Unmanned Systems, Vol. 1, No. 2 (2013) 277–296
© World Scientific Publishing Company
DOI: 10.1142/S2301385013500088

UNMANNED SYSTEMS
http://www.worldscientific.com/worldscinet/us

Enhanced UAS Surveillance Using a Video Utility Metric

Peter C. Niedfeldt*,§, Brandon T. Carroll*, Joel A. Howard†,
Randal W. Beard*, Bryan S. Morse‡, Stephen Pledgie‡

*Electrical and Computer Engineering Brigham Young University, Provo, UT 84602, USA

†Computer Science, Brigham Young University Provo, Utah 84602, USA

‡Autonomous Systems Group Mosaic ATM, Inc., Leesburg, Virginia 20175, USA

A successful mission for an Unmanned Air System (UAS) often depends on the ability of human operators to utilize data collected from onboard imaging sensors. Many hours are spent preparing and executing flight objectives, putting a tremendous burden on human operators both before and during the flight. We seek to automate the planning process to reduce the workload for UAS operators while also optimizing the quality of the collected video stream. We first propose a metric based on an existing image utility metric to estimate the utility of video captured by onboard cameras. We then use this metric to not only plan the UAS flight path, but also the path of the camera's optical axis projected along the terrain and the zoom level. Since computing an optimal solution is NP-hard and therefore infeasible, we subsequently describe a staged sub-optimal path planning approach to autonomously plan the UAS flight path and sensor schedule. We apply these algorithms to precompute UAS and sensor paths for a surveillance mission over a specified region. Simulated and actual flight test results are included.

Keywords: Path planning; video quality; probabilistic search; TTP.

1. Introduction

Unmanned Air Systems (UASs) offer a platform to strategically position sensors such as electro-optical or infrared cameras, communication equipment, etc. to perform a variety of tasks. The breadth of possible UAS applications and the frequency of their utilization continues to increase. Developing autonomous UAS systems to complete these tasks will reduce the burden and risk for human operators who would otherwise perform them. Current civilian and military applications include Wilderness Search and Rescue,[1–5] law enforcement,[6] surveillance,[7] reconnaissance,[8] and target tracking.[9–11]

A critical need of UAS research is to further reduce the workload for UAS operators. Current UAS technology requires multiple human operators to both plan and fly a UAS, while simultaneously controlling the onboard sensors.[10,12–14] While some aspects of the planning process have been automated, we have not found a method that simultaneously plans all flight and sensor paths. By autonomously planning the UAS flight path, the sensor gimbal angles, and the sensor resolution, UAS operators would only need to focus their attention on the sensor display and make high-level decisions. In this paper, we develop an algorithm to simultaneously plan the flight path and the sensor schedule for surveillance over a region of interest.

Mission goals and other high-level decisions are often based on sensor data collected during the flight. To provide usable video to users, an automated planner predicts and plans paths resulting in high-utility sensor measurements. We define high-utility measurements for an EO/IR camera to be video where the operator can perform detection/recognition/identification (DRI) with a high probability of success. Several image utility metrics have been developed, such as in;[15–18] however, we are unaware of existing metrics that estimate the probability of DRI for aerial video.

Received 28 December 2012; Revised 23 July 2013; Accepted 5 August 2013; Published 12 September 2013. This paper was recommended for publication in its revised form by editorial board member Ping Tan.
Email Address: §pcniedfeldt@gmail.com

The state-of-the-art method for estimating the probability of DRI of an image is the Targeting Task Performance (TTP) metric developed by Vollmerhausen and Jacobs.[17] We extend TTP to estimate video utility by considering spatiotemporal effects, such as blur due to motion. The resulting metric is the video TTP (VTTP) metric and is used to both predict and measure video utility to improve the operator's viewing experience. Prior to flight, optimal flight and sensor paths can be found by maximizing the predicted VTTP metric over the path. During or after flight, the measured utility from collected video can then be computed using the VTTP metric.

The planning algorithms presented in this paper are formulated for surveillance applications but can be extended to probabilistic search and other applications. All UAS probabilistic search missions require path planning. These missions do not have a specified goal state; rather, we desire to maximize the probability of DRI over the UAS path and sensor schedule. A common solution to this NP-hard problem is to use a multi-step look-ahead algorithm to plan the UAS path[3,5,19–21] Unfortunately, this approach alone is not computationally feasible when the dimensionality of the search space includes position of the UAS, gimbal pointing, and sensor zoom level.

Quigley *et al.* present a semi-autonomous search method to locate targets by fixing the gimbal azimuth and elevation and flying the UAS manually until the target is located. After localizing the target, they switch to fully autonomous target tracking algorithms that orbit the target.[9] We extend their work to further reduce the operator workload by autonomously planning both the UAS flight path and the sensor schedule to acquire high-utility video.

Waharte *et al.* develop algorithms that allow a hovercraft to vary the number of terrain cells in the sensor field of view (FOV) by adjusting its altitude, effectively controlling the zoom during the search. They then use a greedy algorithm to control both the sensor resolution and the hovercraft north–east coordinates.[21] We extend their work by allowing our fixed wing aircraft to gimbal the sensor while planning the flight path and the zoom level.

Chung and Burdick propose five methods of path planning for searching applications. Two of interest are the Saccadic search and the Drosophila-inspired search, where the UAS is commanded to point the camera at the locations of highest probability while either ignoring or accounting for gimbal slew-rate constraints, respectively.[20] We improve on these concepts by allowing the planner to maximize video utility along the UAS and sensor paths between high-reward locations.

In,[22] the authors present a solution to autonomously plan a flight path and a gimbal control strategy. When determining the flight path, the control options are constrained to either fly straight or turn left or right at a fixed rate. A receding horizon approach is used to search for the best path over the look-ahead window. The gimbal control strategy is to point at the nearest user-defined object of interest. In the current paper, we allow greater control authority by specifically planning a gimbal schedule and allowing the UAS flight path to optimize over more than a fixed set of paths. In addition, we also control the sensor zoom level and utilize the VTTP metric instead of the TTP metric.

We have improved the sensor path planner to use a technique similar to the chain-based method to plan 2D flight paths developed by Argyle *et al.* in.[23] We extend their work by planning a sensor path with dynamic link-lengths to allow the sensor FOV to dwell in regions of interest. We then plan the UAS path and the zoom schedule according to the sensor path. We first presented the concept of the staged-path planner in,[24] but only in relation to an abstract probability of detection metric. This paper elaborates and extends the research presented in[24] to explain our complete methodology. Specifically, we introduce and describe the VTTP metric used when generating the flight path and sensor schedule.

In this paper we extend the state-of-the-art in four ways:

(i) A novel video utility metric is presented that extends well-established image utility metrics.
(ii) The theoretical, optimal parameterized UAS flight path and sensor schedule that maximizes the reward or utility over the path is formulated.
(iii) The theoretical path planning problem is reformulated into three planning stages that are suboptimal but computationally feasible.
(iv) The planning algorithm in[23] is extended to account for dynamic link-lengths used to plan a sensor path.

We apply the method to a surveillance mission to provide higher-utility video footage of the environment. We begin by describing the problem and our notation in Sec. 2. In Sec. 3, we describe the VTTP video quality metric. Section 4 presents the path planning and scheduling algorithms. We plan a flight path and sensor schedule in simulation and fly the path in hardware, presenting our results in Sec. 5.

2. Problem Formulation and Notation

In this paper, the objective is to survey a given location and the surrounding terrain using a UAS with a gimbaled imaging sensor with zoom capabilities. Let the position and attitude of the UAS and the payload configuration be represented by the state vector x, which evolves according to the dynamic equation $\dot{x} = f(x, u)$ given $x(0) = x_0$ and the control input $u(t)$. We parameterize the control input $u(t)$, or the desired UAS path and sensor schedule, by a set of N waypoints $\Theta = (\Theta_s^\top, \Theta_u^\top, \Theta_\zeta^\top)^\top$ where

$$\begin{aligned}\Theta_s &= (n_{s,1}, e_{s,1}, \ldots, n_{s,N}, e_{s,N})^\top,\\ \Theta_u &= (n_{u,1}, e_{u,1}, \ldots, n_{u,N}, e_{u,N})^\top,\\ \Theta_\zeta &= (\zeta_1, \ldots, \zeta_N)^\top,\end{aligned}$$

and where $n_{s,i}$ and $e_{s,i}$ are the ith north and east sensor waypoints that indicate where the sensor optical axis should intersect the terrain, $n_{u,i}$ and $e_{u,i}$ are the ith UAS north and east waypoints, and ζ_i is the ith focal length waypoint indicating the zoom level of the sensor at the waypoint.

With an abuse of notation, we let $u(\Theta)$ indicate the control input that will be needed to maneuver the vehicle and sensor through the waypoints designated by Θ over a fixed look-ahead horizon T. We define the control input such that the closed-loop UAS dynamics follow a constant roll angle between waypoints, resulting in the UAS flying along a connecting arc. The heading at each waypoint is deterministic if the heading at the previous waypoint is known. Thus, setting the initial heading for the first waypoint in the path recursively determines the headings for all the following waypoints. To calculate the heading ψ_i at $\Theta_{u,i}$, we construct a circle that passes through both $\Theta_{u,i-1}$ and $\Theta_{u,i}$ and is tangent to the initial heading ψ_{i-1} at $\Theta_{u,i-1}$. Figure 1 illustrates the geometry of this problem.

Using geometric principles, it can be shown that the change in heading $\Delta\psi_i$ after traveling from $\Theta_{u,i-1}$ to $\Theta_{u,i}$ is given by $\Delta\psi_i = 2\varepsilon$, where the angle ε is calculated as the angle between the heading ψ_{i-1} and the vector $\Theta_{u,i} - \Theta_{u,i-1}$. The UAS position and pose can be determined by calculating

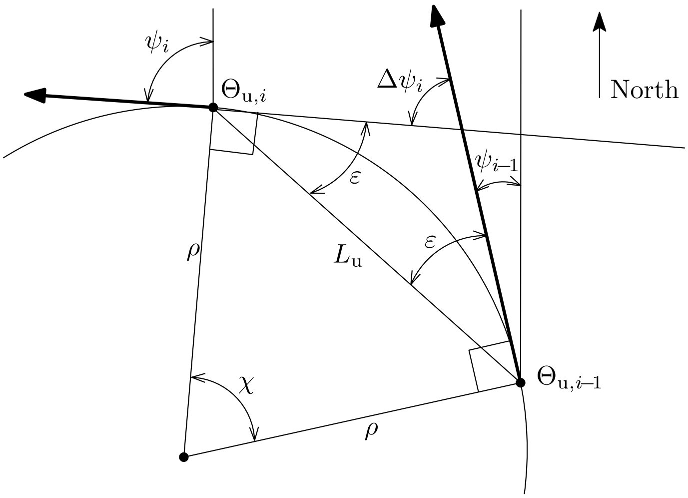


Fig. 1. Geometry used to calculate the heading ψ_i at UAS waypoint $\Theta_{u,i}$ given a previous heading ψ_{i-1} at UAS waypoint $\Theta_{u,i-1}$.

the distance traveled along the arc between $\Theta_{u,i-1}$ and $\Theta_{u,i}$ of the circle with radius $\rho = \sqrt{\frac{L_u}{2(1-\cos(\chi))}}$, where $\chi = \Delta\psi_i$ is the angle between the radii of the circle to each waypoint and $L_u = \|\Theta_{u,i-1} - \Theta_{u,i-1}\|$. We assume that the UAS flies at a fixed altitude to maximize the endurance of the vehicle.[25]

The desired sensor state trajectory is given by linearly interpolating between each waypoint in Θ_s. Note that the sensor path is not parameterized using the gimbal angles, as linear interpolation using that parameterization creates inherently nonlinear motion of the sensor point of interest along the terrain, resulting in optical flow that is unpleasant for the operator to view. The desired zoom state is a linear interpolation between the waypoints defined by Θ_ζ. When planning, we consider either the continuous range of focal lengths $[\zeta_{\min}, \zeta_{\max}]$ or a fixed number of discrete zoom levels.

Let $\xi(t, \tau, \Theta)$ be the family of solutions to $\dot{x} = f(x, u)$ with initial condition $\xi(t, 0, \Theta) = x(t)$ and parameterized by Θ. These paths are planned over the finite look-ahead window $\tau \in [0, T]$, where T is the look-ahead horizon. In other words, the predicted UAS states at time $t + \tau$ when flying a path described by Θ are given by the function $\xi(t, \tau, \Theta)$. The look-ahead horizon is a function of the number of waypoints N and the fixed time between each waypoint Δt, such that $T = N\Delta t$.

Let Z be a set of discrete terrain points, where each $z \in Z$ is an inertially defined north, east, down position that specifies a point in $\mathbb{R}^3$ that resides on the terrain surface. Let z_{base} be the terrain location representing the center of the region of interest. We define the set FOV(x) to be the set of terrain points z that are in the field of view given the state vector x. We desire to know the probability that an operator will be able to correctly perform DRI tasks for each element in FOV(x). Since we cannot determine the true probability of DRI, we model it using the VTTP metric, which will be formally defined in Sec. 3.

The VTTP metric estimates the probability of an operator successfully performing a DRI task if an object were located at terrain point z. The specific VTTP value is determined by the state vector x and a set of parameters $\mathcal{P}$ that include the terrain elevation model, the dimensions of potential objects of interest, and other parameters that will be discussed in Sec. 3. Therefore, the VTTP value is given by the function VTTP$(z, x, \mathcal{P}) \in [0, 1]$. Section 2.1 introduces an evolving reward map to denote regions of interest in the map. In Sec. 4, we describe a path planning algorithm that maximizes the reward weighted by VTTP$(z, x, \mathcal{P})$ along a path to plan the UAS flight path and sensor schedule to observe regions of interest with higher video utility.

2.1. *Reward map*

When performing surveillance missions there may be regions of the terrain that are more important to observe than others. We desire to view these regions of interest frequently and with higher quality to provide better utility to the operator. This capability is enabled by an evolving reward map, which represents the reward available to the system if it observes the terrain point z. The reward map is denoted as $r(t, z, x)$, which indicates the available reward at the terrain point z at time t. The reward map is updated using sensor measurements according to the current state vector x.

In this paper, and without loss of generality, we define the evolution of $r(t, z, x)$ as

$$\dot{r}(z, x) = \omega(z) e^{-\frac{1}{2}(z - z_{\text{base}})^\top \Sigma_r^{-1} (z - z_{\text{base}})}, \tag{1}$$

or a two-dimensional Gaussian distribution centered on a region of interest or home base z_{base}, with covariance matrix Σ_{r} weighted by $\omega(z)$, where each $\omega(z) > 0$. The weighting factor $\omega(z)$ is defined such that each terrain point z can have a unique priority level. Again without loss of generality, we choose a $\omega(z)$ that assigns one of two reward levels $\overline{\omega}$ or $\underline{\omega}$ to each location z to represent either a high or low reward area, respectively. Note that the reward grows at a constant rate for each terrain point z.

The reward map is updated after each sensor measurement. We want to capture how well we view the terrain points z in the FOV, so we use the VTTP metric to scale the reward in the FOV. This update is expressed as

$$r^+(t, z, x) = \begin{cases} r^-(t, z, x) & z \notin \text{FOV}(x) \\ r^-(t, z, x) - \gamma_{\text{r}} r^-(t, z, x) \text{VTTP}(z, x, \mathcal{P}) & z \in \text{FOV}(x), \end{cases} \tag{2}$$

where the reward is unchanged when the point z is not in the FOV, and decreases by γ_{r} times the observed reward when z is in the FOV, where $0 < \gamma_{\text{r}} \leq 1$ and is a tuning parameter describing the maximum amount of reward that can be collected per frame.

Figure 2 represents the reward over the test flight area used in Sec. 5. The locations z that are part of the road network are defined to be high reward. The figure displays an example of what a reward map initialized to zero looks like after evolving for 20 s. Note that the specific parameters in this section and all subsequent parameters are summarized in Table A.1 in Appendix A. Also note that terrain coordinates are described using the Universal Transverse Mercator geographic coordinate system, based in meters.

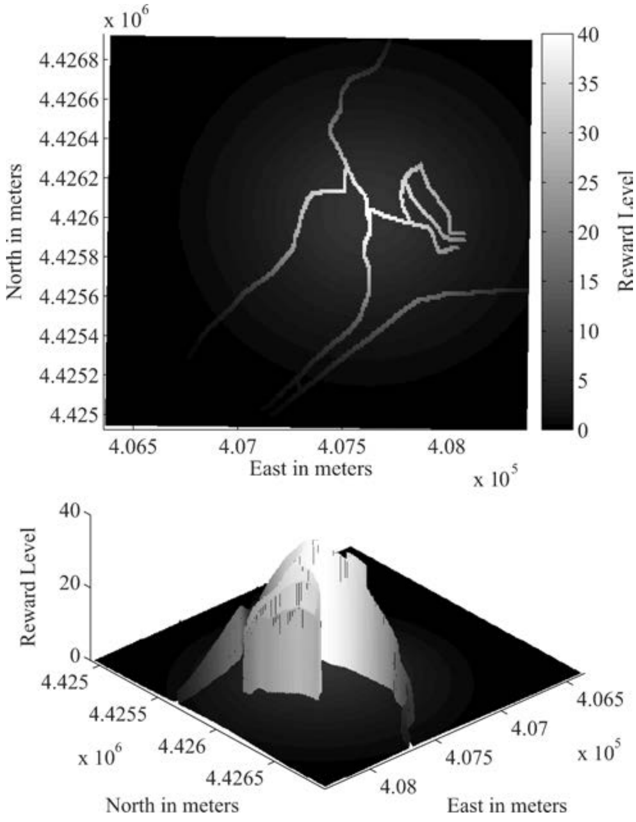


Fig. 2. Top and side view of the evolving reward function after 20 s.

2.2. *Optimal UAS and sensor planning*

Given the reward map r described above, we want to find the parameterized path Θ that maximizes the reward-weighted VTTP metric collected along the path. The reward-weighted VTTP metric collected by the sensor at time $t + \tau$ is given by

$$R(t, \tau, \Theta, \mathcal{P}) = \sum_{z \in \text{FOV}(\xi(t,\tau,\Theta))} r(t + \tau, z, \xi(t, \tau, \Theta)) \text{VTTP}(z, \xi(t, \tau, \Theta), \mathcal{P}) dz. \tag{3}$$

In other words, $R(t, \tau, \Theta, \mathcal{P})$ describes the amount of instantaneous reward observed within the sensor field of view at time $t + \tau$.

The optimal path that results in the maximum reward collected over the look-ahead window $0 \leq \tau \leq T$ is given by

$$\Theta^* = \arg\max_{\Theta} \int_{\tau=0}^{T} R(t, \tau, \Theta, \mathcal{P}) d\tau. \tag{4}$$

Unfortunately, this problem is NP-hard and can only be solved in trivial cases. In Sec. 4, we present a suboptimal strategy for approximately solving this problem.

3. Video Utility Metric

We desire to model the probability of an operator successfully performing DRI tasks using a video stream. For this purpose we propose the VTTP metric, which is a measure of video utility and relates to the probability of DRI. In other words, the VTTP metric quantifies the ability of an operator to perform a particular DRI task using the sensor video stream. We calculate the VTTP values for the points on the terrain map that fall within the field of view of the camera. These VTTP values then determine how much reward is collected from the corresponding points on the reward map.

For each terrain point visible in a video frame, the VTTP calculation consists of a TTP calculation modified by additional components. These additional components account for various facets of the video imaging and the viewing process that TTP does not take into account. We develop two components for the VTTP metric: the motion M and the observer viewing model O. The motion component of the VTTP metric discounts the probability of DRI based on the motion blur induced by the egomotion of the sensor. The observer viewing model discounts the probability of DRI when the object is far from the video frame center. We also develop a method for estimating the contrast at a terrain point location, which is needed to compute the TTP metric.

Given a flight path and sensor schedule the predicted VTTP value can be calculated for each terrain point in the FOV as if a hypothetical object existed at that point. The ability to predict the VTTP value given a planned path allows our planner to search for a path that maximizes the expected reward for the video, thereby improving the operator's ability to detect objects within the video stream during flight. Once a path has been executed, we use the VTTP metric to evaluate how well areas on the map were actually seen. Future work will use this evaluation to close the loop for the next stage of planning so that poorly covered areas can be revisited in real time.

When planning a path, we calculate the predicted VTTP values for points in the field of view as

$$\mathrm{VTTP_p}(z,\xi,\mathcal{P}) = \mathrm{TTP_p}(z,\xi,\mathcal{P})M_\mathrm{p}(\xi,\mathcal{P})O_\mathrm{p}(\xi,\mathcal{P}), \quad (5)$$

where $\xi(t,\tau,\Theta)$ is the predicted telemetry and the subscript 'p's denote predicted values. Once the video has been obtained from following a path, we calculate the actual VTTP as

$$\mathrm{VTTP_a}(z,x,\mathcal{P}) = \mathrm{TTP_a}(z,x,\mathcal{P})M_\mathrm{a}(v)O_\mathrm{a}(x,\mathcal{P}), \quad (6)$$

where $x(t)$ is the actual telemetry, v is the video stream, and the subscript 'a's denote actual values. Each of the components of the VTTP metric in Eqs. (5) and (6) requires certain types of knowledge about the situation. Table 1 summarizes the knowledge that is needed to compute each

Table 1. Requirements to calculate each component of VTTP.

Component	Elevation map	Predicted telemetry	Actual telemetry	Captured video	Object dimensions	Sun position	Ambient and direct lighting	Viewing model
Predicted								
$\mathrm{TTP_p}$	•	•			•			
Lighting (L_p)	•					•	•	
Motion (M_p)	•	•						
Observer model (O_p)		•						•
Actual								
$\mathrm{TTP_a}$	•		•		•			
Motion (M_a)				•				
Observer model (O_a)			•					•

component. The remainder of this section describes the various components comprising the VTTP metric. The TTP image metric and the underlying contrast estimation are described in Sec. 3.1. In Secs. 3.2 and 3.3 we describe the effects of motion and human viewing tendencies. Finally, in Sec. 3.4 we begin the model validation process by comparing actual and predicted VTTP values during a flight and by conducting a small user study.

3.1. *Targeting task performance*

The Targeting Task Performance (TTP) metric is an image utility metric developed by Vollmerhausen and Jacobs as an improvement over Johnson's criteria.[17] The TTP metric estimates the probability of DRI for humans viewing static images. It accounts for the size of the object in the FOV and considers the contrast between the object and the background over all relevant spatial frequencies, using either EO or IR imagery. Because the TTP metric was designed for still images, it does not take into accountthe spatiotemporal nature of video. We build upon the foundation laid by the TTP metric to develop the VTTP metric.

We use the notation $\mathrm{TTP}(z,x,\mathcal{P})$ to denote the TTP value of a hypothetical object located at the terrain point z given the UAS states x and a subset of the parameters $\mathcal{P}$, including the dimensions of the object of interest, a contrast model, and empirically determined parameters as explained in.[22] When searching for an unknown object, we calculate the TTP value according to the dimensions of the smallest possible object that the operator wishes to detect. The TTP metric can be calculated in four main steps. In the first step, we calculate the viewed area of the object of interest in the image frame assuming it were at the terrain point z, denoted $A(z, x, \mathcal{P})$, where $\mathcal{P}$ contains an estimate of the dimensions of the object of interest.

Second, the contrast $\Gamma(z)$ of the object with respect to the background is estimated. In[22] we compute the relative contrast of an object of interest with respect to the background using the Phong Illumination model. Unfortunately, this requires making numerous assumptions about the object and the background's lighting and texture characteristics, which may not be known. In an effort to reduce the number of uncertain parameters, we propose using the lighting information $\Lambda(z)$ to predict the contrast in an image. This method requires no prior knowledge of the appearance model of the object or background and is described in detail in the Predicted Contrast section that follows.

In the third step we compute the TTP integral proposed in.[17] We use the Minimum Resolvable Contrast curve $\mathrm{MRC}(\kappa, \mathcal{P})$ to quantify the human eye and the sensor's ability to detect contrast at a spatial frequency κ. An exact computation of $\mathrm{MRC}(\kappa, \mathcal{P})$ is given in.[26] However, we approximate $\mathrm{MRC}(\kappa, \mathcal{P})$ using a parabola as described in[22] given by

$$\mathrm{MRC}(\kappa, \mathcal{P}) = \exp\left(\frac{-a_2 \pm \sqrt{a_2^2 - 4a_1(a_3 - \kappa\measuredangle_{\mathrm{pixel}})}}{2a_1}\right),$$

where a_1, a_2, and a_3 are parameters defining the curve and $\measuredangle_{\mathrm{pixel}}$ is the field of view of a single pixel. The resulting TTP integral is given by

$$\Phi(z, x, \mathcal{P}) = \int_{\kappa_{\mathrm{low}}}^{\kappa_{\mathrm{cut}}} \left[\frac{\Gamma(z)}{\mathrm{MRC}(\kappa, \mathcal{P})}\right]^{\frac{1}{2}} d\kappa, \tag{7}$$

where κ_{low} and κ_{cut} are the lower and upper bounds of relevant spatial frequencies, and $\Gamma(z)$ is the contrast between the object and the background.

The final step in computing the TTP metric is to estimate the probability of DRI according to a logistics curve. The parameters of the logistics curve are empirically determined according to both the type of task being performed and the type of imaging system. The TTP value is given by

$$\mathrm{TTP}(z, x, \mathcal{P}) = \frac{(N_{\mathrm{resolved}}/N_{50})^E}{1 + (N_{\mathrm{resolved}}/N_{50})^E}, \tag{8}$$

where

$$E = b_1 + b_2^{(N_{\mathrm{resolved}}/N_{50})}$$

and

$$N_{\mathrm{resolved}} = \frac{\sqrt{A(z, x, \mathcal{P})}\,\Phi(z, x, \mathcal{P})}{s_{\mathrm{d}}(z, x)},$$

and where $A(z, x, \mathcal{P})$ is the viewed area of the object of interest, $s_{\mathrm{d}}(z, x)$ is the current slant range or Euclidean distance from the UAS to the terrain point, and N_{50} is an empirically determined factor that represents the average number of cycles necessary for a group of analysts to correctly identify objects 50% of the time. The coefficients b_1 and b_2 are statistically determined from laboratory and/or field data.

3.1.1. *Predicted contrast*

A key element of the TTP calculation is to determine the contrast between an object of interest and the background. Calculating the contrast requires prior knowledge of the coloring and reflective properties of the object of interest. Terrain reference imagery can then be used to predict the contrast of an object against known, and presumably static, terrain. However, one purpose of the VTTP metric is to predict video quality, even when *a priori* information about objects within the scene is unavailable.

We predict the contrast using information about the scene lighting to avoid relying on unknown target characteristics. As Ward describes in[27], higher visual contrast can be achieved in brighter lighting conditions than in darker conditions. We thus use a scene lighting model to predict object/background contrast, i.e., $\Gamma(z) \approx \Lambda(z)$. We acknowledge that using the predicted contrast based on lighting, rather than the actual contrast, may reduce the accuracy of the TTP metric. However, without knowledge of object coloring, this at least gives us an estimate that is proportional to the actual contrast.

There are several ways to model scene lighting, with varying complexity and prior knowledge required. Given a terrain map with reasonable resolution the improvements provided by more complex methods become marginal. We use a simple Lambertian lighting model that considers both ambient lighting as a result of cloud and atmospheric scattering and direct lighting as a result of sunlight falling directly on a given area of the surface. We require the user of the system to set a parameter $\alpha \in [0, 1]$ to indicate the relative mix of ambient and direct lighting at the time of flight, which is essentially an estimate of how cloudy the day is. Setting $\alpha = 1$ indicates complete cloud cover, while setting $\alpha = 0$ denotes a clear day with full sunlight. However, note that there is always some ambient lighting due to atmospheric scatter, so even on a clear day a small ambient component should be used.

Using this model, the illumination Λ of each terrain point z is thus calculated as a weighted mix of direct and ambient lighting,

$$\Lambda(z) = (1 - \alpha)D(z) + \alpha, \tag{9}$$

Fig. 3. Orthoimagery (left), orthoimagery with the lighting map overlaid (middle), and the lighting map $\Lambda(z)$ (right) for one of our test areas. The lighting map was computed at 8:20 a.m. on September 15, 2011, near Eureka, UT.

where $D(z)$ is the amount of direct lighting per unit area of the terrain surface. The direct lighting is dependent on the angle made between the direction of incoming sunlight ℓ and the terrain normal $\eta(z)$, subject to terrain shadowing. It is calculated as

$$D(z) = \begin{cases} 0 & \text{If } z \text{ is occluded from sun} \\ \ell \cdot \eta(z) & \text{Otherwise.} \end{cases}$$

To calculate the lighting direction ℓ and resulting shadow locations, we assume a known or estimated position of the sun. This can be gathered from readily available astronomical data, such as from the NOAA Solar Position Calculator.[28] If the sun does not move significantly during the time in which the video is collected, the lighting for each point in the area can be precomputed. A sample precomputed lighting map is shown along with the orthoimagery for the area in Fig. 3.

3.2. *Motion*

Motion of the video sensor causes blur in the video and makes objects more difficult for the human eye to track. Because of this, the utility of the video is affected by the gimbaling of the sensor and the path flown by the UAS. The motion component of the VTTP metric calculates the amount of motion in the image plane in pixels per second and adjusts the VTTP metric by a multiplicative factor accordingly. Since the amount of blur is proportional to the amount of motion in the image plane, the multiplicative factor drops off linearly from 1 to 0 as the motion increases from 0 to a threshold. This section shows how to compute both the predicted motion factor M_p and the actual motion factor M_a. We assume the use of an EO video camera for this derivation, but it applies to other types of video sensors as well.

3.2.1. *Predicted motion*

To predict the value of M_p, we calculate the apparent motion of 3D points as projected into the image plane of the camera. This calculation, known as a motion field, requires knowledge of the predicted state vector $\xi(t, \tau, \Theta)$ as well as the terrain elevation model and camera parameters from $\mathcal{P}$. We compute the motion field by taking the time derivative of the computer graphics equation that projects the terrain points into the camera's image plane,

$$\begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} = CPR_\mathrm{b}^\mathrm{g} R_\mathrm{v}^\mathrm{b} T_\mathrm{i}^\mathrm{v} \begin{bmatrix} z_\mathrm{n} \\ z_\mathrm{e} \\ z_\mathrm{d} \\ 1 \end{bmatrix}, \tag{10}$$

where $(\frac{z_x}{z_z}, \frac{z_y}{z_z})$ is the coordinate of the point in the image plane, C is the camera matrix, P is the projection matrix, R_b^g is the rotation matrix from the body frame to the gimbal frame, R_v^b is the rotation matrix from the vehicle to the body frame, T_i^v is the translation matrix from the inertial to the vehicle frame, and $(z_\mathrm{n}, z_\mathrm{e}, z_\mathrm{d})$ is the inertial north, east, down coordinate of the terrain point. The camera and projection matrices are defined as

$$C = \begin{bmatrix} \frac{\zeta}{\mathrm{dim}_x} & 0 & o_x \\ 0 & \frac{\zeta}{\mathrm{dim}_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

where ζ is the focal length, (o_x, o_y) is the coordinate of the camera's origin, and dim_x and dim_y are the width and height of a pixel on the camera's sensor. The projection matrix rearranges the axes from NED coordinates to X/Y image coordinates. The motion vector of the terrain point in the image plane (m_x, m_y) is found by taking the derivative with respect to time of the point's image plane coordinate,

$$\begin{bmatrix} m_x \\ m_y \end{bmatrix} = \frac{d}{dt}\left(\frac{1}{z_z}\begin{bmatrix} z_x \\ z_y \end{bmatrix}\right) = \frac{1}{z_z^2}\begin{bmatrix} z_z\dot{z}_x - \dot{z}_z z_x \\ z_z\dot{z}_y - \dot{z}_z z_y \end{bmatrix}, \tag{11}$$

where $\dot{z}_x$, $\dot{z}_y$, and $\dot{z}_z$ are obtained by time differentiating Eq. (10),

$$\begin{bmatrix} \dot{z}_x \\ \dot{z}_y \\ \dot{z}_z \end{bmatrix} = (\dot{C}PR_{\mathrm{b}}^{\mathrm{g}}R_{\mathrm{v}}^{\mathrm{b}}T_{\mathrm{i}}^{\mathrm{v}} + CP(\dot{R}_{\mathrm{b}}^{\mathrm{g}}R_{\mathrm{v}}^{\mathrm{b}}T_{\mathrm{i}}^{\mathrm{v}} + R_{\mathrm{b}}^{\mathrm{g}}\dot{R}_{\mathrm{v}}^{\mathrm{b}}T_{\mathrm{i}}^{\mathrm{v}} + R_{\mathrm{b}}^{\mathrm{g}}R_{\mathrm{v}}^{\mathrm{b}}\dot{T}_{\mathrm{i}}^{\mathrm{v}}))\begin{bmatrix} z_{\mathrm{n}} \\ z_{\mathrm{e}} \\ z_{\mathrm{d}} \\ 1 \end{bmatrix}.$$

Note that we assume that the terrain point $(z_{\mathrm{n}}, z_{\mathrm{e}}, z_{\mathrm{d}})$ does not change with time. While there may be moving objects within the surveillance region during actual flight, the position and velocities of these objects are not known *a priori* and are not considered. In the following section, when estimating the blur observed in the video, the blur due to moving objects is included in the optic flow calculation of the captured video. We do not assume that the camera matrix C is time-invariant, because the focal length can change with time if the camera has zoom capabilities. The projection matrix P is assumed to be time-invariant.

Once the motion of the terrain point has been calculated, the discount factor M_{p} is calculated for that point as

$$M_{\mathrm{p}}(\xi, \mathcal{P}) = \max\left(0, 1 - \frac{\sqrt{m_x^2 + m_y^2}}{m_{\mathrm{T}}}\right), \tag{12}$$

where m_{T} is the threshold amount of motion, and where m_x and m_y are the motion vector components calculated in Eq. (11). Note that M_{p} does not affect the VTTP value when there is no motion and zeros out the VTTP value when the motion exceeds the threshold.

3.2.2. *Actual motion*

To calculate the actual motion in a captured video, we use the pyramidal Lucas–Kanade optical flow algorithm implemented in OpenCV.[29,30] Although we could calculate a motion field using the actual telemetry, error from sensor measurements and imperfect syncing between the video and telemetry would make this calculation less accurate. The optical flow algorithm does not depend on the telemetry because it directly tracks features from frame to frame within the video. We configured the optical flow algorithm to search for 60 features with a minimum distance apart of 60, a quality level of 0.01, a 60 × 60 search window, and six pyramidal levels. These settings gave us a relatively uniform spread of features across the frame and worked well for all the situations that we test.

Because the motion tends to be fairly consistent throughout the frame, we simplify our algorithm by using the average motion of all the tracked features in the frame. Without this simplification, we would need to match the tracked features up with the nearest terrain point projected into the image plane. The equation for the actual motion discount factor is given by

$$M_{\mathrm{a}}(v) = \max\left(0.1 - \frac{w}{N_{\mathrm{f}} m_{\mathrm{T}}}\sum_{i=1}^{N_{\mathrm{f}}}\sqrt{m_{x,i}^2 + m_{y,i}^2}\right), \tag{13}$$

where v is the video stream, w is the frame rate of the video, N_{f} is the number of tracked features, and $(m_{x,i}, m_{y,i})$ is the motion of the ith feature between one frame and the next.

3.3. *Observer viewing model*

When watching video, it is well documented that humans tend to spend more time looking at areas near the center of the video frames. Because of this, they are more likely to detect objects that are located in the middle of the frame. This effect is described by Tatler, and is now commonly known as the center-bias effect.[31,32] While there are many factors affecting where human operators will direct their attention on a video screen, Judd *et al.* found that a Gaussian distribution is a good model of where observers focus their attention.[33]

The VTTP metric incorporates this center-bias effect in order to more accurately estimate the probability of a human operator successfully performing DRI tasks using actual and predicted flight video. We utilize the result presented by Judd *et al.*, by applying a scalar discount to the VTTP value for objects that are further from the center of the image according to a Gaussian distribution, with covariance Σ_0. The scalar discount, or observer viewing model $O(\xi, \mathcal{P}) \in [0, 1]$, is given by

$$O(\xi, \mathcal{P}) = e^{\left(-\frac{1}{2}([x,y]^\top - [o_x, o_y]^\top)^\top \Sigma_0^{-1}([x,y]^\top - [o_x, o_y]^\top)\right)}, \tag{14}$$

where o_x and o_y are the coordinates of the center of the image. The image center coordinates and observer model covariance are containedin the user defined parameter set $\mathcal{P}$.

3.4. *Validation*

We performed two preliminary validations of the VTTP metric. First, we verified that the predicted and actual VTTP values for a flight path are consistent with each other. This consistency is necessary to ensure that flight paths that maximize the predicted video utility also maximize the attained utility. Second, we performed a small user study to verify that the actual VTTP values are correlated with human assessment of video utility. Together, these validations show that the VTTP metric can be used to plan paths that improve the viewer's experience.

3.4.1. *Predicted versus observed VTTP*

To compare the predicted and actual VTTP metric, we compute $\mathrm{VTTP_p}$ from a simulated path and $\mathrm{VTTP_a}$ using actual video and telemetry. We also consider the predicted VTTP metric computed using actual telemetry instead of a planned path, which we denote as

$$\mathrm{VTTP}_{\mathrm{p}^+}(z,x,\mathcal{P}) = \mathrm{TTP_a}(z,x,\mathcal{P})M_\mathrm{p}(x,\mathcal{P})O_\mathrm{a}(x,\mathcal{P}).$$

The difference between this equation and $\mathrm{VTTP_a}$ is the method we use to estimate the motion discount. This allows us to visualize how well the apparent motion predicts optical flow.

The values for $\mathrm{VTTP_p}$, $\mathrm{VTTP}_{\mathrm{p}^+}$, and $\mathrm{VTTP_a}$ computed over a short path are shown in Fig. 4. We observe that $\mathrm{VTTP_p}$ acts much like an upper bound since it models ideal flight. Using actual telemetry, we observe jitter due to turbulence and other disturbances that reduces the video utility. When we compare $\mathrm{VTTP}_{\mathrm{p}^+}$ to $\mathrm{VTTP_a}$, we conclude that M_p is an adequate approximation of M_a. These results indicate that $\mathrm{VTTP_p}$ is a reasonable approximation to $\mathrm{VTTP_a}$ and can be used to plan paths that maximize video utility.

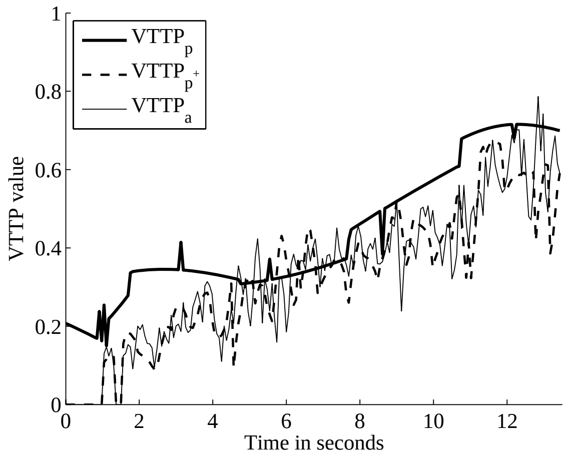


Fig. 4. Using simulated and actual flight data, we compare $\mathrm{VTTP_p}$ to $\mathrm{VTTP_a}$ for a single terrain point. We also compute $\mathrm{VTTP}_{\mathrm{p}^+}$, which is computed using actual telemetry but no video. In this flight, the UAS is approaching the specified terrain point. The increasing VTTP values near imply that the slant range has the largest effect on the VTTP value. The $\mathrm{VTTP_p}$ curve is smoother because it assumes ideal flight and does not model disturbances such as wind gusts. The spikes in the $\mathrm{VTTP_p}$ curve should be examined, but are probably caused by the system switching between waypoints.

3.4.2. *User study*

In order to use the VTTP metric with confidence, we need to know whether there is a correlation between the VTTP values and the observer's ability to perform DRI tasks. Ideally, a user study should be conducted where experienced UAS operators could perform DRI tasks on a wide variety of videos. Unfortunately, for this stage of the project we did not have access to trained UAS operators, nor could we attempt to cover the breadth of possible video scenarios that operators are asked to observe. However, we have performed a small user study where users were asked to rate a set of videos for DRI quality.

For the user study, 10 video clips with durations of 10–14 s were selected where a charcoal-gray extended cab truck was in the video stream for the duration of each clip. The videos were selected to cover many, but not all, combinations of standard viewing conditions. There were videos where the sensor was zoomed in or out, where the truck was either near or far, where the truck was parked in the middle of the road or off to the side, where the truck was in a field or an area with tall shrubs, and in the middle of the day or when there were shadows. In all cases, however, the truck remained stationary.

Twenty participants were administered a test we developed similar to a test developed by Likert.[34] In our test, participants were asked how hard it was for them to detect and identify the truck in the video clip, where the presentation order of the videos was randomized. Users scored one of five preferences: Very Easy, Easy, Average, Hard, or Very Hard. We compared their responses to the average VTTP value of the truck location during the videos. The participants were all college students and most were familiar with the project and affiliated with the BYU MAGICC lab, where this research was being conducted. However, project members did not participate in the study.

We cannot remove response bias from the user's ratings, which makes it impossible to correlate user answers with a numerical value to compare against the VTTP metric. Instead, for each user we compute the difference in preference of all pairwise videos. Also, we compute the difference in VTTP scores for each pairwise combination of videos. We expect that videos with a small difference in Likert rating to have a small difference between their respective VTTP values. When the user preference is vastly different between videos, we expect the difference between the VTTP values to be larger.

In the original analysis, we found that there were three videos that skewed the results due to modeling error. Two videos were given high scores by users because the truck

Table 2. User study results comparing the user preference difference and the corresponding VTTP values between pairwise videos. Notice that as the preference difference increases, so does the mean of the TTP differences for those same videos.

Preference Difference	Mean VTTP difference	Std. deviation of VTTP difference	Number of samples	Significance
0	3.83×10^{-19}	0.0940	290	$p < 0.0001$
1	0.0534	0.1217	153	$p < 0.0001$
2	0.1178	0.1191	110	$p < 0.0001$
3	0.1631	0.1131	63	$p = 0.0152$
4	0.2006	0.0688	19	$p = 0.1757$

was large in the FOV due to high zoom and close proximity. However, the truck was constantly shaking around in the video and received a low VTTP value due to large optical flow. This implies that the motion threshold m_{T} should be scaled inversely proportionally to the ground sample distance (GSD), where the GSD is the distance between pixel centers as measured on the ground. A third video scored a high VTTP value but received a low score from users because it was hard to distinguish the truck parked up against the road near large bushes. This shows that the metric should be modified to include the effects of clutter.

Removing these three outlier videos from the study, we compare the user preference difference with the mean VTTP difference. We note in Table 2 that as the difference in user preference increases, so does the mean VTTP difference. The standard deviations of the VTTP differences for each test are reasonable, and the p-values from the Student T test give a measure of confidence that these values are statistically significant from the adjacent preference difference. Notice that the statistical significance is weakened when there are fewer samples. While more user studies are required to increase confidence in the validity of the VTTP metric, we feel that there is evidence to support its effectiveness at estimating video utility.

4. Staged Path Planning

The UAS and sensor path planning problem is NP-hard and the optimal path Θ^* described in Eq. (4) can only be found in trivial cases. To reduce the dimensionality of the search space, we propose a novel, staged approach to plan the UAS flight and sensor paths by estimating the sub-optimal path $\Theta^\dagger$. We decompose the planning into three stages and use standard nonlinear optimization techniques to solve each sub-problem. In Stage 1 we plan the sensor path $\Theta_{\mathrm{s}}^\dagger$, In Stage 2 we plan the UAS path $\Theta_{\mathrm{u}}^\dagger$ given a fixed sensor path, and in Stage 3 we plan the zoom schedule $\Theta_\zeta^\dagger$ given that the sensor and UAS paths are fixed.

We first plan the sensor path to provide higher-utility video for the operator based on how well we can observe the areas of interest defined by the reward map. We then plan the UAS path, which adjusts the UAS flight path to maximize the VTTP weighted reward of the sensor path. Lastly, we plan the zoom level to adjust the FOV and the ground sample density. In other words, we solve the following optimization problems in order:

$$\Theta_{\mathrm{s}}^\dagger = \arg\max_{\Theta_{\mathrm{s}}} \int_{\tau=0}^{T} R(t,\tau,[\Theta_{\mathrm{s}},\Theta_{\mathrm{u}}^1,\Theta_\zeta^1],\mathcal{P})d\tau, \quad (15)$$

$$\Theta_{\mathrm{u}}^\dagger = \arg\max_{\Theta_{\mathrm{u}}} \int_{\tau=0}^{T} R(t,\tau,[\Theta_{\mathrm{s}}^\dagger,\Theta_{\mathrm{u}},\Theta_\zeta^2],\mathcal{P})d\tau, \quad (16)$$

$$\Theta_\zeta^\dagger = \arg\max_{\Theta_\zeta} \int_{\tau=0}^{T} R(t,\tau,[\Theta_{\mathrm{s}}^\dagger,\Theta_{\mathrm{u}}^\dagger,\Theta_\zeta],\mathcal{P})d\tau, \quad (17)$$

where for Stage 1 we assume the UAS path and zoom schedule are given by the known paths Θ_{u}^1 and Θ_ζ^1, while for Stage 2 we assume that the zoom schedule is given by Θ_ζ^2.

By dividing the planning process into multiple stages we reduce the dimensionality of the optimization problem described by Eq. (4); however, each stage remains a difficult NP-hard problem. In fact, each stage is still more difficult than the Traveling Salesman Problem, due to the underlying nonlinearities and the time-evolving structure of the search space. For this reason, we utilize two simplifications to find solutions more quickly. First, we use a receding horizon approach with a finite time horizon T, where we plan N waypoints and fly the first N_{fly} waypoints before re-planning. Note that the reward map is updated after each time step based on the predicted flight paths. Second, we use the full, time-varying reward map only when planning the final zoom path in Stage 3. During Stages 1 and 2, we use a static reward map at time t to find the optimal sensor and flight path over the time horizon of T seconds. In the subsections below, we describe algorithms to estimate each optimal sub-path.

4.1. *Stage 1: Sensor planning*

The maximization problem to estimate the optimal sensor path is posed in discrete time as

$$\begin{aligned} &\textbf{maximize}_{\Theta_{\mathrm{s}}} && \sum_{\tau=0}^{T} R_1(t,\tau,[\Theta_{\mathrm{s}},\Theta_{\mathrm{u}}^1,\Theta_\zeta^1],\mathcal{P}) \\ &\textbf{subject to} && \text{Sensor Dynamic Constraints,} \end{aligned} \quad (18)$$

where we optimize over a modified version of Eq. (3)

$$R_1(t,\tau,\Theta,\mathcal{P}) = \sum_{z\in \mathrm{FOV}(\xi(t,\tau,\Theta))} r(t,z,x_0)\,\mathrm{VTTP}(z,x_0,\mathcal{P}), \quad (19)$$

where the underlying reward map does not vary with the time of flight parameter τ. During Stage 1, the flight path and the zoom path have not yet been planned, so we use the initial UAS states x_0 to account for the viewing conditions. Errors induced by this estimate are negligible if the time horizon T is not too large. The sensor dynamic constraints can be physical limits on the movement of the sensor or they can represent constraints imposed by the operator to improve the viewing experience. In this section, we describe the method by which we estimate the optimal path Θ_s.

We estimate the solution to Eq. (18) by applying a planning technique presented by Argyle *et al.*, where they use a chain-in-a-forcefield technique to plan a UAS path. In their work, the links in the chain are defined as the UAS flight path waypoints acted on by various forces used to estimate the optimal path. Some of the forces acting on the links include link-length constraints, turn angle constraints, and the gradient of an underlying probability density function.[23]

We let a chain composed of N links, $\mathbf{c} = [z_1, z_2, \ldots, z_N]$, represent the sensor path Θ_s, where the position of the ith link is z_i. Argyle *et al.* use a probability density function to calculate the gradient force acting on the individual links z_i. Instead, we use the time-invariant, TTP-weighted reward map $r_t(z)$ given by

$$r_t(z) = r(t,z,x_0)\,\mathrm{VTTP}(z,x_0,\mathcal{P}),$$

or in other words the reward at time t of terrain point z weighted by the VTTP value. By multiplying the static reward by the VTTP value given the initial conditions, we account for the effects of slant range, grazing angle, occlusions, and contrast when planning Θ_s, while avoiding the complexity of the time-varying gradients.

Argyle *et al.* apply a link-length constraint to enforce equal distances between UAS flight path waypoints. When planning the path of the sensor optical axis, a specific link-length distance is not desired. We extend the work by Argyle *et al.* to allow for *dynamic* link-lengths so that the sensor footprint can dwell in regions of high reward while minimizing the time spent in regions of low reward. A high-level, one-dimensional illustration of this concept is shown in Fig. 5.

We now calculate the forces that act on each link z_i in the chain. Let g_i be the unit vector pointing in the same direction as the gradient, such that

$$g_i = \frac{1}{\beta}\left(\frac{\partial}{\partial z} r_t(z)\Big|_{z=z_i}\right),$$

where β is a normalizing constant. The unconstrained dynamics for link i become $\ddot{z}_i = \gamma_1 g_i$, where γ_1 is a tunning parameter. Since the gradient is calculated from a discretized grid, we use a method similar to the Downhill Simplex Method described in,[35] which is a variant of the Flexible Polyhedron Method first proposed in.[36] In essence, our algorithm constructs a triangle from three adjoining terrain points and estimates the gradient of the reward map from the reward corresponding to those points.

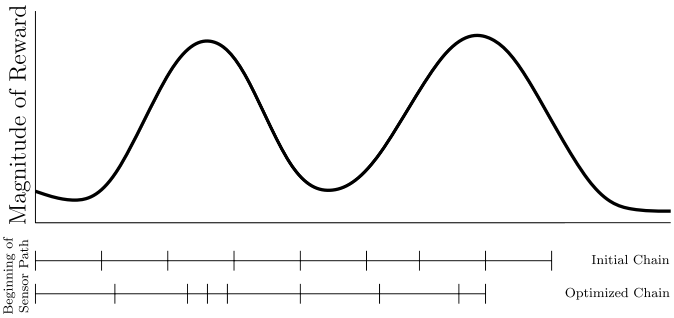


Fig. 5. High-level concept of a sensor path in a 1D world, where the tick marks are waypoints defined every Δt seconds. Notice how the optimized sensor path has waypoints that have shifted to clump in the regions of highest reward, while satisfying a maximum link-length constraint and allowing the path to find and explore other nearby local optima. This results in the sensor footprint dwelling over the regions of high reward-weighted TTP, while moving quickly through the regions of low reward-weighted TTP.

The remaining forces acting on each waypoint z_i are the link-length forces $\lambda_i(z_{i-1})$ and $\lambda_i(z_{i+1})$ relative to the previous and next waypoints in the chain, respectively. Conceptually, there is a correlation between the link-length distance and themotion blur of the image due to sensor and vehicle dynamics. Obviously, a large link-length distance would induce large motion blur, reducing the video quality of terrain in the sensor FOV. The parameter $L_{\max}$ describes the maximum acceptable link-length distance for a given altitude and a particular operator. A minimum link-length distance can also be defined to keep the sensor footprint moving. Applying these constraints, the link-length distance between waypoints of the sensor optical axis will fall between

$$0 \le L_{\min} \le \|\Theta_{s,i}, \Theta_{s,i-1}\| \le L_{\max}. \quad (20)$$

A third parameter L_{nom} is also defined to be the nominal link-length distance for average conditions.

Using the three parameters $L_{\min}$, $L_{\max}$, and L_{nom}, a link-length force is generated based on distance between waypoints, which falls in one of three regions. First, a repelling zone $[L_{\min}, L_r)$, when the two waypoints are too close together. Second, an attracting zone $(L_a, L_{\max}]$, when the two waypoints are too far apart. Finally, there is a nominal zone $[L_r, L_a]$ where we do not apply any link-length forces. The bounds L_r and L_a are determined by the reward level, given by

$$L_r = L_{min} + (1 - r_t(z\pm))L_{nom},$$
$$L_a = L_{max} - r_t(z\pm)L_{max},$$

where we abuse notation and let $r_t(z\pm)$ be the reward level at either the next or previous link in the chain, depending on which force we are currently calculating.

The link-length forces $\lambda_i(z_{i-1})$ and $\lambda_i(z_{i+1})$ due to the previous and next waypoints, respectively, are given by

$$\lambda_i(z_{i-1}) = \begin{cases} \frac{(z_i - z_{i-1})}{s_{prev}} & \text{if } (s_{prev} < L_{min}) \\ \frac{(L_r - s_{prev})}{(L_r - L_{min})}\frac{(z_i - z_{i-1})}{s_{prev}} & \text{if } (s_{prev} > L_{min}) \\ & \quad \text{and } (s_{prev} < L_r) \\ -\frac{(s_{prev} - L)}{(L_{max} - L_a)}\frac{(z_i - z_{i-1})}{s_{prev}} & \text{if } (s_{prev} < L_{max}) \\ & \quad \text{and } (s_{prev} > L_a) \\ -\frac{(z_i - z_{i-1})}{s_{prev}} & \text{if } (s_{prev} > L_{max}) \\ 0 & \text{otherwise,} \end{cases}$$

$$\lambda_i(z_{i+1}) = \begin{cases} -\frac{(z_{i+1} - z_i)}{s_{next}} & \text{if } (s_{next} < L_{min}) \\ -\frac{(L_r - s_{next})}{(L_r - L_{min})}\frac{(z_{i+1} - z_i)}{s_{next}} & \text{if } (s_{next} > L_{min}) \\ & \quad \text{and } (s_{next} < L_r) \\ \frac{(s_{next} - L_a)}{(L_{max} - L_a)}\frac{(z_{i+1} - z_i)}{s_{next}} & \text{if } (s_{next} < L_{max}) \\ & \quad \text{and } (s_{next} > L_a) \\ \frac{(z_{i+1} - z_i)}{s_{next}} & \text{if } (s_{next} > L_{max}) \\ 0 & \text{otherwise,} \end{cases}$$

where $s_{prev} = \|z_i - z_{i-1}\|$ and $s_{next} = \|z_{i+1} - z_i\|$. The constrained dynamics of link z_i can now be written as

$$\ddot{z}_i = F = \gamma_1 g_i + \gamma_2 \lambda_i(z_{i-1}) + \gamma_3 \lambda_i(z_{i+1}), \tag{21}$$

where γ_i are tuning parameters. To avoid offsetting link-length forces, we require $\gamma_2 > \gamma_3$. Figure 6 shows graphically the piecewise linear function describing both the regions of attraction and repulsion and the magnitude of the link-length force for waypoints in both high and low reward regions.

To better explore the search space, four initial paths are used to seed the planner, with the best resultant path selected to be Θ_s. Each initial path originates from the initial $n_{s,0}$ and $e_{s,0}$ coordinates, with N equidistant links spread to the edge of the defined map. In many situations, the initial path will violate the link-length constraints, but as the forces act to satisfy them, the path will explore many local maxima to find the best nearby path. To overcome nonsmooth reward maps, we can also downsample r_t to create an intermediate path to seed the full resolution planner to obtain the final path.

Figure 7 shows a simulated sensor path in a reward field consisting of many local minimum. The parameters for this particular simulation were $L_{min} = 2$, $L_{max} = 11$, and $L_{nom} = 5$ and acted on a chain starting at the coordinates $[97, 105]$. Note that the best path found passes through two local maxima, and that the waypoints are bunched in the high reward regions.

4.2. *Stage 2: UAS planning*

The next step is to plan the UAS path Θ_u given the sensor path Θ_s^*. The optimization problem at this stage can be stated as

$$\begin{aligned} \underset{\Theta_u}{\textbf{maximize}} \quad & \sum_{\tau=0}^{T} R_2(t, \tau, [\Theta_s^\dagger, \Theta_u, \Theta_\zeta^2], \mathcal{P}) \\ \textbf{subject to} \quad & \text{Vehicle Dynamic constraints} \\ & \text{Optional No-Fly Zone constraints.} \end{aligned} \tag{22}$$

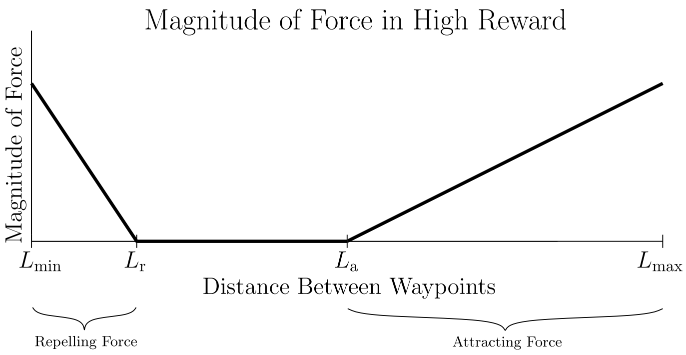


Fig. 6. Link-length constraints of a link in both high and low reward areas. Notice that links in high reward are allowed to clump together without experiencing a repelling force, and they are discouraged from being too far apart. The opposite is true for links in low reward areas to encourage them to find the high reward areas.

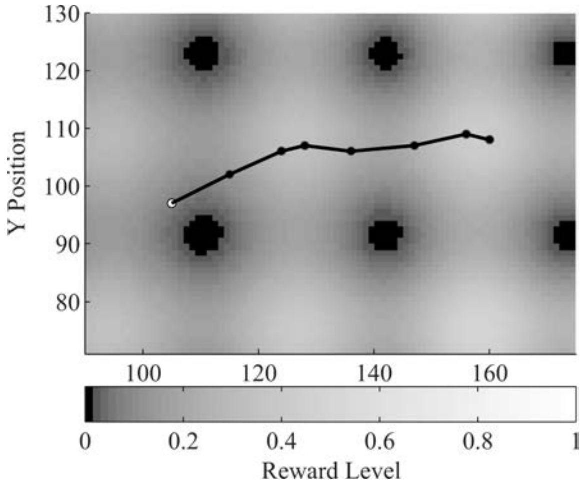


Fig. 7. Sensor path planned from the initial location represented by the white waypoint. Notice that the path traverses two local maximum and that the dynamic link-length constraints are evident as waypoints are spread apart in lower reward and bunched together in high reward.

We again use a static reward map by modifying Eq. (3) to become,

$$R_2(t,\tau,\Theta,\mathcal{P}) = \sum_{z\in \mathrm{FOV}(\xi(t,\tau,\Theta))} r(t,z,\xi(t,\tau,\Theta))\mathrm{VTTP}(z,\xi(t,\tau,\Theta),\mathcal{P}),$$

where the reward map depends only on time t, and not the flight parameter τ.

We define the vehicle constraints according to fixed-wing UAS dynamics, and define a maximum change in heading $\Delta\psi_{\max}$ and a maximum distance between UAS waypoints L_{u}. These constraints are given by

$$|\Delta\psi_i| < \Delta\psi_{\max}$$
$$\big|\|\Theta_{\mathrm{u},i},\Theta_{\mathrm{u},i-1}\| - L_{\mathrm{u}}\big| < \Delta L_{\mathrm{u}},$$

where ΔL_{u} is a tolerance in the length between UAS waypoints to allow the solver the flexibility to adjust paths to find a solution. The path Θ_ζ^2 is simplified such that only one terrain point z, defined by Θ_{s}, is in the FOV. This greatly reduces the computation time needed to calculate the reward-weighted VTTP value for each terrain point z in the FOV. Effects due to zoom are considered in Stage 3.

To solve this problem, we utilize a method based on the sequential quadratic programming (SQP) nonlinear solver proposed in.[37] Since the solution space has many local minima, we seed the solver with five different initial trajectories ranging from hard left and right turns to maintaining a straight orbit. Figures 8 and 9 are examples of UAS paths generated from Stage 2. Note that the results indicate

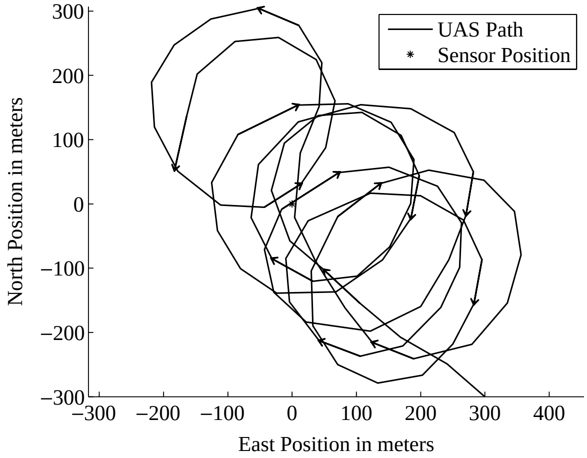


Fig. 8. UAS path given a stationary sensor location. Note that the center of the sensor footprint is at $(0,0)$, and the initial UAS position is at $(300\mathrm{E},-300\mathrm{N})$ with a velocity of 17 m/s and an initial heading of -45°. Total simulated flight time is 400 s.

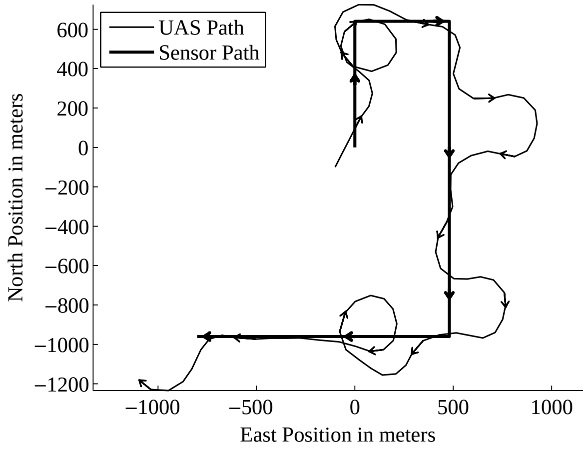


Fig. 9. UAS path given the above sensor path starting at $(0,0)$ and has a velocity of 10 m/s. Note that the initial UAS position is at $(-100\mathrm{E},-100\mathrm{N})$ with a velocity of 17 m/s and an initial heading of 45°. Total simulated flight time is 400 s.

that the planner tends to reduce the line of sight distance to the terrain point.

4.3. *Stage 3: Zoom planning*

The zoom path Θ_ζ is planned during the third stage of the path planner. In this stage we use the time-varying reward map r to let the planner adjust the zoom level to maximize the reward collected during the path. We wait until this stage to vary the reward map because we do not need to make any assumptions about Θ_{s} and Θ_{u} since they were planned in Stages 1 and 2.

We originally posed the optimization problem as defined in Eq. (17). However the planner tended to zoom out as much as possible, probably because even at low resolution the VTTP metric was high enough to capture much of the reward in the sensor FOV due to the low line of sight distances. To allow the operator more direct control of the zoom level, we pose the optimization problem for Stage 3 as

$$\begin{aligned}\underset{\Theta_\zeta}{\textbf{maximize}} \quad & \sum_{\tau=0}^{T} \min\Bigg(R(t,\tau,[\Theta_s^\dagger,\Theta_u^\dagger,\Theta_\zeta],\mathcal{P}),\\ & \qquad \gamma_\zeta \frac{R(t,\tau,[\Theta_s^\dagger,\Theta_u^\dagger,\Theta_\zeta],\mathcal{P})}{N_z(\tau)}\Bigg) \qquad (23)\\ \textbf{subject to} \quad & \Theta_\zeta \le \zeta_{\max}\\ & \Theta_\zeta \ge \zeta_{\min},\end{aligned}$$

where $N_z(\tau)$ is the number of terrain points z in the sensor FOV at time $t+\tau$ and $\gamma_\zeta > 0$ is a tunning parameter. In other words, at each time step the planner maximizes the minimum between the total reward in the FOV and average reward in the FOV scaled by γ_ζ. The scalar value γ_ζ allows the user to control how much the system tends to zoom in or out. High values result in zoom paths that tend to zoom out, while lower values allow the system to zoom in.

Since the time to compute Θ_ζ constitutes the largest amount of time spent planning a path, we consider three different methods of controlling the zoom. The simplest method would be to use a fixed zoom level during the entire flight. The most complex uses estimated gradient information to aid a standard SQP nonlinear solver.[37] Another method is to use a greedy search at each waypoint, selecting one zoom level from a fixed number of possible settings.

The implementation of SQP solver took the longest time to compute because it required multiple computations of the reward level in the FOV during a single iteration of the planner in order to compute the gradients. To reduce the computational complexity, we used the greedy search to implement Eq. (23). This does not allow the planner to optimally account for future viewing opportunities, but we believe that these effects are minimal.

5. Results

We apply the staged-path planner described in Sec. 4 to find a parameterized path Θ that maintains surveillance on a specific location of interest and the surrounding area, especially the adjoining road network. The path Θ consists of 100 waypoints characterizing the UAS flight path, the sensor path, and the sensor resolution. Section 5.1 explains how we implement these algorithms on a standard desktop to develop a flight plan and sensor schedule. Section 5.2 discusses our testbed UAS and compares flight test results to predicted results. Finally, in Sec. 5.3 we discuss the computational complexity of this technique as well as some advantages and disadvantages of using this algorithm.

5.1. *Planning and simulation*

The purpose of the staged path planner is to reduce the effort needed to prepare for and fly a surveillance mission by autonomously computing both a flight path and a sensor schedule. We implement Eqs. (18), (22) and (23) on a

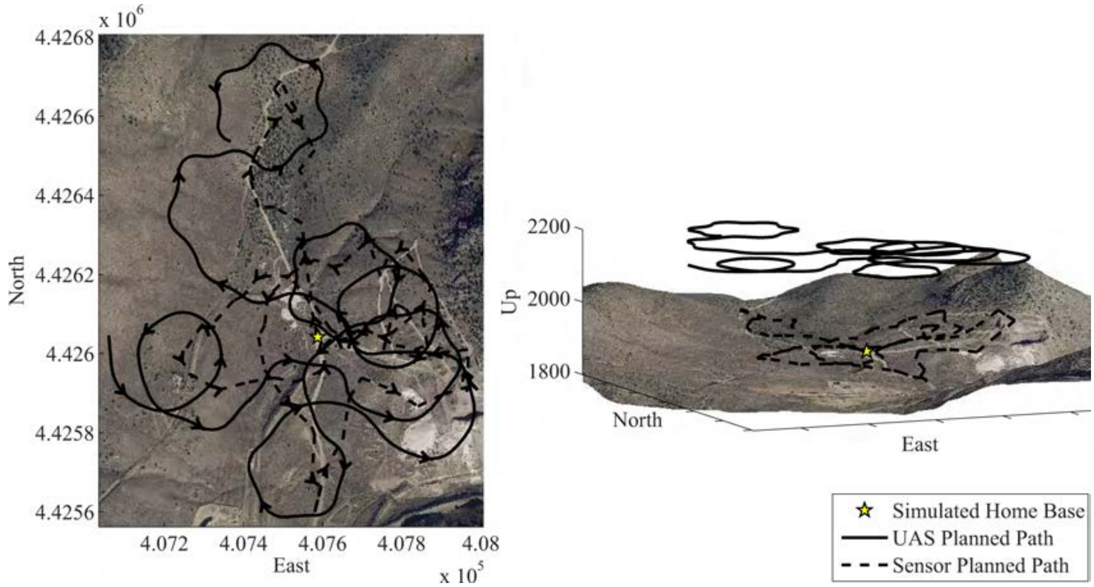


Fig. 10. (Color online) UAS and camera path planned in simulation to perform surveillance.

standard desktop in Matlab. The set of Z discrete terrain points defining the region of interest are available from the US Geological Survey online database.[38] Specifications and other parameters are listed in Table A.1 of Appendix A.

Figure 10 presents the paths generated from the staged planner. Notice that for the majority of the simulation the camera path followed the high reward road network, occasionally crossing low reward areas to switch to other high reward regions. Also, notice that due to the time evolving nature of the reward map, the simulated home base marked by the yellow dot was visible on three separate occasions. Finally, it is also evident that the UAS path tends to minimize the line of sight distance to the sensor path, resulting in loops around the sensor path.

5.2. *Flight test*

The goal of the flight demonstration was to fly the planned path shown in Fig. 10 and compare the predicted coverage to what was actually observed. We used a Zagi style airframe with a 72 in. wingspan equipped with a Kestral™ autopilot and a Sony FCB - IX11A Block color camera, which has a 640×480 pixel imager, with focal lengths that can range from 4.2 mm to 42 mm. The camera was mounted on a gimbal that deployed below the body of the UAS, as shown in Fig. 11. We used Virtual Cockpit™ as the ground station controlling the UAS and OnPoint™ software to record and synchronize video with the telemetry. The autopilot, ground station, and image recording software are all produced by Procerus Technologies.

During flight, we interfaced MATLAB with Virtual Cockpit to send the next set of desired waypoints based on the staged planner. The UAS control surfaces, gimbal servos, and zoom level are controlled to follow their respective paths using onboard low-level control algorithms. Unfortunately, we can send only one UAS waypoint, camera waypoint, or zoom level every 300 ms. We therefore set an arrival radius of 80 m which triggered the UAS to proceed to the next waypoint. Meanwhile, the camera gimbal and zoom levels were adjusted every 600 ms. Real-time telemetry received from the UAS triggered when to change between camera and zoom waypoints, and the algorithm linearly interpolated the desired sensor position between the waypoints.

Figures 12–14 show the planned versus actual paths when flown with our testbed UAS. When flying the paths, the UAS system endured an average wind speed of 3.66 m/s from a heading of 305°, which caused error between the simulated and actual UAS path. The worst overshoot due to wind was about 80 m, but typical errors were less than half the maximum error. The errors in the sensor paths shown in Fig. 13 are more pronounced due to wind gusts and other turbulence during flight. However, the low level gimbal pointing algorithms did a reasonable job of stabilizing the sensor gimbal and following the path. The zoom paths shown in Fig. 14 are very similar. Differences seen in Fig. 14 are due to the simulation expecting a constant time between each waypoint, and then actual flight resulting in varied time between waypoints; this correlates to a time-shift and a time-scaling of the zoom path.

Errors in the UAS and gimbal position affect the shape of the FOV on the ground, changing where the reward is collected within the reward map. To view these effects, we include the evolving reward map after 25, 50, 75, and 100 waypoints of both the simulated and actual flight test in Fig. 15. While the planned and actual reward maps are very similar, there are differences due to errors caused by modeling errors and disturbances while following the planned paths.

Fig. 11. UAS testbed and the extended gimbal.

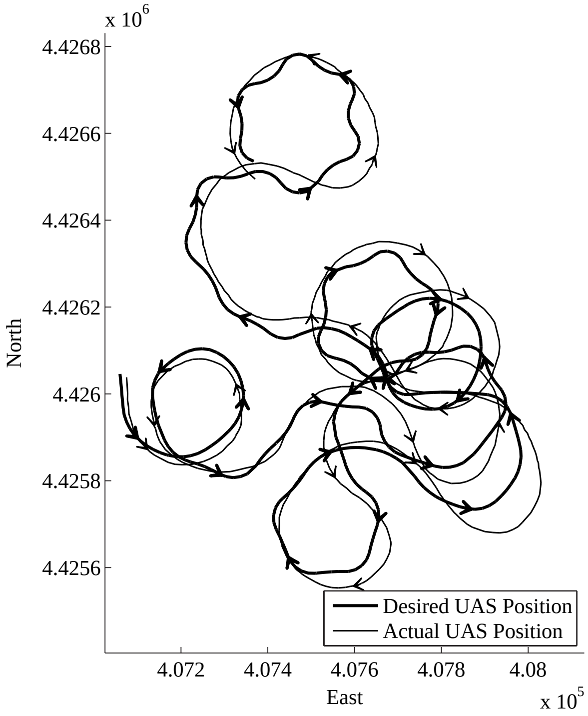


Fig. 12. Comparison of the planned and actual UAS path.

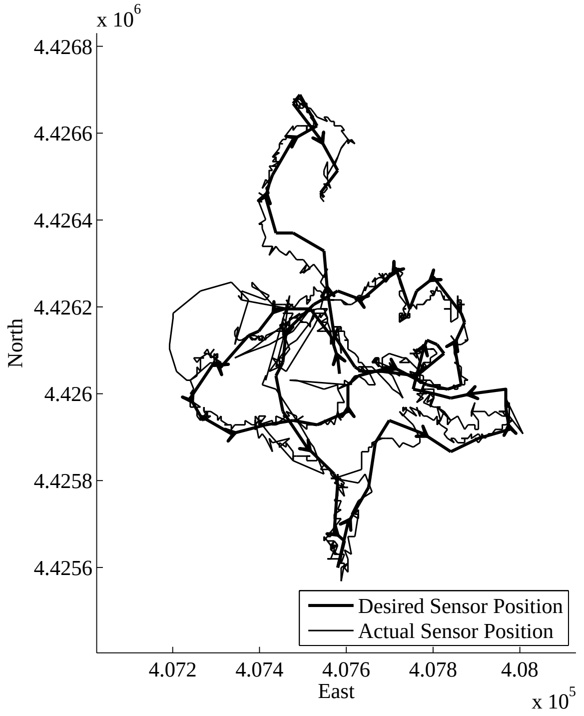


Fig. 13. Comparison of the planned and actual sensor path.

5.3. *Analysis and future work*

We have successfully demonstrated our ability to plan and implement an autonomous UAS flight path, the sensor gimbal schedule, and the sensor zoom schedule to perform surveillance over a desired region of interest. We do not claim these paths to be optimal, but the generated paths are intuitive and provide usable video to the operator. Unfortunately, there does not exist in the literature an alternative method that we can compare our algorithm against. We present our results as a motivation for others to improve

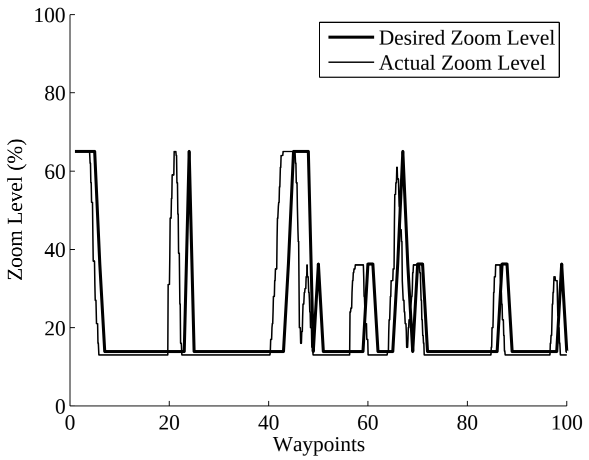


Fig. 14. Comparison of the planned and actual zoom levels.

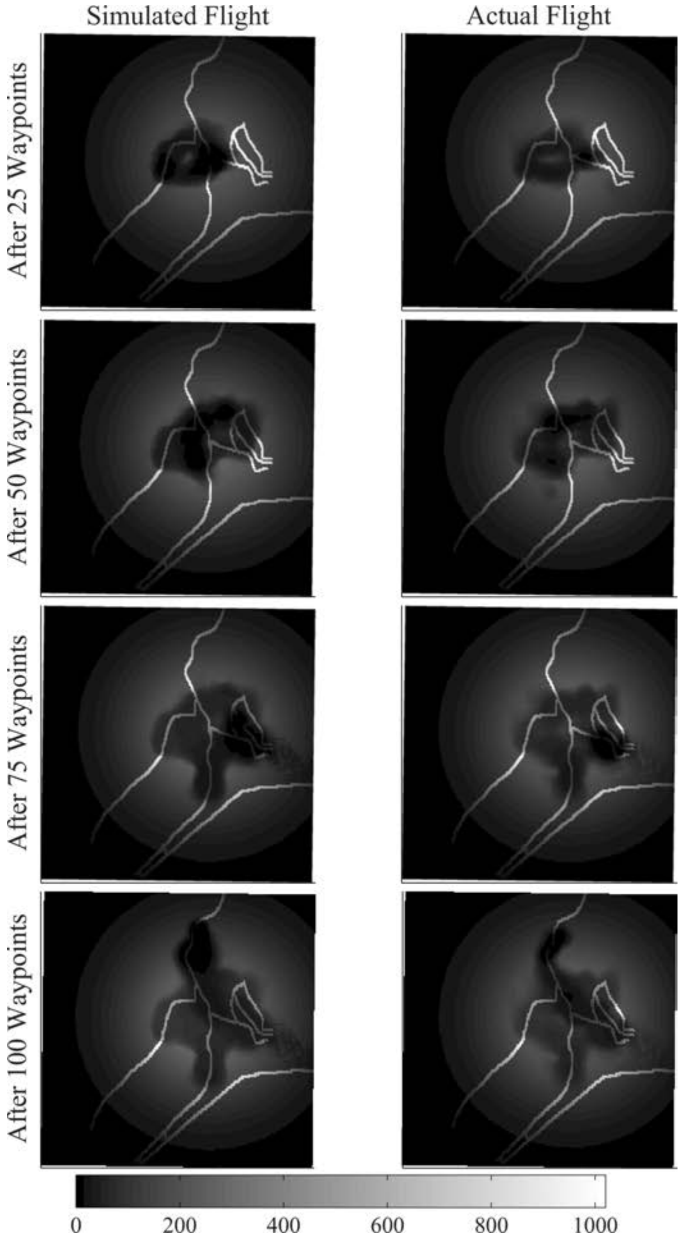


Fig. 15. Reward map evolution after flying 25, 50, 75, and 100 waypoints for both simulated and actual flight.

upon our work, with several areas of possible improvement listed below.

The computational complexity of this technique remains high. The planned path shown in Fig. 10 was generated in about 2.5 h using an Intel Core2 Duo processor running at 3.16 GHz. The subprocess requiring the majority of the computation is the calculation of the zoom path. In Stage 3, the VTTP value is calculated for each terrain point in the camera field of view, and is computed for each discrete zoom level possibility. Though not presented here, using a gradient-based method to plan the zoom level took much longer, about 24 h, due to the repeated calculation of the observed reward needed to compute the gradient.

One area of future research is to develop real-time algorithms in order to include a feedback loop. By using video feedback, the system would be robust to disturbances such as wind gusts and other turbulence by planning paths that account for areas that were not viewed adequately during the initial exploration. At the expense of increased computational complexity, an intermediate technique that would allow the paths to converge to more optimal solutions would be to iterate through Eqs. (15)–(17), using the paths from the previous iteration to seed the next solution.

We acknowledge that the planned UAS flight paths only maintain C1 continuity between the waypoints. In other words, at each waypoint there is a discontinuity in the commanded roll angle. We expect this to cause minor transient effects; however, these effects are smoothed during real flight. Flying connected arcs are also an improvement over other existing methods such as Dubins paths, which require up to three discontinuities in the commanded roll angle between each waypoint.[39,40]

The VTTP metric should be subjected to a more thorough verification and validation process to ascertain its validity as a model of probability of DRI tasks. Also, the various parameters used within the metric need to be tuned. A well tuned metric might eliminate the need to include γ_ζ in Eq. (23), such that we could let $\gamma_\zeta = 0$ and still plan a satisfactory zoom schedule. Finally, we could consider the uncertainty in the parameter set by averaging over their individual distributions to obtain

$$\mathrm{VTTP}(z,x) = \int_{\mathcal{P}} \mathrm{VTTP}(z,x \mid \mathcal{P}) p(\mathcal{P}) d\mathcal{P},$$

where $p(\mathcal{P})$ is probability density function over the parameter $\mathcal{P}$.

The VTTP metric can be improved by refining its current components or through the addition of new components. We set a constant motion threshold in the motion component, but in reality the threshold should scale proportionally to $\frac{\zeta}{s(z,x)}$, where ζ is the focal length and $s(z,x)$ is the slant range from the sensor to the point on the terrain. A clutter component using, for example, one of the clutter metrics used by Henderson *et al.* could be added to model the difficulty of finding the object in a cluttered versus a noncluttered environment.[41] The actual lighting of an area could be estimated from the video by taking both the brightness of the pixels and the exposure time of the camera into account. A Fourier or wavelet transform of the video could be passed into a machine learning algorithm, like a neural network, trained to account for factors not covered by other components of the metric. There is also potential for detecting moving objects and accounting for the effects of the shadow cast by the object itself.

6. Conclusion

This paper has extended existing image utility metrics to develop the VTTP metric used to quantify the utility of video imagery. We have shown through a user study that relative differences in VTTP values correlate with differences in the user preferences. We provide the theoretical framework to plan optimal paths for the UAS path, the sensor path on the ground, and the zoom scheduling. To bridge theory and application, this paper presents a staged approach to successively plan the sensor path along the terrain, the UAS flight path, and the sensor zoom level. Computation in real time in flight is not yet possible, but these algorithms were successfully applied to precompute a UAS path and sensor schedule to survey a specified region in both simulation and hardware.

One of the objectives of this work was to reduce the workload for UAS operators. Currently, for any surveillance mission, one or more operators determine a flight plan, then manually control the sensor gimbaling and the zoom level while monitoring the video screen for potential objects of interest. In performing flight tests simulating these movements, we found that three people could perform these tasks comfortably: one person directing the sensor and zoom movements, one person controlling the UAS path, and the final person focused completely on DRI tasks and directing the UAS and gimbal movements. In contrast, our system requires the operator to load the flight parameters and then execute the planning algorithms. Upon launching the UAS, the planned path is executed autonomously and the operator is free to focus on the video stream. However, we believe that much more work could be done to reduce the operator workload further, and encourage others to improve on the methods presented here.

Acknowledgments

This research was made with Government support in part by Army SBIR Contract W911W6-09-C-0047 through the Aviation Applied Technology Directorate (AATD) and also through the DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

Appendix A

Table A.1 contains a complete list of the parameters used to generate the results presented in this paper. The parameters listed in Table A.1 fall into three categories: Hardware specifications, empirically determined parameters, and user-defined parameters. Parameters defining

Table A.1. Parameters used to plan UAS flight and sensor paths.

Simulation and flight test parameters		
UAS		
Initial north position		4,426,046 m
Initial east position		407,059 m
Altitude (Height above launch)		300 m
Initial heading		-180°
Velocity		17 m/s
Camera		
Minimum focal length	ζ_{min}	4.2 mm
Maximum focal length	ζ_{max}	42 mm
Center x pixel	o_x	320 pixels
Center y pixel	o_y	240 pixels
Pixel width	dim_x	4.65 μm
Pixel height	dim_y	4.65 μm
Pixel FOV	$\measuredangle_{\mathrm{pixel}}$	$2\arctan(\frac{(o_x \mathrm{dim}_x)}{2\zeta})/o_x$
Reward		
Frame rate	w	15 frames/sec
High reward weighting	$\overline{\omega}$	2
Low reward weighting	$\underline{\omega}$	0.4
Reward covariance	Σ_{r}	$\begin{bmatrix} 150{,}000 & 0 \\ 0 & 150{,}000 \end{bmatrix}$
Home base location	z_{base}	$\begin{bmatrix} 40{,}758\,\mathrm{m} \\ 442{,}605\,\mathrm{m} \end{bmatrix}$
VTTP		
Reward collection rate	γ_{r}	1
Object dimensions (height, width, depth)		$2\,\mathrm{m} \times 0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$
Lowest spacial frequency	κ_{low}	0.1
Highest spacial frequency	κ_{cut}	5000
MRC curve empirical parameters	$[a_1,\ a_2,\ a_3]$	$[-0.0235,\ 0.0742,\ 1.4445]$
TTP integral empirical parameters	$[b_1,\ b_2]$	$[1.33,\ 0.23]$
Minimum line pairs on the object	N_{50}	30 pairs
Ambient lighting parameter	α	0.2
Number of image features to track	N_{f}	60 features
Motion magnitude threshold	m_{T}	40 pixels/frame
Observer model covariance	Σ_{O}	$\begin{bmatrix} 40{,}000 & 0 \\ 0 & 22{,}500 \end{bmatrix}$
Staged path planner		
Number of waypoints in lookahead window	N	7 waypoints
Waypoints to fly before re-planning	N_{fly}	4 waypoints
Time between each waypoint	Δt	4 s
Minimum sensor slew distance	L_{min}	0 m
Maximum sensor slew distance	L_{max}	100 m
Ideal sensor slew distance	L_{nom}	20 m
Chain parameters	$[\gamma_1,\ \gamma_2,\ \gamma_3]$	[1, 2, 1]
Max heading change	$\Delta\psi_{\mathrm{max}}$	0.1545 rad/s
Flight path waypoint tolerance	ΔL_{u}	8 m
Zoom parameter	γ_ζ	550

the UAS velocity or the camera imager are defined solely by the hardware specifications. Empirically determined parameters, such as the parameters needed for the VTTP metric, are parameters that have been verified in existing work through extensive user studies, as in.[17] Finally, the remaining user-defined parameters are those that the operator can tune to affect the performance of the planner to suit their preferences or their needs.

References

[1] T. J. Setnicka and K. Andrasko, *Wilderness Search and Rescue* (Appalachian Mountain Club, Boston, MA, 1980).

[2] F. Bourgault, T. Furukawa and H. Durrant-Whyte, Optimal search for a lost target in a bayesian world, in *Proc. Int. Conf. Field and Service Robotics* (2003), pp. 209–222.

[3] S. Hansen, T. McLain and M. Goodrich, Probabilistic searching using a small unmanned aerial vehicle, *Proc. of AIAA Infotech@Aerospace* (2007).

[4] M. Goodrich, B. Morse, C. Engh, J. Cooper and J. Adams, Towards using UAVs in wilderness search and rescue: Lessons from field trials, *Interact. Stud.* **10**(3) (2009).

[5] L. Lin and M. A. Goodrich, UAV intelligent path planning for wilderness search and rescue, *IEEE Int. Conf. Intelligent Robots and Systems* (2009).

[6] C. Bolkcom, Homeland security: Unmanned aerial vehicles and border surveillance, Technical Report, CRS Report for Congress, Code RS21698 (2008).

[7] M. Kontitsis and K. Valavanis, A UAV vision system for airborne surveillance, in *Proc. 2004 IEEE Int. Conf. Robotics and Automation*, Vol. 1 (2004), pp. 77–83.

[8] N. Ceccarelli, J. Enright, E. Frazzoli, S. Rasmussen and C. Schumacher, Micro UAV path planning for reconnaissance in wind, in *Proc. 2007 American Control Conf.*, July 2007, pp. 5310–5315.

[9] M. Quigley, M. Goodrich, S. Griffiths, A. Eldredge and R. Beard, Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera, in *Proc. 2005 IEEE Int. Conf. Robotics and Automation* (2005), pp. 2600–2605.

[10] F. Rafi, S. Khan, K. Shafiq and M. Shah, Autonomous target following by unmanned aerial vehicles, in *Proc. SPIE Defence and Security Symp.* (2006), pp. 6230–6242.

[11] J. Kim and Y. Kim, Moving ground target tracking in dense obstacle areas using uavs, in *Proc. IFAC World Congress* (2008).

[12] G. Limnaios, N. Tsourveloudis and K. Valavanis, Introduction, in *Sense and Avoid in UAS: Research and Applications*, ed. P. Angelov (John Wiley & Sons, Ltd, Chichester, UK, 2012), pp. 3–34.

[13] Unmanned Systems Integrated Roadmap FY2011-2036, Technical Report, Report of the Office of the Secretary of Defence, DOD, USA (2011).

[14] J. Franke, V. Zaychik, T. Spura and E. Alves, Inverting the operator/vehicle ratio: Approaches to next generation UAV command and control, in *Proc. of AUVSI* (2005).

[15] J. Johnson, Analysis of image forming systems, *Image Intensifier Symp.*, AD 220160 (1958), pp. 244–273.

[16] J. Johnson and W. Lawson, Performance modeling methods and problems, in *Proc. IRIS Imaging Systems Group* (1974), pp. 40–44.

[17] R. Vollmerhausen and E. Jacobs, The targeting task performance (TTP) metric: A new model for predicting target acquisition performance, Technical Report, US Army CERDEC (2004).

[18] B. Morse, C. Engh and M. Goodrich, Aerial video coverage quality maps and prioritized indexing, in *Proc. 5th ACM/IEEE RAS Int. Conf. Human-Robot Interaction* (2010).

[19] F. Bourgault, T. Furukawa and H. Durrant-Whyte, Coordinated decentralized search for a lost target in a bayesian world, in *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (2003), pp. 48–53.

[20] T. H. Chung and J. W. Burdick, A decision-making framework for control strategies in probabilistic search, in *Proc. 2007 IEEE Int. Conf. Robotics and Automation*, April 2007, pp. 4386–4393.

[21] S. Waharte, A. Symington and N. Trigoni, Probabilistic search with agile UAVs, in *Proc. 2010 IEEE Int. Conf. Robotics and Automation* (IEEE, 2010), pp. 2840–2845.

[22] P. Niedfeldt, R. Beard, B. Morse and S. Pledgie, Integrated sensor guidance using target probability of operator detection, in *Proc. 2010 American Control Conf.* (2010), pp. 788–793.

[23] M. Argyle, C. Chamberlain and R. Beard, Chain-based path planning for multiple UAVs, *2011 IEEE Conf. Decision and Control* (2011).

[24] P. C. Niedfeldt, B. T. Carroll, R. W. Beard and S. Pledgie, A staged path planner for an unmanned air system performing surveillance and tracking, in *Proc. AIAA Guidance, Navigation and Control Conf.* (2012).

[25] G. P. Kladis, J. T. Economou, K. Knowles, J. Lauber and T.-M. Guerra, Energy conservation based fuzzy tracking for unmanned aerial vehicle missions under a priori known wind information, *Eng. Appl. Artif. Intell.* **24** (2011) 278–294.

[26] J. S. Accetta and D. L. Shumaker, *The Infrared and Electro-Optical Systems Handbook* (Infrared Information Analysis Center, Ann Arbor, MI. and Bellingham, WA, 1993).

[27] G. Ward, A contrast-based scalefactor for luminance display, in *Graphics gems IV*, ed. P. S. Heckbert (Academic Press Professional, 1994), pp. 415–421.

[28] C. Cornwall, A. Horiuchi and C. Lehman, NOAA solar position calculator, Avalable at http://www.esrl.noaa.gov/gmd/ grad/solcalc/azel.html.

[29] J. Shi and C. Tomasi, Good features to track, in *Proc. 1994 IEEE Conf. Computer Vision and Pattern Recognition* (1994), pp. 593–600.

[30] J. Bouguet, Pyramidal implementation of the lucas kanade feature tracker description of the algorithm, Technical Report, Intel Corporation, Microprocessor Research Labs (1999).

[31] B. Tatler, The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions, *J. Vis.* **7** (2007) 1–17.

[32] A. Borji and L. Itti, State-of-the-art in visual attention modeling, *IEEE Trans. Pattern Anal. Mach. Intell.* (2012).

[33] T. Judd, K. Ehinger, F. Durand and A. Torralba, Learning to predict where humans look, *IEEE 12th Int. Conf. Computer Vision* (2009), pp. 2106–2113.

[34] R. Likert, A technique for the measurement of attitudes, *Arch. Psychol.* **22**(140) (1932) 1–55.

[35] W. H. Press, S. A. Teukolsky, W. Vetterling and B. P. Flannery, *Numerical Recipies 3rd Edition: The Art of Scientific Computing* (Cambridge University Press, 2007).

[36] J. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.* **7**(4) (1965) 308–313.

[37] P. Gill, W. Murray and M. Wright, *Practical Optimization* (Academic Press, London, 1981).

[38] USGS national map viewer http://nationalmap.gov/viewer.html.

[39] L. Dubins, On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, *Am. J. Math.* **79**(3) (1957) 497–516.

[40] E. Magid, D. Keren, E. Rivlin and I. Yavneh, Spline-based robot navigation, *2006 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, 2006), pp. 2296–2301.

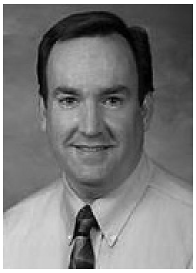
[41] J. Henderson, M. Chanceaux and T. Smith, The influence of clutter on real-world scene search: Evidence from search efficiency and eye movements, *J. Vis.* **9**(1) (2009) 1–8.

Peter C. Niedfeldt received his B.S. degree in electrical and computer engineering in 2009 from Brigham Young University, where he is currently a Ph.D. candidate with an expected graduation of August 2014. He has interned at Raytheon Missile Systems, Guidance Navigation and Control Division, Tucson, AZ and was a summer researcher at the Air Force Research Laboratory, Air Vehicles Directorate, Dayton, OH. His primary research interests are path planning for autonomous vehicles and robust target detection and tracking.

Brandon T. Carroll received his B.S. from Brigham Young University with a double major in Electrical Engineering and Computer Science in 2010. He then joined the MAGICC lab at BYU, studying control theory and computer vision. He completed an M.S. in Electrical Engineering in 2012. Currently, Brandon is a Ph.D. student at Georgia Tech and is studying applications of digital signal processing in the poultry industry.

Joel A. Howard received his B.S. degree in Computer Science from Brigham Young University in 2010. Since 2010, he has been studying at Brigham Young University with an expected graduation of August 2013 with his M.S. degree in Computer Science. His primary research focus is extending current image manipulation techniques, specifically image completion, to handle stereo image pairs.

Randal W. Beard received the B.S. degree in electrical engineering from the University of Utah, Salt Lake City in 1991, the M.S. degree in electrical engineering in 1993, the M.S. degree in mathematics in 1994, and the Ph.D. degree in electrical engineering in 1995, all from Rensselaer Polytechnic Institute, Troy, NY. Since 1996, he has been with the Electrical and Computer Engineering Department at Brigham Young University, Provo, UT, where he is currently a Professor. In 1997 and 1998, he was a Summer Faculty Fellow at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. In 2006 and 2007 he was a visiting research fellow at the Air Force Research Laboratory, Munitions Directorate, Eglin AFB, FL. He is a past Associate Editor for the *IEEE Control Systems Magazine* and the *Journal of Intelligent and Robotic Systems*, and is currently an Associate Editor of the *IEEE Transactions on Automatic Control*. His primary research focus is autonomous control of miniature air vehicles and multivehicle coordination and control.



Bryan S. Morse is a Professor of Computer Science at Brigham Young University. Dr. Morse holds a Ph.D. in Computer Science from the University of North Carolina at Chapel Hill. He has published more than 70 journal and conference papers in various areas of computer vision, image processing, and computer graphics, and he routinely serves as a reviewer for the major journals and conferences in these fields. He has been a visiting researcher at the National Library of Medicine, with whom he still maintains collaboration. He has twice chaired the IEEE Workshop on Applications of Computer Vision and currently serves as vice-chair of the IEEE Computer Society's Technical Committee for Pattern Analysis and Machine Intelligence.

Dr. Stephen Pledgie has served as Chief Scientist of Mosaic ATMs Autonomous Systems Group (ASG) since 2008. He also serves as Vice President of Sentix Corporation, a subsidiary of Mosaic ATM that specializes in Department of Defense technologies. Dr. Pledgie has previously held positions at SET Corporation and with the Air Force Research Laboratory, Munitions Directorate. His interests include optimal planning, guidance, and control of airborne and ground-based sensors to improve navigation performance and maximize the utility of motion imagery. His principal education and training is in the areas of robotics, optimal control, and dynamic optimization. He brings a unique, human-centered perspective to advanced technology projects. Dr. Pledgie is a graduate of the University of Delaware, having earned a B.S. in Electrical Engineering and M.S. and Ph.D. degrees in Biomechanics and Movement Science.