Fast model based feature matching technique applied to airport lighting


Published in:
IET Science, Measurement and Technology

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date: 06. Jun. 2019
Fast model-based feature matching technique applied to airport lighting

J. Niblock  J.-X. Peng  K. McMenemy  G.W. Irwin

School of Electronics, Electrical Engineering and Computer Science, Queen’s University Belfast, Ashby Building, Belfast BT9 5AH, UK
E-mail: jniblock01@ee.qub.ac.uk

Abstract: The use of image processing techniques to assess the performance of airport landing lighting using images of it collected from an aircraft-mounted camera is documented. In order to assess the performance of the lighting, it is necessary to uniquely identify each luminaire within an image and then track the luminaires through the entire sequence and store the relevant information for each luminaire, that is, the total number of pixels that each luminaire covers and the total grey level of these pixels. This pixel grey level can then be used for performance assessment. The authors propose a robust model-based (MB) feature-matching technique by which the performance is assessed. The development of this matching technique is the key to the automated performance assessment of airport lighting.

The MB matching technique utilises projective geometry in addition to accurate template of the 3D model of a landing-lighting system. The template is projected onto the image data and an optimum match found, using nonlinear least-squares optimisation. The MB matching software is compared with standard feature extraction and tracking techniques known within the community, these being the Kanade–Lucus–Tomasi (KLT) and scale-invariant feature transform (SIFT) techniques. The new MB matching technique compares favourably with the SIFT and KLT feature-tracking alternatives. As such, it provides a solid foundation to achieve the central aim of this research which is to automatically assess the performance of airport lighting.

1 Introduction

Landing is one of the most demanding flight regimes in fixed-wing aircraft operations. It has been estimated that landing accidents constitute 41.1% of all hull losses. Hull loss is defined as aircraft damage beyond economical repair. A significant portion of night landing accidents can be attributed to the confusing illusions experienced by the pilot regarding the aircraft position with respect to the runway [1]. Furthermore, in recent years, there has been a substantial increase in the demand for air transport, which has resulted in larger sizes of aircraft and increased flight frequency. This requires higher performance and better maintenance of the airport landing lighting pattern [2], which is used to guide and direct aircraft to a safe landing in all types of weather conditions. One such way of achieving this is to regularly monitor airport lighting with the aim of highlighting and repairing any under-performing luminaires (A complete lighting unit consists of a lamp or lamps together with the parts designed to distribute the light, to position and protect the lamps and to connect them to the power supply.) [3].

Airport landing lighting consists of ground-based runway luminaires and elevated approach luminaires (A complete lighting unit consists of a lamp or lamps together with the parts designed to distribute the light, to position and protect the lamps and to connect them to the power supply). [4]. Over the years, several land-based measurement systems such as the mobile airfield light monitoring system [MALMS (http://www.tmsphotometrics.com/)] and photometric airfield calibration [PAC (http://www.gsilight.com/ppaclabcombo.htm)] have been developed, which are
capable of assessing the performance of inset runway lighting using light meters and cameras, respectively. Typically, these systems work by collecting data as a vehicle (with onboard light sensor technology) drives over, or past, the rows of luminaires. Unfortunately, the major limitation of these systems is that they are incapable of assessing the performance of the approach lighting system (ALS) where the luminaires are raised between 2 and 13.75 m above the ground level.

In 1976, Milward [6] was the first person to acknowledge the potential of an aerial imaging system to assess the performance of airport luminaires [7, 8]. McMenemy and Dodds [9] describe the progress towards an aerial-based imaging system capable of monitoring and assessing airport landing lighting. A vision system was located in the cockpit of an aircraft in order to acquire a video sequence during a normal approach to an airport. The 5 min video sequence was then split into its component frames and analysed offline. The work concluded that if the performance assessment of the airport landing lighting is to be implemented autonomously, then high-quality industrial cameras are required.

This paper extends and builds on that preliminary work to create fully autonomous, image-based, feature identification and tracking software capable of assessing the performance of the airport lighting in real time. In order to assess the performance of the luminaires, it is first necessary to uniquely identify and track each luminaire in the acquired images of the landing lighting pattern. The details of the identification and matching software developed by the authors are addressed in this paper. The ALS is used to test the developed software because the luminaires are elevated above the ground and as yet no method of autonomously assessing their performance exists.

1.1 Airport lighting pattern layout

There are strict guidelines for the location and positioning of luminaires within the airport landing lighting pattern. The relevant standards are documented in the International Civil Aviation Organisation’s (ICAO) standards and recommended practices’ documents [5]. Milward [6] also presents an overview of the standardised lighting system used in the UK and Europe, or, as it is technically known, the Calvert System. For the purposes of this paper, a CAT I ALS [10, 11] is assumed, as illustrated in Fig. 1.

The ALS consists of a 900 m coded line of white lights, on the extended centreline of the runway, and five crossbars at 150 m intervals. The bars decrease in width towards the runway threshold, with lines through the outer lights of the bars converging to meet the runway centreline 300 m upwind from the threshold.

Because of the close proximity of the ALS luminaires illustrated in Fig. 1, it is assumed that if the identification and tracking software is capable of robustly extracting and tracking each ALS luminaire, then it is a straightforward task to extend the techniques to the runway lighting. The next section introduces a literature survey into the existing tracking techniques.

1.2 Autonomous feature tracking

Feature tracking is among the most familiar and long-standing activities in computer vision and image processing [12]. To date, feature tracking has been extended to a number of applications including video surveillance [13–15], medical research [16, 17], image reconstruction [18] and orientation [19, 20], trajectory measurement [21, 22] and autonomous road-lighting performance assessment [23].

This paper proposes a fast, robust and autonomous feature-matching technique that is capable of tracking sequences of airport landing lighting data during a normal descent to an airport. McMenemy and Dodds [9] proposed tracking the luminaires by first estimating the position of the aircraft. From this estimated position, a pin-hole camera model is used to extrapolate the 2D image location of each individual luminaire in the airport lighting pattern. A search around each expected luminaire (represented by a

![Figure 1 CAT I approach lighting pattern](https://www.ietdl.org)
blob of bright interconnected pixels) coordinate is performed to find the associated bright pixels in the image. These bright pixels are summed to represent a luminaire’s total grey level. This work has two major limitations. As the acquired image data used were of poor quality, feature-extraction results were not comprehensive, which in turn required excessive manual intervention to locate luminaires within the image. Thus, the execution time required to run a sequence of images was excessive at $\sim 40\, \text{s/frame}$.

The International Civil Aviation Authority has published a recommendation specifying that the measurement of luminous intensity, beam spread and orientation of the luminaires, included in approach and runway lighting systems for a precision approach runway CAT I, II or III, should be assessed autonomously. The measurements should be undertaken by a mobile measuring unit of sufficient accuracy to analyse the characteristics of the individual luminaires [24]. In real-time systems, only a small percentage of the resources can be allocated to tracking, the rest being required for the preprocessing stages such as recognition, trajectory interpretation and reasoning. Therefore fast feature-tracking software is desirable such that the image flow can be operated in real time.

Feature-tracking algorithms commonly use richly textured features, such as edges, within an image [18, 20, 25]. Algorithms, such as the Kanade–Lucas–Tomasi (KLT) [26], search for the intensity gradient over a predefined search area or window. This approach has a number of drawbacks when related to the identification of airport landing lighting. First, luminaires further away from the vision sensor have a smaller pixel coverage than the ones closer to it and so a dynamic search window is required to accurately quantify the performance measure of the luminaires. Secondly, the KLT and similar techniques are prone to mismatches as the blobs (extracted features) have similar gradients. In this research, each luminaire, within an image frame, is allocated a single feature, which is uncommon. Normally, the object of interest is composed of a number of well-textured features [14, 20, 27], thus allowing the feature tracking to place in context within the scene. Other approaches match strictly on proximity and similarity [14]. For example, Teknomo et al. propose matching results based on the distance from one object to another and the similarity between the object features using a static camera. However, airport landing lighting comprises some 200 luminaires (for a CAT I ALS) located according to strict standards set by the ICAO [5]. Commaniciu et al. [28] and Hue et al. [27] assume a non-rigid target, whereas this work uses a rigid target with a dynamic observer. The dynamic observer can in turn introduce vibrations to the acquired image data. Matching blobs of light by proximity alone can therefore lead to inaccurate correspondences between image frames. Furthermore, if the features associated with each blob (such as grey level and pixel count) are considered, the uniform nature of the ALS can lead to blobs in close proximity having similar characteristics. This additional information will therefore still lead to mismatched data, as illustrated in Fig. 10.

The software proposed in this paper is based on the model-based (MB) matching methodology [29]. The model (a 3D set of luminaires positions) is matched to the image sequence (a series of 2D sets of blobs extracted from an image), where each blob is ideally a single luminaire from the ALS and there exists a one-to-one correspondence between the model and the image data. To do this, we propose a model for the camera projection system that will map the template onto the extracted image data. Pressigout et al. [20] calculate the $(X, Y, Z)$ camera position in 3D space when working in a controlled lighting environment and assessing the orientation of everyday objects such as boxes. However, this application assumes a dynamic platform with associated movement and varying lighting conditions. As such the yaw, pitch and roll $(\psi, \theta, \gamma)$ are estimated in order to accurately ascertain the position and pose of the camera for each acquired image. Chatterji et al. [30] currently use six degrees of freedom to locate the aircraft position in relation to the runway. We take this work one step further by using an optimised least-squares solution to facilitate fast, optimised identification of luminaires. Furthermore, the new software presented is fully autonomous, thus requiring no human intervention, thereby decreasing the execution time. To ensure that the blobs are correctly assigned to the model, an error coefficient is calculated for each iteration. This is minimised using nonlinear, least-squares optimisation to produce a one-to-one match between the template and the image data.

### 1.3 Algorithm performance evaluation

The main aim of this paper is to present a software that outperforms existing feature-tracking methods in terms of successfully identifying and tracking extracted features and its associated execution time. With this application, it is paramount that the luminaires are correctly matched so that luminaire histories are not confused. That is, a luminaire is correctly assigned the same label in all successive images in which it appears. The proposed MB matching technique is therefore compared with two globally recognised tracking alternatives. Synthetic data are used to assess the effectiveness and gauge the performance of the proposed feature extraction and matching techniques. In order to simulate a real-life approach to an airport, a 3D model of the ALS is presented. This uses actual
lighting positions from a Calvert lighting pattern in the UK to reconstruct a realistic model of the approach luminaires during a normal (3') approach to the airport.

The KLT algorithm [26] is well accepted for feature tracking. This method defines the measure of match between fixed-sized feature windows in the past and current frame as the sum of squared intensity differences over the windows. The displacement [31] is then defined as the one that minimises this sum. For small motions, a linearisation of the image intensities leads to a Newton–Raphson style minimisation.

The second technique used for comparison is the scale-invariant feature transform (SIFT) [25]. Local invariant features allow small portions of cluttered images under arbitrary rotations, scalings, change of brightness and contrast and other transformations to be efficiently matched. The idea is to break the image into many small overlapping pieces, each of which is described in a manner invariant to the possible transformations. Each part can be individually matched and the matching pieces put back together.

The following section introduces the existing techniques used for tracking sequences of airport landing lighting images. The limitations of the existing tracking techniques are highlighted using synthetic data before introducing the new MB software in Section 3, which was developed especially for this application. The MB software is tested on the synthetic ALS data in Section 3.5 and also on actual image data in Section 4. A discussion on the effectiveness of the new MB software is given before offering some concluding remarks on the new software.

2 Existing tracking algorithms

This section gives an overview of two established tracking techniques that were selected to track airport luminaires within an image sequence.

2.1 KLT algorithm [26, 31]

In general, any function of three variables \( l(x, y, t) \), where the space variables \( x, y \) and the time variable \( t \) are all discrete and suitably bounded, can represent an image sequence. However, images taken with small time differences are usually strongly related to each other, because they refer to the same scene taken from only slightly different viewpoints.

This correlation is usually expressed by saying that there are patterns that move in an image stream. Formally, this means that the function \( l(x, y, t) \) is not arbitrary but satisfies the property

\[
I(x, y, t + \tau) = l(x - \xi, y - \eta, t)
\]

where a later image at time \( t + \tau \) can be obtained by moving every point in the current one at time \( t \) by a suitable amount. The amount of motion needed \( \delta = (\xi, \eta) \) is called the displacement of the point \( p = (x, y) \) between the time instants \( t \) and \( t + \tau \), and generally is a function of \( x, y, t \) and \( \tau \) [26].

An important problem in finding the displacement \( \delta \) of a point from one frame to the next is that a single pixel cannot be tracked, unless it has very distinctive brightness with respect to its neighbours. In fact, the value of the pixel can change because of noise, and hence be confused with adjacent pixels [26]. Consequently, it is often difficult or even impossible to determine where a particular pixel went in the subsequent frame using only local information. To overcome this, the KLT algorithm tracks windows of pixels, looking for the ones that contain sufficient texture. Formally, redefining \( J(p) = l(x, y, t + \tau), l(p - \delta) = l(x - \xi, y - \eta, t) \) and dropping time for brevity, the local image model is then represented by

\[
J(p) = l(p - \delta) + n(p)
\]

where \( n \) denotes noise. The displacement vector \( \delta \) is then chosen so as to minimise the residue error, defined by a double integral over the given window area \( \mathcal{W} \)

\[
\epsilon = \int_{\mathcal{W}} [l(p - \delta) - J(p)]^2 w \, dp
\]

In (3), \( w \) is a user-defined weighting function. In the simplest case, \( w \) could be set to 1 or could be a Gaussian-type function chosen to emphasise the central area of the window. Although there are several methods to minimise the residue in (3), a linearisation (truncated Taylor expansion) method [31] is used here, when the displacement \( \delta \) is much smaller than the window size.

2.2 Scale-invariant feature transform [19, 25]

This is a computer vision technique to extract distinctive features from images, applicable to matching different views of an object or scene and object recognition. The features are assumed invariant to image scale or rotation, and partially invariant to changing viewpoints and changes in illumination. The name SIFT, refers to the fact that the image data are transformed into scale-invariant coordinates relative to local features.
They are well localised in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter or noise. Large numbers of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with a high probability against a large database of features, providing a basis for object and scene recognition.

The cost of extracting these features is minimised by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test.

The major steps in the computation of the image features include the following.

1. **Scale-space extrema detection.** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify the potential interest points that are invariant to scale and orientation.

2. **Keypoint localisation.** It is used at each candidate location, and a detailed model is fit to determine the location and scale. Keypoints are selected based on the measure of their stability.

3. **Orientation assignment.** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that have been transformed relative to the assigned orientation, scale and location of each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor.** The local image gradients are measured at the selected scale in the region around the keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

The invariance of SIFT towards different image transformations such as image rotation, scale changes (zoom) and off-plane rotations makes it ideal for application to airport lighting.

### 2.3 Adapting the KLT and SIFT techniques

By default, the KLT software ([http://www.ces.clemson.edu/stb/klt/](http://www.ces.clemson.edu/stb/klt/)) accepts a series of portable gray map image files as the input file to produce a portable pixel map file of results. A number of alterations were necessary to ensure compatibility of airport lighting images in uncompressed bitmap (BMP) format. Furthermore, as noise effects, especially from the horizon and runway markings, (Fig. 2) are common in the practical images, a rectangle was used to segment the region of interest (ROI), which is the airport approach lighting pattern visible to the vision sensor. Without this step, false (redundant) features were sent to the KLT software making the tracking process less efficient in the sense that unnecessary information was extracted and tracked. Since the SIFT software was able to accept scale-variant BMP images, to maximise the number of correctly identified and extracted features, the image segmentation described above was again employed.

The next section details how a 3D model of the ALS luminaires was constructed using an image processing graphics rendering package.

### 2.4 Synthetic data

Virtual modelling and scene rendering are particularly useful in applications where it is not viable to continually test new techniques on a real-life model. To aid testing and comparison of the tracking alternatives, a 3D model of the ALS was generated. Here, 2D information taken from a UK airport is adapted to include luminaire height information, obtained from the ICAO standards [5], and rendered into a 3D model of the lighting pattern using OpenFX software ([http://www.openfx.org/](http://www.openfx.org/)).

The objective of the 3D model is to replicate an approach (either normal or at differing angles of approach) to the airport and create a set of image data in order to replicate what a camera would capture from the cockpit of an aircraft during an actual approach to the airport. The model offers a low-cost solution for testing the SIFT and KLT matching.
techniques, to highlight any limitations they may have. As strict guidelines are enforced on lighting pattern layout by the ICAO, the ALS model created for this work can be used to represent any CAT I ALS in the UK and Europe.

The following sections show how the SIFT and KLT techniques perform on a sequence of 200 synthetic images for a perfect (noiseless) set of approach data and corrupted (noisy) approach data. The existing techniques are first tested on noiseless data where all the background and motion noise have been removed. The luminaires are in such close proximity and have similar characteristics, in terms of pixel coverage and grey level, thus making it a difficult matching problem. In order to make the synthetic data more realistic, background and motion (vibration) noise are injected into the image data.

A number of criteria are used to assess performance. Success rate, shown by (4), is used to measure the quality of the matching results. This compares the number of successfully identified features ($S_T$) to the number of expected features ($S_T$), that is, the number of luminaires that should be present in the image.

$$\text{Success Rate}(\%) = \frac{S_T}{S_T} \times 100 \quad (4)$$

Sinha et al. [32] have shown that both the KLT and SIFT techniques can be utilised in real-time applications. As such, quantifying the execution time ($t$) is useful for judging the practicality of the new MB tracker.

Both the image matching techniques were run using the Mathworks MATLAB software on a 3 GHz CPU with 1 GB RAM and some sample results follow.

With respect to the two sets of synthetic image data defined in this section, a number of assumptions are made. First, noiseless data:

1. a standard Calvert category I (CAT I) ALS is utilised;
2. all luminaires are present within the ALS;
3. all sources of noise have been removed, such as background noise and the horizon.

A second set of data termed noisy data are used. With the noisy data, two types of noise are added to the image sequence in order to simulate an actual approach. First, a noise value in the form of uniformly distributed, white noise (100%) is added to the background of the image. This constitutes uniform white noise mixed with the underlying signal; so, for example, if the dynamic range of the signal $X_{noNoise}$ is 0–1, the noise is added;

$$X_{noise} = X_{noNoise} + 0.5 \times (\text{noise\%}/100) \text{ and clamped to the range } 0-1.$$  

The second type of noise is motion blur (100%). This is created by rendering a number of subframes between each frame and averaging the result to simulate vibrations of the aircraft and pitch (header) movement.

2.4.1 KLT results: The KLT software was run on the synthetic image sequence with a 63% success rate for noiseless image data and 59% for noisy image data. Fig. 3 depicts the number of expected luminaires in each frame against the number of luminaires that are extracted and successfully identified using the KLT technique. The success rate for both noiseless and noisy data is illustrated in Fig. 4, which shows that the KLT identifies ~60% of the luminaires within the image data. The effects of adding noise to the image data are minimal, but it is worth noting that they do have a detrimental effect on the success rate. This is
understandable because of the fact that some of the features may merge due to their close proximity and thus be classified as a single feature instead of multiple features.

With this in mind, it was found that an initial problem with the KLT technique was that a static number of features were being tracked throughout the image sequence. This application relies on luminaires being successfully dropped as the aircraft passes over them and they leave the image data acquired by the camera. To accommodate this, the number of features was updated dynamically, which overcomes the problem of numerous features constituting a single luminaire in the latter images. The number of features dropped is recorded and subtracted from the total number of luminaires to obtain $S_T$, that is, the number of expected luminaires.

The execution time for both the noiseless and noisy data are compared in Fig. 5 with the detailed timing data given in Table 1. The results show a number of trends. First, as the number of features to track decreases so does the mean time per frame. The mean time starts at $\sim 63\,\text{s/frame}$ and finishes at just over $50\,\text{s/frame}$ for both the noiseless and noisy data. These timing results are excessive with a mean execution time per frame of $\sim 60\,\text{s}$. It is clear from the KLT code that it has not been optimised for large uncompressed image data. This coupled with the fact that the simulations were run in the MATLAB image processing environment means that, because of inefficient memory handling, the results are not optimised. If the KLT was run using a C++ environment, such as Microsoft Visual Studio, the timing results could be much improved. Tests showed that the KLT had a mean execution time per frame of $\sim 3\,\text{s}$ using visual studio.

![Figure 5 Execution time for KLT on the synthetic ALS](image1)

![Figure 6 Identified against template luminaires for SIFT on the synthetic ALS](image2)

<table>
<thead>
<tr>
<th>Algorithm Name</th>
<th>Total time, s</th>
<th>Mean time, s</th>
<th>Standard deviation, s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noiseless data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLT</td>
<td>11 872.0</td>
<td>59.4</td>
<td>4.1</td>
</tr>
<tr>
<td>SIFT</td>
<td>2086.9</td>
<td>10.5</td>
<td>1.3</td>
</tr>
<tr>
<td>MB</td>
<td>625.0</td>
<td>3.1</td>
<td>0.6</td>
</tr>
<tr>
<td><strong>Noisy data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLT</td>
<td>11 325.0</td>
<td>56.6</td>
<td>7.8</td>
</tr>
<tr>
<td>SIFT</td>
<td>2099.6</td>
<td>10.6</td>
<td>1.3</td>
</tr>
<tr>
<td>MB</td>
<td>610.3</td>
<td>3.1</td>
<td>0.7</td>
</tr>
<tr>
<td><strong>Actual data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB</td>
<td>456.2</td>
<td>2.3</td>
<td>1.2</td>
</tr>
</tbody>
</table>
2.4.2 SIFT results: Fig. 6 depicts the number of expected luminaires in each image frame compared with the number of luminaires that are extracted and successfully identified using the SIFT technique for both the noiseless and noisy synthetic ALS image data. It is apparent from this figure that the SIFT software has not produced good luminaire identification results. This is highlighted by the large gap between the template (expected) number of luminaires and the successfully identified luminaires extracted from the synthetic ALS data. These results are further backed up by looking at the success rate illustrated in Fig. 7. The mean success rate for the noiseless data was 22% and just under 20% for the noisy data. This means that only one out of every five luminaires is correctly identified using the SIFT software.

However, one trend is apparent and worth noting: as the frame number increases so does the success rate. This is due to the fact that as the frame number increases, the aircraft (camera) comes closer to the runway and hence luminaires fall out of the field of view of the camera. As there are less luminaires to track, the SIFT software identifies fewer features within the image data. When there are fewer features in the images, the matching task becomes more simplistic and greatly minimises the probability of a mismatch.

The SIFT software produced better results than the KLT in terms of timing. Fig. 8 shows the timing information for the synthetic ALS data. It is apparent from the figure that the timing data fluctuates between 10 and 11 s/frame. This is reinforced in Table 1 with a mean execution time of ~10.5 s/frame. This result is better than the KLT; however, if a real-time system is to be realised, the software needs to run faster and should be implemented in a C++ environment.

2.5 Limitations of existing tracking techniques

From the application of the SIFT and KLT techniques on the synthetic ALS data, it is evident that they both have a number of disadvantages in terms of successfully matching the luminaires. The results in Fig. 7 show that the SIFT is the least successful matching technique, with only 20% of the blobs being successfully matched. However, and more importantly, the techniques are prone to mismatched data. Fig. 9 shows the ideal scenario where the same image is used to show how a one-to-one correspondence between two consecutive images looks. It can be clearly seen that the figure is composed of two frames and a series of horizontal lines going from one feature to the corresponding feature in the opposite frame. This denotes a successful match. Now, suppose the frame on the right in Fig. 9 is exchanged for the next frame in the image sequence. This produces the results shown in Fig. 10. This figure shows that the SIFT
algorithm has successfully found a one-to-one correspondence between the two frames, but this time there are a number of mismatches represented by the diagonal lines. That is, a luminaire which is assigned a label in the first image will be assigned a different label in the next image. This application depends on the condition of the same luminaire being assigned the same label in successive image frames, the reason being that the main objective of the work is to assess the performance or intensity of the airport luminaires. Research has shown [8] that it is possible to estimate luminous intensity from an image using the pixel grey-level information extracted from each luminaire. To enhance the accuracy of this estimate, it is important to collect the pixel grey level over a sequence of images. As such, it is important to correctly identify luminaires, otherwise their histories (and later luminous intensity performance data) will become confused.

With this in mind, the next section proposes the use of a template of the ALS luminaires in order to make the identification and tracking process more robust and minimise the likelihood of obtaining mismatched data.

3 MB technique

This section introduces an MB methodology proposed to identify and track airport luminaires through an image sequence. The camera representation, that is, the relationship between the real-world coordinate axis and the ALS plane is detailed in Section 3.1. A template of the ALS luminaire data is generated using information from Belfast International Airport, and the ICAO standards and optimisation techniques are used to superimpose this template onto the acquired image data. An overview of these algorithms is given in Section 3.2. The objective of the optimisation is to achieve a one-to-one match between successive image frames. This implementation of this theory is detailed in Section 3.3. ICAO standards stipulate that a requirement of any automated measurement technique is that it is capable of running in real time and that no manual intervention is required. Autonomous software initialisation algorithms are detailed in Section 3.4 before showing how the MB software performs on synthetic data.

3.1 Camera representation

A pin-hole camera can be mathematically modelled as a projection system and an image sensing system. The projection system projects a 3D object from the real world to the image plane of the camera. The image sensing system converts the projected 2D object on the image plane into a series of pixels through an image sensor array.

Figure 11 Projection system

According to the pin-hole imaging principle [41], Fig. 11 depicts how a 3D-point \((x, y, z)\) expressed in the camera coordinate system \(Oxyz\) is projected to a 2D-point \((u, v)\) on the image plane. From Fig. 11, we can summarise as

\[
\begin{align*}
    u &= f^{-1}x \\
    v &= f^{-1}y
\end{align*}
\]

where \(f\) denotes the focal length of the camera, which may vary as the aircraft approaches the airport in order to capture images of good quality.

Assuming that the image sensor array is evenly configured as orthogonal grids in the image plane, the image sensing system can then be represented as the linear transformation

\[
\begin{align*}
    \hat{p} &= k_\theta + k_u u = k_\theta + \alpha_u x^{-1}y \\
    \hat{q} &= k_\nu + k_v v = k_\nu + \alpha_v x^{-1}y
\end{align*}
\]

where \((\hat{p}, \hat{q})\) denotes the coordinates of the pixel corresponding to the 2D-point projection \((u, v)\) in the image plane given in (5), \(k_u\) and \(k_v\) the number of pixels per unit distance in the \(u\) and \(v\) directions, \(\alpha_u = k_u f\) and \(\alpha_v = k_v f\), \((k_\theta, k_\nu)\) represents the image coordinates of the intersection of the optical axis and the image plane, referred to as the principal point.

However, the approaching template is normally defined with the 3D coordinates of the 120 luminaires given in the airport coordinate system \(O_aX_aY_aZ_a\).
Transformation from the airport coordinate system to
the camera one normally includes a translation and a
rotation as shown
\[
\begin{bmatrix}
x
y
z
\end{bmatrix} = R(\gamma, \theta, \psi) \begin{bmatrix}
x - x_0
y - y_0
z - z_0
\end{bmatrix}
\tag{7}
\]
where the translation \( o = [x_0, y_0, z_0]^T \) contains the
coordinates of the camera position (defined at the
original point of the camera coordinate system, see
Fig. 11) in the airport coordinate system. The vector
\( t = [x_1, y_1, z_1]^T \) contains the coordinates of an
approaching template luminaire in the airport
coordinate system and \((\gamma, \theta, \psi)\) denote the yaw,
pitch and roll of the camera system, respectively (i.e. the
three Euler angles of the camera coordinate system
relative to the airport coordinate system). \( R(\gamma, \theta, \psi) \)
is the corresponding rotation matrix, which is formed
by three sequent rotations around the Z-, Y- and X-
axes as in (8)
\[
\begin{align*}
R(\gamma, \theta, \psi) &= R_z(\psi)R_y(\theta)R_x(\gamma) \\
R_x(\gamma) &= \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \gamma & \sin \gamma \\
0 & -\sin \gamma & \cos \gamma
\end{bmatrix} \\
R_y(\theta) &= \begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{bmatrix} \\
R_z(\psi) &= \begin{bmatrix}
\cos \psi & \sin \psi & 0 \\
-\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\end{align*}
\tag{8}
\]
where \( R_z(\psi), R_y(\theta) \) and \( R_x(\gamma) \) are the corresponding
rotation matrices around the Z-, Y- and X-axes,
respectively.

Note that the translation \( o \) is determined by the
position of the aircraft, whereas the three Euler angles
are determined by its altitude. These parameters are
important and should be identified when the aircraft
needs to be positioned.

### 3.2 Camera parameter identification

In the previous section, a camera model was detailed
consisting of four linear parameters \((k_0, \alpha_x, k_\alpha, \alpha_x)\)
and six nonlinear ones \((\psi, \gamma, \theta, x_0, y_0, z_0)\). Given a
set of blobs \((p(i), q(i)), i = 1, \ldots, N\), identified from
an image frame (extracted from an image sequence)
and the corresponding luminaires \((x(i), y(i), z(i)), i = 1, \ldots, N\);
representing the \( N \) blobs in two
column vectors \( p = [p(1), \ldots, p(N)]^T \) and \( q = [q(1), \ldots,
q(N)]^T \) and the \( N \) corresponding template luminaires
in three column vectors \( x_i = [x_i(1), \ldots, x_i(N)]^T \),
\( y_i = [y_i(1), \ldots, y_i(N)]^T \) and \( z_i = [z_i(1), \ldots, z_i(N)]^T \),
the ten parameters can be identified using nonlinear
optimisation to minimise (9) with respect to these
parameters
\[
E = e_u^T e_u + e_v^T e_v \rightarrow \min
\]
s.t. \( e_u = p - k_\omega 1 - \alpha_u u \) \( e_v = q - k_\omega 1 - \alpha_v v \)

In (9), \( I \) denotes the column vector of \( N \) unity elements;
the \( i \)th element of \( e_u \) and \( e_v \), denoted as \( e_u(i) \)
and \( e_v(i) \) are the matching errors of the \( i \)th blobs/luminaire pair
in the \( u \) and \( v \) image directions, respectively; \( u \) and \( v \) are
given in (10).
\[
\begin{align*}
u &= [x^{-1}(1)y(1), \ldots, x^{-1}(N)y(N)]^T \\
v &= [x^{-1}(1)z(1), \ldots, x^{-1}(N)z(N)]^T
\end{align*}
\tag{10}
\]
with the relationship between \( x(i), y(i) \) and \( z(i) \) and
\( x_i(i), y_i(i), z_i(i) \) given in (7).

This is a nonlinear optimisation problem for which
many methods of solution exist. One such method is
detailed by Pressigout et al. [20]. However, for the
purposes of this work, the Levenberg–Marquardt
(LM) method [33, 38–40] is used to estimate the
optimal camera parameters.

The next section introduces the theory behind
producing a one-to-one correspondence between the
template luminaires and the extracted image blobs.

### 3.3 Correspondence between blobs and
template luminaires

Identification of the camera parameters is based on a
given set of correspondence pairs \((p(i), q(i))/\{x(i),
y(i), z(i)\}, i = 1, \ldots, N\). However, the correspondence
problem is a long-standing fundamental one in
computer vision [18], and so far there is no general
reliable solution available. In this section, a simple
iterative procedure is introduced.

Suppose \( M \) raw blobs are extracted from an image,
denoted as \((p(i), q(i)), i = 1, \ldots, M\). Given an
image of the approach lighting template simulated in
the camera model, the problem then is, to denote the
positions of the 120 luminaires in the simulated image
as \((x(i), y(i), z(i)), i = 1, \ldots, 120\). For each
luminaire, the closest blob is selected as the
correspondence. This involves finding the closest blob
to each luminaire, say the $i$th according to (11)

$$j(i) = \arg \min_{1 \leq k \leq M} d(i, k), \quad i = 1, \ldots, 120$$

(11)

where $d(i, k)$ is the squared-distance between the $i$th luminaire and $k$th blob

$$d(i, k) = (\hat{p}(i) - p(k))^2 + (\hat{q}(i) - q(k))^2$$

$$\hat{p}(i) = \hat{p}(x_i(i), y_i(i), z_i(i))$$

$$\hat{q}(i) = \hat{q}(x_i(i), y_i(i), z_i(i))$$

(12)

Here $(\hat{p}(i), \hat{q}(i))$ is the position of the $i$th template luminaire in the simulated image generated in the camera model (6).

However, the solution in (11) is definitely not a one-to-one correspondence set. There is the probability that more than one blob corresponds to the same luminaire since the raw blobs extracted include not only the model luminaires but also those that can arise from the background (such as the horizon) and random image noise. Niblock et al. [4] propose that other common problems leading to mismatches between the model and image data occur from factors such as inadequate sensor resolution, reflections causing stray noise, luminaires leaving and entering the field of view of the sensor and occlusions, where a luminaire is obstructed by an obstacle from the field of view of the sensor. In any case, the closest blob is always selected as shown in (12) and the missing blobs are thus identified. In addition, a distance threshold is applied such that only those correspondence pairs with a distance not greater than the threshold are selected. The distance threshold is normally set at a few pixels or slightly more, depending on the accuracy of the camera model. Once a correspondence pair set is selected, the camera model parameters are then updated using the LM method detailed by Peng et al. [33]. Then a new correspondence pair set is selected based on the updated camera model. This update process is iterated until the correspondence pair is no longer changed, that is, it remains consistent.

3.4 Algorithm initialisation

The previous iterative algorithm is based on an initial condition, which can either be an initial correspondence pair set or an initial camera model parameter set.

For an application on image sequences, after the matching loop on the first image, the camera model parameter set is identified using the previous image, and these parameters are applied as the initial camera model parameter set for the current matching loop. For the matching loop, on the first image or for applications on a single image, an initial correspondence pair set must be identified.

To select an initial correspondence pair set, the line having the most number of collinear blobs on it, referred to as the principal line, is identified first. This is implemented using the Hough transform [34, 35] over the 2D coordinates of the blobs.

Note that, during an approach, the aircraft enters the template along the centreline (orthogonal to the bars), hence the yaw $(\psi)$ is normally no more than a few degrees. This means that the template bars are approximately horizontal, corresponding to the Hough parameter $\alpha = 90^\circ$. The Hough parameter space $(\rho, \alpha)$ is thus selected as $\rho \in \mathbb{R}$ and $80 \leq \alpha \leq 100$, where $\rho$ represents the perpendicular distance from the origin and $\alpha$ the angle with the normal.

Secondly, the raw blobs are clustered into lines that are parallel to the principal line based on their perpendicular distance to the principal line. That is, for each cluster of blobs, the differences between the perpendicular distances to the cluster centre (the mean value of the perpendicular distances) are within a given threshold value.

Once the principal line is identified, the model bars are then identified according to their relative position and the number of collinear blobs. In detail, the template bar corresponding to the principal line is identified according to the number of collinear blobs on the principal line.

For example, Fig. 2 shows an image captured during an approach to Belfast International Airport. Compared with the template, the principal line is identified as bar 4, which is composed of 16 luminaires (two wing bars of seven luminaires and two central luminaires), and the above three lines can then be identified as bars 3, 2 and 1, respectively. Once the bars are identified, an initial correspondence pair set can be selected, whereas the other raw blobs are not used for initialisation.

Note that, for initialisation, it is not necessary to uniquely identify each luminaire from the image. The ALS is composed of five crossbars of luminaires. These crossbars need to be identified and matched to the template. This reduces the effects of background noise, such as the horizon and cloud cover illustrated in Fig. 2, thus making the algorithm more robust. The full correspondence (including the centreline luminaires and the crossbars) is produced and updated iteratively using (12) in the parameter optimisation procedure detailed below.
The iterative algorithm for image sequences is summarised as follows.

1. Set the loop counter \( k = 1 \), read the image data of the frame, \( k \) and extract the raw blob set from the image \( k \), denoted as \( B(k) \). Select an initial correspondence pair set, denoted as \( I(k, 0) \).

2. Set loop counter \( i = 1 \).

3. Update the model parameters by (9) in LM method.

4. Select a new correspondence pair set \( I(k, i) \).

5. Check the changes in the correspondence pair set. If \( I(k, i) \neq I(k, i-1) \), let \( i \leftarrow i + 1 \) and go to step 3. Otherwise, go to step 6.

6. Let \( k \leftarrow k + 1 \). If the image sequence is completed, terminate the process. Otherwise, read the image data of frame \( k \) and extract the raw blob set \( B(k) \); select the correspondence pair set \( I(k, 0) \) based on the current model, and go to step 2.

The next section shows how the MB software performs on synthetic image data detailed in Section 2.4.

### 3.5 Synthetic results

The MB technique proved successful in identifying, matching and tracking the luminaires in the synthetic ALS image data. Fig. 12 shows the results for both the noiseless and noisy image data compared with that of the template. It is worth noting that for the noiseless image data a 100% success rate was obtained. Further details of the success rate is illustrated in Fig. 13. The results for noiseless data were also good with a success rate of 98%. The lowest success rate was 86% in frame number 127 and this was due to the fact that a number of luminaires had been merged because of the effects of vibration on the image data. This is a data extraction problem as opposed to a matching problem.

When noise is injected into the images, the quality of the airport ALS decreases dramatically. One of the advantages of using an MB methodology is that even if a luminaire is faulty and hence missing from the image data, it can still be flagged up and accounted for, as illustrated in Fig. 19. The KLT and SIFT techniques are not able to account for missing luminaires, but can only replace them if they were present in a previous image.

The execution time results in Fig. 14 compare the timing information for the noiseless data with that of the noisy data. Once again, a number of trends exist between the noiseless and noisy data. First, as the
frame number increases, the execution time per frame decreases. This was a trend that was also apparent in the KLT timing results. It is also worth noting that for the first frame the execution time is greater than that of the subsequent frames. This is because the tracking software has to locate the target. Once the target has been identified, a priori information is used to locate the target in subsequent frames, hence decreasing the execution time. The results show that the timing information is $\sim 3$ s/frames for both the sources of data which is a notable improvement in the KLT and SIFT techniques. Table 1 gives a more detailed analysis of the timing results for all the three techniques for both the noisy and noiseless image data.

3.6 Discussion

The synthetic image data results show that the MB matching provides the best results in terms of luminaire identification and matching. In comparison, the KLT approach provides a good set of matching results, whereas the SIFT software is the least successful, with a success rate of $\sim 20\%$. Furthermore, these results prove that it is possible to have almost perfect matching using the MB technique. In contrast, the KLT and SIFT techniques are prone to mismatches in luminaire data. A major requirement of this work is that the identification and matching data are accurate and robust enough to assess each luminaire’s performance in the complete lighting pattern. The results suggest that the new MB matching technique is the best approach to track the synthetic ALS data.

Of the two types of noise injected into the images, the motion blur had a larger detrimental effect on the success rate. This was because in some instances luminaires were smeared and corrupted to an extent that they merged with neighbouring lights. The random white background noise had little effect on the success rate, as the luminaires have a much larger pixel coverage/grey level than the random noise and thus thresholding the images eradicates most of the background noise.

The author would like to acknowledge the fact that the more efficient timing data can be obtained by running the code in C++ and not in the MATLAB environment. Timing results presented in [36] show that the KLT can have an average execution time of $\sim 2$ s/frame. However, for a fair comparison, all algorithms were run from the MATLAB environment. Thus, memory handling inefficiency has led to non-ideal timing data.

For the MB matching to be deemed a success, it is necessary to test it on actual image data taken from a normal approach to an airport.

4 Actual data

This section presents some sample results taken from an approach to a UK aerodrome (Belfast International Airport). The images were captured using a monochrome CMOS [37] vision sensor with manual lens mounted on a vacuum-based platform to minimise the effects of vibration. The imager was...
connected to an Intel Pentium IV processor with controller software used to act as an interface between the processor and the vision sensor. A USB2.0 standard acts as a communication/power mechanism between computer and vision sensor. Fig. 15 gives an overview of the imaging system. For the purposes of this section, all other luminaires not in the ALS were turned off.

As with the synthetic image data, the success rate and execution time were computed for 200 actual images each having the typical format shown in Fig. 2. The frame number for the results of the actual image data goes from 360 to 560 and covers the complete ALS. To ensure that the camera captured all the approaches in detail, a number of redundant images were captured before and after the ALS to ensure that the imaging system was working correctly and capturing quality images before the approach luminaires came into the field of view of the camera.

The results given in Fig. 16 have a mean success rate of 76%. The main reason for the success rate being low is due to the feature extraction algorithms not being able to uniquely identify luminaires that have merged due to the effects of vibration. A further problem was caused by a number of luminaires being occluded from the camera by a temperature meter housed outside the aircraft. This meter covered a complete wing bar of the luminaires. These occluded luminaires are not seen by the imaging system and as such were assumed to be missing; however, unlike previous tracking algorithms, their position was still recorded in case the luminaire returned in a subsequent frame. The template against luminaire extraction information is presented in Fig. 17.

The execution timing data given in Fig. 18 show that the MB algorithm for the practical and the synthetic data are similar. The same trend of increased execution time for the first frame exists and then steadies to a mean time of \(~2\,\text{s/frame}\). Table 1 contains more detailed timing information. One thing to note is that the mean time per frame is actually smaller for the real data. This is surprising, because it should be greater because of the effects of spurious noise in the image data and the fact that the real image data have a marginally higher resolution. However, when the data were analysed, the synthetic data were captured in colour and thus the RGB values had to be averaged in order to access the grey level of any given pixel. The actual image data, for the approach to Belfast International Airport, were carried out with a monochrome camera and hence the information was already in the correct format.

![Figure 18](image1.png)

**Figure 18** *Execution time for MB software on the actual ALS*

![Figure 19](image2.png)

**Figure 19** *Missing/merged luminaires (left image) and optimisation results (right image)*

Note how the luminaires are still identified by the MB matching algorithm even though they are not present in the image data.
4.1 Discussion

The proposed MB matching technique performs well on the practical image data with the success rate being marginally reduced compared with that of the noisy synthetic data. The main problem came from luminaires that were occluded by an object external to the imaging system and merged image data. This is highlighted in Fig. 19. However, the MB technique shows how conclusive the results are for the luminaires that have been successfully extracted, with all the luminaires being uniquely identified and no mismatched data apparent.

Detailed extraction techniques need to be investigated for performance assessment. However, if the luminaire is fully occluded by an external object, it will not be in the image frame and even the best extraction algorithms will not be able to identify it. In this case, the only option is to sample at a higher frame rate and capture more images than is necessary in the hope of obtaining enough images of the luminaires earlier/later in the image sequence. Although the image analysis is not able to extract an occluded luminaire, at least the MB technique can detect that the luminaire is temporarily missing and can uniquely identify it.

The synthetic model is a good approximation for the actual data in terms of the timing analysis. The actual data were faster with a mean time of 2.3 s/frame as opposed to the synthetic data at 3.1 s/frame.

5 Conclusion

This paper details the foundation work into building an aerial-based imaging system designed to autonomously assess the performance of airport landing lighting during a typical approach to an airport. In order to realise a commercial solution, a number of assumptions have been made. First, only the ALS is tested. The approach lighting was chosen because of the close proximity of the luminaires and the difficulty this poses for the unique identification of the luminaires. In addition, the approach lighting is elevated from the ground and therefore its performance cannot be assessed using the conventional ground-based methods such as MALMS that have evolved over the past few years [4].

A literature survey of suitable existing tracking software was conducted and two techniques, namely KLT and SIFT algorithms, were deemed suitable for the application of identifying and tracking airport luminaires. In order to assess how these techniques perform, a synthetic model was created to replicate a normal approach to an aerodrome, where the noise injected into the image data could be controlled.

The paper shows that the existing techniques had limitations, particularly with mismatched data, and therefore an MB technique for matching a template (a predefined set of 3D lighting data from a UK Calvert lighting pattern) to the synthetic image data was presented. The success rate and the execution time for the MB software compared favourably with two alternative approaches in the literature.

In order to assess the robustness of the algorithms and make the synthetic data more realistic, noise (in the form of background and motion) was injected into the synthetic image data. Again, the proposed MB matching algorithm compared favourably with the KLT and SIFT algorithms. A final test was performed on the actual image data with a success rate of 76% and a mean execution time of 2.3 s/frame. These results are encouraging and suggest that an MB matching methodology is the best approach, where all luminaires in the ALS have to be uniquely identified and matched. It can also be concluded that while the synthetic model produces a good approximation to the actual measured images, it could be updated to allow for more severe cases of vibration when one or more luminaires merge.

Finally, more work has to be undertaken into tracking the runway luminaires and CAT II/III lighting patterns, where supplementary luminaires are added to aid the pilot when the visibility is poor during an approach to the airport. However, it is our belief that an aerial-based imaging system is capable of uniquely identifying the luminaires in the ALS. This successful identification and tracking of the luminaires from an image sequence forms the cornerstone on which future research can be based. Having developed this MB method, the next stage is to use the extracted luminaire grey-level data to autonomously assess the performance of the lighting pattern.

6 Acknowledgment

The authors wish to thank the EPSRC (Grant: EP/D05902X/1), the Royal Academy of Engineering and the Royal Society for financial backing.

The contribution of Flight Precision and Belfast International Airport for providing flight time in order to collect airport lighting data is also gratefully acknowledged.

7 References


[34] BALLARD D, BROWN C: ‘Computer vision’ (Prentice-Hall, 1982), Ch. 4


