

A Vision System for Intelligent Mission Profiles of Micro Air Vehicles

Sinisa Todorovic, *Student Member, IEEE*, and Michael C. Nechyba, *Member, IEEE*

Abstract—Recently, much progress has been made towards the development of small-scale aircraft, known broadly as Micro Air Vehicles or MAVs. Up until recently, these platforms were exclusively remotely piloted, with no autonomous or intelligent capabilities, due at least in part to stringent payload restrictions that limit on-board sensors. However, the one sensor that is critical to most conceivable MAV missions, such as remote surveillance, is an on-board video camera and transmitter that streams flight video to a nearby ground station. Exploitation of this key sensor is therefore desirable, since no additional on-board hardware (and weight) is required. As such, in this paper we develop a general and unified computer vision framework for MAVs that not only addresses basic flight stability and control, but enables more intelligent missions as well. Our paper is organized as follows. We first develop a real-time feature extraction method called Multiscale Linear Discriminant Analysis (MLDA). MLDA explicitly incorporates color in its feature representation, while implicitly encoding texture through a dynamic multiscale representation of image details. We demonstrate key advantages of MLDA over other possible multiscale approaches (e.g. wavelets), especially in dealing with transient video noise. Next, we show that MLDA provides a natural framework for performing real-time horizon detection. We report horizon-detection results for a range of images differing in lighting and scenery, and quantify performance as a function of image noise; furthermore, we show how horizon detection naturally leads to closed-loop flight stabilization. Then, we motivate the use of Tree-Structured Belief Networks (TSBNs) with MLDA features for sky/ground segmentation. This type of segmentation augments basic horizon detection and enables certain MAV missions where prior assumptions about the flight vehicle's orientation are not possible. Again, we report segmentation results for a range of images and quantify robustness to image noise. Finally, we demonstrate the seamless extension of this framework, through the idea of visual contexts, for the detection of artificial objects and/or structures, and illustrate several examples of such additional segmentation. This extension thus enables mission profiles that require, for example, following a specific road or the tracking of moving ground objects. Throughout, our approach and algorithms are heavily influenced by real-time constraints and robustness to transient video noise.

Index Terms—vision-based control, real-time control, image segmentation, object recognition

I. INTRODUCTION

OVER the past several years, much progress has been made towards the development of small-scale aircraft, known broadly as *Micro Air Vehicles* or *MAVs*. As these systems and the miniaturized electronics that make MAVs

possible are maturing, interest in MAVs has accelerated substantially for applications ranging from battlefield surveillance, smart munitions and real-time bomb damage assessment, to forest-fire reconnaissance, surveys of natural disaster areas and inexpensive traffic and accident monitoring. At the University of Florida, researchers in aerospace and computer engineering have established a long track record in designing, building and test-flying innovative and rugged MAV and small UAV flight vehicles [1]–[4]; for example, Figure 1 shows some of our recently developed MAVs.

Up until recently, these platforms were exclusively remotely piloted, with no autonomous or intelligent capabilities. To develop these capabilities in MAVs, we are faced with some challenges unique to small-scale aircraft. First, compared to their more traditional and larger model aircraft cousins, MAVs are much less stable and harder to control. Second, wind gusts can typically be equal to or greater than the forward air speed (e.g. 10-15 m/s) of the MAV itself. Thus, an average wind gust can immediately affect a dramatic change in the vehicle's flight path. Third, MAVs have a stringent payload limit; for example, our six-inch (15cm) MAV platform (pictured in Figure 1) weighs only approximately 60g. As such, even with the ongoing miniaturization of electronic components, sensors that are available for larger platforms are not currently practical for use on some of our smaller MAVs. In choosing sensors appropriate for MAVs, we must always balance the value (i.e. information content, accuracy) of a given sensor with the consequent weight penalty associated with that sensor.

The one sensor that is critical to most conceivable MAV missions, such as remote surveillance, is an on-board video camera with a transmitter that streams the video to a nearby ground station. Exploitation of this rich and important sensor is therefore desirable, since no additional on-board hardware (and weight) is required. In the literature, recently, several on-board vision systems for MAVs and unmanned aerial vehicles (UAVs) have been reported. For example, in [5], the authors present a real-time vision system for a rotor-craft UAV to land



Fig. 1. (a) six-inch (15cm) UF MAV; (b) six-inch (15cm) UF MAV in flight with view through its on-board camera; (c) 24-inch (61cm) MAV.

Manuscript received July 11, 2003;

The authors are with the Department of Electrical and Computer Engineering, Benotn Hall, University of Florida, Gainesville, FL 32611-6200, e-mail: {sinisha, nechyba}@mil.ufl.edu, <http://mil.ufl.edu/mav>

onto a known landing target. Their algorithms include linear and nonlinear optimization schemes for model-based camera pose estimation. Next, in [6], a vision-based hierarchical approach to path planning of UAVs is proposed, where a UAV builds and updates a virtual 3-D model of the surrounding environment by processing image sequences. Further, in [7], the authors present a motion-sensing visual system for UAVs and MAVs to follow terrain and avoid obstacles. In their approach, signals from the 20-photoreceptor on-board eye, together with inertial and rotor RPM signals, are processed in real-time to achieve automatic terrain-following flights. The mentioned prior work demonstrates significant improvements in flight control and navigation of MAVs and UAVs, when visual information is introduced as an input to the controls system. Despite these successes, however, the principal limitation of the mentioned efforts is their narrow applicability; that is, these systems have been designed to achieve specific goals without modularity or readily implementable extensions.

Therefore, in this paper we develop a general and unified computer vision framework for MAVs that not only addresses basic flight stability and control, but enables more intelligent missions, such as moving-object tracking and localization, as well. Figure 2(a) gives an overview of our MAV computer vision system, while Figure 2(b) shows an example of the desired output for the system. First, we seek to extract relevant features from the flight video that will enable our higher level goals. Then, we apply this image representation towards horizon detection and sky/ground segmentation for basic flight stability and control. Finally, we extend this basic framework through the idea of visual contexts to detect artificial objects and/or structures on the ground.

Horizon detection and tracking enable basic flight control through the recovery of the two degrees of freedom most critical for stability – namely, the *bank angle* and the *pitch angle*. Sky/ground segmentation disambiguates which regions in the horizon-detected image correspond to ground and sky, respectively. While under normal flight conditions and with proper initialization, this step may not always be necessary, other times it will be critical towards properly estimating the state of the MAV. Consider, for example, a MAV that is ejected from some munition prior to impact for immediate bomb damage assessment. Upon ejection, we cannot guarantee that the orientation of the MAV will be right side up. Sky/ground segmentation also provides a better estimate of the two dominant regions when the horizon does not correspond to approximately a straight line, as is the case, for example, when flying at low altitudes.

Once we have a basic two-region segmentation of the flight video, we can ensure basic stability and control of the MAV, without any additional inertial sensors. Moreover, this segmentation offers important *contextual* priors to further segmentation and detection algorithms. For example, roads cannot appear in the region identified as sky, while other flight vehicles will most likely appear in the identified sky region. Therefore, depending on the particular mission profile for the MAV, our initial sky/ground segmentation, necessary for stability and control, also narrows the search space for subsequent segmentation necessary to implement intelligent

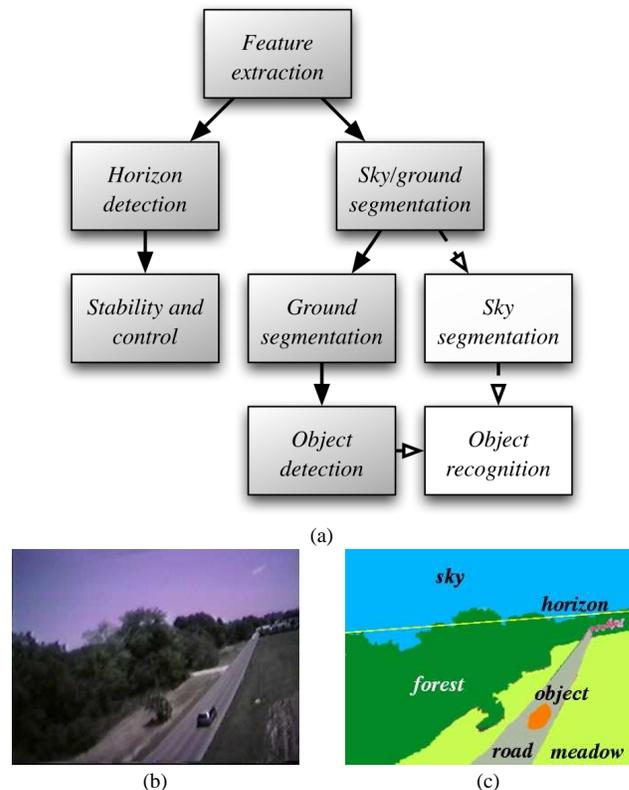


Fig. 2. (a) Overview of the MAV computer vision system (white blocks indicate future work); (b) MAV flight image; (c) sample desired output of the system for the image in (b).

MAV behaviors. In traffic monitoring, for example, we may wish the MAV to follow a flight path over a specific highway, or track a particular make and model car. While we do not specifically address object recognition in this paper, we do provide examples of detection of *artificial objects*, such as a road and a minivan.

The most important factors that inform our choices in designing the above vision system are: (1) real-time constraints, (2) robustness to noise, and (3) a careful examination of those features that lead to reliable segmentation for our application. Below, we review some of the recent literature in natural scene analysis that inform our choices in light of these constraints.

In [8], an unsupervised segmentation algorithm based on Markov random field models for color textures was presented. The authors provided experimental results that illustrated the advantages of using color texture models on color images of natural scenes. With regard to texture, Randen *et al.* [9], reported that for images where many textures with subtle spectral differences exist, as in our case, spectral decomposition by a filter bank (i.e. a multiscale approach) yields consistently superior results over other texture analysis methods. Along these lines, in [10], wavelet-domain hidden Markov models for processing natural scene images were shown to be very efficient, especially in the presence of video noise. On the other hand, recent findings on human vision and natural image statistics undermine these arguments for a wavelet-based approach in image analysis [11], [12]. It was reported that unlike wavelets, cortical cells are highly sensitive to

orientation and elongation, along with the location and scale of stimuli. Moreover, the basis elements which best “sparsify” natural scenes are, unlike wavelets, highly direction-specific. Hence, recently many new image analysis methods, such as wedgelets, ridgelets, beamlets, etc. have been proposed [13], [14]. Aside from the multiscale and localization properties of wavelets, these methods exhibit characteristics that account for concepts beyond the wavelet framework, as does our approach to feature extraction and image representation – namely, *Multiscale Linear Discriminant Analysis (MLDA)* [15].

While we defer a detailed discussion of MLDA until Section II, we note that MLDA provides a harmonic, multiscale representation of images, consisting of piecewise constant regions separated by smooth boundaries. It explicitly incorporates color in its feature representation, while implicitly encoding texture through a dynamic representation of image details. Furthermore, this representation substantially compresses the information content of images, while simultaneously preserving essential details in the image. Not only have we shown that this approach runs in real-time, but MLDA’s sparse representation of images also significantly speeds up subsequent stages in the computer vision system.

Given our feature representation (i.e. MLDA), we must choose a statistical modeling framework that allows us to perform the various segmentation tasks shown in Figure 2. Two main types of prior statistical models have been investigated in the image modeling literature – namely, noncausal and causal Markov random fields (MRF). The most commonly used form of these models is a tree-structured belief network (TSBN). For example, in [16], image segmentation was performed using Bayesian image analysis with TSBNs as prior models. To connect the labeled fields to the image data, the authors used local predictions of pixel classifications from neural networks. Also, in [17], a TSBN-based algorithm for man-made structure detection in natural scene images was proposed. Given the prior success of TSBNs in natural scene analysis and segmentation, we also rely on TSBNs in our segmentation work.

As with MLDA, we defer a detailed discussion of TSBNs in our work (see Section IV). At this point, however, the reader may wonder whether simpler statistical models would suffice for our tasks. Let us examine the sky/ground segmentation problem. Although modeling sky and ground regions in images may seem intuitively easy, it is, in fact, a very challenging task. Depending on lighting, weather, landscape, etc., the appearance of the sky and ground can vary enormously. Given the complex variations in our two image classes (i.e. sky and ground), our experimental results suggest that it is important to represent both local as well as regional interdependencies in the feature space [18], [19]. Therefore, our choice of TSBNs appears to be well justified, since TSBNs can describe neighborhoods in an image at varying scales. Moreover, the inference algorithm for TSBNs, known as Pearl belief propagation [20], is accurate and fast, and can be executed in real-time.

Our paper is organized as follows. In Section II, we define MLDA and motivate its applicability to our system. Next, in Section III, we show how MLDA naturally allows us to perform horizon detection, and we report results on vision-

stabilized flights. Then, in Section IV, we discuss the problem of sky/ground segmentation and our application of TSBNs to this task. Finally, in Section V, we demonstrate additional ground segmentation, detecting the road and an object (i.e. minivan) on the road, under the same framework as was developed for flight stability and control.

II. MULTISCALE LINEAR-DISCRIMINANT ANALYSIS

For our task, we seek to identify features that lead to improved segmentation performance without unnecessarily increasing computational complexity. Our feature selection was largely guided by extensive experimentation [18], from which we conclude that a prospective feature space must:

- 1) span both color and texture domains,
- 2) analyze both local and regional properties of pixels,
- 3) have high directional selectivity, and
- 4) meet real-time constraints.

As we discussed in Section I, these requirements might be satisfied using the wavelet transform, a popular multiscale orientation-selective filtering method. However, we consider other image analysis tools, also mentioned in Section I, due to significant shortcomings in the wavelet treatment of edge structure, which is of particular interest for our horizon-detection task. In Figure 3(a), we show that large wavelet coefficients cluster around smooth contours, which renders the wavelet transform inefficient for representing geometric regularity [13], [21]. Thus, wavelets do not support our basic assumption that images consist of smooth regions separated by smooth contours. Moreover, in the presence of video noise, wavelets exhibit notorious inefficiency in differentiating true edges from noise edges in the image. In Figure 3(b), we illustrate that due to video noise, wavelets detect false ripples and edges, which could cause substantial errors in horizon detection.

Recently, we proposed Multiscale Linear Discriminant Analysis (MLDA) as a feature extraction method [15]. Not only does MLDA overcome the shortcomings of wavelets, but it also incorporates color information (unlike wavelets),

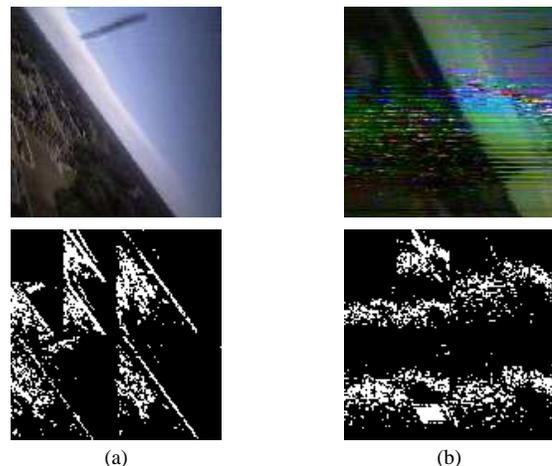


Fig. 3. Two-level wavelet transform: (a) clustering of wavelets with large magnitudes; (b) wavelets are not resilient to bursts of video noise.

satisfying the aforementioned requirements for the feature space. An MLDA atom w is a piecewise constant function on either side of a linear discriminant that intersects a square in vertices v_i and v_j , as illustrated in Figure 4. The discriminant (v_i, v_j) divides the square into two regions with values μ_0 and μ_1 , the mean vectors of pixel values $[red, green, blue]^T$ in RGB color space of the two regions. Decomposing the image into a number of dyadic squares and finding their corresponding MLDA atoms, we obtain the MLDA dictionary over a finite range of locations, orientations and scales, as shown in Figure 4. The MLDA image representation – that is, our feature extraction – is performed by searching through the MLDA dictionary for the atoms that best represent the analyzed image with respect to two important criteria: (1) *discrimination* and (2) *parsimony*.

Addressing the first criterion (i.e. discrimination), we seek the direction (v_i, v_j) , characterized by the maximum Mahalanobis distance J between two regions in a square, where

$$(\hat{v}_i, \hat{v}_j) : J = \max_{(v_i, v_j)} \{(\mu_0 - \mu_1)^T (\Sigma_0 + \Sigma_1)^{-1} (\mu_0 - \mu_1)\}, \quad (1)$$

and Σ_0 and Σ_1 denote RGB covariance matrices for the two regions.

Note that the computational cost of an exhaustive search over a finite set of linear discriminants $\{(v_i, v_j)\}$ can be reduced by updating the relevant statistics only with pixel values of delta regions (areas between two consecutive candidate linear discriminants).

As the size of squares decreases, we achieve better piecewise linear approximation of boundaries between regions in the image, as illustrated in Figure 5. Therefore, the analyzed image is decomposed into dyadic squares of varying sizes which results in the MLDA tree \mathcal{T} . The MLDA tree \mathcal{T} is *incomplete*, because atom generation stops at different scales for different locations in the image. The leaf nodes of \mathcal{T} store the final MLDA image representation.

To control the generation of children dyadic squares, we impose the second optimization criterion (i.e., parsimony) as

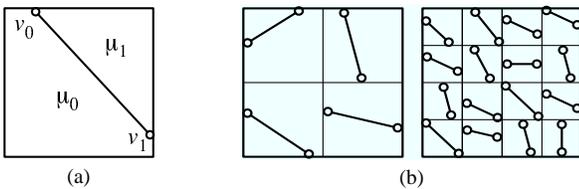


Fig. 4. (a) MLDA atom; (b) MLDA dyadic decomposition.

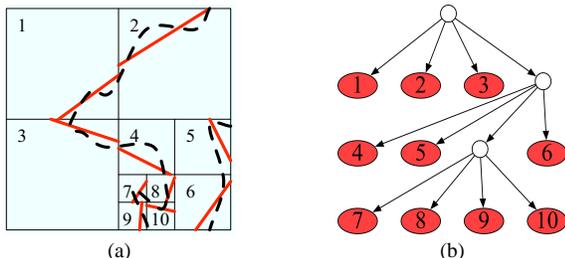


Fig. 5. (a) MLDA image representation: the dashed line depicts the actual curve; (b) corresponding MLDA tree: ellipsoid nodes represent the leaf MLDA atoms.

a counter-balance to accuracy. We define the cost function $R(\mathcal{T})$ to measure the parsimony of \mathcal{T} ,

$$R(\mathcal{T}) = \sum r(w_t) + \alpha |\mathcal{T}|, \quad (2)$$

where $r(w_t)$ is the inverse of the Mahalanobis distance computed for the corresponding terminal node w_t , $r(w_t) = 1/J$, $|\mathcal{T}|$ denotes the number of terminal nodes in \mathcal{T} , and α represents the complexity cost per leaf node. Clearly, an exhaustive search in tree space for the minimum cost function is computationally prohibitive. Therefore, we implement the one-step optimization procedure [22].

Instead of stopping at different terminal nodes, we continue the MLDA decomposition until all leaf squares are small in size, resulting in a large tree. Then, we selectively prune this large tree upward using the cost function $R(\mathcal{T})$. From expression (2), it follows that we can regulate the pruning process by increasing α to obtain a finite sequence of subtrees with progressively fewer leaf nodes. First, for each node $w \in \mathcal{T}$, we determine α_w for which the cost of a subtree \mathcal{T}_w is higher than the cost of its root node w , as follows:

$$\begin{aligned} R(\mathcal{T}_w) \geq R(w) &\Rightarrow \sum r(w_t) + \alpha_w |\mathcal{T}_w| \geq r(w) + \alpha_w \cdot 1, \\ &\Rightarrow \alpha_w = \frac{r(w) - \sum r(w_t)}{|\mathcal{T}_w| - 1}. \end{aligned} \quad (3)$$

Then, the whole subtree \mathcal{T}_w under the node w with the minimum value of α_w is cut off. This process is repeated until the actual number of leaf nodes is equal to or less than the desired number of leaf nodes.

We note that the pruning process described above is computationally fast and requires only a small fraction of the total tree construction time. Starting with a complete tree, the algorithm initially trims off large subtrees with many leaf nodes. As the tree becomes smaller, the procedure tends to cut off fewer nodes at a time. In Figure 6, we illustrate the efficiency in image representation of the fully optimized MLDA tree, as compared to the unpruned MLDA tree. While there is almost no degradation in accuracy with complexity-cost pruning, we achieved a significant reduction in the number of terminal MLDA atoms. Note that the pruned MLDA representation in Figure 6 consists of only $2 \cdot 128 = 256$ mean RGB vectors, and yet the image information is rich enough for subsequent object recognition and segmentation. Thus, applying MLDA as a feature extractor, we can significantly reduce redundant image information, so as to be able to meet real-time constraints. At the same time, we incorporate our basic assumptions for

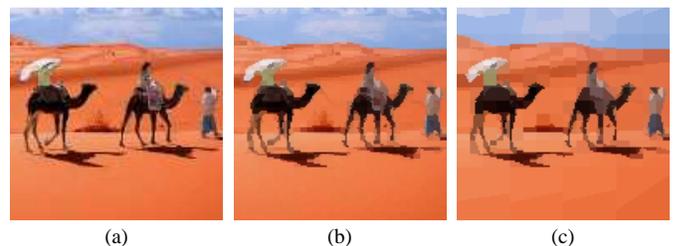


Fig. 6. MLDA efficient image representation: (a) original 256×256 image – 65,536 pixels; (b) no pruning – 1024 leaf nodes; (c) with pruning – 128 leaf nodes.

horizon detection – namely, that (1) the horizon line appears as approximately a straight line, and that (2) the horizon line separates the image into two homogeneous regions.

Clearly, the MLDA representation is appropriate for computer-vision tasks where color is critical. Though not as obvious, MLDA also implicitly encodes information about spacial frequencies in the image (i.e. texture) through the process of tree pruning. Furthermore, the MLDA tree can easily be examined for spacial interrelationships of its linear discriminants, such as connectedness, collinearity and other properties of curves in images.

At this point, the observant reader might conclude that the horizon-detection problem is solved by computing the root atom of the MLDA tree. Note that given our two horizon assumptions specified above, the extracted linear discriminant (\hat{v}_i, \hat{v}_j) is the optimal solution for the horizon estimate. This seemingly obviates the need for MLDA tree expansion and pruning. There are, however, several non-ideal factors that argue for the full tree construction process. In the presence of noise, the root node by itself may not provide reliable pixel statistics. Therefore, in order to detect the horizon, it is necessary to examine discriminants at finer scales of the MLDA tree. For example, in Figure 7, we show that the discriminant of the root MLDA atom does not coincide with the true horizon due to video noise. Expanding the MLDA tree corresponds to image filtering and leads to more accurate positions for the linear discriminants, which then present more accurate evidence of the true horizon’s location in the image.

Had we employed some de-noising algorithm before MLDA and had horizon detection been our sole task, we could, of course, have omitted the construction of the MLDA tree and saved critical time for flight control. As we shall see in subsequent sections, however, the MLDA tree representation gives rise to a host of multiscale classification algorithms for sky/ground segmentation and object detection. In other words, while MLDA is a natural representation for horizon detection, it also supports more advanced computer vision tasks that are desirable for more complex mission scenarios. As such, it plays a dual role in our overall system.

III. HORIZON-DETECTION ALGORITHM

A. Introduction

In this section, we illustrate how MLDA performs in horizon detection for flight stability and control. The two degrees of

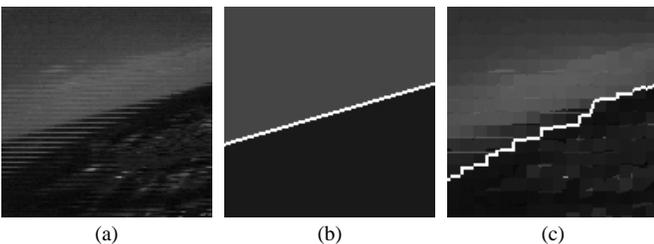


Fig. 7. MLDA efficiently filters video noise: (a) noise degraded original image; (b) root MLDA atom with the discriminant not equal to the true horizon; (c) MLDA atoms at a finer scale are clues for the true horizon position.

freedom critical for stability – namely, the *bank angle* ϕ and the *pitch angle* θ^1 – can be derived from a line corresponding to the horizon as seen from a forward facing camera on the flight vehicle. Thus, accurate horizon detection is essential for our flight stability and control system, requiring no additional inertial sensors.

As we mentioned at the end of the previous section, the MLDA framework offers an efficient solution to the horizon detection problem without prior image processing, such as de-noising, etc. First, the MLDA image representation is found for the cost function R , as defined in equation (2). The precise value of R is largely dependent on the video transmission quality for a given location, weather conditions, etc. Then, linear discriminants are extracted by exhaustively analyzing all possible directions, determined by a finite set of points \mathcal{V} along the perimeter of the image. Each pair of points $(v_i, v_j) \in \mathcal{V}$ defines a direction with slope a_{ij} along which MLDA atoms are examined. If a_{ij} coincides with the slope of an MLDA discriminant (within a predefined margin), then that linear discriminant is extracted as a part of the analyzed direction (v_i, v_j) . Finally, among the candidate sets of the extracted MLDA discriminants, we choose the direction (\hat{v}_i, \hat{v}_j) for which the sum of Mahalanobis distances J , computed as in (1), is maximum. For the special case of only one MLDA node, the algorithm simplifies to examining a single MLDA discriminant, and works well for relatively clean video. Given the solution (\hat{v}_i, \hat{v}_j) , we trivially obtain the corresponding *bank angle* and *pitch percentage* pair $(\hat{\phi}, \hat{\rho})$, which is then used for flight control.²

B. Horizon-detection examples

Figure 8 illustrates several examples of the horizon-detection algorithm at work, while Figure 9 illustrates a more detailed example, plotting J as a function of the bank angle and pitch percentage, and the consequent classification of pixels as sky and ground in RGB space. Figure 10 illustrates two further examples of horizon detection for two particularly noisy flight images. In all cases, we show only the set of extracted linear discriminants, from which the horizon position and orientation are derived. Additional examples and videos can be found at <http://mil.ufl.edu/~nechyba/mav>.

Our horizon-detection algorithm has been demonstrated to run at 30Hz on a 2.4GHz x86 processor with a down-sampled image resolution of 128×128 . To accomplish real-time processing, vertices, v , along the perimeter of the image, are set 4 pixels apart ($V = 32$ vertices on one side), which results in the following number of (v_i, v_j) pairs, $n^0(V)$, at the root level of the MLDA tree:

$$n^0(32) = \left(4 \cdot \binom{V-1}{2} \right) - 4 \cdot \left(\binom{V}{2} \right) \Big|_{V=32} = 5642. \quad (4)$$

¹In practice, we actually recover the closely related pitch percentage ρ , which measures the percentage of the image above the horizon line.

²The transformation from (\hat{v}_i, \hat{v}_j) to $(\hat{\phi}, \hat{\rho})$ assumes that the on-board camera is aligned with the principal axes of the flight vehicle. In that case, ϕ for the flight vehicle corresponds to the angle between the horizon line and the horizontal axis, and the pitch percentage ρ corresponds to the fraction of pixels in the sky region over the total number of pixels in the image [23].

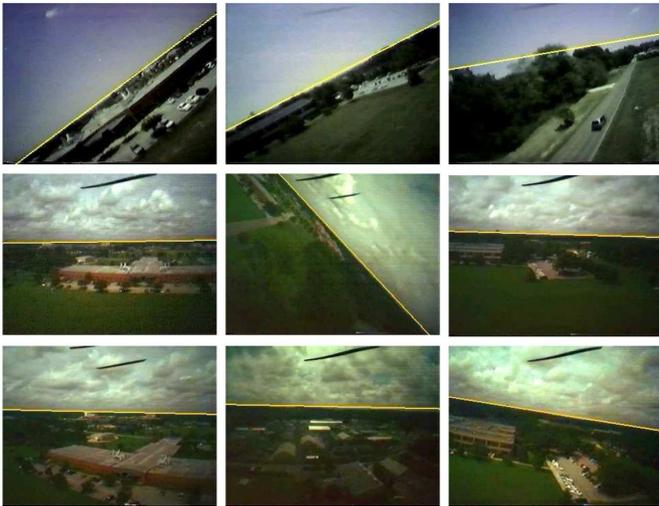


Fig. 8. Various horizon-detection examples under different lighting conditions (sunny and cloudy), and with varying degrees of video transmission noise. For each example, the yellow line indicates the algorithm’s horizon estimate.

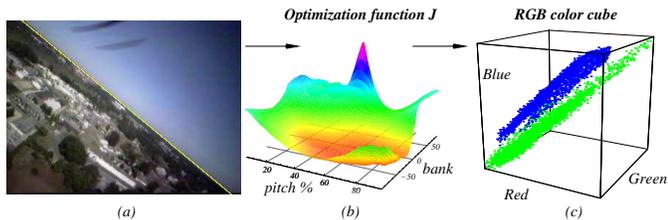


Fig. 9. (a) original image, (b) optimization criterion as a function of bank angle and pitch percentage and (c) resulting classification of sky and ground pixels in RGB space.

Of course, computational complexity increases when examining additional levels of the MLDA tree. If computing power is more limited, we have shown only slightly reduced performance for 64×64 images, where the root search resolution is only $n^0(16) = 1290$.

At different times of the day, and under both fair and cloudy conditions, we have gathered hours of video on-board our MAV, flying under manual control over terrain that includes roads, buildings large and small, meadows, wooded areas, and a lake. For these real-world data, our horizon-detection algorithm correctly identifies the horizon in over 99.9% of cases.

We have conducted several types of performance characterization experiments, where we have systematically tested our algorithm’s resilience to typical aberrations in flight images, such as noise degradation and color rotation. First, adding a uniformly distributed random variable, $U[-a, a]$, to RGB pixel values, we have measured the detection error as a percentage of the image area between the true and estimated horizon positions. The averaged detection results shown in Figure 11 are obtained for the image in Figure 9. Obviously, MLDA is very resilient to moderate noise amplitudes and multilevel MLDA handles image noise more efficiently than MLDA with fewer levels. Similar results have been obtained for additive Gaussian image noise. In the second type of experiments, we have considered chromatic involution, that is, color mappings,

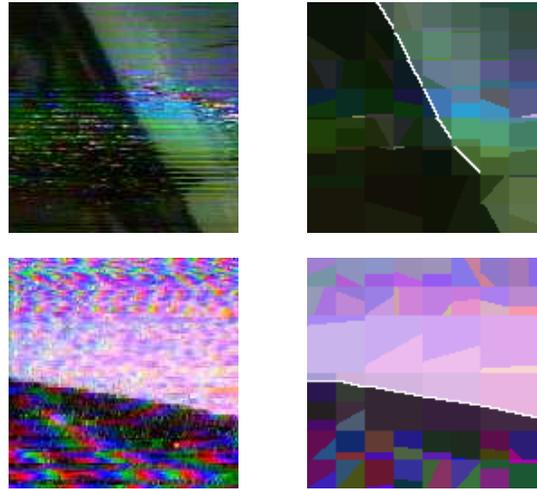


Fig. 10. Noise-degraded MAV flight images: first, the MLDA image representation is found; then, the linear discriminants are extracted along the direction with the maximum Mahalanobis distance.

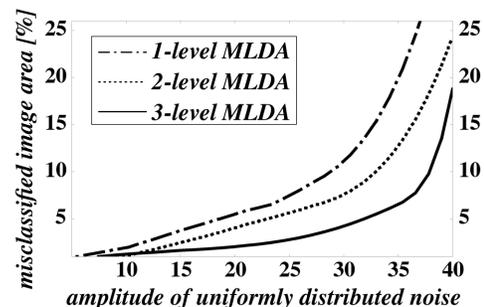


Fig. 11. Averaged detection error: percentage of the image area between the true and detected horizon positions as a function of the amplitude a of the uniformly distributed additive noise $U[-a, a]$. Horizon detection is performed by 1-, 2-, and 3-level MLDA for the image in Fig. 9.

where (1) brightness, (2) relative angles between colors and (3) saturation are unchanged. From our results we conclude that MLDA-based horizon detection is invariant to chromatic involution. Moreover, MLDA is invariant to any isotropic transformation in the RGB color space that keeps relative distances among pixel values intact.

C. Recovery from video noise

At this point, the reader might be wondering whether a full search of the $\mathcal{V} = \{(v_i, v_j)\}$ line-parameter space for each image in the video sequence is really required once flying, since the horizon at the current time step should be very close to the horizon at the previous time step; perhaps speed improvements could be made by limiting this initial search. There is, however, at least one important reason for conducting a full search with every frame, rather than just a partial search in the neighborhood of the previous frame’s horizon estimate. Assume, for example, that the algorithm makes an error in the horizon estimate at time t ; then, at time $t + 1$, a limited search could permanently lock us into the initial incorrect horizon estimate, with potentially catastrophic results.

We illustrate this idea in Figure 12, where we show four frames from a flight sequence over the University of Florida

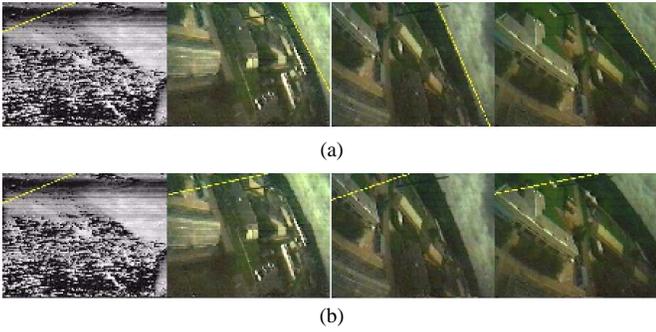


Fig. 12. (a) Video noise in the first frame does not influence subsequent horizon estimation when performing a full search of the line-parameter space for every frame; (b) when consecutive searches are restricted to a limited range around the previous frame's horizon estimate, however, the horizon detection algorithm is not able to recover from the first noise-corrupted estimate.

campus. Note that the first frame is severely corrupted by video transmission noise, so that the horizon estimate for the first frame in the sequence is far from the true horizon. In this case, deepening of the MLDA tree does not yield better performance; therefore, here, we present analysis only of the root node. In Figure 12(a), a full search is conducted for every frame, so that the noise-corrupted estimate of the first frame does not influence subsequent horizon estimates. In Figure 12(b), however, we restrict the search for the horizon line to a limited range around the previous frame's horizon estimate; note that with the limited search, subsequent horizon estimates, continue to exhibit large error. Thus, a full search of the line-parameter space guards against this type of cascading failure due to single-frame errors.

Finally, we point out that since we perform our full search through the line-parameter space \mathcal{V} for a single frame such that consecutive horizon hypotheses are very close to one another, we need only compute the full statistics for the Mahalanobis distance, as in (1), for the first horizon hypothesis [24]. Subsequent computations are incremental, and typically involve only a few pixel values that have to be added to one class and subtracted from the other class. As such, when implemented efficiently, the full search through line-parameter space is significantly less computationally burdensome than might be expected.

D. Self-stabilized flight

We have previously implemented closed-loop control, based on horizon detection and tracking, for one of our MAVs [23], [25]. Figure 13 illustrates the experimental setup for these test flights. The video signal from the MAV is transmitted from the plane through an antenna to a ground-based computer, where all vision processing is performed. In manual mode, the plane is controlled during flight by a remote human pilot through a standard radio link. In autonomous mode, the human input is limited to providing a desired heading (through a joystick interface), while the MAV is stabilized through a PD feedback controller which sends control surface commands to the MAV through a custom designed interface over the same radio link. The MAV used for these test flights is the one depicted in Figure 1(c).



Fig. 13. Experimental setup for video-based flight control.

Due to computational limitations (900 MHz x86 processor) at the time, horizon detection during these flight tests was conducted with only the MLDA root node. This limitation forced the introduction of an *ad hoc* error/noise detection scheme that was based on short-term histories of sky and ground appearances [23], [25]. Furthermore, since sky/ground modeling was not integrated into the system, prior to launch, the MAV had to be oriented right-side up such that the horizon was in the field-of-view of the on-board camera and the sky occupied the top part of the video image. Finally, horizon estimates over time were passed through a simple Kalman filter to reduce control surface jitter.

Figure 14(a) plots a 72-second run of actual flight data, where the flight vehicle was under vision-guided control above the University of Florida campus (the full length of the flight exceeded 10 minutes, and was primarily limited by low battery power). During this flight, the MAV was instructed to execute a trajectory that consisted of straight line segments, followed by left-bank turns (to keep the MAV within range of the receiving video antenna). For comparison, we also plot a 65-second segment of manual (human-controlled) flight in Figure 14(b). (Videos corresponding to these and other flight segments can be viewed at <http://mil.ufl.edu/~nechyba/mav/>) The same vision-based control system also successfully flew over substantially different terrain at a Special Ops demo over Fort Campbell, Kentucky, where audience members, who had never previously controlled any type of aircraft (e.g. model airplane, MAV, etc.) successfully kept the MAV in the air for extended flight times.

From Figure 14, we observe that the human-controlled flight is much more erratic than the self-stabilized flight with respect to both the bank angle and pitch percentage. It is important to note that the erratic nature of the human-piloted flight is not caused by inexperience on the part of the pilot, who had flown our MAVs and small UAVs for many hours over a number of years; in fact, he was one of our most experienced pilots. Rather, the seemingly erratic nature of the human-piloted flight indicates that controlling MAVs and small UAVs is significantly more difficult than controlling larger, more traditional model airplanes, which tend to exhibit much more

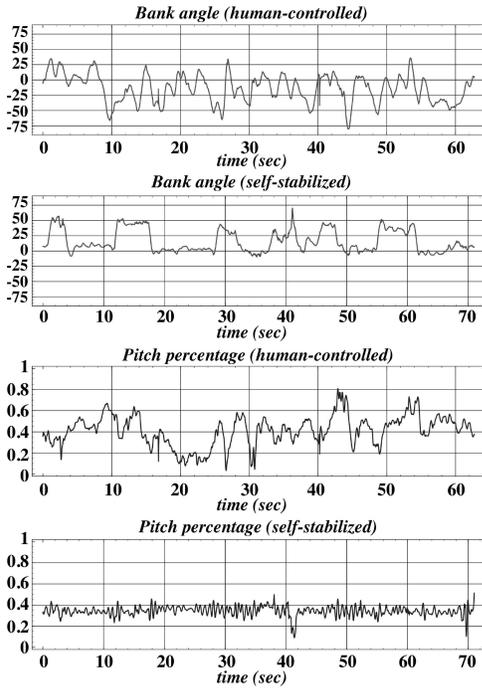


Fig. 14. Bank angle and pitch percentage for a typical human-controlled flight and self-stabilized flight (sequence of level-flight and left-turn segments).

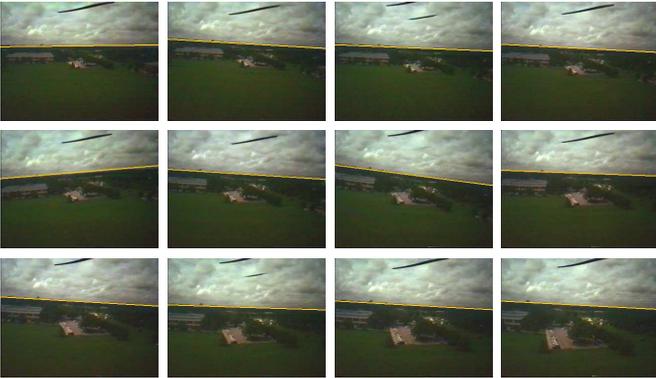


Fig. 15. Image sequence of self-stabilized level flight (7.5 seconds, images are 20 frames apart). Note that the black lines at the top of the image are the propeller.

passive stability.

Thus, qualitatively, even our simple PD control system provides much more stable control than that of our best human pilots, both in terms of steady, level flight, and in coordinated turns. As illustrated by Figure 14(b), human pilots can typically not hold the plane on a steady, level heading for more than a few fractions of a second; under vision-guided control, however, we are able to fly long straight segments that are limited only by the range of the video transmitter (see Figure 15, for example, for a 7.5 second stretch of self-stabilized straight and level flight).

IV. SKY/GROUND SEGMENTATION

As we have seen in the previous section, overall, the horizon-detection algorithm works well. The algorithm does not, however, determine which region corresponds to sky and

which corresponds to ground; it only determines the line separating the two regions. Under benign flight conditions, we can mitigate this deficiency by assuming that initially the sky occupies the upper part of the image. For complex mission scenarios, however, this may be an incorrect assumption with potentially fatal consequences to the flight vehicle. For example, we are currently working on deploying MAVs on munitions for post-impact bomb damage assessment. In this case, the MAV would separate from the munition prior to impact, and an upright attitude with respect to the ground cannot be guaranteed. Correct identification of the sky and ground regions, rather than just the line separating them, therefore, takes on increased importance.

Our approach to sky/ground image segmentation is based on building statistical prior models of both classes. The parameters of the prior models are learned through extensive training, which can be performed off-line³. Once learned, the parameters fully characterize the likelihoods of image classes, given the pixel values. These likelihoods are then submitted to a Bayes classifier, which performs image segmentation [26].

We choose tree-structured belief networks (TSBNs) as the underlying statistical models to describe the sky and ground classes. There are several reasons for this choice of model. Successful sky/ground segmentation implies both accurate classification of large regions, as well as distinct detection of the corresponding boundaries between them. To jointly achieve these competing goals, both large and small scale neighborhoods should be analyzed, which can be achieved using the multiscale structure of TSBNs. Further, as discussed in the Introduction, it is necessary to employ a powerful statistical model that is able to account for enormous variations in sky and ground appearances, due to video noise, lighting, weather, and landscape conditions. Prior work cited in the Introduction clearly suggests that TSBNs possess sufficient expressiveness for our goals. Finally, TSBNs, as opposed to other multiscale statistical models, such as factorial hidden Markov models, dynamic trees, etc., have a fixed-tree structure for which the corresponding inference algorithms are computationally very efficient [20], [27], which is of crucial importance, given our real-time constraints.

A. Tree-structured belief networks

The TSBN consists of nodes i , say at the scale ℓ , vertically connected to one parent $\rho(i)$ at the coarser scale $\ell - 1$, as well as to its four children $\chi(i)$ at the finer scale $\ell + 1$, as depicted in Figure 16. As is customary for TSBNs [16], [28], to each node i we assign an observable random variable y_i , which, in our case, represents an MLDA atom at the corresponding position in the MLDA tree. Recall that the MLDA tree is incomplete (see Figure 5), because atom generation stops at different scales for different locations in the image. Consequently, the TSBN model is incomplete as well.

The distribution of the observable variable y_i is controlled by the hidden state variable x_i , which takes values in a set of C image classes. We assume that y_i is conditionally independent of all other observable and hidden random variables given

³In this case, there are no real-time constraints for training.

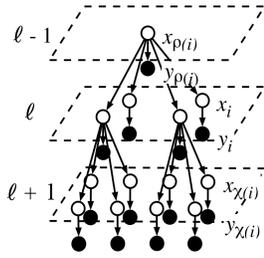


Fig. 16. The TSBN model: hidden state variables are depicted as white nodes and observable variables (MLDA atoms) are represented as black nodes.

its associated state x_i . Furthermore, we impose the Markov property such that x_i is conditionally independent of the entire tree, given its parent state $x_{\rho(i)}$. In Figure 16, edges connecting nodes depict these statistical interdependencies, forming the recognizable tree structure.

If we assume an M -component Gaussian mixture density for likelihoods $p(y_i|x_i)$, the TSBN is fully characterized by the following set of parameters Θ :

- 1) prior probability of the root node $P(x_0)$, $x_0 \in C$,
- 2) transition probability tables $P(x_i|x_{\rho(i)})$,
- 3) mean μ_m , covariance Σ_m , and priors p_m of the components in a mixture of Gaussians $N(y_i, \mu_m, \Sigma_m)$.

In order to simplify computation and to avoid the risk of overfitting the model, we assume that the statistical parameters at the same scale are equal for all nodes.

The TSBN model parameters Θ can be trained by the Expectation-Maximization (EM) algorithm [28], [29], using the Pearl belief propagation scheme [20], [27]. For space reasons, here, we omit the details of this algorithm. The EM algorithm iteratively estimates Θ , maximizing the likelihoods $p(y_i|x_i)$ at all scales. Once Θ is estimated for a given class, it is possible to compute for each node the following M -component Gaussian mixture:

$$p(y_i|x_i) = \sum_{m=0}^{M-1} p_m N(y_i, \mu_m, \Sigma_m). \quad (5)$$

Consequently, we are able to perform Bayesian classification. Denoting the collection of all pixel labels at the finest scale with $X^L = \{x_i^L\}$, where $x_i^L = \{sky, ground\}$, the optimal sky/ground segmentation \hat{X}^L is given by,

$$\hat{X}^L = \arg \max_{X^L} \{P(X|Y)\}, \quad (6)$$

$$= \arg \max_{X^L} \{P(X)P(Y|X)\}, \quad (7)$$

$$= \arg \max_{X^L} \left\{ \prod_{\ell=L}^1 \prod_i P(x_i^\ell | x_{\rho(i)}^{\ell-1}) p(y_i^\ell | x_i^\ell) \right\}. \quad (8)$$

In expression (8), $p(y_i^\ell | x_i^\ell)$ is readily computable from (5). Computation of the transition probabilities of class labels $P(x_i^\ell | x_{\rho(i)}^{\ell-1})$ is nontrivial, as discussed thoroughly in the literature [16], [19], [30]. Since the full derivation of the transition probabilities is beyond the scope of this paper, here, we assume that $P(X)$ is available, implementing a Viterbi-like algorithm, as proposed in our prior work [19].

B. Sky/ground segmentation examples

For training our statistical models, we recorded two sets of 500 sky and ground images. We carefully chose the training sets to account for the enormous variability within the two classes, as illustrated by the representative training examples in Figure 17. After experimenting with different image resolutions, we found that reliable segmentation was achievable for resolutions as coarse as 64×64 pixels. For each image, first the MLDA representation was found and then the corresponding TSBN model was trained. The training time for the 1000 images of both classes takes less than 40s on an 2.4GHz x86 processor. Depending on the number of levels in the MLDA tree, correct segmentation (i.e. testing) takes only 0.03s–0.07s, as shown in Figure 18.

Recall that the MLDA tree is incomplete; therefore, increasing the number of levels does not increase the computational cost exponentially. Clearly, the algorithm runs faster for a small number of MLDA terminal nodes, but at the cost of increased segmentation error. The number of MLDA-tree levels and the number of MLDA leaf nodes are parameters that are preset according to the desired performance goals. Note that while we are very close to meeting our 30Hz processing goal, 30Hz performance is not crucial for sky/ground segmentation, as long as this segmentation is updated sufficiently often. In between segmentation updates, horizon detection suffices for flight stability and control.

Having trained our sky and ground statistical models, we tested the algorithm on 300 natural scene images. For accuracy, we separated the test images into three categories of 100 images each: (1) easy-to-classify sky/ground appearances (e.g. clear blue sky over dark-colored land), (2) challenging images due to landscape variability, and (3) noise-degraded images. In Figure 19, we illustrate classification performance on representative test images from each category. To measure misclassification, we marked the “correct” position of the horizon for each image by visual inspection and then computed the percentage of erroneously classified pixels. For each category, we set the number of MLDA-tree levels to 6 and the number of leaf nodes to 64, 128, 32, respectively. In Table I, we

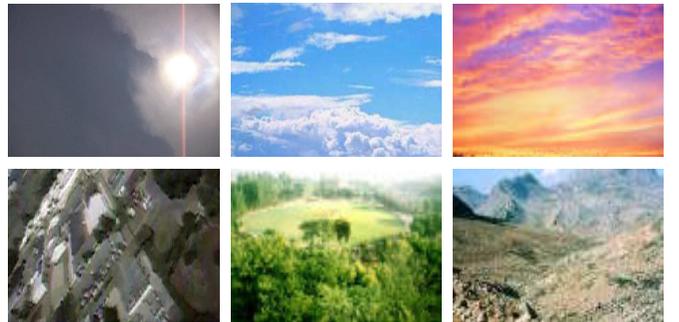


Fig. 17. Training images of the sky and ground classes.

TABLE I
PERCENTAGE OF SKY/GROUND MISCLASSIFIED PIXELS

category I	category II	category III
2% - 6%	2% - 8%	5% - 14%

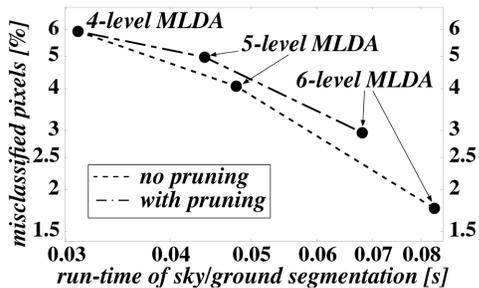


Fig. 18. Percentage of sky/ground misclassified pixels as a function of the run-time for 4-, 5- and 6-level MLDA and the 64×64 subsampled image in Fig. 9. For MLDA with pruning the number of leaf nodes is set to 64 regardless of the number of levels.

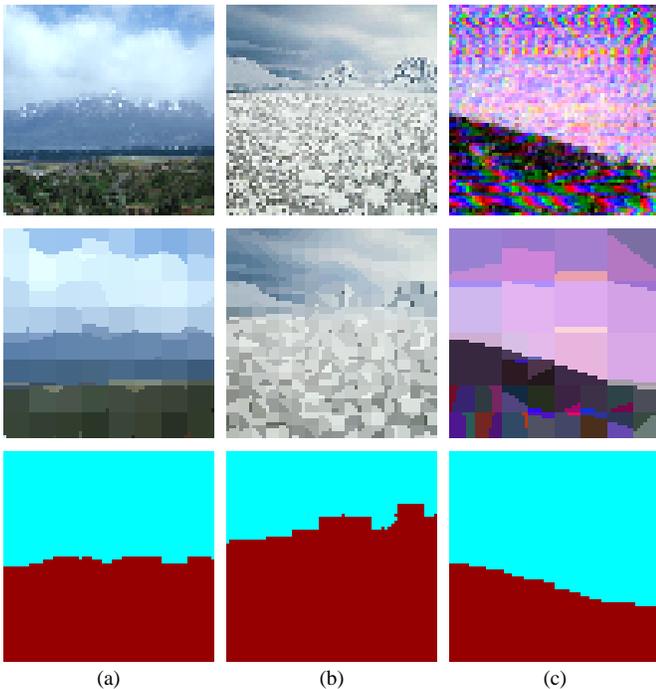


Fig. 19. Sky/ground segmentation for the three categories of test images: (a) mountain view (category I); (b) water surface covered with broken patches of ice similar in appearance to the sky (category II); (c) noise-degraded MAV flight image (category III).

summarize our segmentation results. Averaging results seemed inappropriate, because only a small number of test images in each category generated the larger error percentages.

Next, we tested the influence of uniformly distributed additive noise on sky/ground segmentation performance. We averaged the percentage of misclassified pixels when a uniformly distributed random variable, $U[-a, a]$, was added to RGB pixel values. The results presented in Figure 20 were obtained for the 64×64 subsampled category I image in Figure 9, where both 5- and 6-level MLDA with pruning had maximally 64 leaf nodes. Note that TSBNs for MLDA with pruning outperform TSBNs for MLDA without pruning for large amplitudes of additive noise. This phenomenon is a consequence of the better filtering properties of MLDA with pruning.

We also tested our algorithm on MAV flight videos. While a flight sequence does not offer the same variability in sky/ground appearances as our test images above, these results

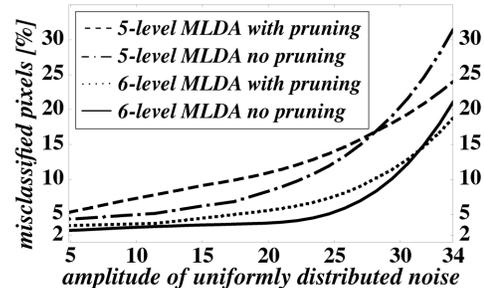


Fig. 20. Averaged segmentation error for the image in Fig. 9 as a function of the amplitude a of the uniformly distributed additive noise $U[-a, a]$. MLDA with pruning filters noise and achieves better performance than MLDA without pruning for large amplitudes of additive noise.

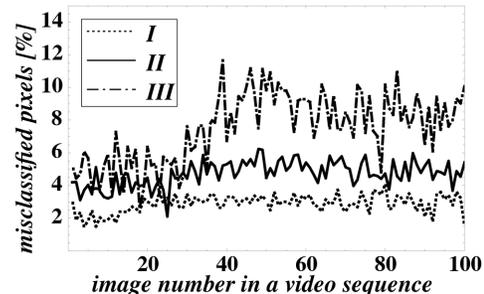


Fig. 21. Percentage of misclassified pixels for three video sequences of 100 images each.

can give us insight into the performance of the algorithm on a sequence of flight images with small changes from one image to the next. Here, we also consider three types of flight video: (1) easy-to-classify sky/ground appearances, (2) challenging landscapes, and (3) noise-degraded images. Each video sequence is down-sampled, such that the time interval between two consecutive images is 1s. The classification results for the three flight videos are presented in Figure 21. Obviously, the variance of the segmentation error is the highest for the third type of video. The increase in error for the third video type, starting at image number 40, coincides with a change in lighting conditions, which changed the “usual” color scheme of the sky and ground. The complete videos can be found at <http://mil.ufl.edu/~nechyba/mav>.

V. ENABLING SMART MAV MISSIONS

While horizon detection and sky/ground segmentation allow for basic flight stability and control, we have not yet considered trajectory planning of the MAV’s flight path in response to events on the ground (or possibly in the air). For example, a particular mission profile might require that the MAV track a moving vehicle on the ground, or fly above a specific highway. As such, we wish to extend our computer vision system for better situational awareness and *artificial* object detection (e.g. a car). However, rather than treat this problem separately from the algorithms for flight stability and control, our approach exploits these computations – both the feature representation (i.e. MLDA) and the whole-image segmentation into sky and ground regions. Hence, the overall system is not a patchwork of independent algorithms, but rather a unified framework for the various tasks in Figure 2.

In our approach to artificial object detection and localization, we seek to exploit the idea of *visual contexts* [31] – a low-dimensional representation of the whole image. Having previously identified the overall type of scene, we can then proceed to recognize specific objects/structures within the scene. Thus, objects (e.g., cars, buildings), the locations where objects are detected (e.g., road, meadow), and the category of locations (e.g., sky, ground) form a taxonomic hierarchy. There are several advantages to this type of approach. Contextual information helps disambiguate the identity of the object despite the poverty of scene detail in flight images. Second, visual contexts significantly reduce the search required to locate specific objects of interest, obviating the need for an exhaustive search for objects over various scales and locations in the image. Finally, as discussed in Section II, our low-dimensional image representation is very resilient to video noise, yielding better object-detection results.

For each image in a video sequence, we compute the MLDA representation, that is, MLDA atoms that represent observable random variables (RVs) $Y = \{y_i\}$. To each y_i , we assign a hidden RV x_i that represents *location*. Finally, we complete the hierarchical structure with the RV *category*, z_i , noting that such a taxonomy can easily be extended to meet different application requirements. In Figure 22, we present the proposed statistical model. Both location and category RVs take values in finite, application-oriented mutually dependent sets. For example, the category equal to “ground” encompasses the following values for locations: road, forest, lawn, apartment complex, etc. For notational simplicity, we denote with Y , X , and Z the collection of all MLDA atoms in the MLDA tree, their locations and their categories, respectively. Note that this approach is reminiscent of building the TSNB models for the sky and ground classes. Here, we are simply generalizing the meaning of classes to any label for locations and categories. Thus, the results from Section IV-A are readily applicable.

In order to perform localization, we resort to a Bayesian formulation, where our goal is to compute $P(X, Z|Y)$. According to the model in Figure 22, it follows that:

$$\hat{X}^L = \arg \max_{X^L} P(X, Z|Y) = \arg \max_{X^L} P(Y|X)P(X|Z)P(Z), \quad (9)$$

where L denotes the finest scale. As in Section IV-A, we assume statistical independence among y_i ’s given the location x_i , as well as independence among x_i ’s given the category z_i . The likelihoods $p(y_i|x_i)$ and $p(x_i|z_i)$ can be modeled as a

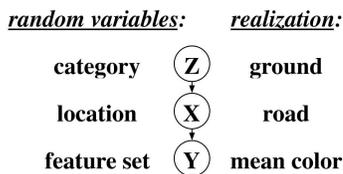


Fig. 22. A graphical model for location recognition and categorization: Y represents all observable MLDA atoms, X denotes their location values and Z the corresponding category. X and Z form a taxonomic hierarchy.

mixture of Gaussians. It follows that

$$\hat{X}^L = \arg \max_{X^L} \prod_{\ell=L}^1 \prod_i p(y_i^\ell | x_i^\ell) p(x_i^\ell | z_i^\ell) P(z_i^\ell | z_{\rho(i)}^{\ell-1}), \quad (10)$$

where $z_{\rho(i)}^{\ell-1}$ denotes the parent of z_i^ℓ . Recall that to learn all the model parameters, we resort to the EM algorithm, using the Pearl belief propagation scheme [20], [27]. Once the model is trained, we can perform location recognition, choosing the realization of the RV X^L for which $P(X, Z|Y)$ is maximum.

Clearly, the proposed statistical model, depicted in Figure 22, can be extended to enable the vision system to detect the presence of artificial objects. Had we used an image analysis tool for feature extraction other than MLDA, we could have used the new model as in Figure 23(a). MLDA is not as efficient for objects as for contexts, because we lose some image details computing the mean values for MLDA atoms (see Section II). Nevertheless, we can resort to examining spacial interrelationships of linear discriminants of MLDA atoms, such as connectedness, collinearity and other properties of curves in the image. Thus, we use one set of features – namely, mean color values μ_0 and μ_1 of MLDA atoms, for inferring values of location X and category Z . The other set of features, more specifically, the geometric properties of slopes a of MLDA discriminants are used in computing object indicator RVs O . Separate dependencies of X and O on different feature sets is illustrated in Figure 23(b).

Similar to the explained procedure above, we derive the corresponding distributions for artificial object detection. Note that due to the limitations of MLDA, our intention is not to actually recognize and differentiate between particular objects, but just to detect the presence of artificial objects, which tend to be characterized by different spatial interrelationships than natural objects (such as a bush). Thus, we examine the slopes of MLDA atoms and look for collinearity and orthogonality. Since we are specifically interested in locating *artificial* objects, we cannot detect these objects simply through color analysis alone (e.g. differences in backgrounds and foreground objects). Because we do object detection and not recognition, the o_i random variables take on binary values $\{0, 1\}$.

In Figure 24, we present the proposed inference procedure on two examples, for which we define the following sets: $z_i \in \{\text{sky, ground}\}$, $x_i \in \{\text{lawn, road, forest}\}$, and $o_i \in \{0, 1\}$. Next, having computed the distributions of Z and

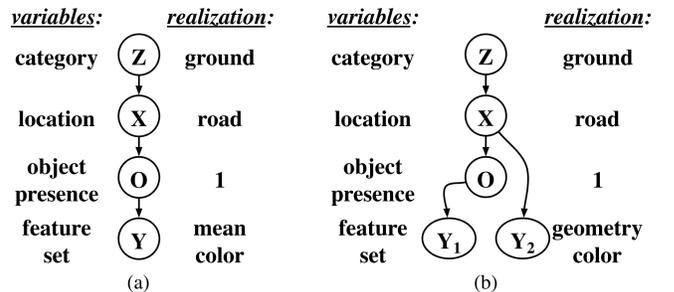


Fig. 23. Extended graphical model for object recognition: (a) inadequate model for MLDA; (b) separating the feature set for location X and object indicator O random variables.

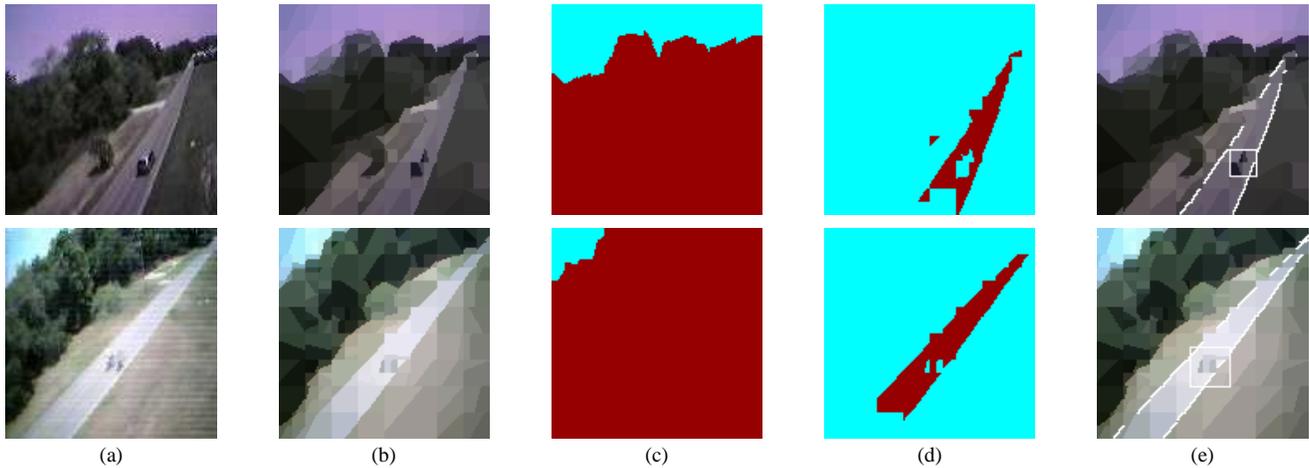


Fig. 24. Object detection: (a) original image; (b) MLDA representation; (c) result of categorization; (d) location recognition; (e) detection of artificial objects.

X , using expressions (9) and (10), we performed location recognition and categorization. Finally, having examined the spacial frequency of orthogonality of discriminants in the location of interest, we detected the artificial object/structure. In this example we considered only the location “road” and objects on the road, though the algorithm enables simultaneous detection of arbitrarily many artificial objects in different locations.

VI. CONCLUSION

In this paper, we presented a unified computer-vision system for enabling smart MAV missions. We first developed MLDA as a feature extraction tool for subsequent image analysis. Next, we demonstrated MLDA’s applicability to real-time horizon detection, and reported flight test results for vision-stabilized flights. Then, we demonstrated the use of TSBNs in segmenting flight images into sky and ground regions. Finally, we extended this basic framework through the idea of visual contexts to detect artificial objects and/or structures.

The work presented in this paper is but one part of our on-going research effort to develop intelligent and mission-capable MAVs. Together with other researchers, we are also working on recovering 3D scene structure of the ground from MAV flight videos. Additionally, work is on-going in sensor integration (e.g. GPS, INS) on our larger platforms (24” maximum dimension), flight vehicle modeling and characterization, and advanced control algorithm development. We expect that the totality of our efforts will eventually enable MAVs not only to fly in obstacle-free environments, but also in complex 3D environments, such as urban settings.

While the focus of this paper has been guided by the requirements of flying vehicles, in principal, the framework described herein certainly is applicable to other robotic platforms as well. Unmanned ground vehicles capable of executing goal-directed, intelligent missions could readily benefit from our unified vision system. Although horizon detection and sky/ground segmentation for stability and control will typically be less critical in ground vehicles, as opposed to flight vehicles, these early steps in our vision framework could nevertheless be exploited to aid higher level goals. Depending

on particular mission profiles, unmanned ground vehicles may well be tasked with, for instance, tracking activity in the air, or pursuing and localizing ground objects of interest. In these cases, sky/ground segmentation can still be the top hierarchical level of a mission-specific defined taxonomy. Furthermore, while points of view obviously differ between ground and flight vehicles, identification of roads and or moving objects, as demonstrated for flight images in Figure 24, will obviously also be critical in many unmanned ground vehicle missions.

ACKNOWLEDGMENTS

This work was supported in part by grants from NASA Langley, the Air Force Office of Sponsored Research, the U.S. Air Force and U.S. Special Operations Command. We also want to acknowledge the work of the entire University of Florida MAV research team for their support of this work, especially Peter Ifju, Andy Kurdila, Rick Lind, Scott Kanowitz, Jason Grzywna and Jason Plew.

REFERENCES

- [1] P. G. Ifju, S. M. Ettinger, D. A. Jenkins, and L. Martinez, “Composite materials for Micro Air Vehicles,” in *Proc. SAMPE Annual Conference*, Long Beach, May 2001.
- [2] D. A. Jenkins, P. G. Ifju, M. Abdulrahim, and S. Olipra, “Assessment of controllability of Micro Air Vehicles,” in *Proc. 16th Int’l Conf. Unmanned Air Vehicle Systems*, Bristol, United Kingdom, April 2001.
- [3] W. Shyy, D. A. Jenkins, and R. W. Smith, “Study of adaptive shape airfoils at low Reynolds number in oscillatory flows,” *AIAA Journal*, vol. 35, pp. 1545–48, 1997.
- [4] R. W. Smith and W. Shyy, “Computation of aerodynamics coefficients for a flexible membrane airfoil in turbulent flow: A comparison with classical theory,” *Phys. Fluids*, vol. 8, no. 12, pp. 3346–3353, 1996.
- [5] C. S. Sharp, O. Shakernia, and S. Sastry, “A vision system for landing an Unmanned Aerial Vehicle,” in *Proc. IEEE Int’l Conf. on Robotics and Automation*, Seoul, Korea, May 2001.
- [6] B. Sinopoli, M. Micheli, G. Donato, and T. J. Koo, “Vision based navigation for an unmanned aerial vehicle,” in *Proc. IEEE Int’l Conf. on Robotics and Automation*, Seoul, Korea, May 2001.
- [7] T. Netter and N. Franceschini, “A robotic aircraft that follows terrain using a neuromorphic eye,” in *Proc. IEEE/RSJ Int’l Conf. Intelligent Robots and Systems*, Lausanne, Switzerland, September 2002.
- [8] D. K. Panjwani and G. Healey, “Markov random field models for unsupervised segmentation of textured color images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 939–954, 1995.

- [9] T. Randen and H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, 1999.
- [10] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain Hidden Markov Models," *IEEE Trans. Image Processing*, vol. 10, no. 7, pp. 1056–1068, 2001.
- [11] D. L. Donoho and A. G. Flesia, "Can recent innovations in harmonic analysis 'explain' key findings in natural image statistics?" *Network: Computation in Neural Systems*, vol. 12, no. 3, pp. 371–393, 2001.
- [12] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. of Optical Soc. of America, A*, vol. 4, pp. 2379–2394, 1987.
- [13] D. L. Donoho, "Wedgelets: Nearly-minimax estimation of edges," *Annals of Statistics*, vol. 27, no. 3, pp. 859–897, 1999.
- [14] A. G. Flesia, H. Hel-Or, E. J. C. A. Averbuch, R. R. Coifman, and D. L. Donoho, "Digital implementation of ridgelet packets," in *Beyond Wavelets*, J. Stoeckler and G. V. Welland, Eds. Academic Press, 2002.
- [15] S. Todorovic and M. C. Nechyba, "Multiresolution linear discriminant analysis: efficient extraction of geometrical structures in images," in *Proc. IEEE Int'l Conf. Image Processing*, Barcelona, Spain, September 2003.
- [16] X. Feng, C. K. I. Williams, and S. N. Felderhof, "Combining belief networks and neural networks for scene segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 467–483, 2002.
- [17] S. Kumar and M. Hebert, "Man-made structure detection in natural images using a causal multiscale random field," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Madison, WI, June 2003.
- [18] S. Todorovic, M. C. Nechyba, and P. Ifju, "Sky/ground modeling for autonomous MAVs," in *Proc. IEEE Int'l Conf. Robotics and Automation*, Taipei, Taiwan, May 2003.
- [19] S. Todorovic and M. C. Nechyba, "Towards intelligent mission profiles of Micro Air Vehicles: Multiscale Viterbi classification," in *Proc. 8th European Conf. Computer Vision*, Prague, Czech Republic, May 2004.
- [20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. San Mateo: Morgan Kaufmann, 1988, ch. 4, pp. 143–236.
- [21] J. K. Romberg, M. Wakin, and R. G. Baraniuk, "Multiscale wedgelet image analysis: fast decompositions and modeling," in *Proc. IEEE Int'l Conf. Image Processing*, Rochester, NY, September 2002.
- [22] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.
- [23] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-guided flight stability and control for Micro Air Vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617–40, 2003.
- [24] S. M. Ettinger, "Design and implementation of autonomous vision-guided Micro Air Vehicles," M.S. Thesis, University of Florida, Electrical and Computer Engineering, August 2001.
- [25] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-guided flight stability and control for Micro Air Vehicles," in *Proc. IEEE Int'l Conf. Intelligent Robots and Systems*, vol. 3, Lausanne, Switzerland, 2002, pp. 2134–40.
- [26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley & Sons, 2000, ch. 2, pp. 20–82.
- [27] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge, MA: The MIT Press, 1998.
- [28] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using Hidden Markov Models," *IEEE Trans. Sig. Processing*, vol. 46, no. 4, pp. 886–902, 1998.
- [29] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [30] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain Hidden Markov Models," *IEEE Trans. Image Processing*, vol. 10, no. 9, pp. 1309–1321, 2001.
- [31] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proc. Int'l Conf. Computer Vision*, Nice, France, October 2003.



Sinisa Todorovic received his B.S. degree at the University of Belgrade, Serbia, in 1994, and his M.S. at the University of Florida, in 2002, both in Electrical and Computer Engineering. He is currently a member of the Center for Micro Air Vehicle Research pursuing his Ph.D. degree at the University of Florida. His research interests include: multiscale statistical modeling in computer vision, probabilistic graphical models, and multiresolution signal processing theory. He has published approximately 10 refereed conference and journal papers.



Michael C. Nechyba received his B.S. degree in Electrical Engineering at the University of Florida in 1992, and his Ph.D. in Robotics at Carnegie Mellon University in 1998. Upon completion of his thesis, Michael joined the Department of Electrical and Computer Engineering at the University of Florida as assistant professor in August 1998. There he serves as associate director of the Machine Intelligence Laboratory. His research interests include machine learning, computer vision and robotics. Currently, Michael's research is focused on two primary areas:

(1) vision-based and sensor-based autonomy for Micro Air Vehicles (MAVs), and (2) direct brain-machine interfaces (BMIs). These research efforts are supported through a number of federal grants from AFOSR, U.S. Air Force, U.S. SOCOM, NASA Langley and DARPA. He has published approximately 30 journal and refereed conference papers.