Analyzing visual fidelity in flight simulation software using Game Engine with Feature Mapping

Rishabh Kaushik
Principal Engineer, Software Engineering
Salt Lake City, Utah
rishabh.kaushik@collins.com

Ankur Rathore
Principal Engineer, Software Engineering
Salt Lake City, Utah
ankur.rathore1@collins.com

ABSTRACT

The integration of Game Engine Technology into Flight Simulation Software has brought about a significant shift in the level of realism that can now be achieved. This is primarily due to advancements in rendering and graphics. As a result, pilots, instructors, and flight training schools are increasingly adopting and utilizing this hybrid technology. This can be seen in the growing demand for Extended Reality (XR) training systems. In these XR simulation software, visual fidelity is of utmost importance as it allows trainees to immerse themselves in the simulation's intricate details. As this trend continues, it becomes crucial to explore techniques for analyzing the visual elements in the scene that contribute to realism and the effectiveness of the simulation.

In this paper, we explore a specific technique that uses Feature Matching principles to perform a comparative analysis between a simulation based on a Game Engine and real flight visual data. Feature matching involves identifying unique keypoints or local features in each image and then finding corresponding matches between these keypoints across the images. The work involves identifying instances of visual objects belonging to specific classes that are of utmost importance to a pilot trainee when performing a flight task. This approach offers a more calculated method for assessing the mapping of features between the two datasets. We utilize different metrics to evaluate the effectiveness and success rate of feature extraction in two sets of sample data, as well as the accuracy of matching keypoints. We anticipate that this paper will serve as a foundation for further research in this field, or this marks the beginning of practical assessments of visual fidelity in relation to the specification of requirements.

ABOUT THE AUTHORS

Rishabh Kaushik is an Extended Reality (XR) Subject Matter Expert and a Principal Software Engineer at Collins Aerospace. His research domain is Image Generators (IGs) for Flight Simulation under Visual Systems. He actively participates in the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) as a member of the Emerging Concepts and Innovative Technologies Subcommittee. Recognized for his contributions to the Simulation and Training industry and the country, he has been granted the prestigious Outstanding Researcher status by United States Citizenship and Immigration Services. Furthermore, he serves on the leadership committee of Collins Aerospace - Mission Systems (Simulation & Training Solutions - STS) DevSecOps Community of Practice (CoP). With two master’s degrees from the University of Utah specializing in Game Engineering, Computer Vision, and Artificial Intelligence, he completed his bachelor’s degree at Birla Institute of Technology, Mesra, India in Computer Science. His passion lies in developing, programming, and maintaining large scale sustainable software architectures by utilizing innovative solutions that incorporate emerging and disruptive technologies.

Ankur Rathore serves as a Principal Software Engineer at Collins Aerospace, specifically in the Mission Systems, Visual Systems division. With over 8 years of experience, Ankur has made significant contributions to the field of Image Generator (IG) and Advanced Visualization. Recently, he has been involved in the development of the Arcus IG, which combines the capabilities of both Game Engine and Flight Simulation technologies. Prior to his research work, Ankur worked as a game developer, publishing games for various platforms such as PC, consoles, and mobile devices. He holds a master's degree in computer graphics, Visualization, and Game Engine Technologies. His primary research interests revolve around bipedal locomotion, kinematics, and heterogeneous computing.

Analyzing visual fidelity in flight simulation software using Game Engine with Feature Mapping

Rishabh Kaushik
Principal Engineer, Software Engineering
Salt Lake City, Utah
rishabh.kaushik@collins.com

Ankur Rathore
Principal Engineer, Software Engineering
Salt Lake City, Utah
ankur.rathore1@collins.com

INTRODUCTION

According to a report by MarketsandMarkets (Game-Based Learning Market Size, Share and Global Market Forecast to 2026), the market size for game-based learning is projected to grow from USD 11 billion in 2021 to USD 29.7 billion by the end of 2026, at a Compound Annual Growth Rate (CAGR) of 21.9%. Moreover, research (GlobalData, 2023) indicates that Virtual Reality (VR) has the potential to outperform conventional approaches by a staggering 400%, particularly in enhancing spatial and situational awareness, managing workloads, making informed decisions, and solving complex problems. This trend is supported by the research done by the Embry-Riddle Aeronautical University (Improved Pilot-Training Program Yields Promising Results, 2022), where they implemented virtual reality training to decrease the duration required for a cohort of 58 students to accomplish their initial solo flight by over 30%, as well as research done by the United States Air Force's (USAF's) T3 program (GlobalData, 2023), launched in 2020, where they developed VR technology for training military pilots that introduced flying lessons, combat readiness exercises, and digital training modules. The first course in July and August 2020 showed that despite similar assessment scores, the training was completed in 12.5 days, 54% faster than the usual 27 days. These sources provide insights into how game engines and Extended Reality (XR) technologies are being leveraged in the simulation and training industry, demonstrating a clear trend towards their increased use and adoption.

Background and Context

Flight simulation fidelity is defined as 'The level at which features of detectable conditions prompt suitable pilot psychomotor and cognitive reactions in a particular task and environment.' (Hess and Siwakosit 2001). Flight simulator fidelity can be broken down into Physical fidelity, Cognitive fidelity, and Functional fidelity. Physical fidelity is determined by how accurately a simulator replicates the physical aircraft's flight deck and overall feel (Allen et al, 1986). Cognitive fidelity pertains to the simulator training environment's capacity to accurately reproduce the cognitive abilities necessary on the aircraft's flight deck (Lee, 2009). Functional fidelity is a measure of how closely the simulator imitates the functionality of the actual equipment (Allen et al., 1986). Research exists for assessing the cognitive and functional fidelity of flight simulators through Pilot Models (Hess & Marchesi, 2009), and subjective pilot fidelity feedback and ratings. Additionally, cognitive, and functional fidelity have also been evaluated by techniques encompassing visual, proprioceptive, vestibular cues, inverse dynamic analysis, and a basic model using human operator visual memory, demonstrated through a rotorcraft repositioning task with varying visual and motion cues (Hess and Siwakosit 2001).

At the same time, the notable advancement of contemporary rendering technology like Game Engines such as Unreal and Unity, along with state-of-the-art Virtual Reality (VR) and Mixed Reality (MR) technologies, requires a framework to assist the flight simulation industry in evaluating the Physical Fidelity aspect of modern Image Generators (IGs) developed using these technologies. We recognize the importance of visual fidelity in influencing Physical fidelity as one of its key components and suggest an initial framework to identify the visual features that contribute to it. We also identify a developmental trend to improve an IG's self-efficacy or 'convincingness', the confidence in one's ability to effectively execute a task by believing in the simulated environment. Typically, a strong sense of self-efficacy can result in favorable performance results (Holbrook & Cennamo, 2014).

Importance of Visual Fidelity in Flight Simulation

Visual fidelity in flight simulation is of utmost importance as it enhances realism, improves training effectiveness, ensures safety, provides cost efficiency, aids in research and development, and influences public perception and marketing. The level of visual detail in flight simulation with accurate visual representations of landscapes, airports, weather conditions, and aircraft behavior are essential for pilot training. By practicing maneuvers, procedures, and emergency scenarios in a simulated environment that closely mirrors real-life conditions, pilots can improve their readiness and competence helping in training effectiveness. Flight simulators are not only used for training but also for testing new aircraft designs, procedures, and software. Accurate visual representations help in identifying potential safety issues or design flaws early in the development phase, reducing risks during actual flights. High visual fidelity ensures that training in a simulator translates well to real-world flying, reducing the need for extensive training in actual aircraft, which is much more expensive and resource intensive. Visual fidelity also plays a role in the public perception and marketing of flight simulators. Realistic visuals can attract more users, whether they are professional pilots, enthusiasts, or researchers, leading to wider trust and acceptance for simulator technologies.

Significance of Integrating Game Engine Technology into Flight Simulation Software

By incorporating game engine technology, flight simulation software can achieve an optimal user experience and immersion. This is done by enabling advanced graphics rendering with features like Physics Based Rendering (PBR) and dynamic lighting, optimizing for real-time simulation and physics accuracy, creating dynamic environments with realistic weather effects, supporting scalability across hardware configurations, facilitating intuitive user interfaces and interactive controls, and integrating VR and AR for enhanced immersion. These elements combine to deliver a more engaging and realistic flight experience for users. This results in a more compelling experience for pilots and trainees, improving the effectiveness of training scenarios by closely mimicking real-world conditions. For example, Game engines are being used to create dynamic environments and simulating various weather conditions. This allows for a more accurate representation of changing weather patterns, atmospheric effects, and environmental interactions like turbulence and cloud formations. Such realism is crucial for training pilots to handle diverse weather scenarios and emergencies.

The integration of game engine technology enables flight simulation software to be scalable and perform efficiently across different hardware setups. This flexibility allows for adjustments in detail and complexity based on the user's hardware capabilities, making simulations accessible to a broader audience ranging from casual users to professional training facilities.

Modern game engines come equipped with advanced physics engines that accurately simulate flight dynamics and aircraft behavior. Game engines have advanced to the point where they can perform various calculations that used to require explicit coding. This supports towards a realistic representation of aerodynamics, handling characteristics, and performance metrics for various aircraft models within the simulation. Pilots can practice maneuvers and procedures with precision, translating their training effectively to real-world flying.

Game engine-based flight simulators with XR integration (Figure 1) provide interactive features like interactive cockpits or video passthrough via masking to allow pilots to physically interact with controls that closely resemble the actual aircraft controls or may even be the authentic controls of the aircraft. This allows for a more immersive and engaging experience for pilots and trainees, further enhancing the overall training effectiveness and user experience.

Figure 1. A game engine-based flight simulator is shown, with the user wearing a Mixed Reality (MR) Headset in a physical cockpit (left) and interacting with the avionics system through video passthrough technology (right).

Purpose and Objectives of the Study

The primary goal of this research is to create an initial structure for assessing the visual accuracy element of flight simulation using computer vision-based algorithms like Feature Matching. This structure aims to facilitate future studies in examining the visual accuracy aspect of flight simulators, without relying on pilot feedback or various Image Generators. The study is centered around two main hypotheses. Firstly, there are specific unknown yet observable features that trainees or pilots employ when carrying out crucial tasks such as landing an aircraft in poor visibility conditions. Secondly, these features can be categorized into distinct object classes using computer vision algorithms like Feature Matching, with each task context necessitating its own unique set of these object classes. The aim of this study is to identify these classes and develop an initial approach to transform the subjective analysis of their visual characteristics into a distinct and deterministic method for evaluating and comparing the disparity between two images, establishing the ‘believability’ of a simulated image.

METHODOLOGY

The methodology consists of these steps: Data Preparation, Data Cleanup and Pre-processing, Feature Matching, Comparative Analysis Approach, and Inferencing. To illustrate this approach, we applied it to a case study involving a pilot landing a passenger plane in low visibility conditions. This was primarily adopted because landing under low visibility conditions is one of the most challenging tasks for pilots. Additionally, this use case helped us reduce the amount of data and streamline the focus on the points of interest (classes of objects). The following sequence explains the different steps in the methodology in order of importance:

Overview of Feature Matching

Feature matching is a crucial process in computer vision that involves identifying correspondences between distinctive points or regions (keypoints) detected in images. The main objective is to locate points in different images that represent the similiar physical point or region within the scene. The typical process of feature matching includes the following four steps:

- **Keypoint Detection:**

Keypoints are specific points or regions within an image that can be robustly detected and are distinctive. Examples include corners, edges, or blobs. Binary Robust Independent Elementary Features (BRIEF) is a fast method for feature descriptor calculation and matching. Common algorithms for keypoint detection include Speeded Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT), Oriented FAST and Rotated BRIEF (ORB), and Features from Accelerated Segment Test (FAST) (Feature Extraction and Image Processing for Computer Vision: Nixon, Mark, Aguado, 2024).

- **Feature Description:**

After detecting keypoints, each keypoint is described by a feature vector that captures its local appearance. This description should ideally be invariant to transformations such as rotation, scale, and changes in illumination. Descriptor algorithms like SIFT, SURF, ORB, and others compute feature vectors based on gradient information, image intensities, or binary patterns.

- **Matching Keypoints:**

Feature matching entails comparing feature descriptors of keypoints from two images to identify corresponding points. Matching is typically achieved by comparing the distance between feature vectors using metrics like Euclidean distance or Hamming distance.

- **Filtering Matches:**

Not all keypoints and their matches are reliable, as outliers can occur due to occlusions, noise, or changes in viewpoint. Techniques like RANSAC are commonly employed to filter out outliers and estimate the transformation model (e.g., homography or fundamental matrix) between images.

Comparative Analysis Approach

To illustrate our approach, we will explain how it can be used to assess the consistency of a real world and simulated landing approach. Our initial objective was to mentally deconstruct the process, following the approach adopted by real-world pilots. This process can be divided into four primary phases, with each phase encompassing a core task (refer to Table 1) that the pilot must successfully complete before progressing to the subsequent phase (as depicted in Figure 2). At Decision Height (DH), which is the altitude reached during an instrument approach, a crucial decision must be made by the pilot. They must choose whether to proceed with the landing approach or initiate a missed approach procedure by going around.

Table 1. The four phases along with their primary tasks and sets of observable features.

Sr. No.	Phase	Primary Task	Set of observable features
1	Approach	Approach the runway at the right positional formation	Precision systems: Instrument Landing System (ILS) + Microwave Landing System (MLS)
2	Landing Flare	Continue Approach or go around based on Decision Height	(Approach + Runway Threshold Identification + Runway Edge) Lights + Precision Systems
3	Touchdown	Continue Approach to land	(Threshold + PAPI + Runway Edge) Lights + Markings + Tire Skid Marks
4	Rollout	Monitor speed and alignment	(Runway End + Taxiway) Lights + Markings

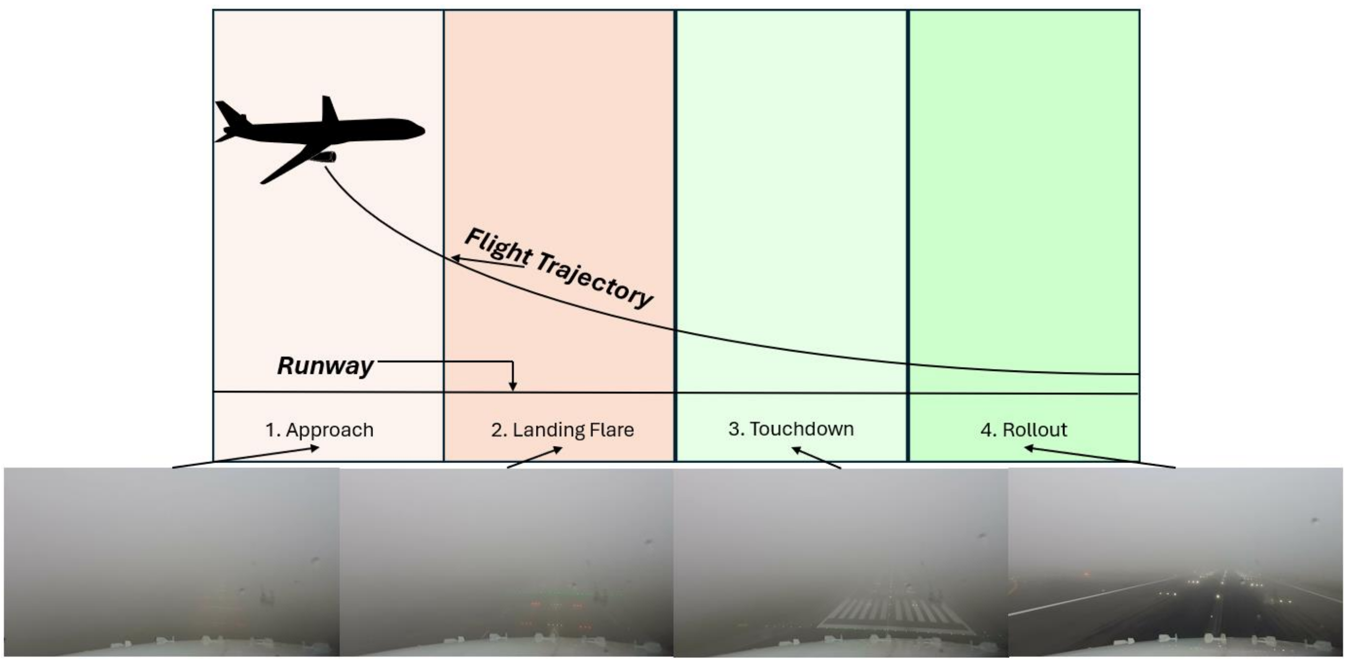

Figure 2. Four main phases of a Pilot's landing exercise along with real-world screenshots taken from the sample set that correspond to these different stages.

Our focus will be on the presence of various types of lights positioned along the runway as a collective entity known as 'runway lights'. We acknowledge that each of these runway light plays a distinct role in facilitating the landing procedure, however the classification of these individual lights would necessitate the utilization of an Object Detection technique based on Artificial Intelligence to assign metadata to the feature-mapped keypoints. Also, due to time and resource constraints, we focused solely on Feature Matching. A larger effort will involve creating a post-processed and cleaned dataset with relevant airport features, training and fine-tuning a transfer learned model for object detection, and developing a new mathematical model to integrate into our current pipeline.

Data Collection and Preparation

The initial batch of images utilized in this research were taken from an Adobe Stock video showcasing a passenger plane making its approach and landing on a runway amidst unfavorable weather conditions and dense fog. The subsequent batch of images were sourced from a company proprietary Game Engine-based Image Generator. We chose to utilize this contemporary IG instead of a conventional one due to its development on a Game Engine that made use of sophisticated Graphics cards, enabling us to conduct data capture and computer vision tasks alongside its rendering capabilities. Moreover, we view this as a universal approach that could be beneficial for systems that are not XR or proprietary. The batches of images fetched from the simulator were further divided into two distinct datasets by simulating landings at different locations: dataset 1 represented a military Air Force runway (Hill Air Force base - KHIF in Utah), while dataset 2 depicted a commercial Airport Runway (Salt Lake City International - KSLC in Utah).

To effectively implement feature mapping, multiple sets of images were collected from both sources, resulting in a substantial sample dataset consisting of 100 images. The configuration within the IG simulator software encompassed various elements such as runway lighting, environmental factors (such as low-visibility fog and volumetric clouds), and synthetic data (including 3D buildings, runway markings, and other physical objects). These settings were meticulously adjusted to closely resemble the real-world conditions observed in the actual landing video.

Implementation details of the comparative analysis

After retrieving the two sample datasets and preparing them for Feature Matching, we proceed to apply the algorithm on two corresponding sets of images. One set consists of real-world images while the other set contains simulator-based images (from one of the airport datasets) that visually resemble the real-world images. To analyze the data, we examine the Hamming distance between these two sets of data points. In computer vision, the Hamming distance is used to measure the similarity between two binary strings, which are sequences of 0s and 1s. This metric calculates the minimum number of substitutions needed to transform one binary string into another. When working on feature extraction tasks, features can be represented as binary descriptors, such as using binary feature descriptors like BRIEF or ORB. The Hamming distance plays a crucial role in aligning these binary descriptors across different images. Please note that the Hamming distance serves as our primary method for encoding the analytical data of the two samples into discrete and deterministic values for analysis. However, we recommend that future studies consider this as a foundation and expand upon it. In our research, we have implemented ORB with Brute Force matching as our algorithm.

Evaluation metrics for assessing feature extraction and matching accuracy

Feature Matching yields a substantial number of results post feature extraction. These results are arranged based on ascending Hamming distance, and the top 10 results are extracted for visualization purposes. Our accuracy metric involves a blend of mathematical and empirical evaluations. We meticulously noted the Hamming distance of the matched features and their respective object classes to showcase contextual and semantic precision. Feature Matching was utilized on collections of images pertaining to a primary task captured at various points during the pilot's execution of said task. The average Hamming distances were then calculated, and the data was graphed and examined across various stages.

RESULTS

Based on our study, we found that Feature Matching provided significant match outputs to the descriptors in any sample set of two images (Figure 3 and 4). The lines drawn between corresponding keypoints or feature points that have been successfully matched between the two images signify:

Correspondence between Points: Each line connects a pair of keypoints (or feature points) that are considered to be the same point in different images. This indicates that the algorithm has identified these keypoints as matching features between the two images. Different colors are used to distinguish between these matches, helping in visualizing the quality and distribution of matches.

Verification and Validation: By inspecting these lines, one can verify whether the feature matching algorithm is correctly identifying corresponding points across different views or frames. It provides a visual confirmation of the accuracy and reliability of the feature matching process. It helps us compare if the connected points are from the same classes of objects.

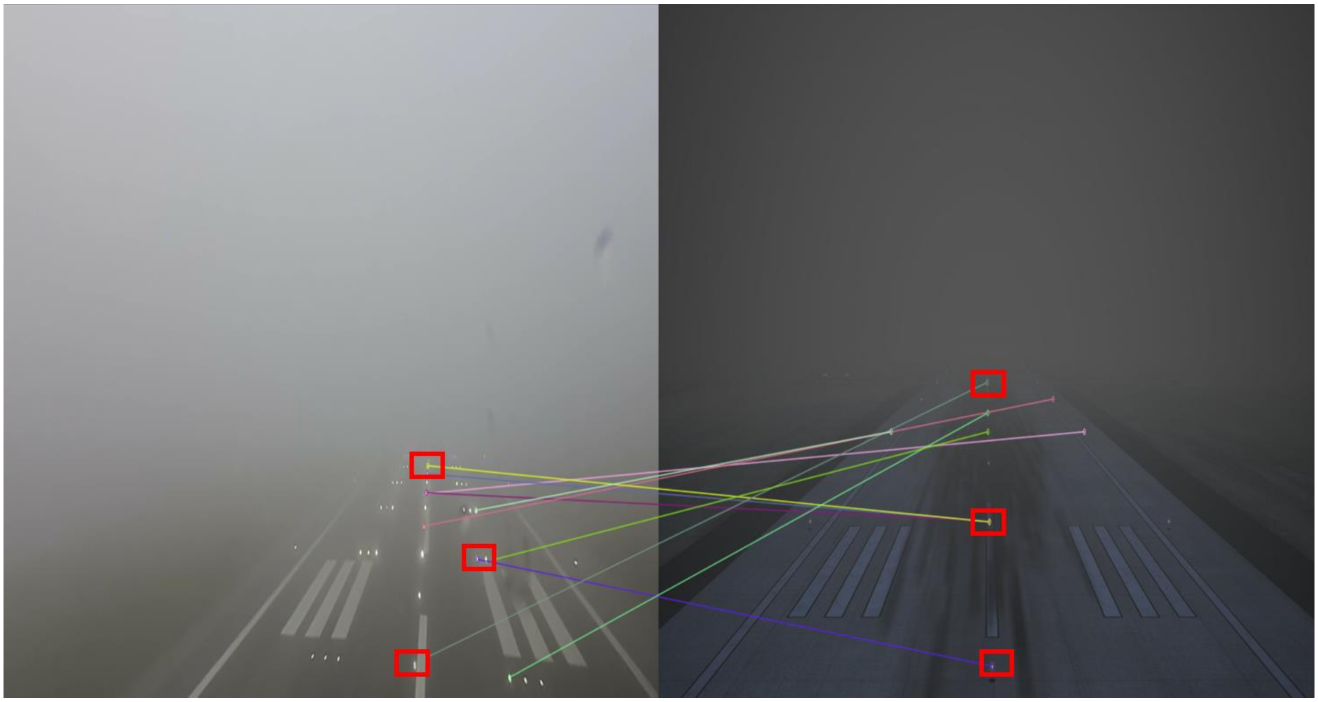

Figure 3. Feature Matching during Phase 3 (Touchdown) in low-visibility landing scenario between real-world (left) and game engine-based flight simulator (right) positioned at Hill Air Force military airport runway, Utah. Red box represents the class of runway light points.

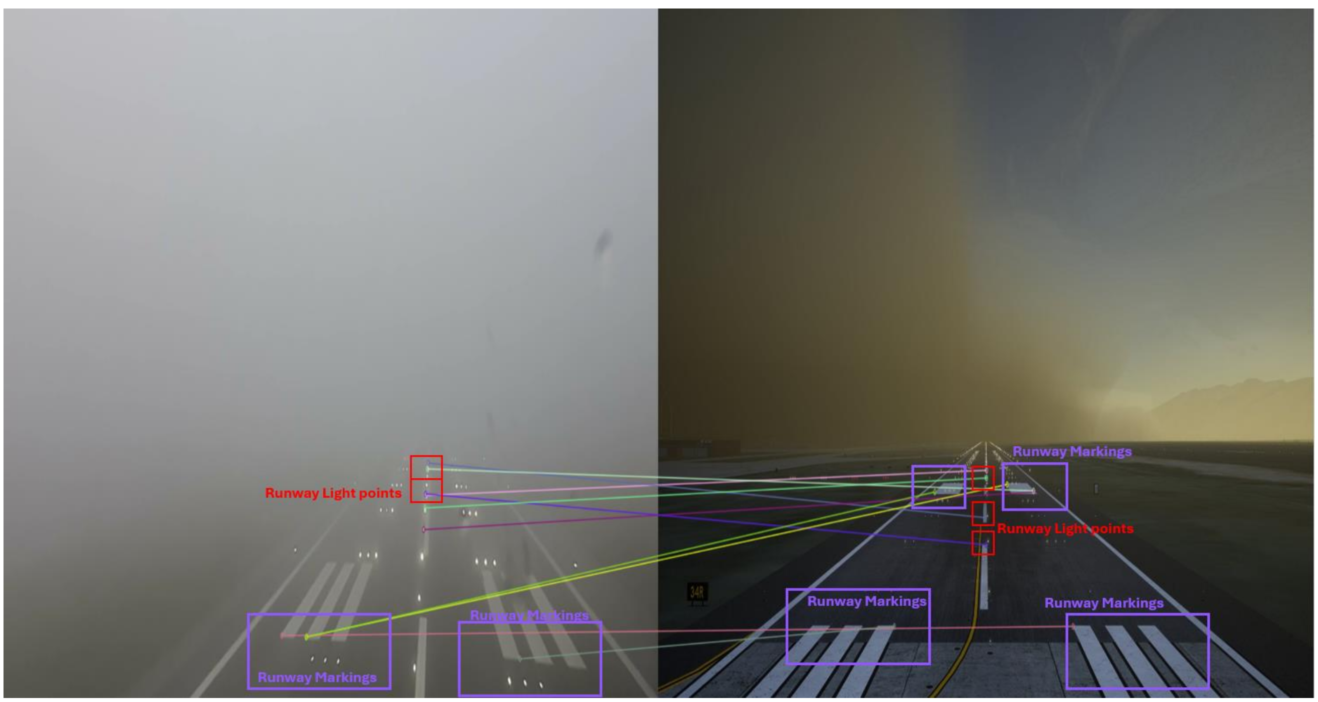

Figure 4. Feature Matching during Phase 3 (Touchdown) in low-visibility landing scenario between real-world (left) and game engine-based flight simulator (right) positioned at Salt Lake City commercial airport runway. Red box represents the class of runway light points and violet are runway markings.

Key Insights

Figure 5 illustrates the mean plot of the Hamming distance, depicting the comparison between the real-world sample set Feature Matched against the military runway (KHIF) – dataset 1 under Phase 3 (Touchdown) of the low visibility landing condition. The total count of feature matches discovered was 32, with the top 10 outcomes exhibiting Hamming Distances less than (<) 55.

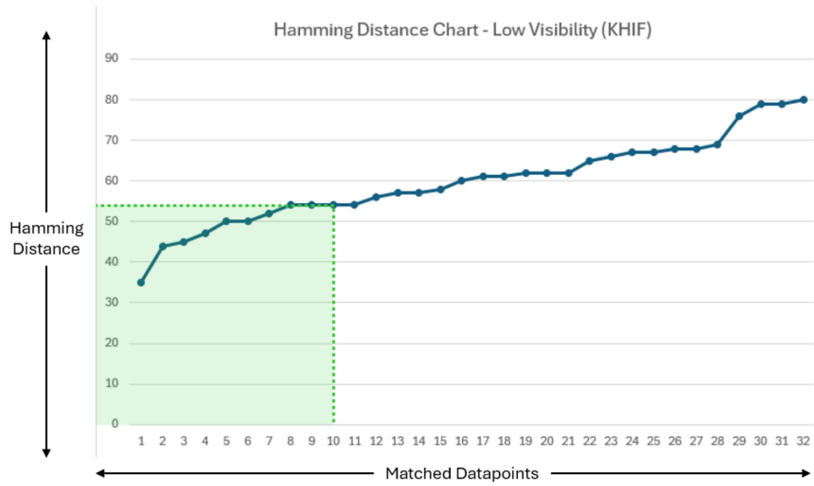

Figure 5. Chart showing the Hamming Distance of the matched datapoints for the average of the Hill Air Force base sample set during Phase 3.

In Figure 6, the average plot of the Hamming distance is depicted, representing the comparison between the real-world sample set Feature Matched and the commercial runway (KSLC) – dataset 2 under Phase 3 (Touchdown) of the low visibility landing condition. The total count of feature matches discovered was 84, with the top 10 results exhibiting Hamming Distances below (<) 40.

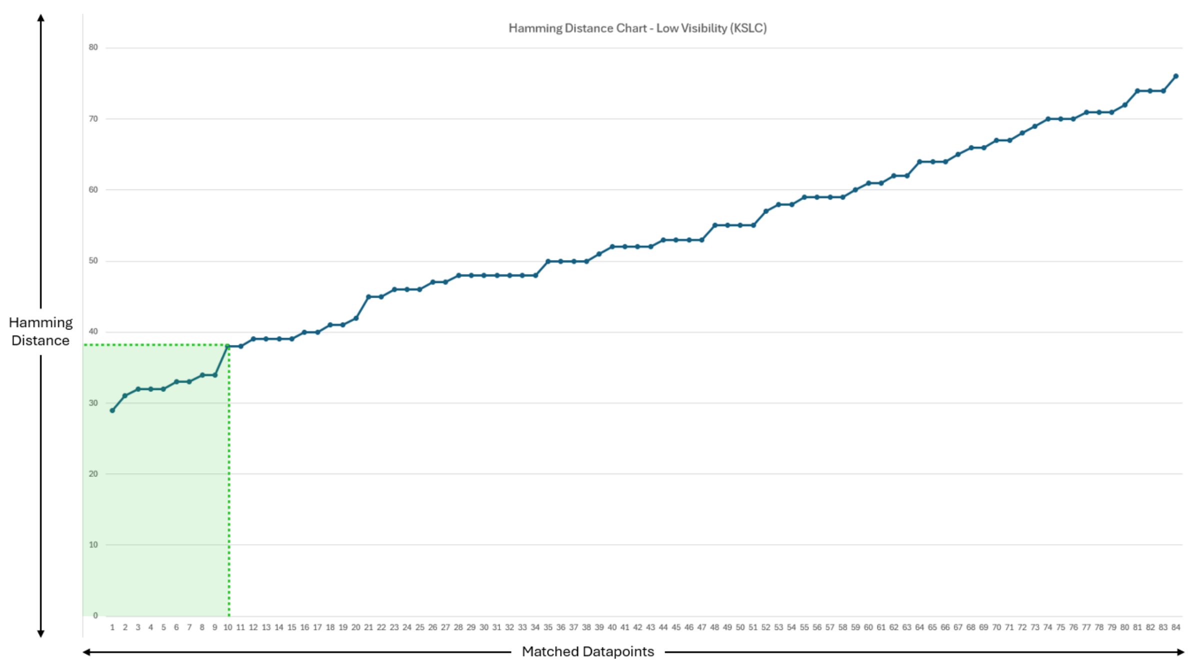

Figure 6. Chart showing the Hamming Distance of the matched datapoints for the average of the Salt Lake City sample set during Phase 3.

We were especially focused on the Touchdown (Phase 3) stage of the low-visibility landing as it represents the initial significant cognitive shift for the pilot to observe a broader range of visual cues.

Technique Effectiveness and Accuracy

Four key observations can be found using this framework:

- When comparing the original video of a passenger jet landing on a commercial civilian airport runway with a military runway dataset like the Hill Air Force base dataset (Figure 5) under low visibility conditions, we find that the number of matched datapoints is only about 38% of what is found when comparing the real-world dataset against a simulated commercial civilian airport runway like the Salt Lake City Airport (Figure 6). In Phase 3, the Salt Lake City runway exhibited a higher number of object classes mapped in the top 10 results of feature matching, including Runway light points and Runway markings, in contrast to the Hill Air Force runway which only had Runway light points being mapped. The integration of this technique with AI such as Object Detection could potentially expand these classes into more specific semantic categories for increased accuracy.
- To validate our two hypotheses, it is essential to first achieve a particular Hamming Distance threshold on the dataset being analyzed for training purposes, which varies depending on the primary task in a specific phase. For instance, when it comes to the Touchdown phase of an aircraft for a low visibility landing, the Hamming distance required should not exceed 40 for the best set of Feature Matched results. This is because the optimal supporting vector result was obtained with a Hamming distance of less than 40 between the two datasets - KHIF (less than 55) and KSLC (less than 40) under similar sets of conditions. The first hypothesis suggests that there are certain observable features that trainees or pilots rely on when performing critical tasks like landing a plane in low visibility conditions. The Feature matched results confirm this, showing that during the **Touchdown phase**, pilots typically focus on the visual fidelity of two important classes of objects: Runway markings and Runway Lights. The second hypothesis states that these features can be translated into distinct classes of objects using computer vision algorithms like Feature Matching, with each task context requiring its own set of these object classes. The line connecting the features between the two sets of images obtained through Feature Matching supports this.
- The optimal support vector for the main task in a particular stage outlined above validates the initial approach of converting the subjective assessment of visual features (class of objects) into a discrete and deterministic values (based on Hamming Distance thresholds) after evaluating and comparing the disparity between two image datasets, determining the ‘convincingness’ of simulated sequences of continuous images produced by a simulator.
- Observable features can be easily translated mathematically using binary descriptors across different images as part of Feature Matching, utilizing processes like ORB. Furthermore, analyzing the lines for the top 10% feature matched results allows us to assess the accuracy by ensuring that the two matched features belong to the same class of objects.
- There are specific subtle observable features that an experienced pilot may use as aids during the landing process, such as Tire Skid Marks on the runway. However, Feature Matching does not consider this class of objects and skips it entirely.

DISCUSSION

Implications for Flight Simulation and Training

This study will help in enhancing XR training efficacy, supporting visual fidelity analysis, assurance of quality and validation, creation and assessment of scenarios, and laying the groundwork for future visual fidelity-based research.

Accurate visual representation is crucial for pilot training in flight simulation, particularly for XR-based training. Inconsistencies in the visual objects used by pilots as aids during tasks can lead to virtual dissonance in VR. Virtual dissonance, in the VR context, refers to the discomfort or disconnect users feel when their sensory experiences in the virtual environment clash with their real-world expectations or sensory input. This can result in cybersickness, causing symptoms like nausea, headaches, and dizziness, impacting immersion, and limiting VR use duration. Failure to address this issue can exacerbate symptoms and create negative associations with VR. The proposed technique can help identify the visual class of objects that are relevant to avoid cognitive dissonance when performing a primary task, thereby heightening immersion and engagement during training sessions, fostering a more effective learning environment. By validating visual fidelity, trainers can guarantee that pilots are immersed in realistic environments that better prepare them for real flight conditions while managing cybersickness.

Utilizing this technique enables the comparison of real-world images or videos with simulated scenes produced by flight simulation software. Through the detection and matching of keypoints and descriptors between real and simulated scenes, developers and evaluators can objectively evaluate how accurately the simulation mirrors real-world visuals. This evaluation encompasses factors like terrain intricacy, landmark precision, texture authenticity, and lighting conditions.

This first pass approach offers a quantitative assessment of visual fidelity, aiding in the validation of simulation software by measuring how closely the rendered scenes resemble actual environments. This method can pinpoint disparities or inaccuracies in the simulation, leading to enhancements in terrain generation algorithms, texture mapping, and lighting models.

Flight simulators frequently replicate diverse scenarios and environments. This study can aid in crafting and evaluating these scenarios by confirming that the simulated environment convincingly compares to the intended real-world setting. Assessing scenarios with this technique assists in refining training scenarios to be more authentic and pertinent to real-world situations.

Based on the aforementioned factors, the initial method proposed carries significant implications for the validation and verification of certified flight visuals. We assert for specific environments the utilization of this research could prove beneficial for the Federal Aviation Authority (FAA) and other flight simulator certification entities in assessing Level D flight simulators in a less subjective manner and employing a more formal analysis methodology. Furthermore, as this approach is independent of pilots and expensive IG hardware, it offers a cost-efficient and reliable means of leveraging flight visuals to ensure visual accuracy. It is recognized that cognitive and behavioral fidelity research and techniques play a crucial role, hence we aim to incorporate a visual fidelity-based approach as another tool to the certification process for modern Game Engine based IGs, particularly those based on XR technology.

Limitations of the Study and Future Work

At the heart of this framework we have utilized Feature Matching and while it is a potent technique for evaluating visual fidelity in flight simulation software. It presents several potential limitations and challenges, some of which are shown in Table 2.

Table 2. Limitations of the Study

Limitation	Description
Computational Intensity	Feature Matching entails processing vast amounts of data, particularly when dealing with high-resolution images or video frames. The computation of matching keypoints and descriptors can be resource-intensive, which may restrict real-time applications or necessitate powerful hardware for efficient processing.
Sensitivity to Environmental Changes	The effectiveness of Feature Matching can be affected by variations in lighting conditions, weather effects, or other environmental changes. Disparities in illumination, shadows, or reflections can impact the reliability of keypoint detection and descriptor matching, resulting in mismatches or false positives.
Sensor and Image Quality	The quality of input images or video frames plays a crucial role in the accuracy of Feature Matching. Low-resolution images, noise, motion blur, or compression artifacts can degrade the performance of keypoint detection and descriptor matching, diminishing the reliability of the technique.
Geometric and Scale Variations	Feature Matching algorithms typically assume a certain level of geometric and scale invariance in keypoints. However, significant geometric distortions or scale differences between real-world scenes and simulated scenes can present challenges for precise matching.
Occlusions and Dynamic Environments	Real-world scenes often involve occlusions and dynamic elements such as moving objects and changing scenery. Feature Matching may encounter difficulties in handling occluded keypoints or matching descriptors across frames where scene content changes rapidly.

Manual Intervention for Validation	While Feature Matching provides automated matching results, validating the accuracy of matches often requires manual inspection and intervention. This process can be time-consuming.

In the future, further advancements in this field will involve integrating this pipeline with Object Detection and conducting real-time video analysis instead of processing images individually. This endeavor will necessitate extensive data generation and refinement, as well as the utilization of neural networks like YOLO (You Only Look Once) (Redmon et al., 2015) and EfficientDet (Tan et al., 2020) to ensure swift and precise analysis. Moreover, the incorporation of new algorithms into the framework will demand a novel mathematical structure capable of facilitating cross-variate analysis while effectively managing existing biases and any potential new biases.

CONCLUSION

Analyzing visual fidelity poses a challenge due to the vast and non-linear observable space. In contrast to behavioral fidelity, which allows us to access a participant's physiological state and create mathematical models for pilots, visual fidelity necessitates a more focused approach to tackle its complexity. Therefore, we opted to break down our methodology into task-based segments. We anticipate that this framework will inspire the industry to invest in research that are independent of users and Image Generators. Additionally, the integration of machine learning will bolster this framework by leveraging Feature Matching in conjunction with Feature Maps derived from neural networks like those utilized in semantic segmentation or object detection. Ultimately, we envision this study as a small contribution towards aiding the military, commercial aviation sectors, and the Federal Aviation Administration (FAA) in promoting innovative methods for supporting classification and certification of modern Game Engine based Flight simulators, as this technological evolution continues to advance with each iteration.

ACKNOWLEDGEMENTS

We would like to thank our simulation and training team in Salt Lake City and Orlando for their support and guidance. Special thanks to Dr. Angus L. M. Thom McLean, III, Senior Technical Fellow at Collins Aerospace for his support and insights.

REFERENCES

[1] Game-Based Learning Market Size, Share and Global Market Forecast to 2026 | MarketsandMarkets. (n.d.). Www.marketsandmarkets.com. https://www.marketsandmarkets.com/Market-Reports/game-based-learning-market-146337112.html

[2] GlobalData. (2023, August 9). AR and VR's role in aviation. Airforce Technology. https://www.airforce-technology.com/analyst-comment/ar-vr-role-aviation/?cf-view&cf-closed

[3] Improved Pilot-Training Program Yields Promising Results. (2022, January 21). Embry-Riddle Newsroom. https://news.erau.edu/headlines/improved-pilot-training-program-yields-promising-results

[4] Hess, Ronald & Marchesi, Federico. (2009). Analytical Assessment of Flight Simulator. Fidelity Using Pilot Models. Journal of Guidance, Control, and Dynamics. 32. 760-770. 10.2514/1.40645.

[5] Hess, Ronald & Siwakosit, W. (2001). Assessment of Flight Simulator Fidelity in Multiaxis Tasks Including Visual Cue Quality. Journal of Aircraft. 38. 607-614. 10.2514/2.2836.

[6] Allen, J. A., Hays, R. T., & Buffardi, L. C. (1986). Maintenance training simulator fidelity and individual differences in transfer of training. Human Factors, 28(5), 497–509.

[7] Lee, A.T., (2009). Flight simulation virtual environments in aviation. Burlington, VT: Ashgate.

[8] Holbrook, H. A., & Cennamo, K. S. (2014). Effects of high-fidelity virtual training simulators on learners' self-efficacy. International Journal of Gaming and Computer-Mediated Simulations (IJGCMS), 6(2), 38-52. doi:10.4018/ijgcms.2014040104

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015, June 8). You Only Look Once: Unified, Real-Time Object Detection. ArXiv.org. https://arxiv.org/abs/1506.02640

[10] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. ArXiv:1911.09070 [Cs, Eess]. https://arxiv.org/abs/1911.09070

[11] Feature Extraction and Image Processing for Computer Vision: Nixon, Mark, Aguado, Alberto: 9780128149768: Amazon.com: Books. (2024). Amazon.com. https://www.amazon.com/Feature-Extraction-Processing-Computer-Vision/dp/0128149760