this Study

The purpose of this study is to provide you an opportunity to grade the relative effectiveness of the four different display methods, A, B, C, and D, that you will be presented with throughout this study. You will first be presented with a training on identifying red umbrellas as well as another training on using the system to select red umbrellas in the left screen and red spots in the right screen. Then you will be presented with the trials and requested to answer some general questions.

Before you begin, please first read and fill out the (1) Consent to be a Research Subject form, (2) Usability Test Clearance Form, and the (3) Pre-Training Questions.

Your Objectives

1. Your first objective will be to learn about the system as well as how to use it comfortably.
2. Once you feel comfortable with the training examples, your trials' objectives will then be to identify and select as many red umbrellas in the left screen and as many red spots in the right screen as you can.
3. Between each trial segment, you will be asked some questions about that segment that will appear on the right screen.

4. At the end of the trials, we ask that you please fill out the general Post-Trials questionnaire.

Training

To begin, you will see on the left screen one of four examples of what a red umbrella may look like in the video. Note that the red umbrella examples are highlighted with a red circle. This red circle will not appear in the video unless you make a selection, at which point the circle will be displayed only very briefly. Please view all four of these red umbrella examples by pressing any key on the keyboard to advance to the following ones.

At this point, you will also be presented with four separate 30 second training segments. Each segment will display aerial video to you using one of the four display methods, A, B, C, or D, in the left screen, and randomly generated spots in the right screen. Each of the four training segments are very similar to each of the 16 trial segments, except...

- The training segments are shorter than the trial segments.
- You can repeat each training segment as many times as you would like before moving on.
- Each training segment will contain only one red umbrella, whereas each trial segment may contain several red umbrellas or none at all.

As soon as you are ready to begin each training or trial segment, please press any key on the keyboard. The video will then begin playing in the left screen and the different colored spots will begin appearing in the right screen. Your objective will now be to do your best to identify and select the red umbrellas as they appear in the left screen while simultaneously selecting each red spot as it may appear in the right screen.

Selecting Red Umbrellas (Left Screen)

Once you believe that you have identified an object in the video that resembles a red umbrella in the video that is being played in the left screen, you can mouse left-click anywhere in the left screen to freeze that frame. Once the frame has been frozen, you can do the following:

1. Move your cursor over the object and then mouse left-click again to select it. Once you select an object, a red circle will very briefly be displayed around the object and then the video will again be displayed.
2. However, if you decide that there really is not a red umbrella in the freeze frame, then you can mouse right-click anywhere in the left screen to cancel the freeze frame, at which point the video will again be displayed.

Freezing the frame will not pause the video. The video will continuously play in the background throughout each segmenteven if you have frozen a frame. So the longer a frame is frozen, the more video you will be missing. After a frame is frozen with the first left-click, as soon as you either left-click again to select an object or right-click anywhere in the left screen to cancel, the frame will be unfrozen and the video will resume playing not from where it was frozen, but where it would have been if had it not been frozen at all.

Please note the following about selecting red umbrellas in the left screen:

- The current frame of the video will always have a green border around it. You can and should click on the red umbrellas that appear both inside or outside of the green framed area.
- You only need to select each umbrella once when you see it in the video; but, if you happen to see again a previously selected umbrella later on in the video segment, you should select it again. However, you are free to select each umbrella as many times as you would like as long as it does not keep you from selecting as many red spots as you can or other red umbrellas.
- If a red umbrella goes out of view and into the black area before you were able to click on it, you can and should still select the black area that you think the red umbrella would be in relation to the current frame.
- You are free to select or deselect spots in the right screen while the video is playing or when a frame is frozen in the left screen.

Selecting Red Spots (Right Screen)

Throughout each segment, different colored spots will be generated and displayed at random time intervals and at random places in the right screen. Anytime that a red spot appears, your objective will be to select it by mouse left-clicking on it. When you mouse left-click anywhere in the right screen, a circled white cross-hair will appear centered on your selection. It is important that the center of these white cross-hairs be touching anywhere within a red spot to count. If you notice that your selection needs adjusted, you are free to adjust that selection by left-clicking as many times as you need to correctly place the cross-hairs over a red-spot until the spots move.

Please note that there will not always be a red spot generated to select. If you need to remove a selection in the right screen, you can mouse right-click anywhere in the right screen to cancel your selection and erase the cross-hairs. Also, the very last spot of each segment does not count. So if you were about to click on it and missed it, no worries.

In Short

In both left and right screens, the mouse left-click is always used to select (select the frame of video to freeze, select the red umbrella, or select the red spot) and the mouse right-click is always used to cancel (cancel the frame freeze, or cancel the spot selection).

The Trials

There will be 16 unique trials presented to you. Each trial will use a different video segment that will be presented to you using one of the four different display methods, A, B, C, or D. Each trial will last about 1 minutes. Start each trial by pressing any key on the keyboard. Do the best that you can at identifying as many red umbrellas and red spots, but remember that identifying and selecting the red umbrellas will be more important than selecting the red spots.

At the end of each segment, please use the screen on the right to answer the general questions for that segment that will appear on your right screen. Each segment will by followed by the same set of on-screen questions, so please read them in the printed questionnaire page titled Post Trial Questions Sheet before you begin the trials so that you can be thinking about them while completing each trial.

You are also encouraged to record any notes or impressions that you had about each segment or display method after each trial on the provided Display Methods Notes Sheet. This sheet has the trial number listed as well as which corresponding display method was used for each trial.

You are also welcome to take a break as needed between each trial. After you answer the post trial questions on the right screen and are ready to continue, please press any key on the keyboard to begin the next trial.

Follow-up Questions

After you have completed all of the 16 trials, please answer the questions about your preferences and general feedback on the provided Follow-up Questions Sheet.

Important

If at any time during this study you begin to experience any physical discomforts or feel the need to withdraw from the study for any reason, we encourage you to do so immediately.

Thank you very much for your participation.

B.2 User Study Questions

Pre-Training Questions

Please check only one choice per question.

1. Do you have any physical limitations that may possibly affect your performance in this user study (e.g. color-blindness, impaired motor skills, etc.)?
O Yes
O No

2. How experienced do you feel that you are with using computers?
O Expert
O Average
O Novice

3. How experienced do you feel that you are with wilderness search and rescue tasks?
O Expert
O Average
O Novice

4. How experienced do you feel that you are with tasks involving searching for things on the ground from high up above in the air (aerial searching tasks)?
O Expert
O Average
O Novice

5. How familiar are you with the research related to this study?
O Never heard of any of it before this user-study.
O I have heard about the research, but I have never seen any of the video display methods before.
O I have never heard about the research, but I have seen some of these video display methods before.
O I know about the research, and I have seen the video display methods before.

6. How familiar are you with others' preferences of the display methods that you will be presented with in this study?
O I know nobody else's preferences.
O I know somebody else's preferences.
O I know a couple other people's preferences.
O I know many peoples' preferences.

Post Trial Questions (to be aware of)

Please do not answer these here, they will be presented to you after each trial, so please keep them in mind while you are completing each trial.

1. How many Red Spots do you think you missed in that segment?
O 0
O 1
O 2
O 3 or More

2. How many Red Umbrellas do you think you missed in that segment?
O 0
O 1
O 2
O 3 or More

3. That segment's DISPLAY METHOD was ________ than the DISPLAY METHOD before it.
O Easier
O About the same
O Harder

Follow-up Questions

Please check only one choice per question; however, you may pick multiple choices, but please state your assumptions.

1. Which DISPLAY METHOD was the most comfortable for you to watch overall?
O No differences.
O A
O B
O C
O D

2. Which DISPLAY METHOD was the least straining (easiest) to watch overall?
O No differences.
O A
O B
O C
O D

3. Which DISPLAY METHOD was the most straining (hardest) to watch overall?
O No differences.
O A
O B
O C
O D

4. Which DISPLAY METHOD made you feel the least oriented overall?
O No differences.
O A
O B
O C
O D

5. Which DISPLAY METHOD made you feel the most oriented overall?
O No differences.
O A
O B
O C
O D

6. Which DISPLAY METHOD do you feel like you could watch the longest overall?
O No differences.
O A
O B
O C
O D

7. Which DISPLAY METHOD would be your preference in a real search situation?
O No differences.
O A
O B
O C
O D

8. Please number with 1-4 each DISPLAY METHOD according to your preference, 1 being your most preferred and 4 being your least preferred.
A ____
B ____
C ____
D ____