Vision Based Landing for Commercial Aircraft

D. Dickmanns[†⋆], *J. Boada-Bauxell*[⋆⋆], *V. Gibert*[⋆⋆] *and F. Schubert*[⋆]
[⋆]*Airbus Group Innovations*
Airbus Defence and Space GmbH, 81663 München, Germany
dirk.dickmanns@airbus.com, falk.schubert@airbus.com
[⋆⋆]*Stability & Control Research*
Airbus Operations SAS, 31000 Toulouse CEDEX, France
josep.boada-bauxell@airbus.com, victor.gibert@airbus.com
[†]Corresponding author

Abstract

This paper presents an approach to vision based landing (VBL) concepts integrating the following contributions:
a) Pilot interactions are used to leverage on the superior object recognition capabilities of humans. This drastically reduces the search space that has to be covered by the vision system. Aircraft data, the known situational context and background information are integrated as well.
b) A dissimilar design approach comprising a combination of diverse image processing (IP) algorithms improving the robustness for the whole distance range from early approach to touchdown and taxiing under varying environmental conditions.
c) Visual servoing for aircraft control using the results presented here is shown in an accompanying paper.[13] For initial testing, simulations with synthetic images have been implemented.

1. Introduction

The landing of an airliner is one of the most critical phases of flight. Ground based systems (ILS/GBAS) enabling assisted landings are today the main way to ease the piloting task during this phase. Thanks to these systems, the deviations of the aircraft from the ideal approach can be computed and used in the guidance laws. However, these systems are expensive; their availability is limited to airports having deployed the required infrastructure. Furthermore additional procedures in air-traffic control are required. In the frame of the future aircraft, AIRBUS is studying the possibility to perform an automatic landing without information from external means, i.e. unequipped or unknown runways.

In the absence of precise navigation means, the aircraft is assumed to be roughly placed near the landing area with the accuracy of current navigation technology like an Inertial Reference System (IRS), VHF Omnidirectional Range/Distance Measuring Equipment (VOR-DME) or the Global Positioning System (GPS). However, these systems are not precise enough to perform an automatic landing on a runway.[21] To cope with these limitations several solutions are explored. Among them, vision is becoming an interesting approach to cope with relative position information.

1.1 State of the Art

During the last two decades, camera and computer vision technology have advanced to a level of performance that is beginning to allow for practical use in aircraft applications. Since long, they can provide pose estimates with the required accuracy under favorable conditions. Early work on VBL has been performed in the 1990ies[30,38] from which the edge based tracking is reused here.

An overview of current methods for search and detection of runways in both high-altitude vertical view and approach view imagery is given by.[16] They also subsume tracking methods with different sensor types and systems such as ILS, MLS, (D)GPS, onboard lasers / radars and cameras.

The image processing methods cited there, older work[25,35] and current literature[18,33] mostly employ local edge filtering (see figure 11), and edge/line formation using Hough Transformation, RANSAC, dynamic programming, least squares fitting and/or thinning to retrieve the lines/edges enclosing the rectangular hardened surface area and the markings of the runway. It is quite easy to come up with first image processing results using those well known standard approaches. However, they are usually limited to favourable environmental conditions and to situations in which the runway is well visible and covers a considerable part of the image.

Experience in another, already commercialized domain, namely vision based driver assistance systems, has shown that the necessary integrity of vision systems poses fundamental and pertinacious problems. From these past experiences significant challenges in the envisaged vision based landing applications are expected to surface only late during higher, non-research, TRL development phases. They may even raise long term research needs.

The most relevant aspect in the development of a vision system is thus to reach sufficient performance figures for the target applications:

- Figures of high integrity as indicated by the system itself under all relevant operating conditions have to be achieved.
- The accuracy of the numerical values of the results must be sufficient for the control task at hand. For most systems, this is usually the somewhat simpler task. Noise (random imprecision) is rarely a problem given an appropriate imaging subsystem: Most are relative errors, hence the absolute errors are reduced during approach.

1.2 Overview

This paper is organized along three aspects to achieve practically relevant detection and false-alarm rates and sufficiently accurate results of vision based landing systems in more diverse and more difficult environmental and situational conditions:

a) Human vision still by far exceeds the capabilites of technical vision systems. Pilot interactions can be used to massively support the IP by drastically reducing the search space that has to be covered by the vision system. Similarly, situational context information like QNH and runway direction can be exploited together with data from conventional aircraft sensors or the flight control system. The use of all this informantion is presented in section 2.

b) A single image processing algorithm is not expected to be sufficient due to the variety of functional requirements posed by the use of pilot interactions and due to the scale variation of the runway that has to be handled from early approach to touchdown and roll out. A combination of diverse image processing (IP) algorithms covering these aspects is detailed in section 3. They are also suitable to provide the information required for pose estimation with enhanced robustness against varying environmental conditions.

c) For development and test of VBL systems, image data together with the desired processing results (ground truth data) are required. Proving that they are sufficiently dependable and accurate for a certain practical use requires test data sets in a volume allowing for statistical performance characterization. To alleviate from the need of numerous test flights, simulations for image rendering have been implemented as part of an assessment methodology. They may become a part of later validation and verification processes. Examples for this synthetic image generation are given in section 4.

A conclusion and some perspectives close this text.

1.3 System Integration

Estimation of the deviations between a camera and a 3D point with a single embedded camera has been proposed in several range identification methods as in[7,24] or even early works from.[12] The principle is that the estimator, initialized with a rough value, converges to one of the deviations of the camera w.r.t. a targeted point. These methods require that there is an image processing algorithm providing enough sufficiently reliable features to feed the estimation algorithms. Generally, range identification problems consider that one point in the image is identified and tracked. Some methods perform the range identification using the information of a line in the image, e.g.[12]

During landing, the estimated deviations can be used as output feedback in order to guide the aircraft[6,14] or as input into a sensor fusion scheme.[17] Hence, image processing is a critical point for detection, tracking and extraction of features.

Computer vision is able to extract sufficiently reliable visual features that can feed visual servoing algorithms for guiding the aircraft. Indeed the results of the IP can feed a pose estimation algorithm computing deviations w.r.t. the approach path to the runway in a representative closed loop simulation. This is presented in an accompanying paper.[13]

2. Additional Information and Data

As a first step of integration, the calculation of the region of interest (ROI) for runway detection was designed and implemented based on an assumed aircraft state with its associated, possibly large, navigation (state estimation) errors as provided by the aircraft flight control system and the navigation system.

Then, an operator interaction giving a rough runway start point and an optional runway direction in the image was integrated. Outliers and spurious detections of detection modules can be grossly reduced by limiting their search space around the pilot interaction positions; the detection modules then merely have to confirm and refine the designated positions. Integrating this valuable input and feedback from the pilot into the vision based landing system also helps keep the pilot in the loop without increasing his overall workload.

2.1 Using Aircraft State

The runway is projected for all $2^6 = 64$ signed combinations of the 6 DOF state estimation errors added to the nominal estimated pose, green in Figure 1. The area in which the runway should appear with x sigma confidence is spanned by the resulting runway images. x has to be selected to find the runway with sufficient probability, usually in the range of 3.0–6.0. The minimal spanning rectangle on the ground is calculated by inverse perspective transform to the assumed ground plane. This requires the height of the aircraft above the ground and the pitch and roll angle to be known (along with the accompanying error (co-)variances). The resulting ground rectangle, which is aligned to the nominal runway, is projected into the image delivering a quadrangular region of interest, blue in Figure 1. The runway position projected into the image using the nominal aircraft pose is white in the middle of the blue ROI.

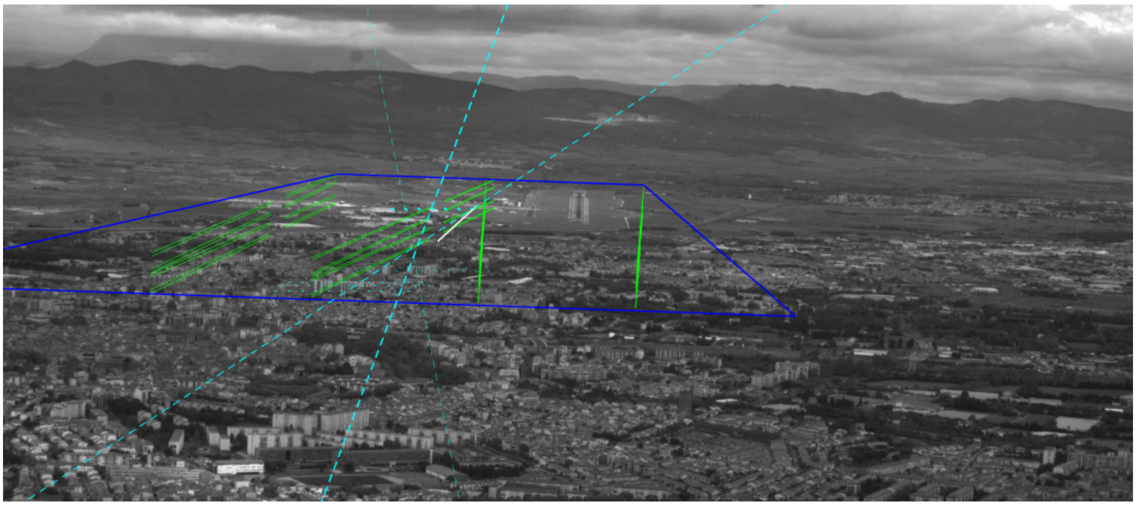

Figure 1: Big ROI on initialization

Calculated region of interest (ROI) using the aircraft pose and large estimation errors, e.g. due to extended GPS loss. Light blue (cyan) are the (unusable) results of a local symmetry based runway detector.

2.2 Operator Interaction

To achieve reasonable detection results in a single ROI with the detection algorithms presented later, the ROI must not be more than 5 to 10 times larger than the runway area. To achieve this, an operator interaction (currently an image position given by a 2D input device and a directional input given by a 1D input slider) can be used to

- place the ROI close to the runway center,
- give a directional hint, which is necessary for greater lateral offset and roll angle and, most importantly,
- reduce the ROI size to values corresponding to the operator input errors. (Some input location error has to be taken into account, this is what will be compensated for later by the detection algorithms.)

Assuming a reasonable error in the operator input, this results in a smaller ROI almost centered to the runway, in which the e.g. a local symmetry based runway detector can deliver dependable results, see Figure 2.

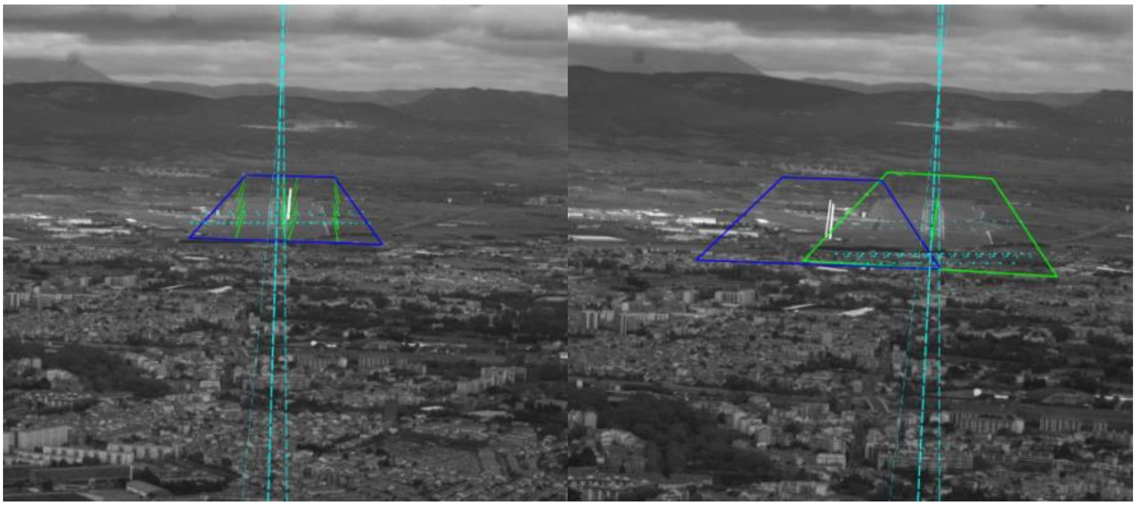

Figure 2: Small ROI and propagation
Reduced ROI centered on runway by operator input (left) and propagated by global registration (right, green). There, the runway has already left the original window (blue).

During approach, the ROI can be predicted in two ways. Firstly, the aircraft motion data can be integrated, which results in a window drifting on the ground (relative to the image content) mainly due to the initial errors in the assumed height and angles, e.g. due to a QNH being not actual, and, to a lesser extent, due to sensor drift. Aircraft inertial data is particularly reliable with good accuracy and very high integrity; the air data system delivers airspeed and barometric altitude. This approach has been implemented in simulation. Secondly, the image window can be propagated by patch or point trackers on the ROI corners or by applying the estimated homography from the global registration.

2.3 Situational Context

If the runway direction is known from the flight plan and/or an airport database, the expected runway vanishing point and hence the direction of possible runway edge elements can easily be derived by a simple forward projection calculation. Then, a directional input fo the runway direction is not necessary. A valid QNH from the airport reduces the altitude error leading not only to a smaller ROI, but also to less drift of predicted image features based on aircraft state — altutude error is the main reason for their drift.

3. Image Processing Algorithms

Detection and tracking of the runway can be considered as separate tasks. Both both must accommodate the possible variation in runway appearance. The initial detection, i.e. the indentification and first localization of the runway is generally the much harder problem. Tracking of a runway in successive images is a considerably simpler problem, because the appearance in previous images is known and can be used to assist in the search. If processing occurs at the desired rate of 25 Hz, the appearance of the runway does not change dramatically from one image to the next.

The previous section showed how a vision based landing system can exploit the superior human recognition capabilities to support vision algorithms. The pilot interaction shall happen as early during approach as possible due to pilot workload on final approach and touchdown. However, during early approach the detection algorithms (section 3.5) may not yet be reliable enough.

In this early phase, although VBL systems might have difficulties in dependably *detecting* the runway, *tracking* algorithms that do not know what a runway is may well be capable of determining the motion of the aircraft and the scaling change of the image. So the designations made by the pilot can be kept fixed to the image content until the runway, still assumed to be close to the tracked designations, becomes large and distinct enough to be detected technically. This is the reason for combining tracking and detection algorithms with pilot interactions.

When aligning the aircraft to the runway, it might happen that during these maneouvers the runway, or the designation done by the pilot, might run out of sight. This is especially relevant when a simple system topology with a single fixed-mounted camera is used as is assumed here to limit overall system complexity. In these situations, global registration or simultaneous localization and mapping (SLAM) methods can still try to keep the ground position of the pilot designations for later reuse when they reappear in the image after the alignment maneouvers.

The final detection and measurement of the runway happens via model matching, be it implicit as in neural nets or state vector machines or by explicit matching of image features to geometrical models e.g. through probabilistic methods. Initially, simple and generic model assumptions are employed, namely the symmetry of the runway and the threshold as a bright block or a block with a specific spatial frequency of brightness change. Later in the approach (and the development process), more detailed model assumptions about the runway shall be used. They have to offer sufficient genericity to cope with runway variants.

Tracking and registration can support detection and measurement by delivering comparably stable, but long term drifting image positions. Aircraft data can be used to predict image positions of features. Both complement the typical error characteristics of detection algorithms, which tend to deliver long term accurate results except some usually rare, but often more extreme outliers. Data Fusion of the tracking positions, the predicted positions and the detections shall be exploited to improve the overall integrity of the results.

Given the wide scale change from early approach to touchdown and landing, diverse and dissimilar detection and tracking methods might have to be adaptively employed, also depending on the type of the image signatures available. Although detection algorithms might be of lesser importance during final approach and touchdown, their use may not be completely dropped because trackers live from a result feedback loop, which gives raise to certification issues.

For the above reasons, a single image processing algorithm is not expected to be sufficient for the tasks and the scale variation of the runway. Hence, a set of complementary algorithms is investigated as sketched in figure 3 and described in the following subsections. With an overlap of application ranges a dissimilar design approach to further increase integrity is feasible.

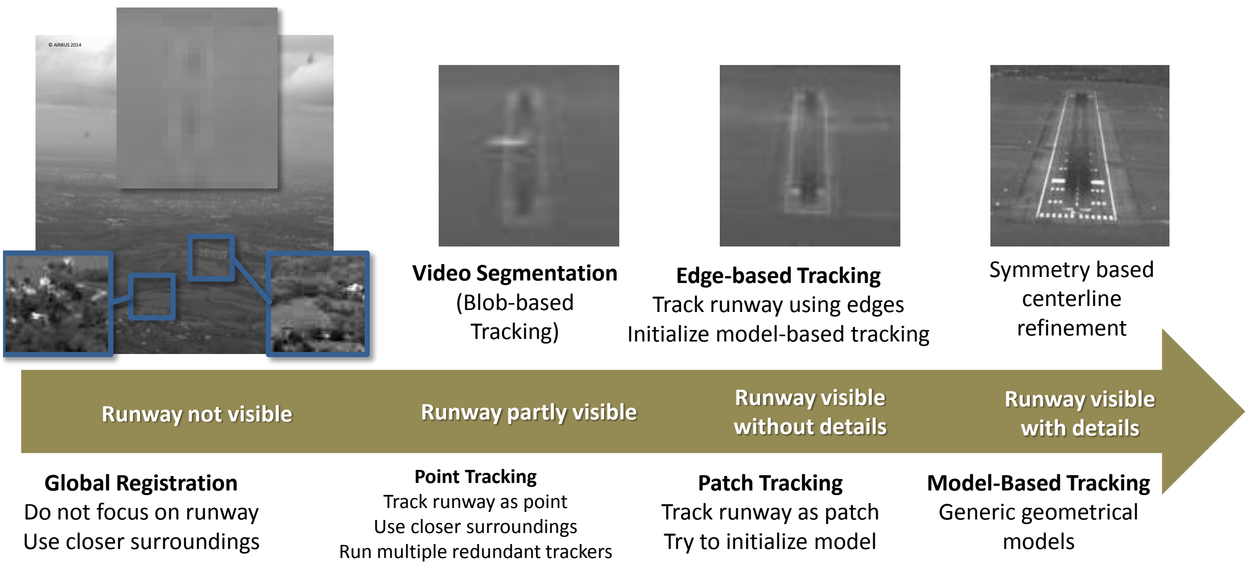

Figure 3: Algorithm selection depending on distance / resolution of the runway

The desired final output to the flight control system must at least be the runway centerline position and slope in the image and the start point of the runway (or any other known point) on this centerline. Additional results like the positions and angles of runway boundaries or markings are helpful.

3.1 Global Registration

Image registration is a mapping process which links image coordinates between two video frames that correspond to the same static scene point. This is achieved by directly or indirectly computing the camera pose which maps all image coordinates onto static scene points. With this mapping it is possible to track a designated static point like a runway within a video. The advantage over commonly used visual tracking algorithms[28] is that this computed mapping is valid for the whole image and not just for the patch to be tracked. Hence if the patch to be tracked, e.g. the runway, moves out of the field-of-view, e.g. due to a flight path correction, this point can still be tracked from frame to frame as it still corresponds to the same static scene point. In that case other static scene points which still remain in the field-of-view are used to compute the registration mapping. Such a tool is also valuable for the vision based landing especially in early approach and little usable signature on the small and possibly faint runway.

The direct way to estimate the pose of the camera relative to the runway is by using full state 6-DOF motion observers like SLAM. These do not require any assumption about the scene and can be performed with a single camera.[8] These algorithms track many discrete points from frame to frame and solve a non-linear system for estimating the extrinsic camera matrix. The individual point tracks do not need to be perfect as outlier detection algorithms and priors for the non-linear solving can be applied to augment information about the camera pose retrieved from redundant (i.e. using stereo cameras) or complementary (e.g. inertial, see above) measurements. The extrinsic camera matrix allows, together with the intrinsic camera matrix estimated by a calibration process, to project each image pixel into the scene and vice versa. Hence all image coordinates, corresponding to static scene points, can be propagated from frame to frame.

An alternative, indirect approach to measure the camera pose exploits the fact that the scenery around a runway can be approximated by a plane. All static scene points which lay on or a close to this plane can be used to compute this mapping between the image coordinates. These approaches are called homography based registration. As opposed to the SLAM approaches mentioned above, the mapping is not achieved by computing the extrinsic camera matrices but by computing a homography. The process is illustrated in figure 4. The mapping is a 3×3 matrix (H in fig. 4) which projects image coordinates ($\vec{p}_i$ and $\vec{q}_i$ in fig. 4) directly from one frame to another[36] without projecting them first into the 3D scene like done in SLAM. Due to the much lower complexity we chose this latter approach for computing the global registration. In the following we give a short overview of the most common methods for homography-based-registration.

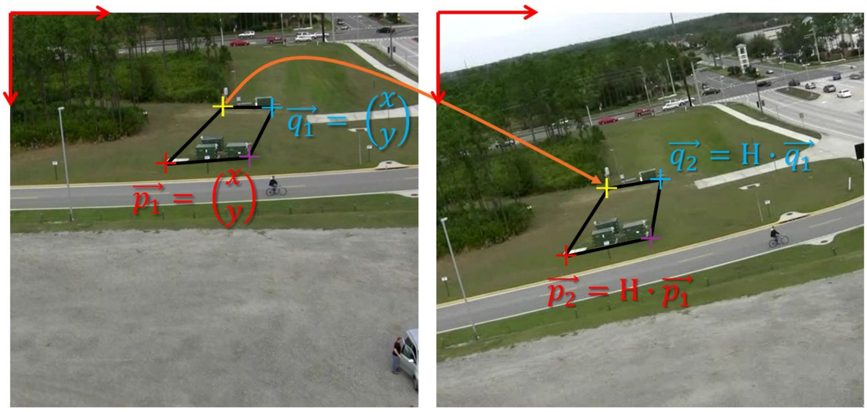

Figure 4: Principle of homography based registration

3.1.1 Overview of Homography Based Registration

The problem of homography based registration has been studied for many decades and many methods are available. However, because of the large number of different approaches very often it is not clear which method should be used. Furthermore, homography based registration is needed in a wide range of applications each imposing different requirements. To cope with these different challenges a number of methods have been proposed in the literature. Since the large field of methods is difficult to assess, we provide a short summary in the following paragraph. In figure 5 the relation of the discussed methods is illustrated.

The first distinction between registration algorithms can be made based on how pixel variations are introduced in two different images. In many cases the movement of the camera or zoom change results in misalignment between two images. This can be described in mathematical terms and it applies to all pixels. Any registration algorithm that builds on such an assumption, where a camera movement globally affects all pixels, is called a global registration method. Parts between two captured images may also change, even if the camera was not moved at all. This might be due to dynamic scene motion, e.g. a walking person. Such a complex movement cannot be described mathematically for all pixels in a global form. Hence, each pixel has its own motion model from one image to another. Methods that estimate the motion of each pixel individually are called local registration methods. In fig. 5 the different categories of local and global registration methods are illustrated. These coarse categories can be further subdivided based on how information about the camera motion is extracted from the images. Either all pixels are used, i.e. dense, or only a sparse subset of pixels is used. For global registration, the first type of methods is called global dense registration. These work either directly on intensity values (i.e. intensity-based methods often called direct methods) or they transform the images into the frequency domain. Frequency-based methods are very robust to noise artefacts, but are limited in the types of global image transformation that can be computed. The second type of methods is called global sparse registration methods,

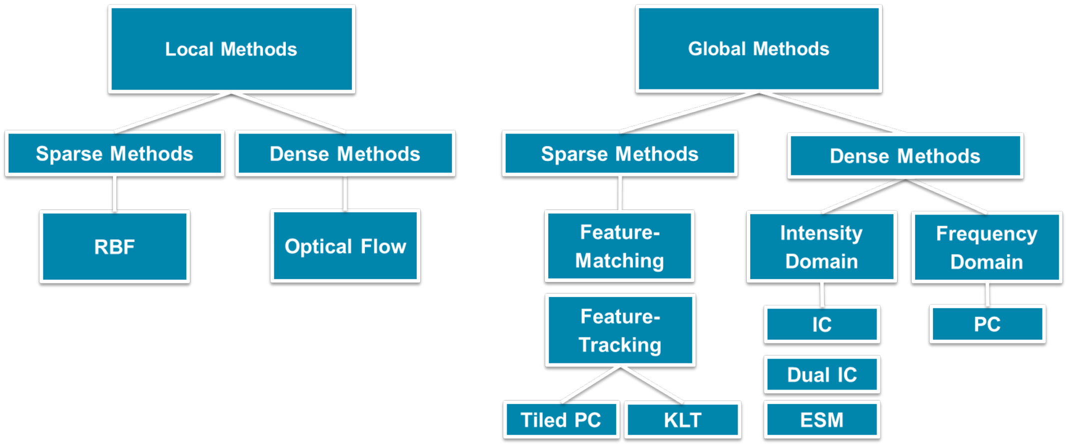

Figure 5: Overview of different registration algorithms.

where sparse locations are either defined by feature detectors or extracted from a defined grid. Although generally in real applications the assumption of a global transform for registering images is too restrictive, many applications rely on such a simplification to ensure accurate registration. This is because the global transform is computed with the support of many measurements, densely or sparsely. Local registration does not have the global support leading to many outliers. In our application of image-based landing this support is an important feature as it allows to track the runway even if it moves out of view. Therefore, we use a global homography-based registration.

Current methods based on local features are by far the most popular ones for image registration[37] and all of them share a similar processing chain. First, sparse local features are detected in the two images to be registered. Second, corresponding features are matched using descriptors computed at the feature locations. Third, a transformation (i.e. 8-DOF homography) is computed. The most time consuming parts are the detection of feature locations and the computation of the descriptors. Although very efficient algorithms exist[3,29] which already allow real-time processing for small image sizes, some applications require even faster algorithms. In the presented vision based landing application, the image registration is only a small component of an algorithmic chain. Hence the computational time available should be as small as possible. Therefore image registration methods need to be much faster than the current state of the art to allow these applications to run at real time as well. In the following will discuss such a registration scheme which has been originally published in.[32]

3.1.2 Tiled Registration Using Point Trackers

We divide the image into subparts, so called tiles, and track the centers of the tiles by a point tracking algorithm. The translational offsets for each tile and the tile centers can be used to generate a matching point pair (the translated centers of the tiles). From these robustly generated point matches a homography can then be estimated. Different algorithms such as the discrete linear transform (DLT)[20] and a non-linear estimation with robust outlier detection can be applied. A schematic illustration of the tiled registration scheme is depicted in figure 6.

The point tracking of the tile centers can be achieved by any popular visual tracker.[28] Patch tracking algorithms have gained much attention in the recent past due to advances in learning concepts (i.e. ferns, boosting, randomized forests, SVMs) applied to tracking schemes called "tracking-by-detection". A good overview of them can be found in.[23] These tracking algorithms typically estimate the position, scale and rotation of the patch and try to solve the general tracking problem which involves many challenges such as pose changes, object deformations, background variations, partial and full occlusions, etc.. However, in many applications like the vision based landing only a few of those challenges are relevant. In our application the biggest challenges lay in the pose change, more specifically the change in scale (i.e. during approach the runway becomes bigger in the image). Occlusions by clouds and other atmospheric effects also need to be handled. Other challenges are not very relevant, for instance the change in background can be neglected as the surroundings of the runway will mostly stay the same except for a few ground objects such as taxiing aircrafts (see figure 7).

An alternative to patch tracking described above is point tracking. Algorithms from this category are much faster to compute. However, they are not as robust to all challenges of the general tracking problem. Furthermore the precision is not achieved by tracking only a single patch perfectly, but rather tracking many points within a patch or area. A global model needs to be applied to know which of the many trackers can be trusted and which not. A simple, generic model is the general motion consensus (i.e. average offset). More advanced models depend on the specific

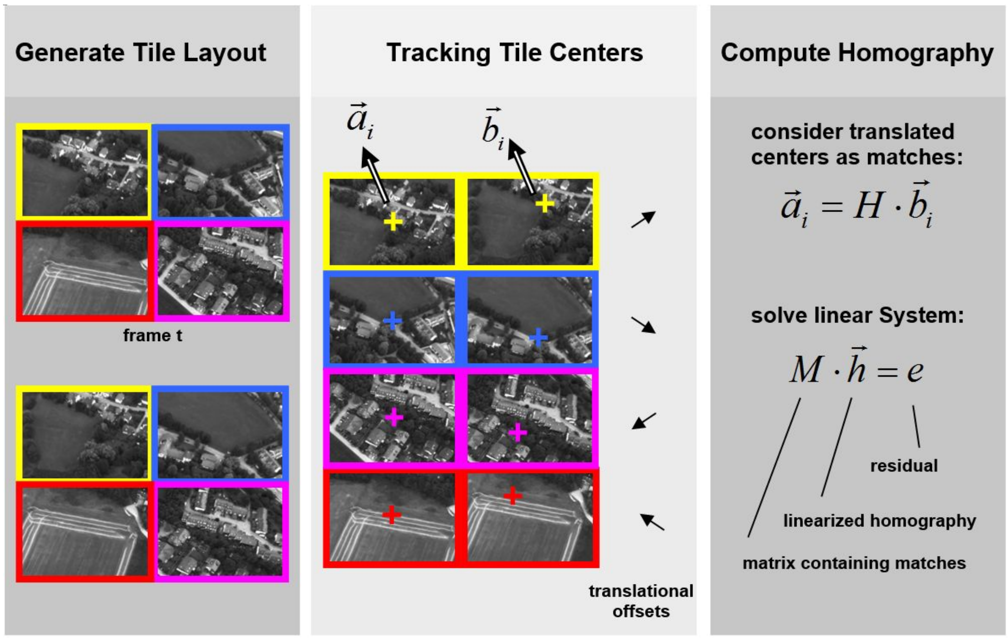

Figure 6: Tiled registration scheme using tiles and point tracking.

application. In the vision based landing the tracking will be applied to points which lay on the ground plane of the world. A homography (i.e. 3x3 matrix) describes the motion of all points that lay on this plane. This concept was discussed in the previous section on registration. The outlier detection step in this computation will identify which of the point trackers are accurate and which not. A single tracking failure will hence not influence the overall performance, which makes this approach more robust in the view of redundancy than the patch tracking algorithms. Therefore, in our application we will use two types of point trackers which are discussed in the following.

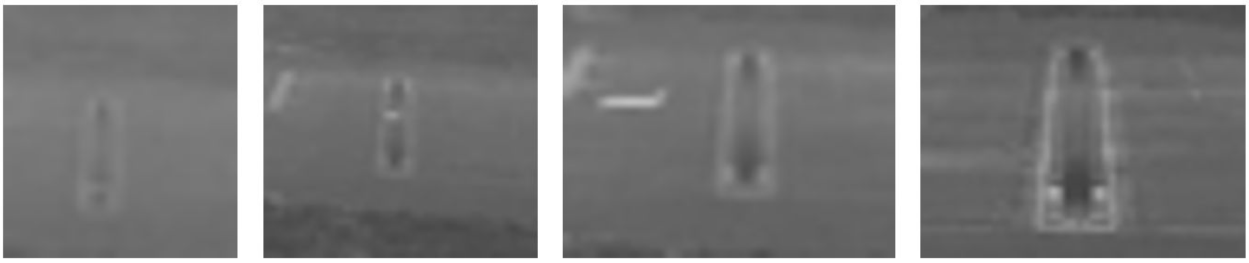

Figure 7: Tracked image patch of runway. The visual background of the runway does not change significantly.

Various algorithms for point tracking exist. A commonly used one is the KLT-tracker.[34] In[32] phase correlation[2] was used to track the tiles as the involved FFTs can be highly parallelized using GPU implementations.

The greatest advantages of phase correlation are its simplicity and robustness. It can handle narrow and larger baselines (given the images overlap) at the same computational cost, it is robust to noise, has the ability to register less textured images (where usually not enough features can be detected) and requires considerably less parameter tuning than feature based and intensity based methods.

In our system we use a slight modification of this phase correlation approach discussed above. Instead of using raw image intensities we employ gradient based features to characterize each tile to be tracked. The size of the tile plus some border defines the search area of maximal translational offsets that can be computed for a single tile. The detection is computed using a correlation similar to the phase correlation as discussed above. The main differences are the online learning capability and the efficient prediction of appearance changes induced by translational offsets. The online learning step updates the correlation reference template during tracking. Hence appearance changes caused by non-translational offsets are slowly incorporated into the reference.

In figure 8 a result of this tiled registration approach is shown. The point trackers are automatically selected to cover most of the image area. The advantage of the global homography is clearly visible. Even points which are not visible yet (e.g. the red points on the right edge of frame $t + k$) can be tracked into the frame as always enough point trackers (i.e. at least 4) are placed in the current field-of-view to compute the homography.

Figure 8: Result of the tiled registration using point trackers. The point trackers are highlighted in blue. The yellow line is the artifical horizon, which is updated by the homography as well as any point in the image like the dark red ones.

3.2 Visual Patch Tracking

Visual patch tracking is a very active research field in which many different, competing methods have been proposed.[27] As many of them focus on similar applications, i.e. generic tracking of manually selected patches like a runway, it is important to understand the properties of the type of tracking category each method falls into. Only then it is possible to select the right algorithm from the large pool of different approaches for a specific task such as VBL. In that sense, tracking benchmarks have become increasingly popular to shed some quantitative light in the pool of methods. However, often the best performing methods rank almost equally well or their ranking differs depending on the benchmark. The best performing methods on the well-known "Online Object Tracking Benchmark"[39] were learning-based tracking by detection methods such as STRUCK[19] and TLD.[22] However, in the same year the "The Visual Object Tracking Challenge"[26] ranked their performance in middle of all methods compared, which was confirmed by the experiments run in the follow-up challenge 2014.[27] As the selection of algorithms that were compared and the pool of video sequences used for evaluation had a significant overlap, the conclusions that can be drawn from these results on the general tracking performance are not obvious for an end user. After all, these benchmarks only demonstrate that an image processing algorithm solves a dataset, but not necessarily a target application. One possible explanation is that the differences in the performance of the best tracking-by-detection algorithms are quite small. The observed variations in ranking are most likely due to subtle differences in the dataset which favor the one or the other algorithm. In figure 9 the result of an online learning SVM with structured output and a part-based model is shown. The patch-tracker accurately follows the position of the runway which was manually selected by the user.

Figure 9: Tracking result of the "part-tracker"[40] outputting only position.

3.3 Video Segmentation

One class of methods which has been underrepresented in the state of the art benchmarks discussed in the previous section are segmentation based tracking approaches. In our view this is due to four reasons. First, many top performing methods do run not in real time.[5,15] Second, the overlap computation using bounding boxes penalizes segmentation based methods as these extract tight object contours, which are usually smaller than the coarse, surrounding ground truth rectangle. Third, the appearance model is often based on color, which is considered a limitation as opposed to gradient based descriptors, because color models tend to perform poorly or simply do not work on grayscale, low-saturation and low-contrast sequences. Fourth, segmentation automatically locks onto object boundaries, which prohibits tracking of patches that are not bound by a clear contour, e.g. the upper part of a human face. In[31] it was demonstrated that these four reasons mentioned above do not always hold true.

On the contrary to the prejudices to segmentation based tracking, there are quite a few advantages. Highly deformable objects are often impossible to describe with a bounding box. Hence, background information is also included in the foreground area, which misleads learning based approaches during the online adaption step and can be a major source of drift. A segmentation step, which extracts the contour, could be very valuable to guide the sampling of positive and negative examples for a co-running tracking by detection method (e.g. the part tracker discussed in section 3.2). Additionally, the contour itself is also a valuable descriptor, which can be used for downstream applications such as pose estimation. For very small objects or those with homogeneous appearance color might be the only valuable descriptor as opposed to the very popular gradient-based ones.

Recently a segmentation based tracking algorithm was presented[4] which achieves impressive performance in terms of accuracy and run-time. The core power of this approach is the intelligent coupling of rigid registration and fast segmentation via level sets. This combination allows very fast computation and yet accurate contour estimation. Both together allow the color based modeling to achieve best performance even in very low contrast scenarios, which are considered intractable for segmentation based tracking. In[31] a quantitative evaluation using the latest datasets in the tracking community was presented. Video-based segmentation can achieve tracking speeds of up to 200Hz for typical object sizes using a single core of a standard 2.3 GHz CPU. In figure 10 the result of this tracking algorithm running on a synthetically rendered runway is depicted.

Figure 10: Tracking result of the "video segmentation" implemented in.[31]

3.4 Edge Based Tracking

Human-made objects often consist of or contain straight or only slightly curved edges, be it shape edges as e.g. for rectangular blocks or surface properties as e.g. for runways with markings. These object features tend to lead to well visible gray value or color steps which in the image extend orthogonally to their gradient direction thus forming image edges. Edges have been extracted since the beginnings of image processing using simple correlation masks as shown in figure 11 on image areas or whole images.

The correlation masks can be closely adapted to edges of certain direction and spatial extent (length, sharpness, smear) and then be applied on only a single line (search path, see fig. 12 left) delivering noise-reduced gradient profiles averaged along the edge, see figure 13 left. They can be viewed as a coarsely discretized version of scaled Gabor Wavelets or a precomputed fixed set of steerable filters (see fig. 12 right) and are often called matched (edge) filters or directional edge measurements. These features have already been employed in.[11,38]

Such directional edge extractors can deliver highly selective, but very sensitive measurements of edge positions and directions. Highly accurate measurements via subpixel and subdirection interpolation have been achieved: way below 0.1 pixel of positional standard deviation (down to 0.01) as well as good angular accuracy below 0.5° (down to

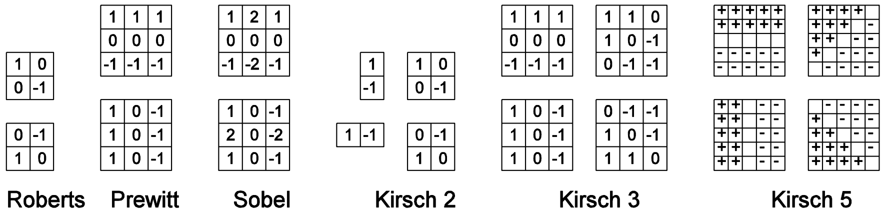

Figure 11: "Classic" edge correlation masks

0.1°) have been proved for edges of reasonable contrast and mask lengths of roughly 10 to 20 pixel as shown in figure 12.

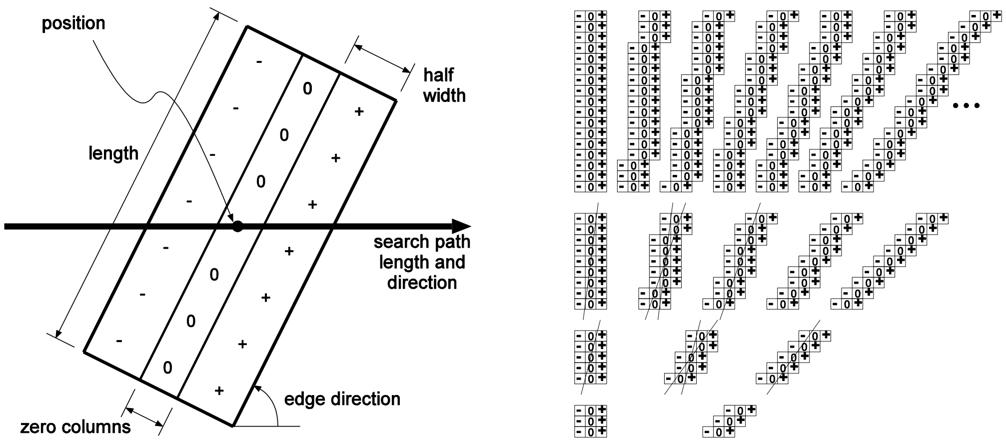

Figure 12: Directional edge measurements
Parameterization of edge masks for adaptation to expected edge directions (matched filters) and example of a mask set implementation for one octant

Some self checking (RAIM) capabilities are provided by e.g. the correlation of positional and angular measurements which gives a hint to non-symmetric gradient moment or to curved edges. All this is achieved to a surprisingly low computational burden when compared to the feature extraction schemes composed of the simpler classic masks plus line formation algorithms widely deployed for runways, and also compared to the fastest point trackers — at the cost of delivering one-dimensional measurements only. Two-dimensional trackers can be composed of two othogonal edge measurements and are useful if the edge assumptions are roughly met by the target areas, as is often the case for runways or parts thereof.

Tracking of edge elements can be based on different levels of intermediate results of the edge measurements. Correlation of column sums or mask responses (the directional brightness profile or gradient profile along a search path) using sum of squared or absolute differences can be used. Alternatively, matching of edge extrema using positional histograms (binning) or any other robust matching scheme can be used. The latter approach reduces the drift problem common to all sorts of purely correlation based approaches by explicitly taking into account information from distinct image features which are assumed to be almost drift free as long as no moving objects are involved.

3.5 Detection Algorithms

The first basic detection algorithms implemented for VBL exploited local symmetry based on "classic" (fig. 11) gradient element extraction plus random sample consensus (RANSAC) for runway centerline refinement (detection) and spatial frequency for threshold detection using frequency domain methods. Real-time capability has been achieved by a GPU-based CUDA-implementation.

Currently the edge based tracking methods provide symmetry based runway detection capabilities through symmetry correlation, that is, correlation of reverse (mirrored) gradient responses to inverse gradient responses, but shall be extended to matching of mirrored gradient extrema to their inverse similar to the robust methods for tracking. This requires directional adaptation of the mask over the search path length for nonparallel edges due to perspective distortion, and reflecting the direction change in the matching.

Figure 13: Edge and model based tracking
Gradient profiles across a runway suitable for tracking through correlation (left) and simple model based detection of threshold and aiming point (right) at the green markers. Spurious detections at the bottom of the search area, detection of the distant aiming point at the top.

For centerline detection results, see figure 2.

Threshold detection as detection of a bright line almost parallel to the horizon is investigated during early approach, see fig. 12 right, while the analogon to the frequency domain methods to detect the threshold when resolvable into stripes using directional edge measurements is under development.

4. Test Data by Image Rendering

A vision based landing system has to undergo a thorough V&V process. The operational boundary conditions have to be analyzed and requirements for such a landing-system derived. Against these, the performance of such systems (detection rates / confusion matrix, accuracy, worst case execution time etc.) based on relevant in-flight videos with accompanying sensor data sets (at least rotational speeds, accelarations, air data system and compass/orientation from the FCS) has to be assessed. Assessment methods have been established in the automotive domain since about a decade[9] .

The main problem with this approach is the availability of reference data, that is, the expected output of the VBL system. While this is not easily solved for real data, image rendering modules can easily provide the ground truth and all necessary coordinate transformations between 3D model and 2D image coordinates from their internal scene and camera model.

For development, test and validation of image based landing systems, synthetic images should be available in two main modes. Initially, images and sensor data should be generated using models and knowledge similar to those internally used within the vision based landing system to see possible consequences of the underlying modelling assumptions: These are “ideal” measurements for initial testing of basic algorithm functionality. Then, data close to reality are necessary for robustness testing under difficult conditions. This should cover especially the currently problematic situations and shall deliver realistically disturbed images and sensor data.

A simple rendering based on a 2D graphics library generates images from the models as they are (or shall be) used for runway detection and tracking. It provides “ideal” images as shown in figure 14 left to test the basic functioning of algorithms under optimal conditions.

A second approach was based on the Ogg Renderer *ogre*. It provided some more realism (fig. 14 right), but was not real time capable. It would not be easy to improve performance beyond the 6-8 Hz achieved. The airport area size is one of the performance limiting factors and cannot be easily enlarged. Integrating a terrain handler required big effort and did not seem feasible.

So for the more challenging, realistic test data, the rendering part from the public domain Flight Gear simulator was separated into a standalone module driven by the aircraft/camera pose parameter plus the lens, imager and camera and some environmental parameters. Flight Gear as test data generator provides relevant benefits, such as reasonable image quality with realistic visual effects like complex scenes, wide variability of lighting conditions and elaborate weather conditions, e.g. flight through parameterized fog and clouds. It is capable of delivering 30 to 60 Hz image refresh rate and provides a scene manager so that worldwide use is possible, even in greater altitude.

There are also some issues and necessary extensions: The runway currently is a single fixed category C texture.

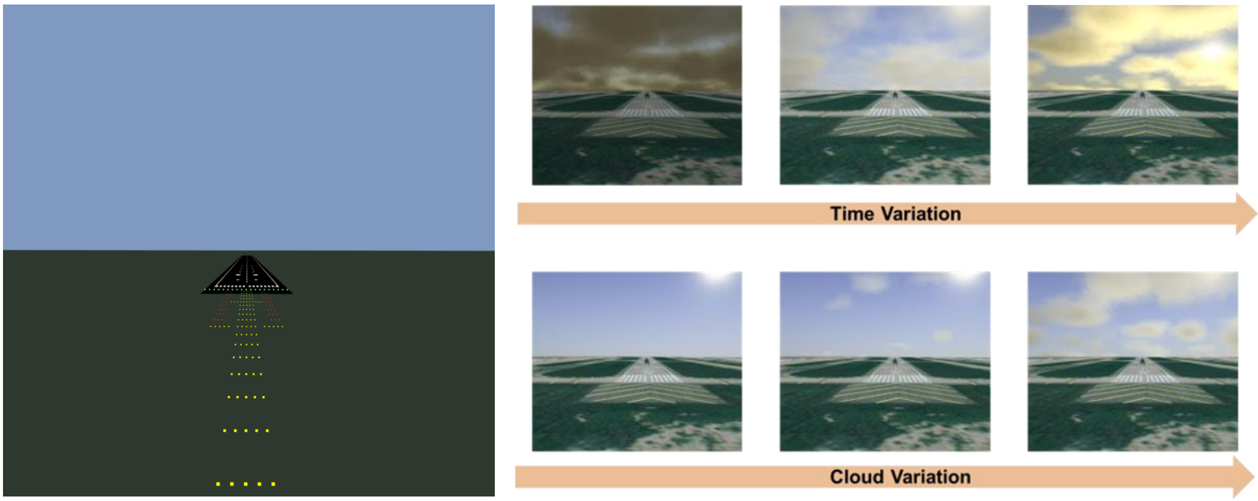

Figure 14: Simple 2D (left) and Ogre-based rendering (right)
simple rendering for initial testing of basic algorithm functionality under "ideal" conditions, Ogre-based for some more realism

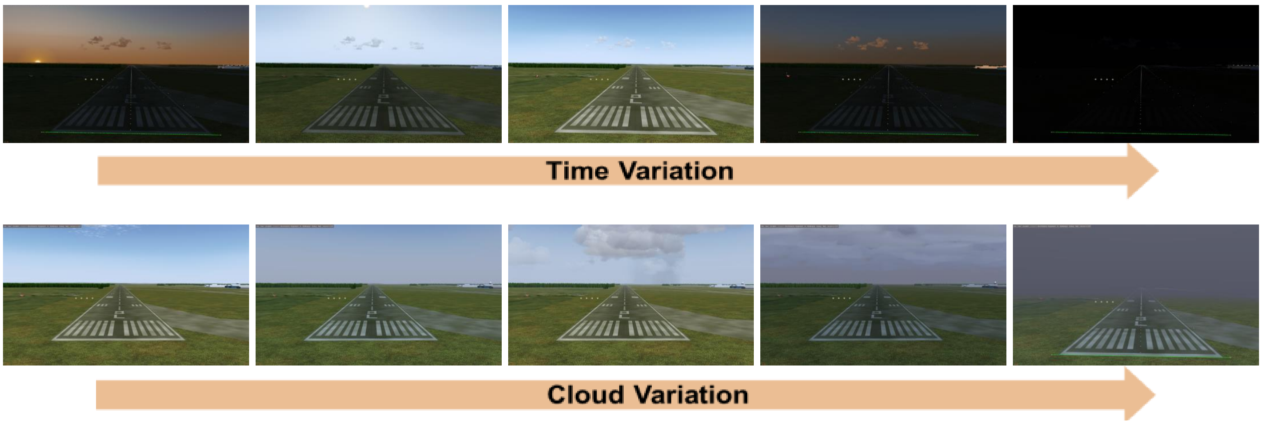

Figure 15: Some of the parameters provided by Flight Gear rendering

A variation of runways and some further important visual effects have to be implemented. For example, a water film and puddles on the runway can add lots of spurious, but strong features. Even under dry conditions, a shaded runway rendering with extended reflectance models like specular and retro reflective components should cover contrast inversion, that is, tire marks becoming bright and runway markings becoming dark when close to the sun reflection point. Also, the current way of integration does not deliver correctly rendered snow and rain as stripes emerging from the movement based vanishing point, but "falling" precipitation only. A free running version with internal aircraft dynamics correctly used will deliver realistic rain and snow.

5. Conclusion and Perspective

Simulation facilities for image data in a wide range of environmental conditions along with a set of relevant image processing algorithms have been integrated into a development and test environment. The image processing results achieved so far have proven useful for aircraft control in simulations[13] during approach. Aircraft data and situational context as provided by the simulation has been used to support the image processing.

Future work shall extend the operational range to the close final approach with touchdown and rolling. The imaging conditions are then quite similar to roadway recognition tasks, hence principles known from the automotive industry shall be leveraged on. The geometrical model of the runway shall be refined in several levels of detail for applicability in the wide scale range and aspect conditions encountered. Additionally, the registration shall be extended to more elaborate ego-motion observation with feature point depth estimation (simultaneous localization and mapping, SLAM) to enhance the precision of the global registration.

References

[1] American Institute of Aeronautics and Astronautics. *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2006.

[2] V. Argyriou. *Advanced Motion Estimation Algorithms in the Frequency Domain for Digital Video Applications*. PhD thesis, University of Surrey, 2006.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *CVIU*, 2008.

[4] Charles Bibby and Ian Reid. Robust real-time visual tracking using pixel-wise posteriors. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *ECCV (2)*, volume 5303 of *Lecture Notes in Computer Science*, pages 831–844. Springer, 2008.

[5] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009.

[6] Laurent Coutard, F. Chaumette, and Jean-Michel Pflimlin. Automatic landing on aircraft carrier by visual servoing. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2843–2848, 2011.

[7] O. Dahl, F. Nyberg, J. Holst, and A. Heyden. Linear design of a nonlinear observer for perspective systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 429–435, Barcelona, Spain, 2005.

[8] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, June 2007.

[9] Dirk Dickmanns, Reiner Wertheimer, Harald Gräf, and Martin Anderschitz. Assessment of lane recognition systems. In[10], 2006. http://www.fisita2006.com/programme/programme/F2006V252.

[10] FISITA. *Proceedings of the FISITA World Automotive Congress 2006 (on CD, http://www.fisita.com/publications/bookstore?id=96)*, Yokohama, Japan, June 2006.

[11] Simon Fürst, Stefan Werner, Dirk Dickmanns, and Ernst Dieter Dickmanns. Landmark navigation and autonomous landing approach with obstacle detection for aircraft. In *System Design Considerations for Unmanned Tactical Aircraft (UTA)*, AGARD-CP-594, pages 20–1 – 20–11, AGARD, 7 Rue Ancelle, 92200 Neuilly-Sur-Seine, France, 1998.

[12] V. Gibert, L. Burlion, A. Chriette, J. Boada-Bauxell, and F. Plestan. Nonlinear observers in vision system : Application to civil aircraft landing. In *European Control Conference (ECC)*. IEEE, 2015.

[13] V. Gibert, L. Burlion, A. Chriette, J. Boada-Bauxell, and F. Plestan. Vision based automatic landing of a civil aircraft by using nonlinear pose estimation). In *6th EUROPEAN CONFERENCE FOR AEROSPACE SCIENCES*, 2015.

[14] V. Gibert and G. Puyou. Landing of a transport aircraft using image based visual servoing. In *9th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, pages 74–79, Toulouse, France, 2013.

[15] M. Godec, P.M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011.

[16] Xiaojin Gong, Lynn Abbott, and Gary Fleming. A survey of techniques for detection and tracking of airport runways. In[1], 2006.

[17] P. Goupil, R. Dayre, and P. Brot. From theory to flight tests: Airbus flight control system trl5 achievements. In *19th IFAC World Congress*, pages 10562–10567, Cape Town, South Africa, August 2014.

[18] N. Gulec and N. Sen Koktas. Enhancement of vision systems based on runway detection by image processing techniques. In *Proc. SPIE 8360, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications IX, 83600T*, 2012.

[19] S. Hare, A. Saffari, and P.H.S Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[21] FAA/William J. Hughes. Wide-area augmentation system performance analysis report. reporting period: July 1 to september 30, 2012. Technical Report 42, NSTB/WAAS T&E Team, October 2012.

[22] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[23] Zdenek Kalal. *Tracking Learning Detection*. PhD thesis, University of Surrey, 2011.

[24] D. Karagiannis, D. Carnevale, and A. Astolfi. Invariant manifold based reduced-order observer design for nonlinear systems. *Automatic Control, IEEE Transactions on*, 53(11):2602–2614, 2008.

[25] Rangachar Kasturi, Sadashiva Devadiga, and Yuan-Liang Tang. A model-based approach for detection of runways and other objects in image sequences acquired using an on-board camera. Technical Report NASA-CR-196424, NAS 1.26:196424, Pennsylvania State Univ.; Dept. of Computer Science and Engineering, University Park, PA, 1994.

[26] M. et al. Kristan. The visual object tracking vot2013 challenge results. In *ICCVW*, 2013.

[27] M. et al. Kristan. The visual object tracking vot2014 challenge results. In *ECCV*, 2014.

[28] Matej et.al. Kristan. The visual object tracking vot2014 challenge results. In Lourdes de Agapito, Michael M. Bronstein, and Carsten Rother, editors, *ECCV Workshops (2)*, volume 8926 of *Lecture Notes in Computer Science*, pages 191–217. Springer, 2014.

[29] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 2010.

[30] Rainer F. Schell. Computer vision for autonomous flight guidance and landing. In *IFAC Symposium Aerospace Control '92*, Ottobrunn, 1992.

[31] Falk Schubert, Daniele Casaburo, Dirk Dickmanns, and Vasileios Belagiannis. Revisiting robust visual tracking using pixel-wise posteriors. In *Computer Vision Systems*. Springer, 2015.

[32] Falk Schubert and Krystian Mikolajczyk. Benchmarking gpu-based phase correlation for homography-based registration of aerial imagery. In *Computer Analysis of Images and Patterns - 15th International Conference, CAIP 2013, York, UK, August 27-29, 2013, Proceedings, Part II*, pages 83–90, 2013.

[33] Jiajia Shang and Zhongke Shi. Vision-based runway recognition for uav autonomous landing. *IJCSNS International Journal of Computer Science and Network Security*, 7(3), 2007.

[34] Jianbo Shi and Carlo Tomasi. Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, 1994.

[35] T. Soni and B. Sridhar. Modelling issues in vision based aircraft navigation during landing. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 89–96, Sarasota, FL, 1994. IEEE.

[36] R. Szeliski, editor. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[37] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. *ECCV*, 2006.

[38] Stefan Werner, Simon Fürst, Dirk Dickmanns, and Ernst Dieter Dickmanns. Vision-based multi-sensor machine perception system for autonomous aircraft landing approach. In *10th Int. AeroSense Symposium: Enhanced and Synthetic Vision*, pages 54–63, Orlando, FL, USA, April 1996. Proc. SPIE 2736.

[39] Y. Wu, J. Lim, and M-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[40] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Part-based visual tracking with online latent structural learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, Oregon, USA, 2013.