

TIME-TO-CONTACT ESTIMATION IN LANDING SCENARIOS USING FEATURE SCALES

G.C.H.E. de Croon, D. Izzo, and G. Schiavone

Advanced Concepts Team, European Space Agency

ABSTRACT

We present a feasibility study on using the scale of SIFT-features for estimating time-to-contact in space landing scenarios. The time-to-contact is a measure of the lander's height divided by its vertical velocity. In this feasibility study the lander is assumed to have a radar altimeter, using the time-to-contact for estimating the vertical velocity. Experiments with virtual zooms on images show that: (1) the image size, the number of frames per second, and the amount of memory are important factors for the success of using feature scales for time-to-impact estimates, and (2) the distribution of estimates has many outliers, necessitating robust estimators such as the median. Experiments with landing scenarios in the PANGU simulator show that the vertical velocity estimates are accurate enough for successful landing in different conditions.

Key words: Autonomous landing; feature scale; SIFT.

1. INTRODUCTION

Autonomous landing is a crucial capability for extraterrestrial exploration missions. Many existing studies focus on the use of passive cameras for state estimation, since they are light-weight and consume relatively little energy. Computer vision methods have been shown to significantly improve the accuracy of the state estimates used for landing (cf. [JWC⁺07, MTR⁺09, SMS10]).

Flying animals are capable of safe and accurate landings, while using limited sensors and processing. Therefore, they have served as inspiration for extraterrestrial landing studies [OBI09, VRVS10, IWSss]. The focus of these studies is on the use of *ventral optic flow*, a measure of the translational velocity divided by the height. Bees are known to use optic flow for controlling their speed and height, also when landing [Pre87, BSZC05, BSZ⁺06]. In [VRVS10] a control law is studied that keeps the ventral optic flow constant, leading to successful simulated landings.

A related measure known to play an important role in animal landings is the *time-to-contact* (e.g., [Lee67,

LDGW93]). In the context of landing, the time-to-contact is a measure of the height divided by the vertical velocity¹. When approaching the ground, there is an increasing expansion of imaged ground features. This expansion is typically measured by means of the optic flow of interest points in the image. The flow is directed away from the focus of expansion (FoE), which indicates the direction of travel. Accurate time-to-contact estimates depend on the FoE either being known (as in a pure vertical descent) or estimated. There are methods for estimating the FoE on the basis of inertial measurements and optic flow vectors, but it remains a difficult problem [TWO96, MGS08].

It is also possible to estimate the time-to-contact using the scales of features in the images. In this article, we use the features obtained with the Scale Invariant Feature Transform (SIFT) [Low04] to this end. Estimating time-to-contact on the basis of SIFT feature scales has the advantages that (i) SIFT features can also be used for determining optic flow and recognizing a landing site, (ii) it may be less sensitive to (subpixel) errors than estimating the time-to-contact on the basis of optic flow, and (iii) SIFT robustly matches features under larger image transformations. Previous studies employing feature scales [CG09, ANC09] have focused on (robotic) driving scenarios. In this article, we present a feasibility study on using feature scales as an information source in extraterrestrial landing scenarios.

The remainder of the article is organized as follows. In Section 2, we explain how the feature size can be used to determine time-to-contact and how it is implemented in our experiments. In Section 3, we apply the method to virtual approaches on the basis of the publicly available images of the lunar surface. The virtual approaches are used to investigate some of the key method parameters required for successful time-to-contact estimates. Subsequently, in Section 4, the time-to-contact estimates are used in simulated landing scenarios. In principle the time-to-contact could be used directly for controlling the vertical thrust on the lander. However, in this feasibility study the lander is assumed to use a radar altimeter for determining the height and the time-to-contact for determining the vertical velocity. Conclusions are drawn in

¹Please remark that this measure is only equal to the actual time to contact if the velocity remains constant, which is typically not the case in extraterrestrial landing scenarios.

2. TIME-TO-CONTACT

In this section, we first explain how feature scale can be used for determining time-to-contact (Subsection 2.1). Subsequently, the algorithm is introduced that estimates the time-to-contact on the basis of the scales of SIFT features (Subsection 2.2).

2.1. Estimation of time-to-contact with scale

Figure 1 shows a schematic of a lander at two subsequent moments in time, t_1 and $t_2 = t_1 + \Delta t$. The lander observes a feature on the ground, illustrated with a black circle. The choice for a ‘spherical’ feature derives from the *assumption* that the scale of the feature does not depend on the direction from which the lander observes it². Under this assumption, the scales of the feature in the images at the subsequent time steps (σ_1 and σ_2) only depend on the distances with the lander (d_1 and d_2). The altitudes of the lander are indicated with z_1, z_2 .

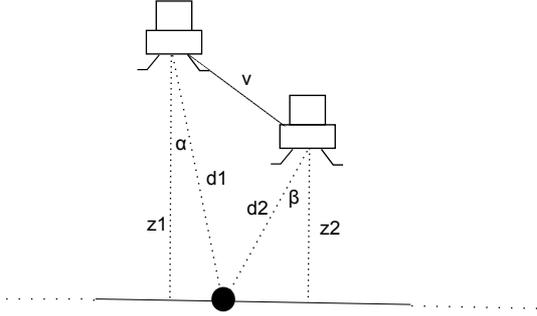


Figure 1. Schematic of a lander observing a feature (black circle) on the ground plane at two different time steps. The scales of the feature in the images can be used for determining the time-to-contact.

The scale and distance are related to each other as follows:

$$d_1 \sigma_1 = d_2 \sigma_2, \quad (1)$$

implying that if the lander halves its distance to the feature, the feature’s scale in the image becomes twice as large. Furthermore, we have the following relations:

$$z_1 = \cos(\alpha) d_1 \quad (2)$$

$$z_2 = \cos(\beta) d_2 = \cos(\beta) \frac{d_1 \sigma_1}{\sigma_2} \quad (3)$$

²This assumption has similar effects as the more common assumption that the observer moves straight towards the center of a planar object with as normal vector the movement direction.

, where $\cos(\alpha)$ and $\cos(\beta)$ rely on the lander’s attitude being known or estimated.

The time-to-contact τ is defined as follows:

$$\tau = \frac{z_2}{z_1 - z_2}. \quad (4)$$

In the equation, τ is expressed in the number of frames. It can be expressed in seconds as follows: $\tau(s) = \tau \Delta t$. The estimated time-to-contact τ is only equal to the actual time to contact under the *assumptions* that (i) the velocity of the lander is constant, and (ii) the landing surface (on which the feature is located) is flat. We note that the first assumption is typically violated. However, this can be resolved either by updating τ frequently enough (as is presumed in this study) or by fitting more complex motion models on the basis of multiple scale measurements [MPN⁺09].

Equation 4 can be expressed in variables observed by the lander as follows:

$$\tau = \frac{z_2}{z_1 - z_2} = \frac{1}{z_1/z_2 - 1} \quad (5)$$

which, with the help of equations 2 and 3, can be transformed to:

$$\tau = \frac{1}{\frac{\cos(\alpha) d_1 \sigma_2}{\cos(\beta) d_1 \sigma_1} - 1} = \quad (6)$$

$$\frac{1}{\frac{\cos(\alpha) \sigma_2}{\cos(\beta) \sigma_1} - 1} = \frac{\cos(\beta) \sigma_1}{\cos(\alpha) \sigma_2 - \cos(\beta) \sigma_1}. \quad (7)$$

If one assumes a camera with a relatively small field-of-view (say $\leq 40^\circ$), $\cos(\alpha)$ and $\cos(\beta)$ will be close to 1. Furthermore, having the time step between the two observations go to 0 leads to the formula used in [CG09, ANC09]:

$$\lim_{\Delta t \rightarrow 0} \tau(s) = \lim_{\Delta t \rightarrow 0} \Delta t \frac{\sigma_1}{(\sigma_2 - \sigma_1)} = \quad (8)$$

$$\lim_{\Delta t \rightarrow 0} \frac{\sigma_1}{(\sigma_2 - \sigma_1)/\Delta t} = \sigma_1 / \dot{\sigma}_1 \quad (9)$$

2.2. Scale Based Vision Algorithm (SBVA)

To automatically find corresponding features in two images is a standard computer vision problem, to which well-known solutions exist. The current state-of-the-art in feature matching between images are the SIFT-features mentioned in the introduction. They are invariant to scale changes, translations, and rotations. The features are located in an image pyramid and have a position, orientation, scale, and a descriptor used for the matching. The scales of the features can be used for determining the time-to-contact τ .

During landing, the Scale Based Vision Algorithm (SBVA) receives a series of subsequent images. Each image is processed to locate SIFT-features, which are

matched to the features of the previous image. The matching is currently implemented in the standard way [Low04], and is thus only based on the feature descriptors (spatial restrictions based on state estimates of the lander and its camera can be added at a later stage).

In general, the scale at time t of the feature, s_t , is regarded as a noisy measurement of the true scale σ_t . It is assumed that the noise is Gaussian with standard deviation η : $s_t \sim \mathcal{N}(\sigma_t, \eta)$. Considering the scale-measurements as noisy implies that one can get a better estimate of the true scales σ_t by performing multiple observations over time. Therefore, each feature is tracked to a maximum number of images (time steps) M , adding a new scale each time that the feature has a match in the new image. The ‘old’ feature’s descriptor is updated with the new one, and the feature is removed if it has no match in the new image.

To estimate τ for a single feature, a maximum likelihood linear fit is determined for the measurements $\{s_1, \dots, s_m\}$ with $\max(m) = M$: $\sigma_t = \dot{\sigma}t + \sigma_0$. The slope of the line represents $\dot{\sigma}$, while the point on the line at t represents σ_t . Since SBVA always uses a linear fit of the scales, there should exist a trade-off between higher accuracy due to the size of the ‘memory’ M and the larger error due to discrepancies between the actual motion and the linear motion model. Figure 2 illustrates the tracking of features over time (top) and the way in which the m measurements per feature are used to estimate $\dot{\sigma}$ (bottom).

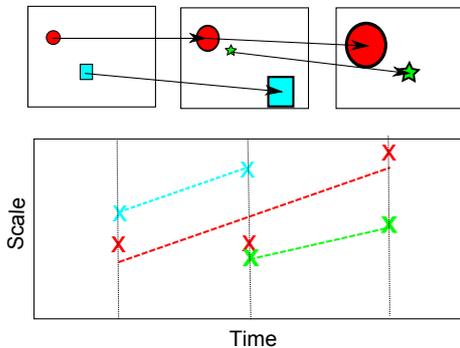


Figure 2. Features are tracked over multiple images, in order to better estimate $\dot{\sigma}$. Top: The red circular feature is tracked over $m = 3$ frames, while the square blue feature and green star feature are only tracked in $m = 2$ images. Bottom: Linear fits (dashed lines) are determined for the scale measurements s_t (crosses).

Each matched feature f results in an estimate of the time-to-contact $\hat{\tau}_f$. The presence of multiple matched features results in a distribution over different time-to-contact values $p(\hat{\tau})$. Both the mean value and the median value of the distribution will be explored as estimation methods. The motivation for using the median value is that the distribution may have a considerable number of outliers, which have a large impact on the mean.

3. ZOOMING IN ON LUNAR IMAGES

In this section, we will report on experiments in which SBVA is applied to zooms of high-resolution lunar images. The zooms will correspond to a linear motion (descent with ‘constant’ speed). Hence, the goal in this section is not to model an actual descent, but to explore the effects of SBVA parameters on the accuracy of the time-to-contact estimates. In particular, the following parameters will be varied:

1. **Estimation method.** The mean and the median value of the distribution will be used as estimators.
2. **Memory size.** A larger memory size should lead to a better accuracy, if the motion is linear (constant descent speed).
3. **Image size.** The following image sizes in pixels are investigated, because they are standard sizes in the PANGU simulator used in Section 4: 512×512 , 256×256 , and 128×128 . Larger image scales may lead to more accurate scale estimates, but also imply a larger computational effort.
4. **Frequency of image acquisition.** During a real descent with a real camera, this would pertain to the number of frames per second. A higher frequency implies smaller transformations between the images, which simplifies matching. However, scale differences between the images will also be smaller, probably taking them closer to the magnitude of the measurement noise.

In Subsection 3.1, the experimental setup is explained. Subsequently, the results are given in Subsection 3.2.

3.1. Experimental setup

For the zoom experiments, the high-resolution image stitch of the Lunar Reconnaissance Orbiter Camera (LROC) has been downloaded³. From the image stitch the center area of 15000×15000 pixels is selected, since it has little perspective effects. The image of the center area has been resized to 5000×5000 pixels for use in the experiments.

Zooms are then made at random places in the image by employing a virtual camera with image dimension 512×512 . Initially, the virtual camera covers a zone of 2500×2500 pixels in the overall image stitch. The zoom then consists of linearly reducing in 23 steps the width and height of the covered area until it covers an area of 333×333 pixels. At every step in the zoom, the covered image area is resized to the camera dimension of 512×512 with bicubic sampling. All images are saved.

³<http://lroc.sese.asu.edu/>

SBVA is applied to the images in sequence, while storing the time-to-contact estimates $\hat{\tau}$. The estimates are compared to a ground-truth time-to-contact value that goes from 26 at the first image to 3 at the last image⁴. For varying the image dimensions, the saved images are loaded and resized. For varying the image capture frequencies a larger step size of 2 is employed, implying that the zoom is performed twice as fast (in 11 steps from 2500×2500 to 333×333). Per parameter setting (M , image size, capture frequency) 10 zooms are performed. Figure 3 shows four images from a zoom sequence on a part of the lunar surface.



Figure 3. Four example images from an image zoom of the lunar surface.

3.2. Results

Figure 4 shows the results for the different estimators for image dimension 512×512 , a step size of 1, and $M = 4$. The light blue lines represent the $\hat{\tau}$ values over time for the mean estimator. The thick blue line is the average over all runs. The black line is the ground truth τ . The median estimator is illustrated with green lines. The bottom graph shows the standard deviation of the $\hat{\tau}_f$ distribution over time.

Figure 4 shows that the mean of the $\hat{\tau}_f$ values results in unreliable estimates that can be in the order of tens of time steps away from the ground truth TTC. The median estimator provide better estimates of τ , although it slightly underestimates the initial large τ values. The median estimator is quite robust over the different approach sequences; the light green lines do not deviate much from the dark green line that represents the average over all approaches. Towards contact, both estimates become more accurate; indeed, the standard deviation of the $\hat{\tau}$ distribution becomes smaller over time. This can be explained by the scale differences becoming larger towards impact. The findings on the errors of the estimators are illustrated by Figure 5: it shows the absolute error $|\tau - \hat{\tau}|$ over time for both estimators.

With the same settings as mentioned above, the maximum memory size is varied as $M = \{1, 2, 3, 4, 5\}$, while estimating τ with the median, and the other settings mentioned above. Figure 6 shows the absolute error $|\tau - \hat{\tau}|$ averaged over all time steps and zooms, for the different memory sizes. The largest decrease of the absolute error takes place until $M = 4$. Further analysis showed that the cause of this finding may lie in the fact that in the

⁴The ground-truth can be determined on the basis of the covered area over time.

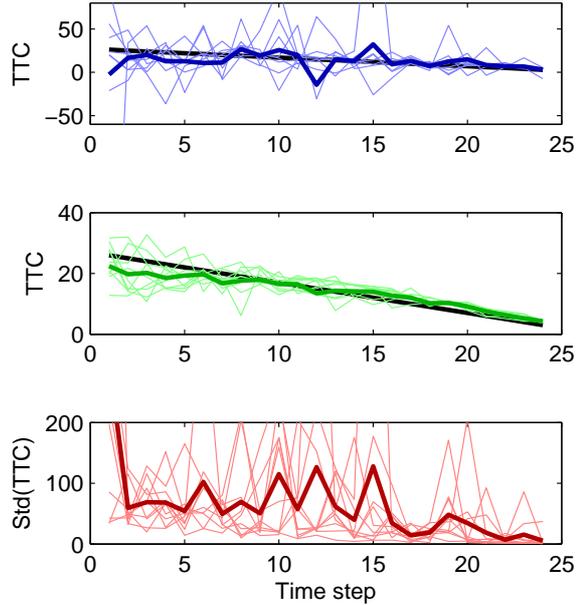


Figure 4. Time-to-contact estimates of different estimators, from top to bottom: mean τ_f value (blue), median τ_f value (green), and the standard deviation of the τ_f distribution (red).

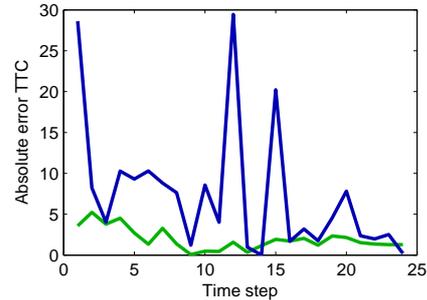


Figure 5. Absolute errors $|\tau - \hat{\tau}|$ over time for the different estimators: mean (blue) and median (green).

zoom data set, features are on average only tracked over 3 images.

Reducing the image size leads to reduced estimation performance. With $M = 4$ and an image step size of 1, the average absolute error $|\tau - \hat{\tau}|$ for the median strategy goes from 1.58 (512×512) to 2.07 (256×256) to 12.24 (128×128). Decreasing the frames per second also has a negative effect on the performance: going from an image step of 1 to an image step of 2 results in an average absolute error of 1.84 (512×512). In a real-time scenario, larger image sizes lead to the processing of fewer frames per second. The selection of the image size therefore represents a trade-off between having detailed scale measurements and having fewer such measurements per second.

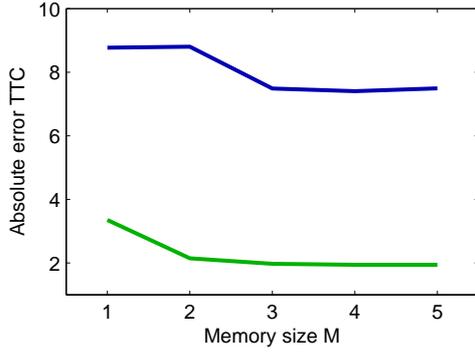


Figure 6. Average absolute errors $|\tau - \hat{\tau}|$ for different memory sizes for the different estimators: mean (blue) and median (green).

4. SIMULATED LANDING EXPERIMENTS

In this section, landing experiments are performed in simulation. First, the specifics of the simulator are explained (Subsection 4.1). Subsequently, the results of the experiments are discussed (Subsection 4.2).

4.1. Experimental Setup

In this subsection, we explain the experimental setup for the simulated landing experiments. The setup involves the generation of a visual environment (Subsection 4.1.1), the physical model (Subsection 4.1.2), and the Scale Based Landing Algorithm (SBLA) that controls the lander (Subsection 4.1.3).

4.1.1. Visual environment

The visual environment is generated with the help of PANGU [PMD04], which has been designed for realistically rendering space environments. PANGU can generate surfaces with different amounts of roughness, and with different densities and characteristics of craters and boulders. The image rendering takes into account the position of the observer and allows the adoption of light conditions typical for space environments.

We presume that the amount of texture is an important factor for the results of the SBVA. In the simulation experiments the texture is varied with (a) the ‘fractal number’ parameter ($F \in \{0.3, 0.5, 0.7, 0.9\}$) and (b) the number of craters per square kilometer ($C \in \{0.1, 0.5, 0.9, 1.3\}$). Figure 7 shows three example images, all with $C = 0.9$, but with fractal numbers $F = 0.3, 0.5, 0.9$ (from left to right).

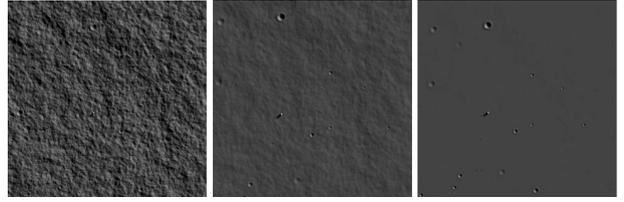


Figure 7. Three example images from the planet surface generated by PANGU, with fractal numbers $F = 0.3, 0.5, 0.9$ (from left to right).

4.1.2. Physical model

To simulate the dynamics of our lander, we use a minimal set of equations able to capture the dynamical features we need for the purpose of this paper. The equations used are:

$$\begin{aligned}
 \dot{v}_x &= u_x/m \\
 \dot{v}_y &= u_y/m \\
 \dot{v}_z &= u_z/m - g \\
 \dot{x} &= v_x \\
 \dot{y} &= v_y \\
 \dot{z} &= v_z \\
 \dot{m} &= -\frac{\sqrt{(u_x^2 + u_y^2 + u_z^2)}}{I_{sp}g_0}
 \end{aligned} \tag{10}$$

where the lander position is denoted by x, y, z , its velocity by v_x, v_y, v_z , its mass by m . The lander state will thus be $\mathbf{x} = [x, y, z, v_x, v_y, v_z, m]$. The thrust components are denoted by $\mathbf{u} = [u_x, u_y, u_z]$, while I_{sp} is the engine specific impulse (i.e. how many seconds the thrusters would thrust at 1N consuming 1kg of fuel) and $g_0 = 9.8065$.

4.1.3. State feedback control

During the descent, the lander is controlled by actuating the thrust vector u_x, u_y, u_z . The control law adopted for the thrust vector is the optimal state feedback $\mathbf{u} = \mathbf{u}^*(\mathbf{x})$ resulting from the solution of the following optimal control problem:

$$\begin{aligned}
 \text{find: } & t_f, u_x(t), u_y(t), u_z(t) \\
 \text{to maximize: } & m(t_f) \\
 \text{subject to: } & u_x(t)^2 + u_y(t)^2 + u_z(t)^2 \leq T_{max}^2 \\
 & \text{Eqq.(10)} \\
 & \mathbf{x}(0) = \mathbf{x} \\
 & z(t_f) = 0, v_z(t_f) = 0 \\
 & v_x(t_f) = 0, v_y(t_f) = 0
 \end{aligned} \tag{11}$$

We denote the solution to the above problem as $\tilde{\mathbf{u}}(t)$. Solving it at each step we extract $\mathbf{u}^*(\mathbf{x}) = \tilde{\mathbf{u}}(0)$ which is then used by the lander to control its descent. We use the algorithm described in [IWSss] to obtain such a state feedback in real time and sensor estimates to evaluate \mathbf{x} . Note that this numerical scheme is able to provide a real

time feedback accounting for optimality principles that can actually be used as a feedback to much more complicated dynamics than that that expressed in Eq.(10). For example one could actually simulate a six degrees of freedom lander with a more accurate thrust model, and still use the solution to the optimal control problem above to control the descent. The discrepancy between reality and the model used by the lander to find a state feedback would, in this case, result in a suboptimal trajectory with a minimal penalty on the consumed propellant [IWSss].

The lander is assumed to use traditional sensors for estimating most of its state variables. In this preliminary study, the height is assumed to be measured, for example with a radar altimeter. The time-to-contact estimate is used for estimating the vertical velocity, with $\hat{v}_z = h/\hat{\tau}$. The vertical velocity estimate is smoothed over the last three time steps and used by the controller for determining the thrust.

4.2. Results

SBLA succeeds in landing successfully under all different texture conditions. First, the results on the most textured surface ($F = 0.3, C = 1.3$) are discussed. The top part of Figure 8 shows the relation of the ground-truth ‘time-to-contact’ $h/-v_z$ (black dashed line) to the estimated time-to-contact $\hat{\tau}$, which is shown both non-smoothed (red line) and smoothed (green line). The estimated time-to-contact corresponds well to the ground-truth time-to-contact. The bottom part shows the resulting vertical velocity estimates (red line for non-smoothed, green line for smoothed) compared with the ground truth velocities (black dashed line). The velocity estimates deviate in the order of 3-5 m/s, which is sufficiently accurate for a successful landing.

As expected, the results depend on the amount of texture in view: less texture leads to less accurate time-to-contact estimates. Figure 9 shows the same results as above, but now for $F = 0.7, C = 0.5$, and Figure 10 shows the results for $F = 0.9, C = 0.1$. Although in both cases the landing is successful, the estimates are further away from the ground truth than for $F = 0.3, C = 1.3$. Interestingly, the results of $F = 0.7, C = 0.5$ resemble the results obtained in Section 3: when the lander enters high gate, the time-to-contact is underestimated, while close to low gate the time-to-contact is slightly overestimated.

Further analysis shows that the main parameter of influence is the fractal number F . Figure 11 shows the average absolute error in the time-to-contact estimate for all different parameter settings of C (y-axis) and F (x-axis). The color scaling is included in the figure. For $F < 0.9$ the average error is rather limited (< 2.2), but setting $F = 0.9$ leads to errors in the order of 7 seconds. The number of craters only seem to be of influence when $F = 0.9$. Actually, when there is quite some surface texture other than the craters, the SIFT feature detector prefers to track other features. Figure 12 shows SIFT fea-

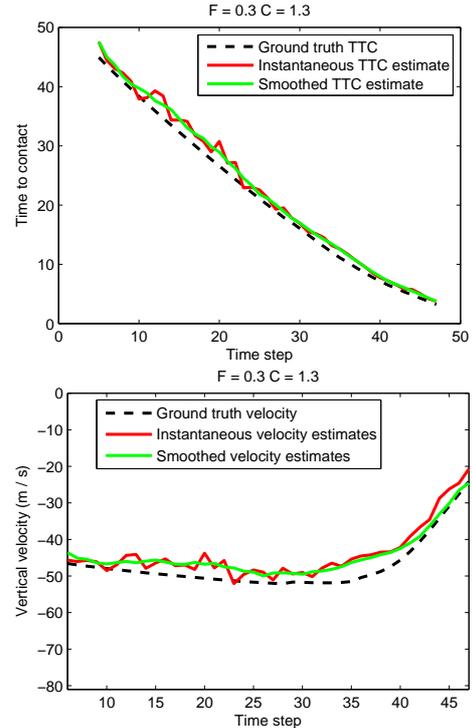


Figure 8. Results for $F = 0.3, C = 1.3$. Top: ground-truth time-to-contact (black, dashed line) and estimated time-to-contact, both non-smoothed (red line) and smoothed (green line). Bottom: ground-truth v_z (black, dashed line) and estimated v_z , both non-smoothed (red line) and smoothed (green line).

tures as circles with their radius proportional to the features’ scales. Most of them are not located on the craters themselves. This may be an advantageous property of SIFT features, since landing sites are typically located on smooth terrain.

5. CONCLUSIONS AND DISCUSSION

From the empirical results, we may draw the conclusion that it is feasible to use SIFT feature scales for estimating ‘time-to-contact’ ($h/-v_z$) in extraterrestrial landing scenarios. Experiments with virtual zooms on images show that: (1) the image size, the number of frames per second, and the amount of memory are important factors for the success of using feature scales for time-to-impact estimates, and (2) the distribution of estimates has many outliers, necessitating robust estimators such as the median. Experiments with landing scenarios in the PANGU simulator show that the vertical velocity estimates are accurate enough for successful landing in different conditions. Planetary surfaces with a more pronounced texture lead to better results than surfaces with little texture.

There are a few important matters that should be addressed by future work in order to prove the usefulness of the approach. First, the SIFT-algorithm is known to

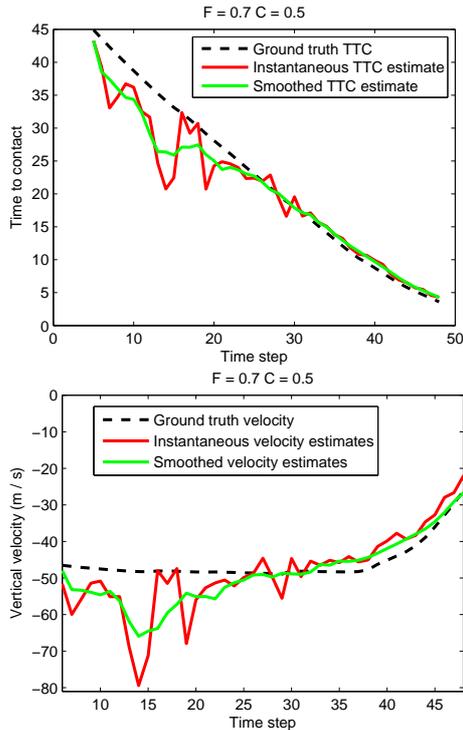


Figure 9. Results for $F = 0.7, C = 0.5$. Top: ground-truth time-to-contact (black, dashed line) and estimated time-to-contact, both non-smoothed (red line) and smoothed (green line). Bottom: ground-truth v_z (black, dashed line) and estimated v_z , both non-smoothed (red line) and smoothed (green line).

be computationally intensive, which may seem to collide with the computational restrictions on space platforms. Fortunately, in recent years fast approximations have been introduced, such as Speeded Up Robust features (SURF) [BETG08]. More recently, Rotation Invariant Fast Features (RIFF) have been introduced, which can be run real-time on mobile phones [TCT⁺10]. Although for this reason real-time computation of SIFT-like features on space platforms does not seem to be a fundamental limit of the technique, the approximations come at a cost in accuracy. In future work, the performance of the faster approximations will have to be tested as well.

Second, as demonstrated by both the zoom-experiments and the landing experiments, the accuracy of the time-to-contact estimates depends on the texture around and at the landing site. Typically, a landing site is selected for being flat and smooth, implying that there is only a moderate amount of texture. A more serious investigation into the images of actual landing sites should provide a better idea of the potential of using SIFT-features. On the one hand, SIFT-features do not necessarily depend on strong features such as craters and rocks (see Figure 11). On the other hand, it has been argued that other types of features better capture the scales in natural scenes, such as the SIRS features introduced in [ANC09]. Further research should establish the reliability of different feature types when landing on smooth terrain.

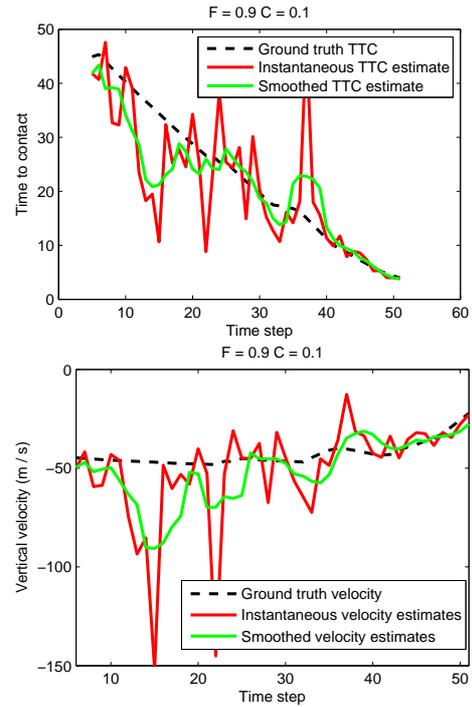


Figure 10. Results for $F = 0.9, C = 0.1$. Top: ground-truth time-to-contact (black, dashed line) and estimated time-to-contact, both non-smoothed (red line) and smoothed (green line). Bottom: ground-truth v_z (black, dashed line) and estimated v_z , both non-smoothed (red line) and smoothed (green line).

Third, the interest in the time-to-contact estimates derives from the fact that animals use such estimates for landing, without having access to sensors that measure absolute height (such as a radar altimeter). Therefore, future work should focus on using the time-to-contact directly for control, without the need for a radar altimeter.

REFERENCES

- [ANC09] G. Alenya, A. Negre, and J.L. Crowley. A comparison of three methods for measure of time to contact. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4565–4570, 2009.
- [BETG08] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.
- [BSZ⁺06] E. Baird, M.V. Srinivasan, S. Zhang, R. Lamont, and A. Cowling. Visual control of flight speed and height in honeybee. In *Simulation of Adaptive Behavior*, pages 40–51, 2006.
- [BSZC05] E. Baird, M.V. Srinivasan, S. Zhang, and A. Cowling. Visual control of flight speed in honeybees. *Journal of Experimental Biology*, 208:3895–3905, 2005.

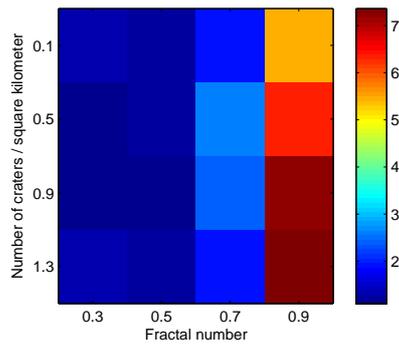


Figure 11. Average absolute errors $|\tau - \hat{\tau}|$ for the different settings of C (y-axis) and F (x-axis).

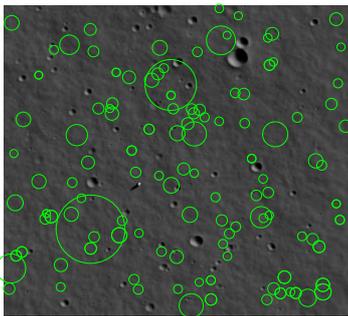


Figure 12. SIFT-features detected in an image generated by PANGU. The circles are centered at the feature positions, while the radius is proportional to the scale.

- [CG09] A. Chavez and D. Gustafson. Vision-based obstacle avoidance using SIFT features. In *ISVC (2)'2009*, pages 550–557, 2009.
- [IWSss] D. Izzo, N. Weiss, and T. Seidl. Constant optic-flow lunar landing: optimality and guidance. *Journal of Guidance, Control, and Dynamics*, in press.
- [JWC⁺07] A. Johnson, R. Willson, Y. Cheng, J. Goguen, C. Leger, M. Sanmartin, and L. Matthies. Design through operation of an image-based velocity estimation system for Mars landing. *International Journal of Computer Vision*, 74(3):319–341, 2007.
- [LDGW93] D.N. Lee, M.N.O. Davies, P.R. Green, and F.R. van der Weel. Visual control of velocity of approach by pigeons when landing. *Journal of experimental biology*, 180:85–104, 1993.
- [Lee67] D.N. Lee. A theory of visual control of braking based on information about time-to-collision. *Perception*, 5:437–459, 1967.
- [Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [MGS08] M. Makhoulta, Y. Gao, and K. Shala. A vision and behaviour based approach for short-range autonomous navigation of planetary rovers. In *ESA Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA, Noordwijk, Netherlands, 2008*.
- [MPN⁺09] D. Muller, J. Pauli, C. Nunn, S. Gormer, and S. Muller-Schneiders. Time to contact estimation using interest points. In *12th international IEEE Conference on Intelligent Transportation Systems, 2009*.
- [MTR⁺09] A.I. Mourikis, N. Trawny, S.I. Roumeliotis, A.E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.
- [OBI09] G. Orchard, C. Bartolozzi, and G. Indiveri. Applying neuromorphic vision sensors to planetary landing tasks. In *Biomedical Circuits and Systems Conference, BIOCAS 2009*, pages 201–204. IEEE, 2009.
- [PMD04] S. Parkes, I. Martin, and M. Dunstan. Planet surface simulation with pangu. In *Eighth International Conference on Space Operations*, pages 1–10, 2004.
- [Pre87] R. Preiss. Motion parallax and figural properties of depth control flight speed in an insect. *Biological Cybernetics*, 57(1–2):1–9, 1987.
- [SMS10] G. Sibley, L. Matthies, and G. Sukathme. Sliding window filter with application to planetary landing. *Journal of Field Robotics. Special Issue: Visual Mapping and Navigation Outdoors.*, 27(5):587–608, 2010.
- [TCT⁺10] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Unified real-time tracking and recognition with rotation-invariant fast features. In *CVPR*, 2010.
- [TWO96] N. Takeda, M. Watanabe, and K. Onoguchi. Moving obstacle detection using residual error of FOE estimation. In *IROS*, pages 1642–1647, 1996.
- [VRVS10] F. Valette, F. Ruffier, S. Viollet, and T. Seidl. Biomimetic optic flow sensing applied to a lunar landing scenario. In *IEEE International Conference on Robotics and Automation (ICRA 2010)*, pages 2253 – 2260, 2010.