SIFT-driven Vision-based Positioning for UAVs: Overcoming GPS Signal Challenges in Urban Environments
İHA'lar için SIFT Güdümlü Görüş Tabanlı Konumlandırma: Kentsel Ortamlarda GPS Sinyal Zorluklarının Aşılması

Emre ŞATIR[1*], Elif DEMİRTAŞ[1], Halim AĞDEMİR[1], Fatma YILDIZ[1]
[1]Department of Computer Engineering, Izmir Katip Celebi University, Izmir, Turkey
ORCIDs: 0000-0002-1950-5549, 0009-0006-0978-0287, 0009-0003-8889-1089, 0009-0004-2605-6456
E-mails: emre.satir@ikcu.edu.tr, elifdemirta00@gmail.com, agdemirhalim4@gmail.com, fatma200320@gmail.com
*Corresponding author.

***Abstract*—Unmanned Aerial Vehicles (UAVs) are increasingly utilized in various industries for tasks ranging from surveillance to logistics. However, GPS signal loss in dense urban environments poses significant challenges to the safe and accurate navigation of UAVs. This paper proposes a vision-based position estimation system that leverages the Scale-Invariant Feature Transform (SIFT) algorithm to provide robust UAV navigation in GPS-denied environments. By detecting keypoints in high-resolution images captured by the UAV's camera and matching these points across successive frames, the system calculates displacement and converts pixel-based movements into real-world metrics. Experimental results show that the proposed method provides a reliable alternative to GPS-based navigation with an average error rate of %6.09. The system's real-time processing and adaptability to complex urban environments make it a viable tool for enhancing UAV navigation in GPS-compromised scenarios.**

***Keywords*—UAV navigation, vision-based positioning, SIFT algorithm, keypoint detection, real-time position estimation**

***Özetçe*—İnsansız Hava Araçları (İHA'lar), gözetlemeden lojistiğe kadar çeşitli sektörlerde giderek daha fazla kullanılmaktadır. Ancak, yoğun kentsel alanlarda GPS sinyal kaybı, İHA'ların güvenli ve hassas bir şekilde gezinmesine önemli zorluklar getirmektedir. Bu makale, GPS olmayan ortamlarda İHA navigasyonunu sağlamak için Ölçek Değişmez Özellik Dönüşümü (SIFT) algoritmasını kullanan görsel tabanlı bir konum tahmin sistemi önermektedir. İHA'nın kamerası tarafından yakalanan yüksek çözünürlüklü görüntülerdeki kilit noktaları tespit ederek ve bu noktaları ardışık karelerde eşleştirerek, sistem yer değiştirmeyi hesaplar ve piksel tabanlı hareketleri gerçek dünya metriklerine dönüştürür. Deneysel sonuçlar, önerilen yöntemin ortalama %6.09 hata oranı ile GPS tabanlı navigasyona güvenilir bir alternatif sunduğunu göstermektedir. Sistemin gerçek zamanlı işlem kapasitesi ve karmaşık kentsel ortamlara uyum yeteneği, GPS kısıtlı senaryolarda İHA navigasyonunu geliştirmek için uygun bir araç haline getirmektedir.**

Anahtar Kelimeler*—İHA navigasyonu, görüş tabanlı konumlandırma, SIFT algoritması, anahtar nokta tespiti, gerçek zamanlı konum tahmini

I. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs) have become crucial assets in various industries due to rapid technological advancements. The application of UAVs in sectors such as defense, agriculture, logistics, mapping, and urban transportation underscores their potential to significantly enhance operational efficiency and safety. Particularly in the context of smart cities and urban air mobility projects, the ability of UAVs to navigate safely has become increasingly important. However, for successful operations, UAVs require precise positioning capabilities. Although GPS (Global Positioning System) is the most widely used positioning technology today [1], its reliability can be compromised in urban environments where signals are weak or obstructed, negatively affecting UAV navigation [2]. In cases where GPS signals are lost, UAVs encounter substantial challenges in accurately estimating their position, thus increasing safety risks [3]. This situation highlights the need for alternative position estimation methods that reduce UAVs' reliance on GPS.

This study proposes a vision-based system to enable UAVs to estimate their position during navigation when GPS signals are weak or unavailable. The system utilizes the Scale-Invariant Feature Transform (SIFT) algorithm [4] to detect changes in the UAV's position by analyzing visual information from the surrounding environment. The SIFT algorithm is a robust keypoint detection algorithm capable of recognizing and matching objects in an image despite variations in scale, rotation, and lighting. By offering a position estimation method that does not depend on GPS, especially in dynamic and complex urban environments, this system enhances UAVs' safe navigation capabilities.

The proposed method relies on detecting keypoints in high-resolution images captured by UAVs during flight and matching these keypoints across successive images [5]. UAVs calculate displacement by processing images captured at regular intervals (e.g., every 100 milliseconds) while maintaining a fixed altitude. This enables real-time data processing. The SIFT algorithm identifies keypoints in each image and tracks changes in the x and y coordinates of these points across successive images, converting these shifts into metric values. Consequently, the UAV's horizontal and vertical displacements along its flight path are transformed from pixel units into real-world measurements (meters), enabling accurate position estimation [6]. To achieve precise displacement calculations, fixed objects in the images are used to determine a pixel-to-meter conversion ratio.

The system is developed using Python programming language [7] and the Open-Source Computer Vision Library (OpenCV) [8]. Python was chosen for its extensive library support and powerful capabilities in image processing. Its compatibility with OpenCV and other scientific computing libraries makes Python an ideal choice for this type of application [9]. OpenCV, an open-source library for computer vision and machine learning, is well-regarded for its high performance, broad community support, and multi-language compatibility. In this study, OpenCV was used to implement the SIFT algorithm for detecting keypoints and processing image data in real time.

This study makes the following contributions to the field of UAV positioning:

- *An Alternative Position Estimation Method that Reduces GPS Dependency:* This work introduces a reliable, image-based position estimation system for UAVs that can function effectively when GPS signals are weak or unavailable, contributing to the literature on UAV positioning techniques.
- *Application of the SIFT Algorithm in Dynamic Environments:* The use of the SIFT algorithm to fulfill real-time positioning needs with high accuracy and reliability presents a new approach to utilizing this algorithm in GPS-denied environments, addressing a gap in current research.
- *Real-Time Navigation Enhancement in Urban Areas:* The development of a vision-based positioning system designed to enable safe navigation of UAVs, especially in dense urban environments, provides significant advancements in the development of autonomous UAV systems with real-time data processing capabilities.

This paper is organized as follows: After the introduction, Section 2 discusses related studies on the subject. We propose our method in Section 3 and describe our experimental setup in Section 4. Finally, Section 5 talks about conclusions and future work.

II. Related Work

Research on UAVs has shown significant improvements in position estimation and motion tracking using vision-based algorithms [10]. In this context, the SIFT algorithm, developed by Lowe in 2004 [4], represented a breakthrough in image processing by providing scale and rotation invariance in the detection of key points in images. SIFT holds a prominent place in literature due to its high accuracy and robustness, particularly in object recognition and tracking systems. In this study, the capabilities of the SIFT algorithm are utilized to achieve vision-based position estimation for UAVs.

The paper published by Mikolajczyk and Schmid in 2005 is a significant study that comprehensively evaluates the performance of local feature descriptors in the field of computer vision [11]. This study presents a comparative analysis of local feature descriptors commonly used in various image processing and object recognition tasks. The authors examine SIFT and other local descriptors frequently employed in recognition and matching problems. By evaluating the performance of different descriptors under various distortions, such as scale, rotation, viewpoint, blur, and noise, they analyze which methods are more effective under specific conditions. It is understood that the SIFT algorithm provides high accuracy and reliability in complex environments like urban areas.

SURF (Speeded-Up Robust Features) is a local feature descriptor and detector algorithm developed by Herbert Bay, Tinne Tuytelaars, and Luc Van Gool in 2006 [12]. This algorithm was designed to operate faster than the previously developed SIFT algorithm and is used in tasks such as image matching and object recognition. Despite SURF's speed advantage, SIFT was selected for this project because it provides several advantages: its ability to provide more accurate and reliable matching results under challenging conditions (e.g., very complex backgrounds, significant rotation, or scaling differences); its capacity to detect and match features better in noisy or low-contrast images; its robust performance against rotation, scale, and perspective changes.

ORB (Oriented FAST and Rotated BRIEF) is an algorithm designed to operate more quickly, making it particularly suitable for real-time applications [13]. Like SIFT, ORB is invariant to scale and rotation, but it has lower computational complexity. ORB runs faster and is more computationally efficient than SIFT, but in terms of accuracy, it may not always perform as well as SIFT. ORB's performance tends to be weaker, especially in low-contrast or noisy images, compared to SIFT.

The relevant paper published in 2011 [14] addresses the challenges of GPS signals in urban environments and their impact on aviation. This study explores the difficulties faced by GNSS (Global Navigation Satellite System) signals in urban environments, particularly in areas known as "urban canyons"—areas with tall buildings. The paper introduces a new positioning technique called "shadow matching" to mitigate these challenges. Shadow matching is a technique that uses the shadows cast by buildings to improve the accuracy of positioning in urban areas. By comparing the predicted shadows from 3D city models with the actual shadows detected by the GNSS receiver, shadow matching helps to determine the most likely position of the receiver when traditional GPS signals are weak or blocked. This technique is particularly useful in dense urban areas, where signal obstructions due to buildings can cause significant inaccuracies in GNSS-based positioning.

The relevant research conducted in 2014 [15] is a study in the field of SLAM (Simultaneous Localization and Mapping). SLAM is a process in robotics and computer vision where an agent (e.g., a robot or camera) determines its own location while simultaneously creating a map of its environment in an unknown setting. This study proposes a system capable of real-time 3D mapping and localization using a single camera in large-scale environments. It has significant applications in fields such as autonomous vehicles, augmented reality, and robotics.

The relevant study conducted in 2016 [16] focuses on the simulation-based evaluation of object tracking and positioning systems for UAVs. This study presents a benchmark and simulation system for object tracking and monitoring systems designed for UAVs, with a specific focus on flight path tracking. The primary goal of this study is to provide a standardized platform to evaluate and enhance the real-time object and target tracking capabilities of UAVs.

The study titled "Autonomous Vision-based UAV Landing with Collision Avoidance using Deep Learning," published in 2021 by Liao et al. [17], presents a novel approach that enables UAVs to autonomously land in challenging, obstacle-rich environments using advanced deep learning techniques and vision-based navigation. Unlike traditional methods that rely heavily on GPS or predefined markers for landing, this approach utilizes real-time visual data processed by YOLOv4 to identify safe landing zones and avoid potential obstacles during descent. The integration of deep learning models with onboard sensors allows the UAV to dynamically adapt to unexpected obstructions and rapidly changing terrain features, enhancing its reliability in complex environments such as forests, urban areas, and uneven landscapes. This work is significant as it demonstrates the potential of deep learning to improve autonomous UAV operations, particularly in applications where conventional landing systems may struggle due to poor visibility or high obstacle density.

While numerous studies have explored vision-based position estimation and UAV navigation, most rely on alternative feature detection algorithms such as SURF, ORB or integrate additional sensors like LIDAR to enhance accuracy. Unlike these approaches, this study specifically focuses on utilizing the SIFT algorithm due to its superior performance in complex and dynamic environments, such as urban areas, where scale, rotation, and lighting conditions frequently change. Furthermore, unlike many existing methods that require extensive computational resources or rely on multi-sensor fusion, the proposed system offers a streamlined, camera-only solution for real-time position estimation without GPS dependency. This distinction positions our method as a lightweight, adaptable, and cost-effective alternative, especially suited for real-time applications in GPS-compromised environments

III. Methodology

In this study, a position estimation algorithm is introduced to determine the estimated location in the event of GPS failures caused by environmental factors (such as tall buildings, dense forests, etc.) or atmospheric disturbances (such as heavy rain, wind, etc.), which can render GPS inoperative. The position changes of UAVs were calculated using a method that converts pixel dimensions from images into real-world measurements without relying on GPS data. This method enables precise tracking of the UAV's position by performing calculations based on images captured by the camera and the known sizes of objects. Thus, even in the case of GPS failures, position estimation can be achieved, providing greater accuracy in autonomous systems.

To avoid errors due to altitude changes when converting pixels to real-world dimensions, objects with standardized sizes worldwide (such as manhole covers or sedan cars) were used. These objects were used as references for unit conversions and point matching of objects in successive images was performed using SIFT. By averaging the changes in the distances of the points from the left and top edges of the frame, the direction and distance the UAV has traveled (in meters) were precisely calculated. This method allows the UAV to accurately determine its position without the use of GPS.

The method used in this study is based on detecting keypoints in high-resolution images obtained by UAVs during flight and matching these points across successive images. UAVs calculate displacement using images captured every 250 milliseconds at a fixed altitude and process this data in real time. The SIFT algorithm detects keypoints in the images and calculates the changes in the x and y coordinates of these points between each image pair, converting these changes into metric values. Thus, the horizontal and vertical displacements made by the UAV along its flight path are transformed from pixel units into real-world measurements (meters), allowing for accurate position estimation. To accurately calculate the displacement, scaling is performed using fixed objects in the images, and a pixel-to-meter conversion ratio is obtained. The data used during testing was obtained from a high-resolution video captured by a drone flying at a constant altitude. Every 250 milliseconds, a frame from the video was extracted and converted into still images. The extracted data was then transferred to a computer for displacement analysis. The computer, functioning as a ground control station, runs algorithms developed in Python using the OpenCV library to process the data. Figure 1 shows four screenshots taken with a period of 250 milliseconds.

The SIFT algorithm was used to calculate the pixel displacement and direction of the UAV between successive images. With SIFT, the keypoints in each image and their corresponding descriptors were extracted. The descriptors were matched between the images using the BFMatcher (Brute-Force Matcher) algorithm. BFMatcher is a feature-matching algorithm commonly used in image processing and computer vision, such as in libraries like OpenCV [18]. The matches obtained were ranked based on the distance between the descriptors to ensure the best matches between the images. An example illustrating this process is shown in Figure 2. For each match, the changes in the x and y coordinates of the keypoints were calculated, and the averages of these changes were taken.

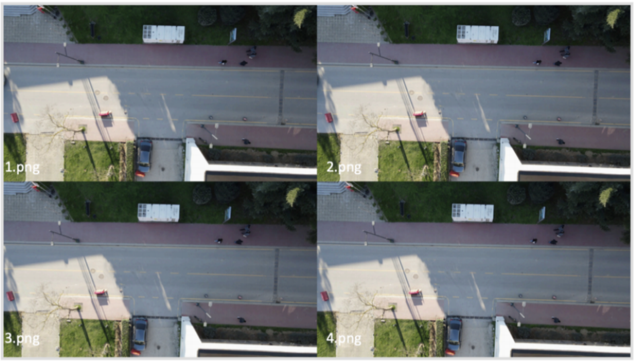

Figure 1: Four screenshots taken with a period of 250 milliseconds.

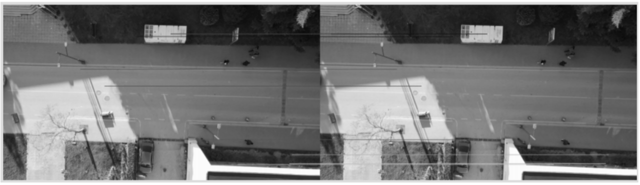

Figure 2: Matching of keypoints detected by the SIFT algorithm in consecutive images.

One of the most critical steps in this study is converting the pixel units of the images into real-world dimensions. Therefore, a pixel size measurement is performed on the reference image. The user is asked to mark two distant edges of a known object in the image. A line is created between these two points. The "matplotlib" library [19] is used to create the line. During this process, the x and y coordinates of the mouse clicks are added to a list called "points." This list stores the coordinates of the selected points. By using the coordinates of the two selected points, the Pythagorean theorem is applied. The modeling of the Pythagorean theorem in this process is shown in Figure 3.

A scale ratio is created by comparing the reference length of the selected object with its real-world size (in meters). This scale ratio represents the real-world length per pixel (meters/pixel). The average coordinate changes between two images, calculated using the SIFT algorithm, are converted from pixels to meters using the mentioned scale ratio. This conversion is essential for translating pixel-based movements into real-world measurements, ensuring accurate positioning. After calculating the displacements for each pair of successive images, these displacements are cumulatively summed. In this way, the total displacement of the UAV relative to its initial reference point can be calculated over the entire flight duration. The cumulative displacements can be tracked separately in both horizontal (x) and vertical (y) directions.

When the UAV moves in a certain direction, stationary objects (such as the ground or buildings) appear to shift in the opposite direction from the camera's perspective. This is

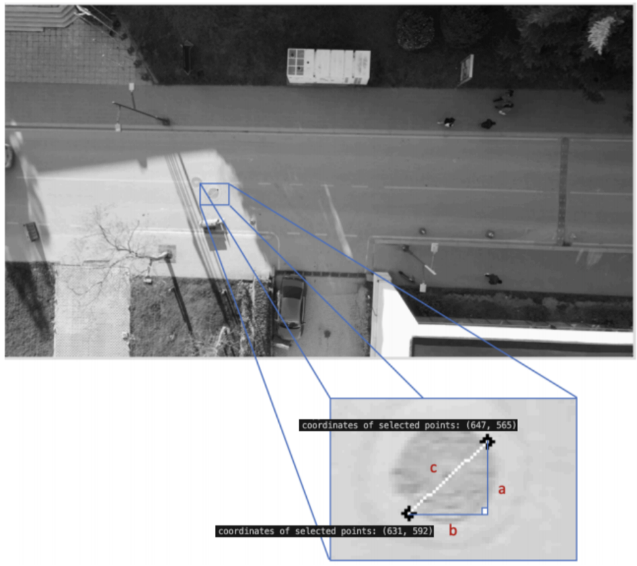

Figure 3: Modeling the Pythagorean theorem on the work.

a effect caused by the change in the position of objects in the camera's field of view due to the movement of the camera. To correct for this, the calculated x and y displacements are multiplied by -1 to invert the direction of the vectors. Multiplying the x and y displacements by -1 corrects the direction of motion, ensuring that the UAV's actual movement is accurately represented.

Figure 4 shows the basic steps of the approach used in the study.

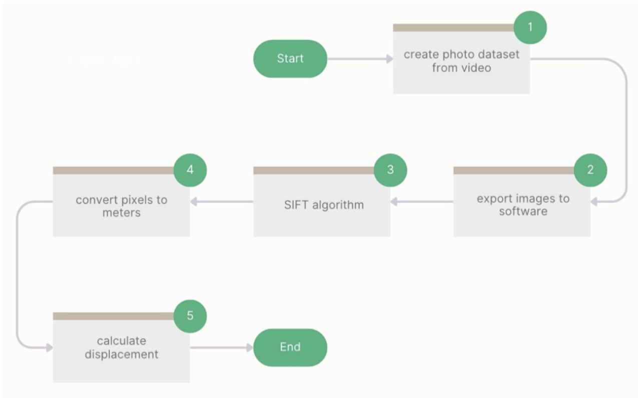

Figure 4: The basic steps of the study.

IV. Experimental Results

To test the accuracy and reliability of the proposed method, a specific experimental procedure was carried out. In this experiment, a standard credit card (8.56 cm x 5.4 cm, or 0.0856 m x 0.054 m), an A4-sized paper (21.0 × 29.7 cm, or 0.21 m × 0.297 m), and a 50 Turkish kurus coin (2.385 cm, or 0.02385

m) were used as references. These objects were chosen for their standardized and easily recognizable dimensions. During the testing process, a phone camera was moved along a single axis of 23 cm (0.23 m) at a constant altitude, capturing images from various heights. These recordings were processed through the steps outlined in Section 3 and provided as input to the algorithm.

The experiment was repeated for three different conditions. The values recorded by the software were measured as 0.2313 m, 0.2456 m, and 0.2049 m, respectively. In this case, it was determined that the distance measured by the software deviated from the actual value by an average of %6.09. This discrepancy provides important data for evaluating the software's precision and accuracy.

As shown in Figure 5, the experimental setup and the reference object used in the first condition are provided as an example. This visual illustrates the height at which the phone camera was held steady and the placement of the reference object. The length of the reference object was used during the software's analysis steps to scale the measurements to real-world dimensions.

Figure 5: Reference object and viewing window with standard credit card dimensions (8.56 cm x 5.4 cm).

Coordinates of the selected points, calculated pixel length and terminal input are shown in Table I. Table II, Table III and Table IV lists the results of the three experiments performed.

Description	**Value**
Coordinates of the selected points (first)	442,189
Coordinates of the selected points (second)	539,191
Length of the reference object in pixels	97.02061636580135
The real-world size of the reference object in meters	0.0856
Real-world size per pixel (meter/pixel)	0.0008822867057168379

Table I: Reference object measurements and calculations.

Table II, Table III and Table IV presents the analysis of where the position changes (Δx and Δy) between two images are calculated in meters. These analyses demonstrate the software's performance in accurately detecting position changes during movement. The variations in Δx are caused by slight hand tremors that affected the camera during manual sliding. Δy, on the other hand, is the main variable tested. That is, Δy represents the intended axis of motion in the experiments,

Image Pair	**Δx (meters)**	**Δy (meters)**
1.jpg - 2.jpg	0.0000000	-0.0000001
2.jpg - 3.jpg	0.04335000	0.00000338
3.jpg - 4.jpg	-0.07655000	0.04753000
4.jpg - 5.jpg	0.01515000	0.06018000
5.jpg - 6.jpg	0.04219000	0.0000823
6.jpg - 7.jpg	-0.08025000	0.00003800
7.jpg - 8.jpg	0.03728000	0.01489000
8.jpg - 9.jpg	-0.06756000	0.00001423
133.jpg - 134.jpg	0.00000982	-0.00004542
134.jpg - 135.jpg	0.02126000	-0.08061000
135.jpg - 136.jpg	-0.05475000	0.0003250
136.jpg - 137.jpg	-0.0006180	0.0202000
Total Position Change	**0.00314971**	**-0.23134907**

Table II: Position changes for Experiment 1 across Δx and Δy values.

Image Pair	**Δx (meters)**	**Δy (meters)**
1.jpg - 2.jpg	0.00001120	-0.00012233
2.jpg - 3.jpg	0.00009129	-0.00007561
3.jpg - 4.jpg	0.00009193	0.00027771
4.jpg - 5.jpg	-0.00006544	0.00192886
5.jpg - 6.jpg	0.00006639	0.00183458
6.jpg - 7.jpg	0.00002087	0.00171599
7.jpg - 8.jpg	0.00002297	0.00266292
8.jpg - 9.jpg	0.00004034	0.00234020
133.jpg - 134.jpg	0.00007454	0.00204640
134.jpg - 135.jpg	-0.00048240	0.00037967
135.jpg - 136.jpg	0.00027767	0.00185112
136.jpg - 137.jpg	0.00008044	0.0008044
Total Position Change	**0.09457952**	**0.24562819**

Table III: Position changes for Experiment 2 across Δx and Δy values.

while Δx captures unintentional horizontal deviations. In the first and third experiments, the camera was moved in the -y direction, while in the second experiment, it was moved in the +y direction. Although the target displacement was 0.23 m, the software measured 0.2313 m, 0.2456 m, and 0.2049 m, respectively, providing insight into the algorithm's accuracy. Based on these results, it was determined that the system deviated from the intended displacement by an average of %6.09. The error percentages for each measurement are presented in Table V.

The obtained error rate demonstrates that the method pro-

Image Pair	**Δx (meters)**	**Δy (meters)**
1.jpg - 2.jpg	-0.00000049	0.00000499
2.jpg - 3.jpg	-0.00001211	-0.00002376
3.jpg - 4.jpg	0.00014920	-0.00000950
4.jpg - 5.jpg	-0.00008253	-0.00174793
5.jpg - 6.jpg	-0.00100477	-0.01007545
6.jpg - 7.jpg	-0.00003070	-0.00009279
7.jpg - 8.jpg	-0.00002253	-0.00147851
8.jpg - 9.jpg	0.00007393	-0.00187626
133.jpg - 134.jpg	-0.00007373	-0.0007373
134.jpg - 135.jpg	-0.00021142	-0.0003851
135.jpg - 136.jpg	-0.00020553	-0.0003838
136.jpg - 137.jpg	0.00005348	-0.0003524
Total Position Change	**-0.00322631**	**-0.2049870**

Table IV: Position changes for Experiment 3 across Δx and Δy values.

Experiment	Error Percentage
1	0.57
2	6.78
3	10.91

Table V: Error percentages for experiments.

vides results close to the reference value, proving its accuracy. It should also be considered that the error rates may be influenced by factors such as the user marking the points, the person measuring the experimental setup, the display screen, and the sensitivity of the SIFT algorithm.

V. Conclusion And Future Work

This study proposes a system that enables UAVs to navigate safely not only when GPS is disabled but also in conditions where GPS is unavailable or unreliable. The system allows UAVs to continuously analyze visual data from their surroundings while flying at a constant altitude to calculate displacements. This ensures that even in scenarios where GPS signals are disrupted, such as in densely urbanized areas, around tall buildings, in heavy traffic, or under adverse atmospheric conditions, the system can maintain safe navigation for UAVs.

The proposed method is designed to be integrated into real-time applications. By continuously analyzing the recorded images through a camera system integrated into the UAV, position changes can be calculated instantly. This approach enables continuous monitoring of the vehicle's motion or flight path. In cases where GPS fails or becomes unreliable, displacement estimation can be performed without relying on such external sources. The information obtained from continuously processing camera data can be effectively used in autonomous systems, safety and navigation systems, and even in urban transportation vehicles. Thanks to its real-time computation and data processing capabilities, this system can track and guide the route of any vehicle in real-time. The real-time processing of camera data ensures immediate feedback, enhancing the responsiveness of autonomous systems in dynamic environments.

The proposed position estimation algorithm has the potential to serve as an effective alternative in the event of GPS failure. However, some challenges were encountered during the study. The proposed system provided accurate results by utilizing reference objects during navigation. However, despite the assumption that the dimensions of the reference objects are standardized, they can vary in the real world. This can negatively affect the accuracy of the position estimation. In other words, uncertainties in the dimensions of the reference objects can impact the reliability of the results.

Maintaining a constant altitude plays a crucial role in the performance of the algorithm. The algorithm is based on the relationship between the real-world size and the pixel length determined in the given image. Any change in the vehicle's altitude disturbs this ratio, reducing the accuracy of the results. In short, if the meters-per-pixel conversion ratio does not remain constant, the performance of the algorithm is negatively affected.

It has been observed that the algorithm performs successfully when the UAV moves forward, backward, left, and right. However, errors in position estimation can occur when the vehicle maneuvers around its axes. When the vehicle rotates in place without changing its position, the keypoints detected by the SIFT algorithm show variations. This variation leads to errors in position estimation. Particularly when the camera's field of view remains at 90 degrees, pixel conversion can be calculated accurately, but if the vehicle performs yaw, pitch, or roll movements, incorrect results may be obtained. Therefore, it is critical to account for the effects of camera movements and implement corrective measures to achieve more accurate position estimations.

It should be noted that the SIFT algorithm used in the proposed system may struggle to detect keypoints in complex or low-quality images, which in turn affects the accuracy of the matching process and, consequently, the estimated position. Therefore, the quality of the captured images is a critical factor that directly impacts the performance of the system.

Some future improvements are being considered for the proposed system. One of these is to prevent the negative impact of altitude changes on the algorithm by integrating the vehicle's altitude data into the code and updating the pixel conversion ratio in real-time. In this way, pixel conversions can be calculated more accurately by taking altitude changes into account, thus improving the overall performance of the algorithm. This integration will enable more reliable results in position estimation and image processing tasks.

Additional algorithms for motion analysis could be developed to accurately define the rotational movement of the vehicle. For example, optical flow techniques [20] or deep learning-based motion classification methods [21] could be used to distinguish misinterpreted position changes. Optical flow techniques can detect subtle changes in the visual field, while deep learning-based motion classification can offer more robust interpretations of complex movements.

Lastly, the accuracy of the estimated position can be improved by integrating data from other sensors, such as gyroscopes and accelerometers, into the system. The use of multiple sensors will provide a more robust and reliable estimation system, particularly in the event of a GPS failure.

In conclusion, using this method, the real-world displacement of the UAV was successfully calculated solely based on camera images. The image processing algorithms and scaling method provided highly accurate position estimation, allowing the UAV's flight path to be effectively monitored. This approach is considered to be of critical importance, especially for applications requiring independent position tracking without relying on external sources such as GPS, and it is believed that it can serve as a foundation for autonomous systems.

Acknowledgements

This study is supported by Izmir Katip Celebi University Scientific Research Project (IKCU BAP) under project number 2024-TETAP-MÜMF-0004.

References

[1] Kaplan ED, Hegarty CJ. Understanding GPS: Principles and applications. 2nd ed. Artech House, 2006.

[2] Groves PD. Principles of GNSS, inertial, and multisensor integrated navigation systems. Artech House, 2013.

[3] Dilaver MB, Cakmak F, Uslu E, Altuntas N, Amasyali MF, Yavuz S. Autonomous navigation algorithm for robocup rrl maneuvering 2 field. Journal of Intelligent Systems with Applications 2019; 2(1): 46-49.

[4] Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2004; 60(2): 91-110.

[5] Yildirim YC, Yeniad M. Usage of machine learning algorithms on precision agriculture applications. Journal of Intelligent Systems with Applications 2020; 3(2): 107-113.

[6] Scaramuzza D, Fraundorfer F. Visual odometry [tutorial]. IEEE Robotics & Automation Magazine 2011; 18(4): 80-92.

[7] Python Software Foundation. Python Language Reference, version 3.x.

[8] Bradski G. The OpenCV Library. Dr. Dobb's Journal of Software Tools 2000.

[9] Kutlu Y, Alanoglu Z, Gokcen A, Yeniad M. Raspberry pi based intelligent robot that recognizes and places puzzle objects. Journal of Intelligent Systems with Applications 2019; 2(1): 85-89.

[10] Ciklacandir S, Ozkan S, Isler Y. A comparison of the performances of video-based and imu sensor-based motion capture systems on joint angles. 2022 Innovations in Intelligent Systems and Applications Conference (ASYU), Antalya, Turkey, 7-9 September 2022, pp. 1-5.

[11] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 2005; 27(10): 1615-1630.

[12] Bay H, Tuytelaars T, Van Gool L. SURF: Speeded up robust features. In: European Conference on Computer Vision, 2006, pp. 404-417. Springer.

[13] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision (ICCV), 2011, pp. 2564-2571.

[14] Groves PD. Shadow matching: A new GNSS positioning technique for urban canyons. Journal of Navigation 2011; 64(3): 417-430.

[15] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM. In: European Conference on Computer Vision, 2014.

[16] Mueller M, Smith N, Ghanem B. A benchmark and simulator for UAV tracking. In: European Conference on Computer Vision (ECCV), 2016, pp. 17-28.

[17] Liao T, Haridevan A, Liu Y, Shan J. Autonomous vision- based UAV landing with collision avoidance using deep learning. arXiv preprint arXiv:2109.08628, 2021

[18] Szeliski R. Computer Vision: Algorithms and Applications. Springer Science & Business Media, 2010.

[19] Hunter JD. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 2007; 9(3): 90-95.

[20] Horn BKP, Schunck BG. Determining optical flow. Artificial Intelligence 1981; 17(1-3): 185-203.

[21] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. Advances in Neural Information Processing Systems 2014; 27: 568-576.