Machine Vision and Applications (2014) 25:211–227
DOI 10.1007/s00138-013-0560-7

ORIGINAL PAPER

Real-time landing place assessment in man-made environments

Xiaolu Sun · C. Mario Christoudias ·
Vincent Lepetit · Pascal Fua

Received: 6 August 2012 / Revised: 21 August 2013 / Accepted: 19 September 2013 / Published online: 28 November 2013

Abstract We propose a novel approach to the real-time landing site detection and assessment in unconstrained man-made environments using passive sensors. Because this task must be performed in a few seconds or less, existing methods are often limited to simple local intensity and edge variation cues. By contrast, we show how to efficiently take into account the potential sites' global shape, which is a critical cue in man-made scenes. Our method relies on a new segmentation algorithm and shape regularity measure to look for polygonal regions in video sequences. In this way, we enforce both temporal consistency and geometric regularity, resulting in very reliable and consistent detections. We demonstrate our approach for the detection of landable sites such as rural fields, building rooftops and runways from color and infrared monocular sequences significantly outperforming the state-of-the-art.

Keywords Automated landing · Hazard detection · Component tree · Image segmentation · Shape analysis

This work was supported in part by the EU myCopter project.

Electronic supplementary material The online version of this article (doi:10.1007/s00138-013-0560-7) contains supplementary material, which is available to authorized users.

X. Sun (✉) · C. Mario Christoudias · V. Lepetit · P. Fua
Ecole de Polytechnique de Lausanne, BC Building,
1015 Lausanne, Switzerland
e-mail: xiaolu.sun@epfl.ch

C. Mario Christoudias
e-mail: mario.christoudias@epfl.ch

V. Lepetit
e-mail: vincent.lepetit@epfl.ch

P. Fua
e-mail: pascal.fua@epfl.ch

1 Introduction

Unmanned aerial vehicles (UAVs) offer many civilian and military applications, and are now a quickly growing industry [13]. While they are still usually remote controlled, automated flight is an attractive alternative that would largely increase their usefulness and reliability. Of particular importance, is the ability to land automatically, which requires the efficient and reliable detection of suitable landing sites. This is actually an old problem as it has been extensively studied for planetary landing in space, but it has been recently revisited for both fixed-wing and rotorcraft UAVs [13,40,49,50].

Landing site detection is generally a time critical decision that must be made reliably and quickly, often in a few seconds or less [16,50]. Active sensors have been widely used for this purpose [23,26,40,41,47] but have severe drawbacks. They are expensive, power-hungry, and often heavy. Their range and resolution are usually limited [41], and special care must be taken when operating in populated areas.

By contrast, passive sensors such as cameras are inexpensive, low power and lightweight. They can operate from a range of flight altitudes and are safe for use in populated urban and rural environments. As a result, many camera-based approaches [5,13,24,37] have also been explored over the years. To achieve real-time performance, most of them rely on simple techniques such as thresholding of local intensity variations or edge density. They completely ignore the global shape of the potential landing area, which is a vital clue for human pilots seeking a landable field in an emergency. As illustrated in Fig. 1, global shape and regularity matter because it is extremely difficult to assess visually whether a piece of terrain is sufficiently flat by other means from above. Human eyes do not provide a long enough baseline for stereo under these conditions and although textural or shading cues can be useful in some cases, such as when a heavily rutted

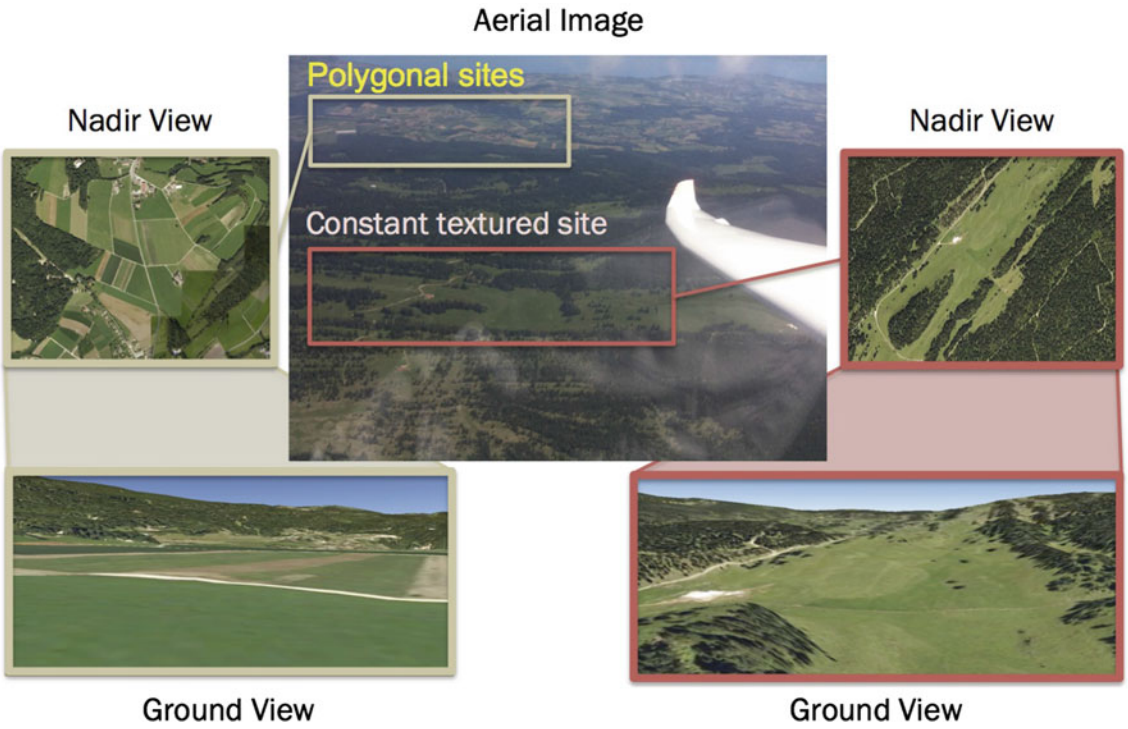

Fig. 1 Importance of shape cues for landing site detection. The center image was taken from a glider flying along a mountain ridge. From the air and based on local appearance, the regions within the *yellow* and *red black regions* both appear to be landable. On the *left* and *right*, we used Google Earth to display nadir views of these two areas and low-altitude oblique views, which are those a pilot would see upon approaching to land. While the area on the *left* is indeed flat and landable, the one on the *right* is not and touching down there would result in a crash. Local appearance is, therefore, insufficient to assess the suitability for landing and finding regular polygonal structures on the ground is required

field produces strong shadows at particular times of day, they are generally unreliable. Typically, they only become useful when very close to the ground and therefore too late to select another field if the chosen one proves unsuitable. Furthermore, obstacles such as ditches and fences that could cause an accident cannot easily be seen either. In the event of an engine failure, light aircraft and helicopter student pilots are therefore trained to look for regular polygonal areas, such as cultivated fields or rooftops, large enough given their respective aircrafts' landing speeds and under the assumption that they are more likely to be flat than irregular ones and less likely to contain hidden obstacles. This is even more important for glider pilots who fly without an engine and can expect to land in unprepared fields, such as those in Fig. 1, several times during their flying careers due to adverse meteorological conditions.

In this paper, we propose a real-time algorithm that emulates this human ability to quickly assess candidate landing sites when flying over man-made environments whose 3D geometry cannot be reliably assessed by shape-from-X methods, for example, in the event of an emergency landing that precludes a controlled flight path to acquire reliable range estimates. Polygonality combined with simple texture measures is then a useful substitute, which has been used in earlier work to detect prepared landing sites such as runways [18], but not unprepared ones.

At the heart of our approach are the maximally stable regions (MSERs) [32], which have proved effective for stable patch detection, which we use in spatio-temporal image volumes to produce polygonal regions. To this end, we introduce a Hough-like voting scheme into the algorithm that builds our spatio-temporal MSERs to guarantee that they have planar sides. Unlike other algorithms that perform global image segmentation [7,45], ours is highly efficient and runs at 5 Hz on 320 × 240 images using commodity hardware, in part because our MSERs are found using an efficient component tree representation [29].

We will demonstrate that our algorithm can reliably detect a wide range of potentially landable areas for both fixed-wing and rotorcraft UAVs, such as rural fields and building rooftops from both nadir and oblique viewpoints. We also show that we can detect runways from low-quality infrared image sequences in which runway markings are not clearly visible with significantly better performance than traditional contour-based methods that rely solely on their rectangularity [10,18,25,43,48].

The remainder of this paper is organized as follows: Related work is first discussed in Sect. 2 and our component tree segmentation algorithm is outlined in Sect. 3. The proposed shape regularity measure is presented in Sect. 4 and an extension to temporal sequences is discussed in Sect. 5. Experimental results are provided in Sect. 6.

2 Related work

There is a long tradition of detecting landing sites from aerial images, which dates back to the inception of our field [36]. Initial work focused on the detection of runways for the automated aerial mapping of airports [25,34,36,46]. Since then many automated landing approaches have been developed that considered the detection of both prepared and unprepared landing sites from both active and passive sensors in a variety of terrains.

Our work is primarily concerned with landing site assessment in man-made environments from images captured by a monocular camera. We begin our discussion with an overview of unprepared landing site detection with both active and passive sensing. Prepared landing site detection is then outlined with an emphasis on runway detection. Finally, a discussion on the use of shape for man-made landing site assessment is provided.

2.1 Unprepared landing sites

Techniques to unprepared landing site detection rely on measurements of surface geometry and appearance to detect and avoid hazards and find suitable landing sites. Many methods have been proposed for the assessment of 3D surface geometry from active range sensors [23,27,40,47]. Johnson et al. [27] propose a hazard map estimation framework using estimates of surface slope and roughness from laser scanner range measurements. Similarly, Howard and Seraji [23] develop a fuzzy logic approach for the classification of terrain into landable and hazardous segments, based on measurements of slope, approach and roughness obtained with least-squares plane fitting applied to LIDAR range data. More recently, Scherer et al. [40] have demonstrated the automated landing of a full-scale rotorcraft UAV using a laser scanning sensor.

While a promising technology for landing place assessment, active sensors have a high energy consumption and a heavy payload. They typically have a restricted operational range of 1km or less, require additional safety considerations in populated areas, and often involve a costly, time-consuming acquisition process making them unsuitable for the applications we target [40,47].

To overcome the limitations of active sensing, many approaches have been developed for the estimation of 3D surface geometry and landing site assessment from passive camera sensors [16,24,38,49]. Camera sensors are inexpensive, low power, lightweight devices that can operate from a large range of flight altitudes and can be safely used in populated areas, making them an attractive alternative to expensive, power intensive active sensors, especially for the smaller UAVs. Still the full acquisition of 3D range estimates from passive sensors in generic environments and operating conditions remains challenging and computationally costly.

Monocular methods usually utilize sparse structure from motion in combination with surface interpolation to estimate 3D terrain geometry [21,28,49]. Hoff and Sklair [21] utilize optical flow-tracked features to obtain range estimates that are incrementally improved with Kalman filtering. Similarly, Templeton et al. [49] propose a recursive multi-frame planar parallax algorithm for dense, real-time 3D surface recovery. Although efficient, these methods are prone to fail when local features cannot be tracked reliably, and require a controlled flight path for reliable range estimation restricting their general applicability. Simpler methods based on homography estimation have been proposed for surface slope estimation that either assume the presence of a flat ground plane [3] or rely on the efficient proposal of candidate landing sites [5]. Stereo vision systems have also been investigated [33,50], however, generally require a large baseline making them less suitable for smaller platforms.

As an alternative to active- or passive-only sensing solutions to landing place assessments, multi-sensor approaches have also been investigated that seek to combine the strengths of each sensing modality [22,37,41]. Pien [37] proposes the use of passive sensing that exploits simple intensity variation measures to segment candidate landing regions provided as input to an active sensor laser range verification stage. Similarly, Serrano et al. [41] advocate for a multi-tiered solution that combines the strengths of passive and active sensors to achieve a diverse capability of operational ranges.

Whether a passive-only or combined-sensing solution is used, the ability to quickly assess candidate landing sites is a crucial step employed by many approaches in the literature [5,13,22,24,37]. Monocular texture analysis techniques have played a predominant role in finding suitable candidate landing sites as they typically involve a fairly simple image processing and are easily amenable to real-time operation [9,13,16,22,37].

Approaches assume that obstacles are indicated by dominant image boundaries, and look for relatively featureless, constant-valued candidate landing regions. Garcia et al. [16] find circular areas exhibiting a low edge density computed using normalized edge histograms. Similarly, Fitzgerald [13] detects candidate landing sites as featureless regions absent of image edges. Howard et al. [22] employ a local intensity variation measure similar to [37] along with a simple thresholding to find candidate sites. Although efficient, these methods do not truly model the object image boundaries, and as seen in our experiments are sensitive to image noise and hallucinated local edges, often requiring a careful threshold selection and choice of edge detection parameters. In this paper, we propose a global image segmentation algorithm that can be computed in real time and significantly improves over the performance of these methods.

2.2 Landing pads and runways

Prepared landing site detection has also received a lot of attention in the literature in particular for the detection of landing pads and runways. Research on landing pad detection has concentrated on the design of landing signatures that can be easily detected and tracked using a monocular camera [4,20,42,44]. Similarly, techniques to runway detection exploit runway markings and region geometry [14,17,18,25,43,48,56]. They typically search over extracted line features to find landing patterns and the runway boundary. Huertas et al. [25] utilize a hypothesis and test formulation based on the detection of "anti-parallel" lines or *apars*. Provided with a set of hypothesis regions, pathway markings are used to detect runways and differentiate them from other airport and ground transportation roads. Shang and Shi [43] apply a horizon detection and intensity thresholding step to identify a runway region of interest followed by Hough line fitting to detect the runway boundary. Similarly, Hamza et al. [18] assume a region of interest and a perspective runway template provided from an external navigation system. Various line-fitting methods are then explored for runway corner detection including Hough voting and a RANSAC least-squares estimation.

However, a known region of interest and runway template is not always readily available, and horizon detection is restricted to oblique viewpoints where the horizon is clearly visible and not obstructed by nearby building or mountain structures. In addition, due to poor visibility or when seen from a distance runway markings are not always apparent. A method that can efficiently detect candidate regions without the use of such additional cues is therefore needed. Moreover, all these approaches do not generalize to unprepared landable sites.

2.3 Shape regularity

In man-made environments, many suitable landing sites can be characterized as featureless, regularly shaped regions. While region shape has been an important cue for the detection of runways, shape has been largely unexplored for the assessment of unprepared landing sites. Although methods have been proposed that search over rectangular or circular regions for unprepared landing site assessment [13,16], these methods do not exploit region shape for the underlying segmentation, and instead apply a template search of known scale and geometry. Unlike previous approaches, we do not assume the existence of a known region of interest or template, and shape is used to guide the underlying image segmentation and detect candidate man-made landing sites. The use of region shape results in accurate landing site detection without the need for distinctive landing patterns or other constraints, however, when available, such additional cues can be useful in combination with our approach at a later verification stage.

3 Component tree image segmentation

In this section, we first provide a brief overview of component tree segmentation mostly using the formulation of [29]. We then discuss extensions we use for finding obstacle-free candidate landing sites.

Image segmentation by thresholding is an instance of a more general class of techniques known as *connected filtering*. Connected filters are morphological operators that can be used to simplify the image while preserving its contours, employed in a variety of applications [29]. Those that commute with thresholding are called *flat* filters [19]. Flat filters remove components whose attributes violate a given criteria, and get their name from the constant-valued regions they detect in an image. Although efficient, they are fairly simple and can be sensitive to varying imaging conditions and noise, especially when only considering the connected components computed at a single predefined threshold, as is typically done to detect landing sites [13,14,16,22].

The *component tree* [8,29] is a *non-flat* filter that considers the relationship between connected components as they *evolve* across an entire threshold range. In this way, it allows for increased flexibility and modeling capacity overcoming many of the limitations of simple flat filtering [29]. Furthermore, highly efficient algorithms exist for their implementation [29,35,51,54] making them well suited for real-time applications.

3.1 Component trees

Component trees are based on the notion of *threshold decomposition* [55]. Let f be a real-valued image defined by the function $f : F \mapsto \mathcal{R}$ where the support $F \subseteq \mathbb{R}^2$. A reconstruction of the image f can be defined using image thresholding

$$f(x) = \max\{t : x \in \mathcal{X}_t(f)\}, \tag{1}$$

where

$$\mathcal{X}_t(f) = \{x \in F : f(x) \geq t\}, \tag{2}$$

is the threshold set of the real-valued image f obtained at threshold t.

Equation (1) decomposes the image into a set of binary images that define a simplified representation.

Let $C_{t,n}$ be the nth connected component of threshold set $\mathcal{X}_t$. Equation (1) can be re-expressed as

$$f(x) = \max \left\{ t : x \in \bigcup_n C_{t,n} \right\}. \tag{3}$$

A connected filter preserves the components of f whose attributes satisfy a certain criterion T,

$$\Phi(\mathcal{X}_t(f)) = \cup\{C_{t,n} : C_{t,n} \text{ satisfies criterion } T\}. \tag{4}$$

An important feature of connected filters is that they only remove components, and unlike other morphological operators they do not alter the component boundaries, a desirable property for image segmentation.

A component tree $\mathcal{T}$ is defined from the components $C_{t,n}$ with one node per component denoted as $n(C_{t,n})$ or simply n. Threshold sets have the important property that $\mathcal{X}_{t+1}(f) \subseteq \mathcal{X}_t(f)$ which implies that for every component $C_{t,m}$ there exists a component $C_{t+1,n} \subseteq C_{t,m}$ [29]. Two nodes $C_{t+1,n}$ and $C_{t,m}$ are linked in the tree if $C_{t+1,n}$ is a *descendant* of $C_{t,m}$ satisfying the above property. The root of the tree $n_{\min}$ is defined by the component $C_{\min}$ that is the superset of all the components in the image found by thresholding the image by its minimum value. The tree is constructed by progressively thresholding the image, linking the nodes between neighboring thresholds, starting at the root.

Component trees can be used to implement either a flat or non-flat connected filtering. A flat filtering only considers the nodes $C_{t,n}$ individually at each level t, whereas a non-flat filtering enforces criteria defined along branches of the tree. A key advantage of component trees is that they can be used to define a selective image filtering that only affects concentrated regions in the image corresponding to branches in the tree, leaving the rest of the image unaltered. A selective filtering is not possible using flat filters and gives component trees a distinct advantage over them, especially when not all objects are well segmented using only a single threshold, as is typically the case. Component trees also generalize previous hierarchical connected filters in the literature having a close connection with *opening trees* [53] and *max-trees* [39]. For a more detailed treatment of component trees, we refer the reader to [29].

An example illustrating the use of flat vs. non-flat filtering is depicted in Fig. 2 where the goal is to segment the two constant circular regions. Flat filtering considers each threshold set individually and, therefore, has difficulty in obtaining the desired segmentation, especially when only considering a single threshold. The boundaries of each region can be detected using multiple thresholds, however, at the cost of added clutter. In contrast, a non-flat tree filtering can easily detect the circular regions since it can exploit the fact that the connected components of these regions remain relatively unchanged across threshold sets compared to other regions in the image.

3.2 Extensions

Image segmentation with component trees is performed by considering the sequence of node attributes found along a branch of the tree, otherwise called an *attribute signature* [29]. While [29] considers signatures defined with respect to tree branches associated with leafs of the tree, in this paper we detect regions in the image by finding attribute extrema along each branch similar to [32]. This allows for the discovery of featureless regions characterized by dominant image boundaries referred to in [32] as MSERs. Whereas [32] only considers local extrema; however, we compute extrema across an entire tree branch as this helps avoid spurious detections.

More formally, let $g(n)$ represents an attribute of node n. Our attribute signature is defined as

$$n(C_{t,n}) \text{ is } \begin{cases} \text{active}, & \text{if } g(n(C_{t,n})) = \min\{g(n(C_{k,m})) : \\ & \quad C_{k,m} \in \mathcal{B}(n(C_{t,n}))\} \\ \text{not active}, & \text{otherwise.} \end{cases} \tag{5}$$

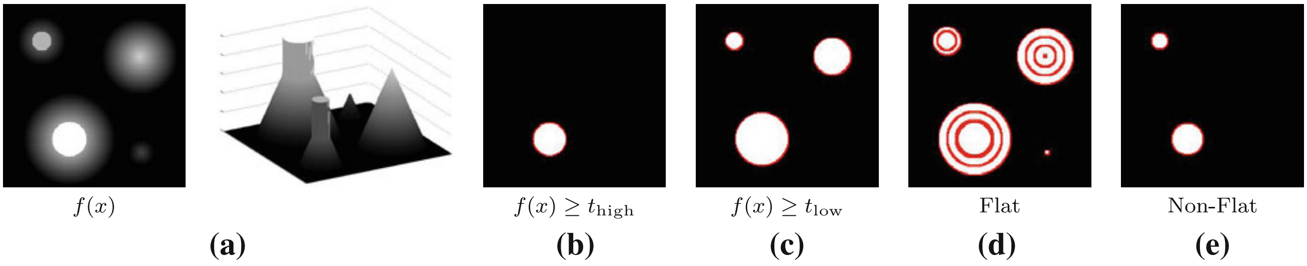

Fig. 2 Selective filtering. **a** An example intensity image and its associated surface. **b**, **c** Flat filtering with a single threshold. **d** Flat filtering across an entire threshold decomposition. **e** Non-flat component tree filtering. Detected regions are indicated in *white* with a *red* contour. Cleanly segmenting the constant circular regions is not possible using a flat filtering with either a single threshold or threshold decomposition. In contrast, a non-flat component tree filter can selectively filter these regions by exploiting the fact that their connected components remain unchanged across a range of threshold sets

where $\mathcal{B}(n)$ is the tree branch containing node n. A node is labeled as *active* if it is to be preserved by the component tree filter, and is labeled as non-active otherwise. Following [32], we use an area variation signature and define $g(n)$ using the area variation between n and its neighboring parent and child nodes. This notion is extended to temporal sequences in Sect. 5. The final image segmentation is then obtained retaining all the active nodes in the tree

$$\Phi_{\mathcal{T}}(f)(x) = \begin{cases} 1, & \text{if } \phi_{\mathcal{T}}(f)(x) \leq f(x), \\ 0, & \text{otherwise}, \end{cases} \qquad (6)$$

obtained with the connected tree filter

$$\phi_{\mathcal{T}}(f)(x) = \max\{t : x \in C_{t,n} \text{ and } n(C_{t,n}) \text{ is active}\}. \qquad (7)$$

Many low-level image representations can be used with our approach including those proposed in the landing site detection literature [16,22,37]. Compared with other representations, we found a combination of intensity and intensity gradient to work best. Using intensity-only can stably segment noisy regions; however, it often overlooks salient edge information. Similarly, an intensity gradient-based segmentation delineates geometric shapes very well, but is more sensitive to noise (see Fig. 3).

We employ a combined intensity and gradient feature that can benefit from the strength of each representation, defined as

$$C = I - \alpha G, \qquad (8)$$

where I is the intensity image, G the gradient image of I, and α is a scale used to weight the gradient image. Under the proposed feature combination, the image gradient helps to guide the component tree segmentation, while still benefiting from the stability of constant intensity regions. In our experiments, we found this combination to give increased stability compared to using either feature alone.

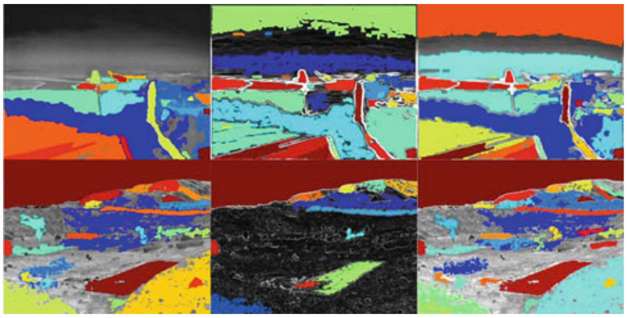

Fig. 3 Combined feature representation. Component tree segmentation applied to the (*left*) intensity image, (*middle*) gradient image, and (*right*) a combined intensity and gradient representation. Detected regions are displayed using a heat map color coding with red indicating highly stable regions and blue low stability. The proposed feature combination benefits from the strengths of each representation and has the ability to exploit image edges while leveraging the stability of constant intensity regions

4 Shape regularity measure

Man-made landing sites can be distinguished by their characteristic, simple shape, often consisting of elongated linear structures. In this section, we introduce a notion of *shape regularity* and use it to segment polygonal regions from the image indicative of man-made landing sites. Many algorithms have been proposed for polygon detection in the literature; however, most of them are restricted to fairly simple polygonal shapes [1,30]. We provide a generic measure of shape regularity applicable for the detection of a variety of polygonal structures.

The regions resulting from component tree segmentation are further filtered according to their shape. A region's shape is considered *regular* if its contour is well approximated by N lines. This concept is illustrated in Fig. 4 using two example contours, one that is regular and one that is not. With our approach, lines are efficiently estimated from each contour using an adaptive Hough voting [12] whose resolution is adapted according to contour size. In particular, the radial resolution and non-maximum suppression threshold are set proportional to the minor radius of an approximating ellipse estimated separately for each region. We found this to give more reliable linear structure detection compared to a fixed-resolution Hough space.

A shape regularity score is computed for each region based on the percentage of contour points that belong to a detected linear structure,

$$L(n) = \sum_{\min\{N,M\}} p_i, \qquad (9)$$

where p_i is the percentage of region contour points voting for line i, and M is the number of detected lines. The linearity

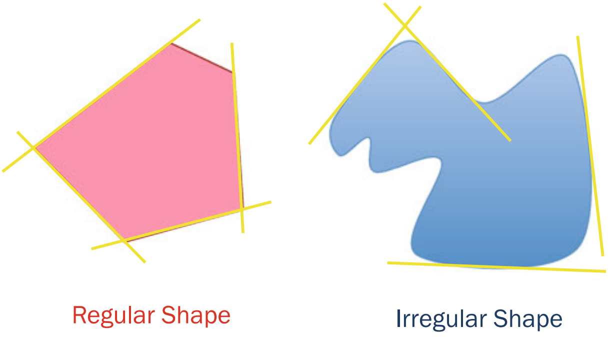

Fig. 4 Shape regularity. Regularly shaped regions are those exhibiting a simple polygonal shape. We consider a region to be regularly shaped if its contour is well approximated by N lines. This is illustrated for two example regions with $N = 4$

score is parameterized by N, defined as the maximum number of detected lines used to compute the score, and with the lines sorted in decreasing order by their dominance, p_i. Intuitively, a higher value of N will assign a higher score to more complex shapes, and can be used to tune the detector to the desired class of polygonal shapes. For example, in the case of rectangular runways one would set $N = 4$.

In order to find dominant polygonal regions, we define the combined score

$$s(n) = L(n)\Gamma(-g(n)) = \frac{L(n)}{1 + \exp\left(-\frac{\mu - g(n)}{\sigma}\right)} \tag{10}$$

where $L(n)$ is the linearity score and $g(n)$ is the area variation score for component tree node n. In Eq. (10), a soft thresholding is applied to the stability score defined by the sigmoid $\Gamma(x)$ with threshold μ and bandwidth σ. This reflects the intuition that the stability score is often only a weak feature, the combined score giving more emphasis to the underlying linearity measure while down-weighting highly unstable regions.

The final shape segmentation signature employed by our approach can be expressed as

$$n(C_{t,n}) \text{ is } \begin{cases} \text{active,} & \text{if } g(n(C_{t,n})) = \min\{g(n(C_{k,m})): \\ & C_{k,m} \in \mathcal{B}(n(C_{t,n}))\} \text{ and} \\ & s(n(C_{t,n})) \geq T \\ \text{not active,} & \text{otherwise.} \end{cases} \tag{11}$$

with detection threshold T. The shape regularity measure defines a flat tree filtering and can be efficiently implemented by restricting its evaluation to the extremal nodes for which it applies.

5 Temporal consistency

We explore the use of temporal consistency in addition to geometric regularity to increase the reliability of our man-made landing site detection. A modified component tree segmentation for image sequences is first discussed. A Hough plane voting scheme is then outlined for assessing the shape regularity of spatio-temporal regions.

Temporal consistency exploits multiple image time instances to help gauge the presence of a suitable landing site. It is particularly well suited for segmenting low-quality images whose image boundaries are noisy and are more reliably extracted by accumulating evidence across many frames. We employ a sliding temporal window to detect landing sites from video. Prior to segmentation, a simple homographic alignment step similar to [3] is first applied to each windowed image sequences to provide a quick, coarse camera motion correction.

Provided an aligned image sequence component tree segmentation is applied to find dominant regions across *both* space and time. These regions are then filtered accordingly to detect candidate landing places. Component tree filtering and segmentation on image sequences proceeds much in the same way it does for a single image.

Let v be a real-valued image sequence defined by the function $v: V \mapsto \mathcal{R}$ where the support $V \subseteq \mathbb{R}^3$ with the third dimension being time. Connected components are found using threshold decomposition on v applying Equation (1). Whereas for a single image, the components are 2D regions for image sequences they correspond to 3D volumes.

Dominant components are assessed using a *volume variation* signature similar to [11]

$$g_v(n_t) = \frac{\mathcal{V}(n_{t+\Delta}) - \mathcal{V}(n_{t-\Delta})}{\mathcal{V}(n_t)}, \tag{12}$$

where $\mathcal{V}(n_t)$ is the volume of the component represented by node n_t and Δ is the threshold difference level. $n_{t+\Delta} \supset n_t$ is the node at threshold $t + \Delta$ of the component immediately including n_t, and $n_{t-\Delta} \subset n_t$ is the component at threshold $t - \Delta$ immediately included by n_t.

The shape regularity measure is extended to evaluate the surfaces of component volumes. Lines used to approximate 2D region contours correspond to spatio-temporal planes in the image sequence. These planes are efficiently detected using a Hough voting procedure. We represent a plane using its normal vector $n = (n_x, n_y, n_t)$ and a point $X_c = (x_c, y_c, t_c)$ belonging to the plane. For any point X in the plane,

$$\begin{aligned}(X - X_c) \cdot n &= (x - x_c)n_x + (y - y_c)n_y + (t - t_c)n_t \\ &= xn_x + yn_y + tn_t - \rho = 0\end{aligned} \tag{13}$$

where ρ is the distance from the origin to the plane. Using spherical coordinates,

$$\rho = x * \cos\theta * \sin\phi + y * \sin\theta * \sin\phi + t * cos\phi \tag{14}$$

where ϕ is the angle between the normal vector n and the t axis, and θ is the angle between the x axis and the projection of n onto the xy plane.

A plane is therefore defined by three parameters: (ρ, θ, ϕ), which form a 3D voting space. Each component surface point casts a 3D surface of votes in this space. Finally, planes are detected as the peaks in the resulting accumulator matrix. We employ an adaptive binning resolution based on component size, and limit ϕ to a small range about 90 degrees to prefer linear structures stable across time. A weighted voting strategy based on the angle between the surface and fitted plane normals is also employed along with a non-maximum suppression to remove excess votes about each peak. Figure 5 displays an example plane fitting result. In the figure, the left plot shows the aligned point cloud, and the right plot the

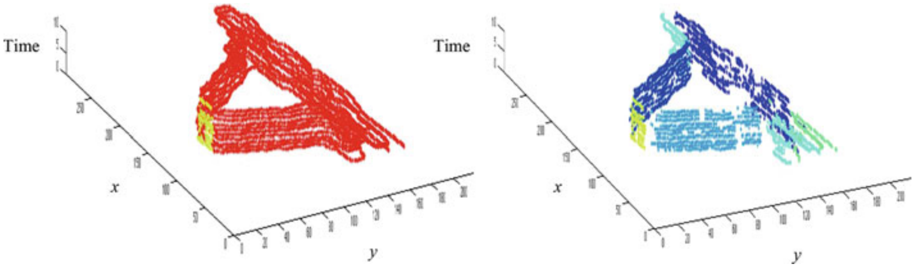

Fig. 5 Spatio-temporal plane detection: (*left*) an aligned spatio-temporal point cloud and (*right*) detected planes whereby the points belonging to each plane are highlighted in separate colors

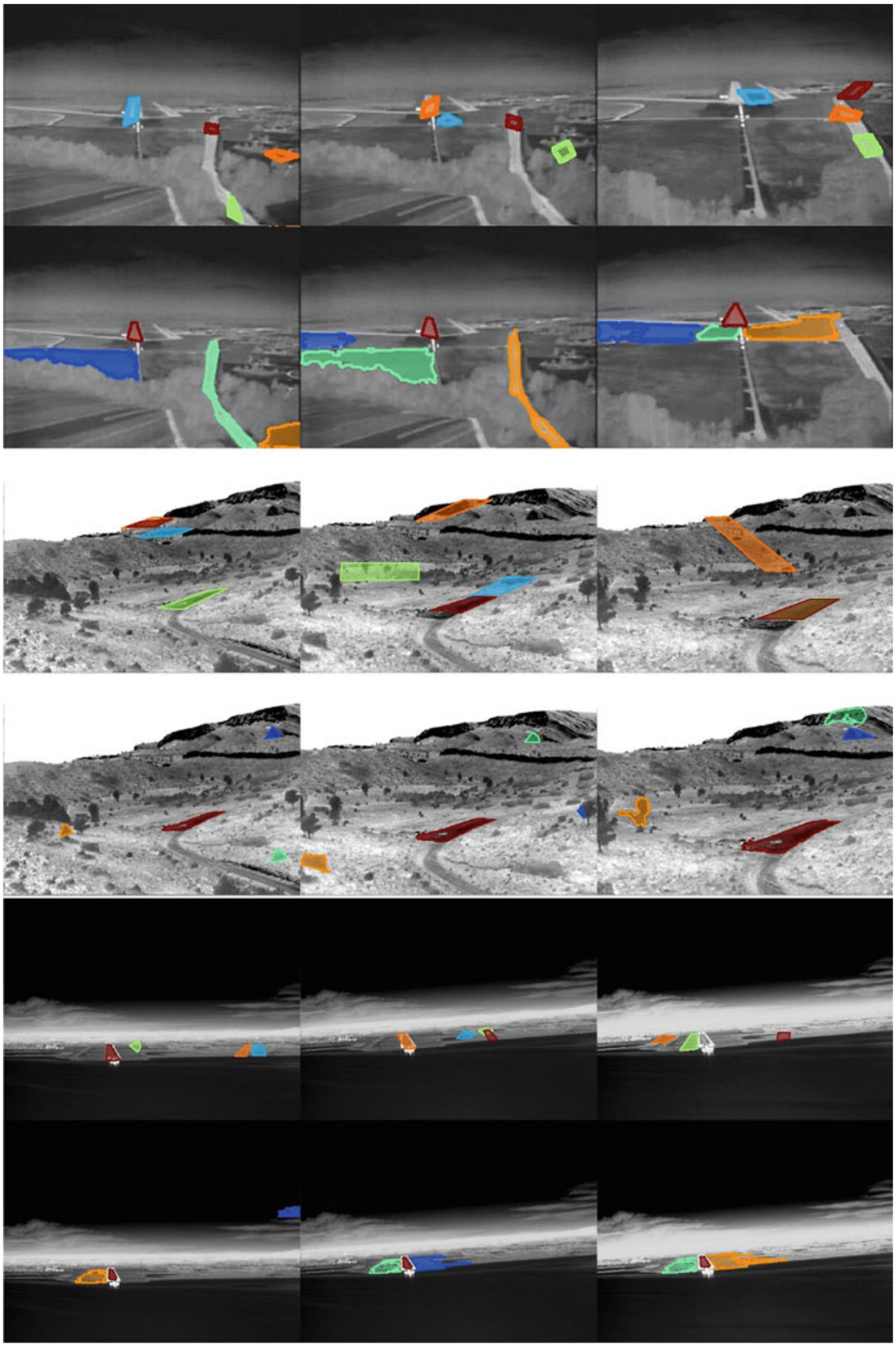

Fig. 6 Runway detection. Results for each sequence are displayed for (*top*) the windowed hough baseline and (*bottom*) our approach (*MP*). *Red, orange, green* and *blue rectangles* denote the top four detections ranked in that order. We supply the corresponding video sequences as supplementary material. Our approach significantly outperforms the baseline technique

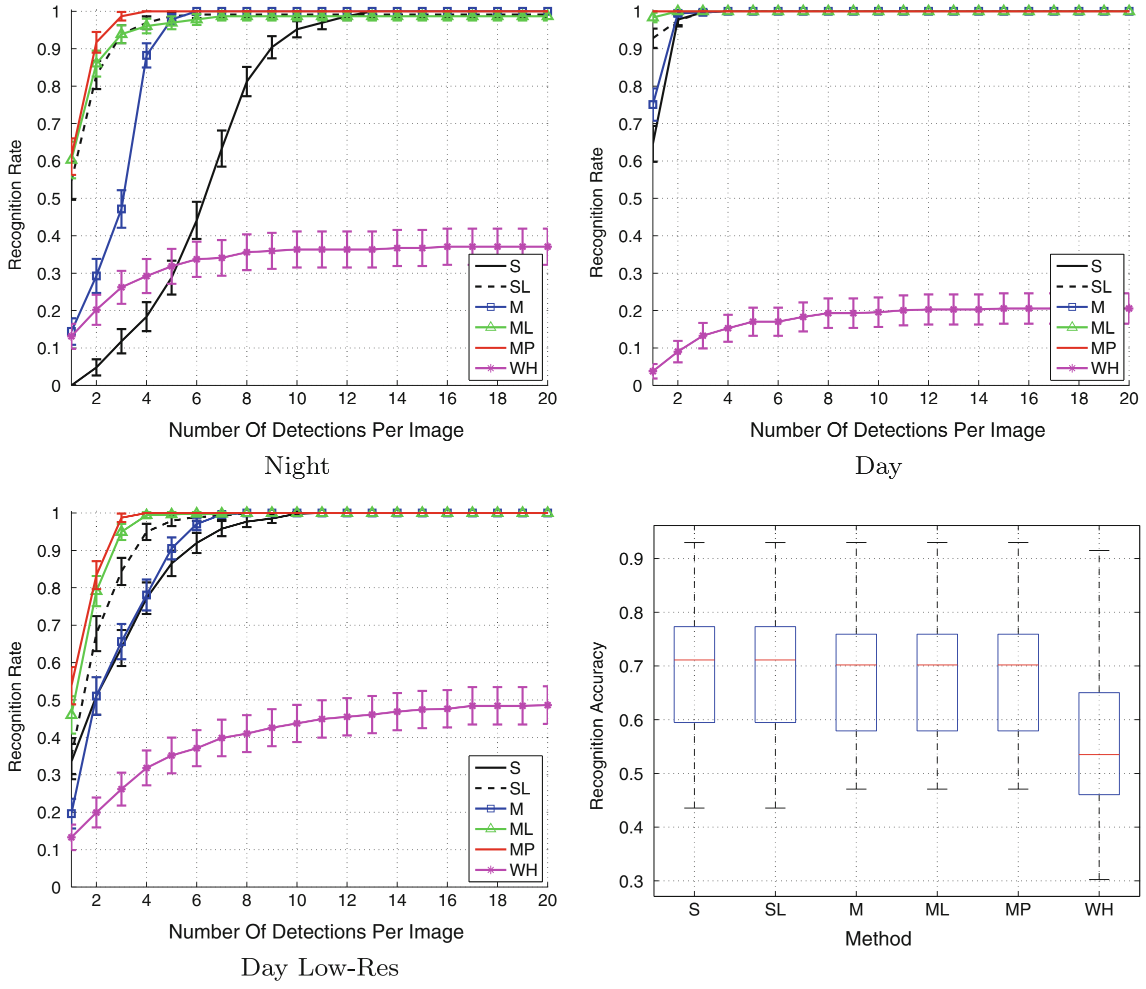

Fig. 7 Runway recognition performance displayed for each method. Five settings of our approach are evaluated, they are: *S* single-frame, *M* multi-frame component tree segmentation without shape regularity, *SL* single-frame segmentation using line detection, *ML* multi-frame segmentation with line detection, *MP* our full approach using plane detection. The use of area-based cues and spatio-temporal consistency result in a large improvement over the baseline method across all three sequences. Temporal consistency is especially important when working from low-resolution imagery. Runway detection accuracy is also shown across all sequences. A *box plot* is provided for each method with the *red bar* and *edges of each box* showing the median, and top 25th and 75th percentile detection accuracy. Compared with the baseline technique (WH), our method results in a more accurate detection of the runway

fitting results with the points of each plane highlighted in a separate color.

6 Experiments

In this section, we demonstrate our approach for the detection of runways, building rooftops and rural fields from aerial infrared and color video sequences. We first discuss our experimental setup and employed baselines. We then present our results on man-made landing site detection that highlight both the reliability and efficiency of our method.

6.1 Experimental setup

Five settings of our approach are evaluated, they are: (S) single-frame and (M) multi-frame component tree segmentation without shape regularity, (SL) single-frame segmentation using line detection, (ML) multi-frame segmentation with line detection, and (MP) our full approach using plane detection.

As an evaluation metric, we use the percent overlap between the ground-truth and detected runway:

$$\text{detection accuracy} = \frac{\text{area of overlap}}{\text{total area}} \tag{15}$$

In our experiments, a landing site is considered detected if the detection accuracy is at least 30 %.

We systematically used a feature combination weighting of $\alpha = 4$, a temporal window size of ten frames, and used stability parameters of $\mu = 0.1$ and $\sigma = 0.2$ for the noisy infrared sequences. The higher quality color videos exhibited less noise, and we therefore set $\mu = 0.1$ and $\sigma = 0.001$. In

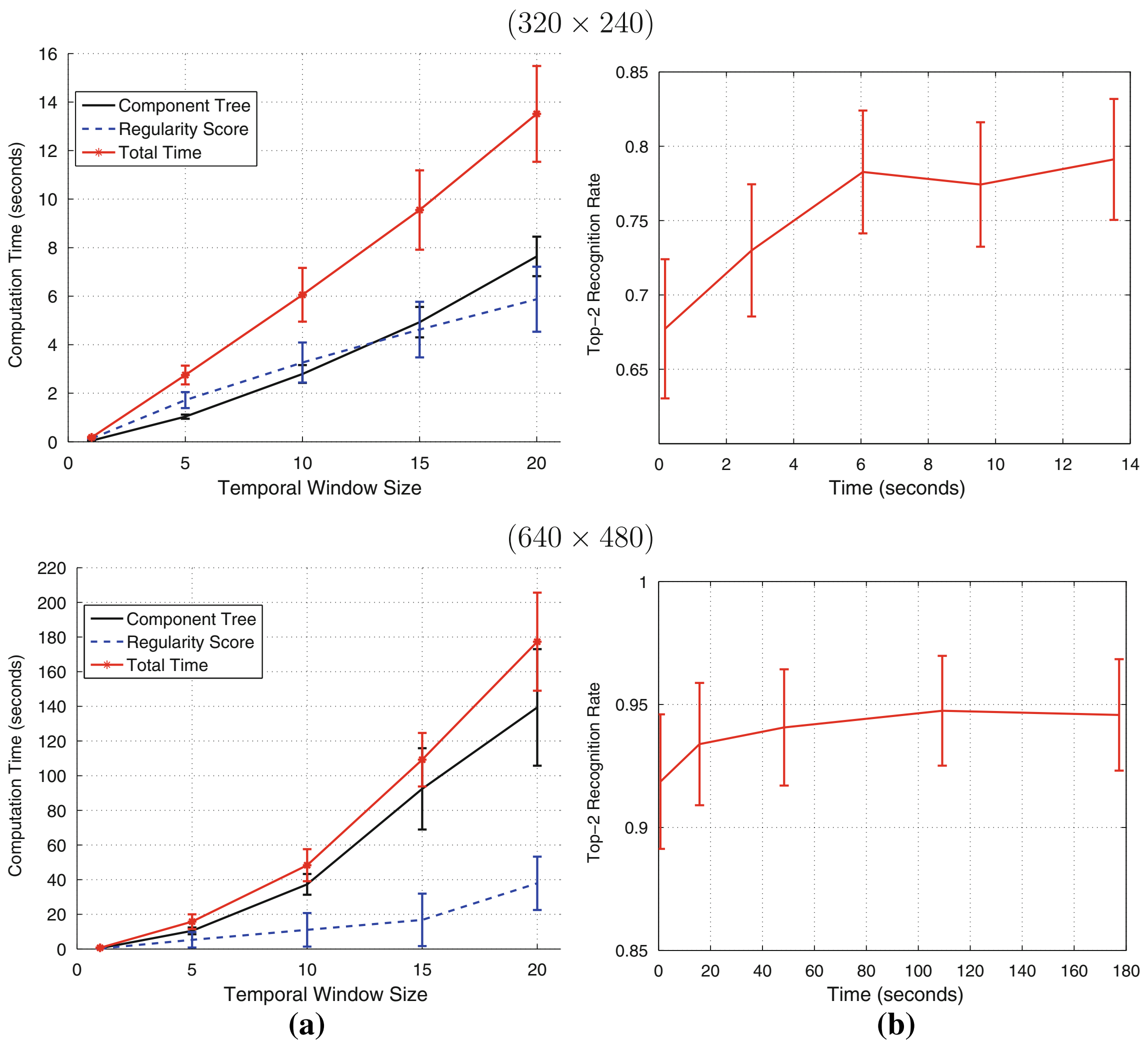

Fig. 8 Performance time. The average computation time is shown across the runway sequences: **a** average time for each step of our approach as a function of input frame count and **b** detection rate as a function of computation time

other words the linearity score was the primary measure for these sequences.

Sequence alignment was performed using SIFT feature matching [31] and homography estimation between consecutive image frames. Although we do not optimize over the efficiency of the alignment pre-processing step, many methods exist for fast feature matching and homography estimation [3,28].

Component tree segmentation as with other segmentation algorithms can often result in small spurious regions. Minimum and maximum region area limits are therefore used, that we assume known for both ours and the baseline methods. This is a reasonable assumption, since in most applications the landing site area is easily available from the landing size requirements of the aircraft, as is typically used in practice [13,16,40]. Similar requirements are used to help constrain our Hough voting step, whereby the angular weighting is set according to the expected region size.

Our implementation of MSER component tree segmentation is based on the vlfeat library [52] and was done in MATLAB using C-code MEX-wrappers. Although efficient, our code can be further improved for even faster performance.

6.2 Baselines

We consider multiple baseline methods for runway and unprepared landing site detection.

For the detection of unprepared landing sites, we consider an intensity variation method that finds candidate regions based on efficient, simple image thresholding [5,22,37]. This measure is evaluated using a spatial sliding window computed densely throughout the image. Intensity variation is defined as

$$I_\sigma(c) = \sqrt{\frac{1}{(2r+1)^2} \sum_{x \in W(c,r)} (I(x) - \mu(c;r))^2} \qquad (16)$$

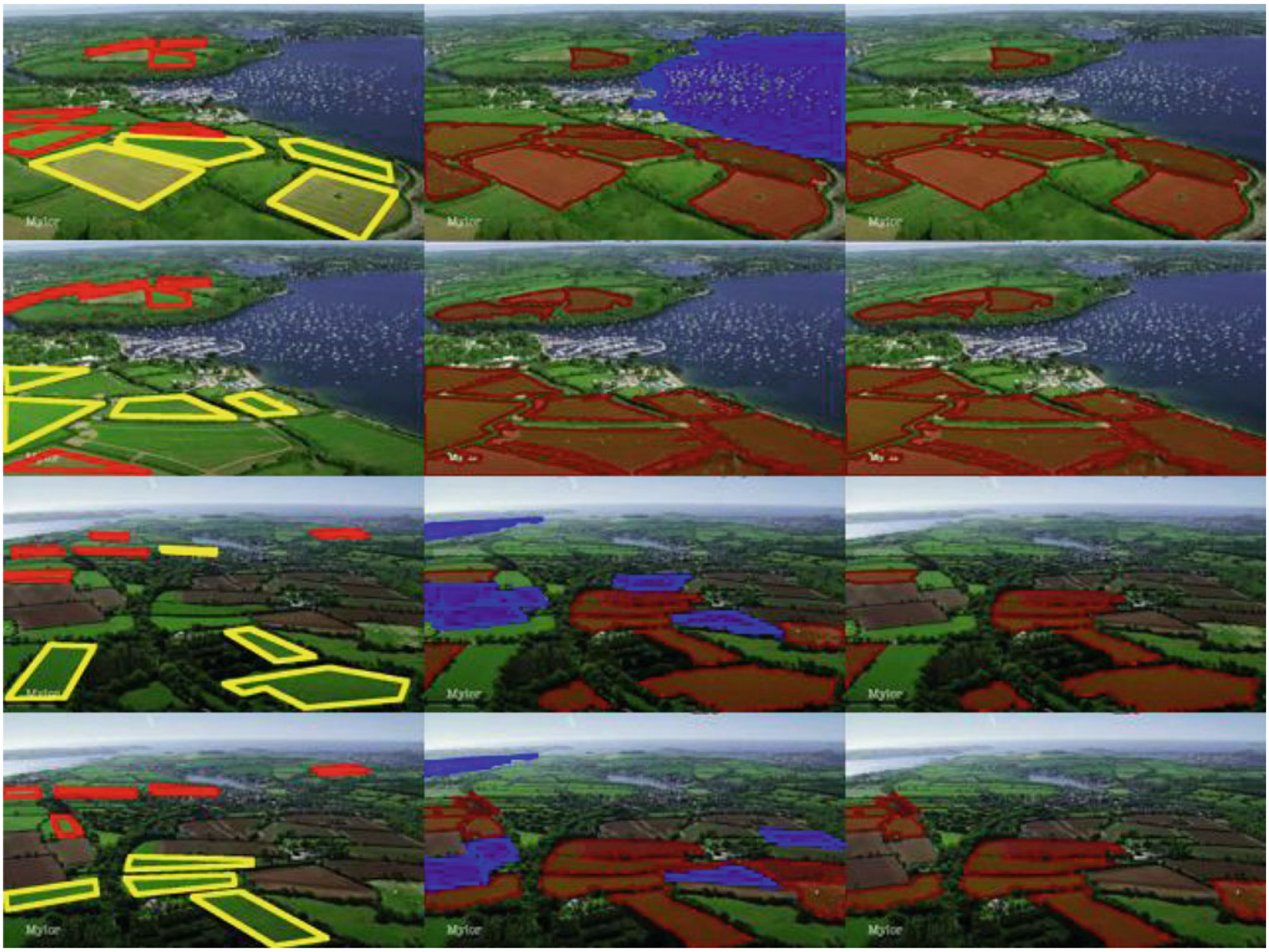

Fig. 9 Landable field detection. Results on landable field dataset: (*left*) pilot annotations with *yellow polygons* signifying top four most landable areas, (*middle*) landable areas detected by our approach, and (*right*) our approach combined with color. We detect a significant number of the landable areas labeled by the expert annotator, especially those clearly visible by the camera. Additional detections appear similar to the expert annotated regions, however, are differentiated by other factors such as field type and slope that are not taken into account by our approach. A simple color feature used in combination with our approach helps to avoid unwanted regions (highlighted in *blue*), and results in detections that more closely resemble the pilot's selections

where $W(c, r) \subset F$ defines an $r \times r$ window centered at c and $\mu(c)$ is the mean intensity within the window. Flat filtering is then performed by thresholding this measure at a pre-specified value. We evaluate this method at different threshold values, and used $r = 5$ throughout our experiments as we found this to give the best results.

We also consider a more sophisticated technique and compare against mean-shift image segmentation [7], using the EDISON segmentation library [6]. Although our method cannot be expected to be significantly more accurate than this approach we show that it performs similarly for finding the boundaries of featureless, unobstructed regions, while being far more efficient.

For runway detection, we compare against the method of [30] that employs a windowed Hough transform for rectangular region detection, and is representative of the approaches that look for landable fields as rectangles. We will refer to it as *WH*.

6.3 Results

Runways We evaluated our approach using the three infrared runway sequences. They consist of mid-resolution (640 × 480) and low-resolution (320 × 240) infrared images, each sequence being made of roughly 250–500 frames. As runways are defined by rectangular planar surfaces, we ran our approach with polygonal complexity parameter $N = 4$.

A qualitative comparison with the baseline on the runway sequences is provided in Fig. 6. The top four recognition results are shown with red indicating the top region. The baseline technique results in many false and missed detections.

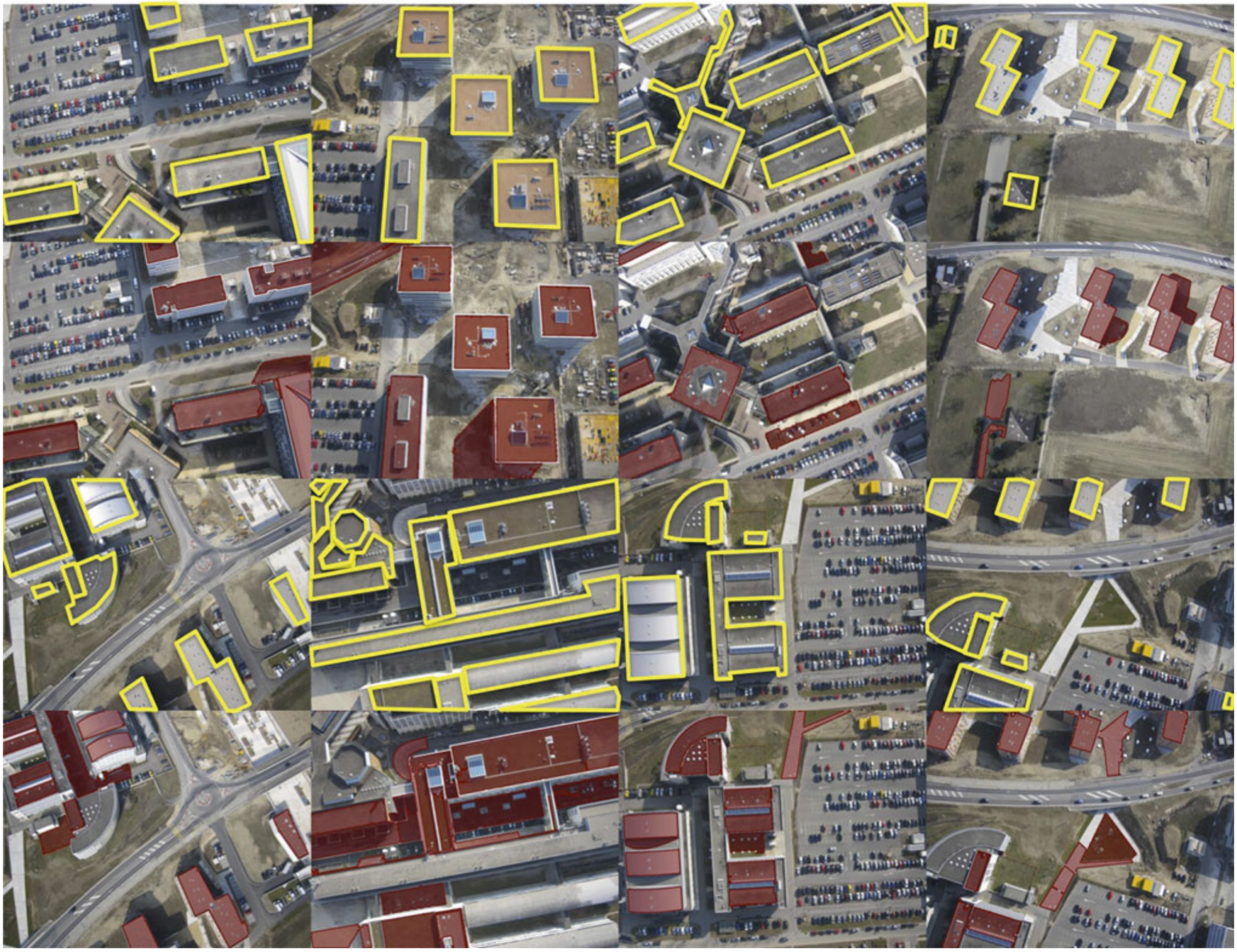

Fig. 10 Rooftop detection. Detected rooftops are highlighted in *red*, ground-truth rooftop annotations are shown in *yellow*. Our approach faithfully detects many of the relatively featureless, regularly shaped rooftops displayed in these images. False detections largely consist of other regularly shaped, constant textured regions including polygonal sidewalks and courtyards

Compared to the baseline, our approach more consistently and accurately detects the runway region.

Figure 7 displays the top-n recognition results for the different approaches. The runway is considered detected if it is found as one of the top n detected regions. Leveraging area-based cues and spatio-temporal consistency results in a significant improvement over the baseline rectangular region detection method. The baseline method was run with knowledge of the ground-truth scale and affine transformation parameters of the runway that are required as input to their method. In contrast, our approach has no knowledge of these parameters and they are estimated automatically as part of the detection, which means that it starts with a handicap.

The use of area-based cues alone results in a significant improvement over the baseline with single-frame component tree segmentation and line detection exhibiting a fairly reasonable performance, detecting the runway as one of the top ten regions in most images. Temporal consistency can result in an even further improvement with our approach, as is especially the case for the low-resolution day sequence whose top-three recognition rate increased from 85 % to nearly 100 %.

Figure 7 also displays the detection accuracy of each method. Once again we gain a significant improvement over the baseline method with our approach resulting in an average detection accuracy of 70 % across the different settings compared to 55 % for the baseline.

Performance time and accuracy with respect to window size is displayed in Fig. 8. Single-frame performance for our approach is well under a second for both image resolutions, with a speed of 5 Hz for 320 × 240 and 1.25 Hz for 640 × 480. While performance time remains reasonable with larger window sizes for 320 × 240, it is more costly for the mid-resolution sequences. Unlike at lower resolutions,

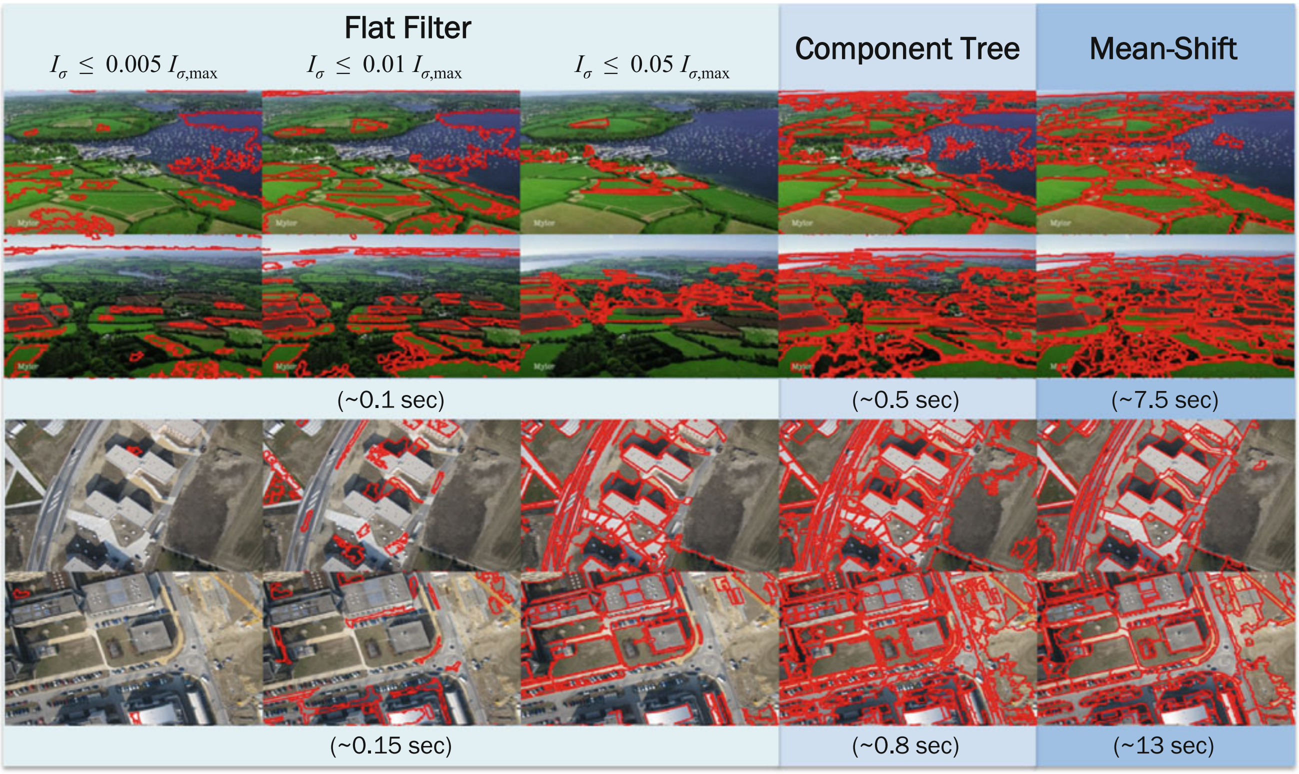
Flat Filter
$I_\sigma \leq 0.005\, I_{\sigma,\max}$
$I_\sigma \leq 0.01\, I_{\sigma,\max}$
$I_\sigma \leq 0.05\, I_{\sigma,\max}$
Component Tree
Mean-Shift
(~0.1 sec)
(~0.5 sec)
(~7.5 sec)
(~0.15 sec)
(~0.8 sec)
(~13 sec)

Fig. 11 Segmentation quality. Results are displayed for our component tree approach and the simple flat filter intensity variation and color mean-shift baselines. Both component tree segmentation and flat filtering are evaluated using intensity variation. Threshold-based image segmentation, although efficient, is highly sensitive to the choice of threshold. The segmentation quality obtained with our approach is similar to that obtained with mean shift, but at a fraction of the computation time

however, temporal consistency is less important for these sequences with a similar accuracy across different window sizes.

Unprepared landing sites We also evaluated our approach for the detection of landable fields, consisting of flat, planar, regularly shaped expanses such as agricultural fields and dirt strips, and building rooftops.

For landable fields, we used a dataset consisting of two aerial sequences of an aircraft flying above a rural area. They consist of 854 × 480 color images each of roughly 85 images in length. The geometry of these fields is more complex than the rectangular runways considered.

In order to assess the performance of our algorithm, we asked a trained glider pilot to label a handful of images from these sequences with the ten most landable areas in each image. Gliders are planes without engines. They sometimes have to land out on unprepared surfaces if they cannot make it back to an airfield due to adverse conditions. Glider pilots are therefore trained to recognize suitable landing spots, which are flat and 300–400 meters long.

Figure 9 displays the detection results of our approach on the pilot-annotated images. Our approach detects a significant number of the landable areas labeled by the expert annotator. Single-frame performance (*SL*) is displayed with multi-frame (*MP*) performing similarly.[1] The top four most landable areas as deemed by the pilot are colored in yellow, most of which are detected by our algorithm. Missed detections mostly consist of distant regions not clearly visible in the image. Similarly, extraneous detections consist of regions with similar appearance and geometry to those annotated by the pilot, and were eliminated by him due to factors not taken into consideration by our algorithm. For example, the pilot took into account the type and slope of the landing surface, preferring grass fields un-occluded by trees to dirt patches. A simple, efficient color thresholding can be used in combination with our approach to prefer green regions to dirt-colored and blue ones, and help avoid unwanted areas like dirt-strips, regularly shaped lakes and sky. These results are also included in Fig. 9 and are seen to more closely resemble the pilot's selections.

For building rooftops, we used aerial images captured from a Sensefly drone[2] flying above the EPFL campus. This dataset consists of a collection of 164 (1000 × 750) images. Figure 10 displays the detections obtained with our approach.

[1] Results on the full sequences are included as part of supplementary material for all datasets. *MP* results for landable fields are also provided.

[2] http://www.sensefly.com.

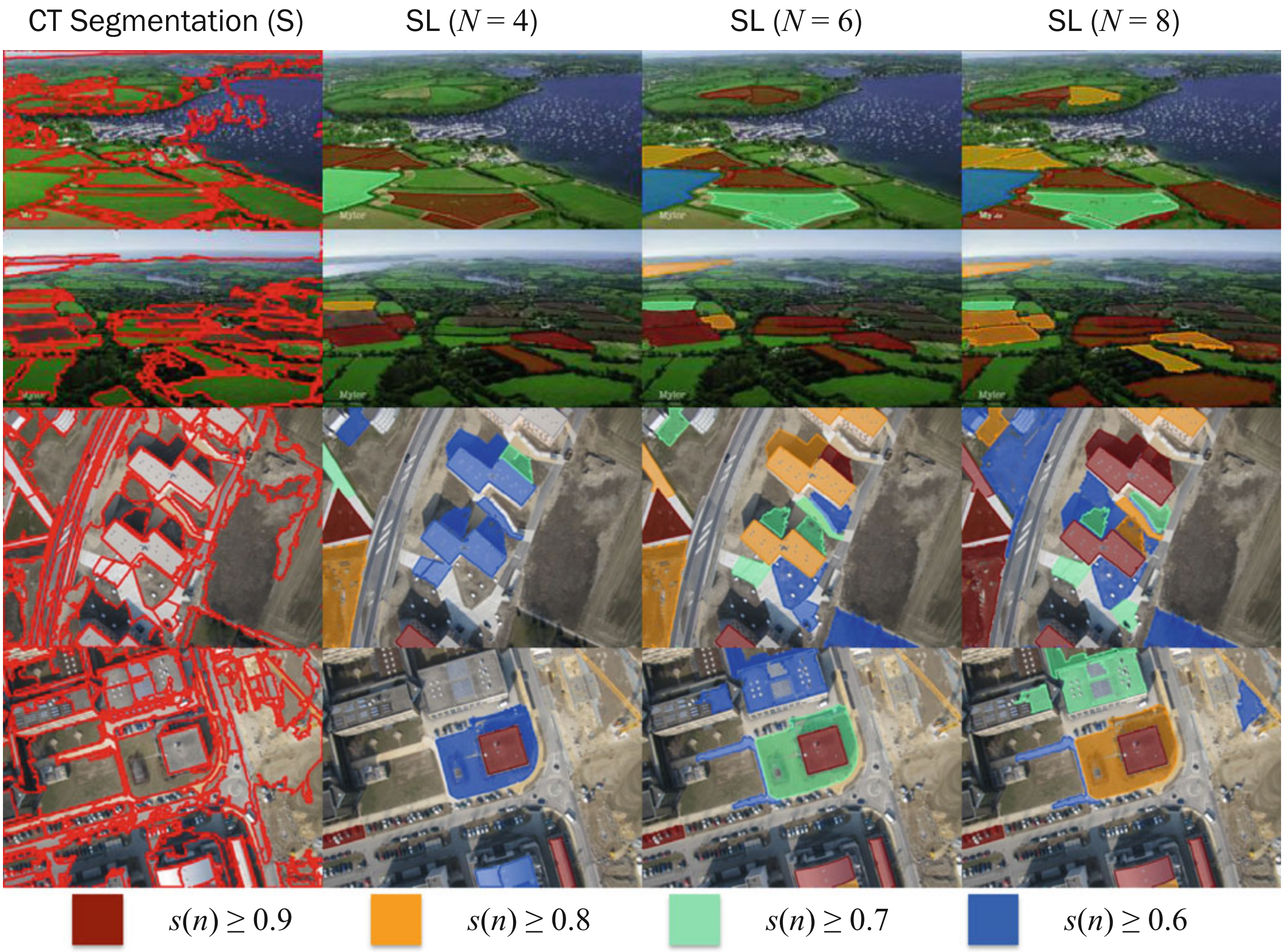

Fig. 12 Shape regularity: (*left*) regions found with component tree segmentation and (*right*) regions scored according to shape regularity with different N. The color coding indicates each region's regularity score $s(n)$. Our approach successfully discriminates regularly and irregularly shaped regions. While $N = 4$ favors fairly simple regions, larger values of N also detects regions of a more complex shape. For reasonably sized N, our approach performs similarly across the different values, as is seen for $N = 6, 8$

Most of the rooftops in these images are detected, our method favoring unobstructed featureless rooftop regions without imposing explicit rectilinearity constraints as we did in earlier work [2,15]. Not surprisingly, false detections largely consist of other regularly shaped constant textured regions, such as building walls and polygonal sidewalks and courtyards. Such detections can be easily filtered at a later detection stage, *e.g.*, ., using image homographies to discriminate vertical from horizontal surfaces.

Segmentation quality and shape regularity A comparison of our component tree segmentation approach to simple flat filtering, both applied to the intensity variation measure, and mean-shift color image segmentation is provided in Fig. 11. Average computation time per image is also shown. The segmentation quality obtained with our approach is similar to mean-shift, however, at a fraction of the computation time. A simple flat filtering, although efficient, is highly sensitive to an appropriate choice of threshold, with a different threshold per dataset giving favorable results. In contrast, our approach can be seen as integrating across this parameter, and the exact same setting of our approach works equally across datasets, while still maintaining a real-time computation time of under one second.

We also compare the performance of our approach for different shape complexity parameters. Figure 12 displays the results on the landable field and rooftop datasets for different values of N. In the figure, the different regions are highlighted according to the employed detection threshold T applied to the shape regularity measure. Using a small complexity parameter $N = 4$ favors fairly simple regions, with larger values of N resulting in regions of increasing complexity. For reasonably sized N, our approach performs similarly across the different values, as is seen for $N = 6, 8$. Comparing with the original component tree segmentation, we see that our approach is able to successfully discriminate regularly and irregularly shaped regions to detect man-made landing sites.

7 Conclusion

We have presented a real-time algorithm for landing site assessment in unconstrained man-made environments that exploits region shape, a critical cue in such environments, in addition to purely local appearance. We rely on a component tree for real-time image segmentation. It does not depend on simple but potentially unreliable image thresholding and yields region boundaries similar to those produced by more computationally expensive techniques. The component tree is complemented by a Hough-like voting scheme to select polygonal regions and extended for multi-frame processing to improve reliability in low-resolution images.

We evaluated our approach on challenging aerial infrared and color video sequences. By jointly leveraging area-based cues and enforcing spatio-temporal consistency and geometric regularity, we achieved reliable detection and assessment of runways, arbitrarily shaped landable fields, and rooftops. We also significantly outperformed our baselines. Our experiments on landable fields involved annotations by an expert pilot. They demonstrate that we can approach human performance and provide insight into the types of visual features that would be useful for further improvements.

Promising avenues of future work include the integration of additional features such as 3D geometry and texture, and the evaluation of our approach within a larger automated landing system. We are also interested in the application of our approach to other vision domains including image registration and matching in man-made environments.

References

1. Barnes, N., Loy, G., Shaw, D.: The regular polygon detector. Pattern Recognit. **43**(3), 592–602 (2010)
2. Bignone, F., Henricsson, O., Fua, P., Stricker, M.: Automatic extraction of generic house roofs from high resolution aerial imagery. In: European Conference on Computer Vision, pp. 85–96 (1996)
3. Bosch, S., Lacroix, S., Caballero, F.: Autonomous detection of safe landing areas for a UAV from monocular images. In: International Conference on Intelligent Robots and Systems (2006)
4. Brockers, R., Buffard, P., Ma, J., Matthies, L., Tomlin, C.: Autonomous landing and ingress of micro-air-vechicles in urban environments based on monocular vision. In: SPIE Conference on Micro- and Nanotechnology Sensors, Systems and Applications III (2011)
5. Cheng, Y., Johnson, A., Matthies, L., Wolf, A.: Passive imaging based hazard avoidance for spacecraft safe landing. In: International Symposium on Artifical Intelligence, Robotics and Automation in Space (2001)
6. Christoudias, M., Georgescu, B., Meer, P.: Synergism in low level vision. In: International Conference on Pattern Recognition (2002)
7. Comaniciu, D., Meer, P.: Mean Shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
8. Couprie, M., Bertrand, G.: Topological grayscale watersheld transformation. In: SPIE Vision Geometry VI (1997)
9. Cuseo, J., Fleming, P., Hoff, W., Sklair, C., Eshera, M., Whitten, G.: Machine vision techniques for planetary terminal descent hazard avoidance and landmark tracking. In: American Control Conference (1991)
10. Di, N., Zhu, M., Wang, Y.: Real time method for airport detection in aerial images. In: International Conference on Audio, Language and Image Processing (2008)
11. Donoser, M., Bischof, H.: 3D segmentation by maximally stable volumes (MSVs). In: International Conference on Pattern Recognition (2006)
12. Duda, R., Hart, P.: Use of the hough transform to detect lines and curves in pictures. In: Communications of the ACM (1972)
13. Fitzgerald, D.: Landing site selection for UAV forced landing using machine vision. Ph.D. thesis, Queensland University of Technology (2007)
14. Fu, Y., Xing, K., Han, X., Zhang, H.: Airfield runway detection from synthetic aperture radar image. In: Congress on Image and Signal Processing (2008)
15. Fua, P., Hanson, A.: An optimization framework for feature extraction. Machine Vis. Appl. **4**(2), 59–87 (1991)
16. Garcia-Pardo, P., Sukhatme, G., Montgomery, J.: Towards vision-based safe landing for an autonomous helicopter. Robot. Auton. Sys. **38**, 19–29 (2001)
17. Gong, X., Abbot, A.L., Fleming, G.A.: A survey of techniques for detection and tracking of airport runways. In: 4th AIAA Aerospace Sciences Meeting and Exhibit (2006)
18. Hamza, R., Ibrahim, M., Ramegowwda, D., Rao, V.: Runway position and moving object detection prior to landing. In: Hammoud, R.I. (ed.) Augmented Vision Perception in Infrared: Algorithms and Applied Systems, Springer, Berlin (2009)
19. Heijmans, H.J.A.M.: Theoretical aspects of gray-level morphology. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 568–582 (1991)
20. Hermansson, J., Gising, A., Skoglund, M.A., Schon, T.B.: Autonomous landing of an unmanned aerial vehicle. Tech. Rep. Linkopings Universitet (2010)
21. Hoff, W., Sklair, C.: Planetary terminal descent hazard avoidance using optical flow. In: International Conference on Robotics and Automation (1990)
22. Howard, A., Jones, B., Serrano, N.: Integrated sensing for entry, descent, and landing of a robotic spacecraft. In: IEEE Transactions on Aerospace and Electronic Systems, vol. 47, no. 1, pp. 295–304 (2011)
23. Howard, A., Seraji, H.: A fuzzy rule-based safety index for landing site risk assessment. In: 5th Biannual World Automation Congress (2002)
24. Huertas, A., Cheng, Y., Madison, R.: Passive imaging based multi-cue hazard detection for spacecraft safe landing. In: IEEE Aerospace Conference (2006)
25. Huertas, A., Cole, W., Nevatia, R.: Detecting runways in complex airport scenes. Comput. Vis. Grap. Image Process. **51**(2) (1990)
26. Johnson, A., Keim, J., Ivanov, T.: Analysis of flash lidar filed test data for safe lunar landing. In: IEEE Aerospace Conference (2010)
27. Johnson, A., Klumpp, A., Collier, J., Wolf, A.: Lidar-based hazard avoidance for safe landing on mars. In: 11th AAS/AIAA Space Flight Mechancis Meeting (2001)
28. Johnson, A., Montgomery, J., Matthies, L.: Vision guided landing of an autonomous helicopter in hazardous terrain. In: International Conference on Robotics and Automation (2005)
29. Jones, R.: Connected filtering and segmentation using component trees. Comput. Vis. Image Underst. **75**(3), 215–228 (1999)
30. Jung, C.R., Schramm, R.: Rectangle detection based on a windowed hough transform. In: Computer Graphics and Image Processing (2004)
31. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **20**(2), 91–110 (2004)

32. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
33. Matthies, L., Huertas, A., Cheng, Y., Johnson, A.: Stereo vision and shadow analysis for landing hazard detection. In: International Conference on Robotics and Automation (2008)
34. Mckeown, D., Harvey, W., Mcdermott, J.: Rule-based interpretation of aerial imagery. IEEE Trans. Pattern Anal. Mach. Intell. **7**(5), 570–585 (1985)
35. Najman, L., Couprie, M.: Building the component tree in quasilinear time. Image Process. **15**(11), 3531–3539 (2006)
36. Nevatia, R., Babu, R.: Linear feature extraction and description. Comput. Graph. Image Process. **13**(3), 257–269 (1980)
37. Pien, H.: Autonomous hazard detection and avoidance. Tech. rep., NASA (1992)
38. Rogata, P., Sotto, E.D., Camara, F., Caramagno, A., Reborado, J.M., Correia, B., Duarte, P., Mancuso, S.: Design and performance assessment of hazard avoidance techniques for vision-based landing. Acta Astronaut. (2007)
39. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Trans. Image Process. **7**(4), 555–570 (1998)
40. Scherer, S., Chamberlain, L., Singh, S.: First results in autonomous landing and obstacle avoidance by a full-scale helicopter. In: International Conference on Robotics and Automation (2012)
41. Serrano, N., Bajracharya, M., Howard, M., Seraji, H.: A novel tiered sensor fusion approach for terrain characterization and safe landing assessment. In: IEEE Aerospace Conference (2006)
42. Shakerina, O., Vidal, R., Sharp, C.S., Ma, Y., Sastry, S.S.: Multiple view motion estimation and control for landing and unmanned aerial vehicle. In: International Conference on Robotics and Automation (2002)
43. Shang, J., Shi, Z.: Vision-based runway recognition for UAV autonomous landing. Int. J. Comput. Sci. Netw. Security **7**(3), 112–117 (2007)
44. Sharp, C.S., Shakernia, O., Sastry, S.S.: A vision system for landing an unmanned aerial vehicle. In: International Conference on Robotics and Automation (2001)
45. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
46. Suetens, P., Fua, P., Hanson, A.: Computational strategies for object recognition. ACM Comput. Surv. **24**(1), 5–61 (1992)
47. Takahashi, M.D., Abershitz, A., Rubinets, R., Whalley, M.: Evaluation of safe landing area determination algorithms for autonomous rotorcraft using site benchmarking. American Helicopter Society 67th Annual Forum (2011)
48. Tandra, S., Rahman, Z.: Robust edge-detection algorithm for runway edge detection. In: SPIE Image Processing: Machine Vision Applications (2008)
49. Templeton, T., Shim, D.H., Geyer, C., Sastry, S.S.: Autonomous vision-based landing and terrain mapping using an Mpc-controlled unmanned rotorcraft. In: International Conference on Robotics and Automation (2007)
50. Theodore, C.T., Tischler, M.B.: Precision autonomous landing adaptive control experiment (PALACE). In: Army Science Conference (2006)
51. Udrea, R.M., Vizireanu, N.: Iterative generalization of morpholigical skeleton. J. Electron. Imaging **16**(1) (2007)
52. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/
53. Vincent, L.: Local grayscale granulometrics based on opening trees. In: Maragos, P., Schafer, R.W., Butt, M.A. (eds.) Mathematical Morphology and Its Applications to Image and Signal Processing. Kluwer Academic Publishers (1996)
54. Vizireanu, D.N., Pirnog, C., Lăzărescu, V., Vizireanu, A.: The skeleton structure—an improved compression algorithm with perfect reconstruction. J. Digit. Imaging **14**, 241–242 (2001)
55. Wendt, P.D., Coyle, E.J., Gallagher, N.C.: Stack filters. Acoust. Speech Signal Process. **34**(4), 898–911 (1986)
56. Zongur, U.: Detection of airport runways in optical satellite images. Master's thesis, Middle East Technical University (2009)

Author Biographies

Xiaolu Sun received both his BS and MS degrees in electrical engineering from Tsinghua University, China in 2007 and 2010,respectively. Afterwards he joined the computer vision laboratory of EPFL, Switzerland, to pursuit a PhD degree. His research interests include computer vision (image segmentation and object detection) and robot vision (vision based navigation).

C. Mario Christoudias received a BS degree in electrical and computer engineering from Rutgers University in 2002 and MS and PhD degrees in computer science from the Massachusetts Institute of Technology in 2004 and 2009. He is currently a postdoctoral researcher in the Computer Vision Laboratory at the Ecole Polytechnique Federale de Lausanne. His main research interests are in the areas of computer vision and machine learning.

Vincent Lepetit is a Senior Researcher at the Computer Vision Laboratory, EPFL. He received the engineering and master degrees in Computer Science from the ESIAL in 1996. He received the PhD degree in Computer Vision in 2001 from the University of Nancy, France, after working in the ISA INRIA team. He then joined the Virtual Reality Lab at EPFL (Swiss Federal Institute of Technology) as a post-doctoral fellow and became a founding member of the Computer Vision Laboratory in 2004. His research interests include vision-based Augmented Reality, 3D camera tracking, object recognition, and 3D reconstruction. He often serves as program committee member and area chair of major vision conferences.

Pascal Fua received an engineering degree from Ecole Polytechnique, Paris, in 1984 and the Ph.D. degree in Computer Science from the University of Orsay in 1989. He joined EPFL (Swiss Federal Institute of Technology) in 1996 where he is now a Professor in the School of Computer and Communication Science. Before that, he worked at SRI International and at INRIA Sophia-Antipolis as a Computer Scientist. His research interests include shape modeling and motion recovery from images, analysis of microscopy images, and Augmented Reality. He has (co)authored over 300 publications in refereed journals and conferences. He is an IEEE Fellow and has been an Associate Editor of IEEE journal Transactions for Pattern Analysis and Machine Intelligence. He often serves as program committee member, area chair, and program chair of major vision conferences.