Fire detection of Unmanned Aerial Vehicle in a Mixed Reality-based System

Shabnam Sadeghi Esfahlani, Silvia Cirstea, Alireza Sanaei and Marcian Cirstea
Department of Computing and Technology
Anglia Ruskin University, Cambridge, UK
Email: shabnam.sadeghi-esfahlani@anglia.ac.uk,
silvia.cirstea@anglia.ac.uk,
alireza.sanaei@anglia.ac.uk,
marcian.cirstea@anglia.ac.uk

***Abstract*—This paper presents a system that combines robotic operating system (ROS) and computer vision techniques for fire detection in a mixed reality environment. We have collected video streams from a mini camera on an Unmanned Aerial Vehicle (UAV), where the navigation data relied on state-of-the-art Simultaneous Localization and Mapping (SLAM) system. The data collected onboard are communicated to the ground station and processed through the robotic operating system. A robust and efficient re-localisation SLAM was performed to recover from tracking failure and frame lost in the received data. The fire detection algorithm was developed based on the colour, movement attributes, temporal variation of fire intensity and its accumulation around a point. A mixed reality environment was used to visualise and test the proposed system. The observation and data analysis confirmed that the UAV could successfully detect fire and flame, fly towards and hover around it.**

Keywords: Mixed Reality, Fire Detection, SLAM

I. Introduction

Recent advances in robotics and remote sensing provides the ability to perform tasks based on robot's sensing without human intervention. The onboard sensor blending facilitates the intelligent quadrotor to operate in cluttered, GPS-denied environments safely. This ability enables us to construct a map of the environment and locate the robot in that. Equipping flying robots with a camera and streaming video inputs could be used to detect fire in large and open areas faster before spreading around [1] and [2]. A moving camera could collect images at a speed of 30 frames per second (fps) [3]. The rapid detection of fire at incipient levels can maximise the probability of successful fire suppression, escape and survivability ([4] and [5]). Infra-red (IR) detection and ultra-violet (UV) detection are two methods to measure the range of flame radiation. IR detects particular flame flicker generated by fire and UV detects its UV radiations. [6] adopted an image processing technique for automated real-time fire detection in video images. The algorithm was based on the transient alteration of fire intensity. A Gaussian-smoothed colour histogram was introduced in [7] to detect the fire-coloured pixels and temporal variation of pixels in each frame. To optimise the detection performance, an erode operation and region growing method could be integrated [8]. The pixel intensity may shift due to global change and fire glimmer. In that case, the pixel-by-pixel intensity deviation could be applied to non-fire colour pixels by first adjusting for the temporal variation. It is performed by determining which pixels are fire candidates, obtaining the average variation in the strength of the colour and subtracting the average value from the pixel variations at each location. [1] proposed a real-time system for automatic fire detection using colour video inputs with an algorithm based on the transient properties of fire. This system has significant advantages over traditional fire detectors, including improved detection, descriptive information about fire's size and growth rate. The algorithm in [7] creates a threshold Gaussian-smoothed colour histogram for increased accuracy of training sequences. [9] adopted the spatial wavelet transforms and static camera monitoring system to analyse periodic behaviour in smoke boundaries and regions. Edges are regarded important due to their sharpness that drives to a decrease in the high-frequency content of the image and causing regional extrema in the wavelet region. [9] illustrated that a decrease in values of regional extrema is a sign of smoke where a scene enhances greyish colour that leads to a reduction in chrominance values of pixels. [10] proposed the spatial difference analysis using a histogram based approach, which focuses on the standard deviation of the green colour band. They found that green colour band is the most discriminative for recognising the spatial colour variation of flames. This can also be seen by analysing the histograms of colours. The value of green colour varies more than red and blue; as a result of the standard deviation of the green colour exceeding 50, the region is labelled as candidate flame. For smoke detection, this technique is not always applicable, because smoke regions often do not show as high spatial colour variation and can cause false detections. Temporal Fourier analysis is an alternative approach to detect flickering flames in that any change in the Fourier domain between $[5-10]$ Hz is a sign of flames. The effectiveness of fire detection technology has significantly improved following the initiation of artificial intelligence methods. The intelligent algorithms of fire detection include cross-correlation, neural networks (NN), fuzzy logic and Hidden Markov Model [11]. In intelligent fire detection systems, a ground work-station is used to perform data processing. As such we propose the use of a nano aerial vehicle (Crazyflie 2.0) which is equipped with a mini camera in a Mixed Reality (MR) environment. In this system, the fire detection algorithm is adopted in that the decision making is conducted in a ground station via ROS operating system. The method used is a temporally extended normalised covariance descriptor that is designed to characterise spatiotemporal video segments. The algorithm automatically removes unreliable fire pixels using erode operations. Some missing fire pixels are detected using region growing methods. The Crazyflie weighs only 27g, fits in the palm of a hand with a wingspan of 9 cm and has two micro-controllers [1]. Due to its limited payload (15g) we used a mini ArduCAM camera for video streaming (the resolution of 320 × 240 pixels). An open-source Robot Operating System (ROS)[2] is adopted and interfaced with Unity game engine to demonstrate the potential of our system in fire detection. Fig. 1 illustrates the nano quadrotor that is used in this study with the ArduCAM set up.

II. Context of the Research

A. Feature Descriptor: Front-End

A front-end of our algorithm consists of the observation function of our system. It identifies landmarks and measures image of landmarks of unknown positions and keyframes (a subset of images) to compute poses [12]. Feature performance or keypoint detection in the front end is responsible for characterising and matching keypoints [13]. Different combinations of detectors and descriptors have been adopted in literature for mapping and localisation using monocular or stereo vision. Binary feature detectors and descriptors have increased performance. A compact binary representation and limited computational requirements make descriptors attractive solutions to many modern applications. These attributes are important for mobile platforms which have limited computation and memory resources [13].

BRIEF (Binary Robust Independent Elementary Features) is a binary vector in that each section is the result of a test within two pixels of a patch of an image. The patches are previously levelled with a Gaussian kernel to lessen noise. BRIEF is proposed by [14] that uses a sampling pattern consisting of various comparisons. The sample features are selected randomly from an isotropic Gaussian distribution concentrated at the feature location. Given its simple construction and compact storage as illustrated in Fig. 2-a, BRIEF requires the lowest computation but relatively larger storage.

ORB (Oriented FAST and Rotated BRIEF) was proposed by [15]. It overcomes the deficit of rotation invariance of BRIEF and is an active binary descriptor which is based on BRIEF descriptor [14]. ORB is built upon the well-known FAST (Features from Accelerated Segment Test) keypoints and feature extraction method . FAST corner detector uses a circle of 16 pixels (a Bresenham circle of radius 3) to recognise if a candidate point is an edge. An integer number labels each pixel in the group. A point is considered to be a corner if a set of adjoining pixels in the loop is brighter than the magnitude of candidate pixel plus a threshold amount or if all are darker than the intensity of applicant pixel minus the threshold value. A weighted averaging of pixel intensities in the local patch estimates regional orientations using an intensity centroid [16]. The orientation is a vector between the feature location and the centroid. The sampling pattern employed in ORB uses pairwise intensity comparisons that maximises the descriptors variance and minimises the correlation under various orientation changes. ORB matches visual features using PTAM that splits tracking and mapping into two separate tasks for real-time keypoint detection [17].

BRISK (Binary Robust Invariant Scalable Keypoints) descriptor was proposed by [18] which provides both scale and rotation invariance which was inspired by BRIEF. It consists of scale-space corner detection with random unity implementation, orientation estimation by regional gradient calculation, and the formation of a binary descriptor from brightness correlations in the keypoint region. To compute the feature locations, it uses the AGAST (Adaptive and Generic Accelerated Segment Test) corner detector[3]. AGAST improves FAST by increasing speed while maintaining the same disclosure accomplishment.

Speeded-Up Robust Features (SURF) was proposed by [19], [20] which is a scale and rotation invariant indicator. SURF relies on integral images for image convolutions. It considers the concentration of the principal existing features and uses a Hessian matrix-based measure for the detector and a distribution-based descriptor. SURF and SIFT rely on a parameterisation of an image area, in that each dimension is a discretisation of a float excluding binary. These two features are both fast and high-quality methods for recognising image keypoints but require high computation and storage as shown in Fig. 2-a.

Scale-Invariant Feature Transform (SIFT)) [21]) extracts distinctive invariant features from images. This can be employed to implement substantial matching between various representations of an object or scene. SIFT is invariant to image scale and rotation and provides robust matching across a substantial range of affine regression. SIFT computes a histogram of locally oriented gradients around the interesting point and stores the bins in a 128D vector. SIFT [22] or, at higher speeds, SURF ([19]) are algorithms invariant to many transformations and are tolerant to any distortions caused by viewpoint change, sudden movement, brightness or contrast variation of 2D image keypoints. Most loop closing operations are performed using SIFT or SURF descriptors [23].

B. Representations for Robot Localization and Mapping: Back-End

SLAMs are navigation solutions which are mapping while tracking locally and globally [24]. SLAM is considered one of the most important steps towards real robot autonomy. SLAM is the problem of estimating both the robots location

[1]https://wiki.bitcraze.io/projects:crazyflie2:index
[2]http://wiki.ros.org/ROS/Introduction
[3]http://www.i6.in.tum.de/Main/ResearchAgast

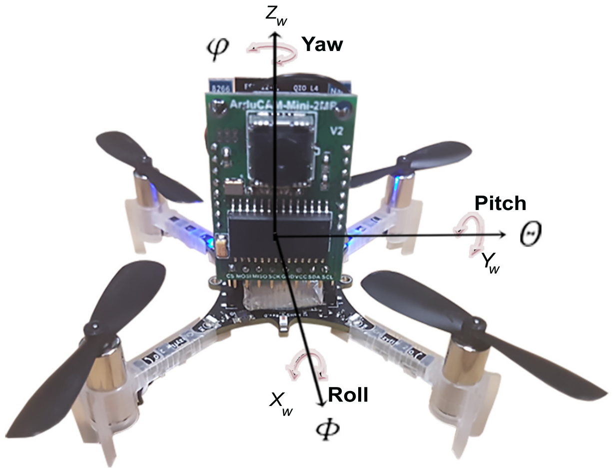

Fig. 1. The Crazyflie 2.0 with ArduCAM-Mini-2MP-V2 and ArduCAM-ESP8266-Nano modules.

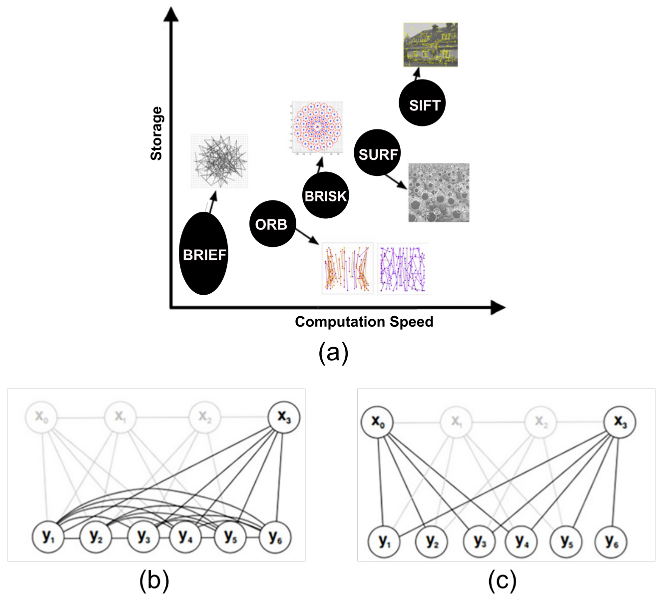

Fig. 2. (a) The computation speed and memory requirement for Real Value parametrisation of front-end features (SURF and SIFT) and the binary descriptors (BRIEF, ORB and BRISK) with an example of their pattern [13]. (b) filter and (c) BA base approach to visualise how inference progressed [22].

and a map of its surrounding environment. In the back-end, a map is created as the sensors (robot) are perceiving the environment. Any noise in the estimate of the robots pose could result in error in the estimation of the map. SLAM algorithms address the parameters relating to sensors (i.e. sonar, laser or vision, a wide or narrow field of view), map representation (i.e. occupancy grid, 2D or 3D, natural or specialised landmarks), robot's and environment's static/dynamic parameters (i.e. indoor or outdoor) and co-ordinate frameworks for sensor measurements and robot control signals over time [25]. Environment dynamics are essential as most algorithms for indoor do not scale well for outdoor applications. Regarding theoretical frameworks, SLAM can be categorised into two main paradigms: filtering and optimization based approach [26]. Some of the filtering methods are: Extended Kalman Filter (EKF) and Rao-Blackwellized particle filters (FastSLAM) [27]. The Extended Kalman Filter (EKF) has been the most common procedure and uses a covariance matrix that incorporates all landmarks [28]. Nevertheless, as the number of landmarks grows this matrix quickly becomes challenging to expand. Particle Filters approximate the posterior distribution over robot poses and manage outliers better than the EKF but suffers from inadequate scaling. Visual bundle adjustment (BA) is an optimisation method that uses Levenberg Marquardt or the Gauss-Newton methods [29] and [30]. Fig. 2-b and Fig. 2-c illustrate a vector of parameters outlining a historical viewpoint of the camera (x_i) and a vector of parameters depicting the attitude of a static feature (y_i). These are links between continuing x_i representing non-visual association regarding local motion.

In the **Filtering approach**, illustrated in Fig. 2-b, all attitudes other than the current one are marginalised out after each frame, however, some features are retained if they might be used over again. The graph does not grow arbitrarily with time, and will not grow at all during repeated movement. Persistent feature variables are added just when new areas are investigated. The drawback of this approach is that the diagram immediately increases inter-connectivity due to the elimination of a past attitude variable. It results in replacing variables with new links between every pair of features. Joint potentials over all of these mutually-inter-connected variables must be collected and updated. The main drawback of filtering approach is the computational cost of propagating joint distributions scales defectively with the number of variables involved. The standard algorithm for filtering employing Gaussian probability distributions is the EKF. The compact inter-connections between features are evident in a single joint density of features stored by a mean vector and extended covariance matrix.

Keyframe Bundle Adjustment (BA) is shown in Fig. 2-c. BA involves obtaining the full maximum likelihood (ML) feature diagram at every new time-step. BA is known to provide accurate estimates of camera localisations as well as a sparse geometrical reconstruction [31]. BA is an iterative method, in which one attempts to fit a non-linear model to the measured data that can be applied to a wide range of reconstruction and optimisation problems using the ML solution.

In this study, the UAV's state is described by its camera and an inertial measurement unit (IMU) sensor. The front-end of our SLAM algorithm employs ORB to detect and match keypoints/edges and orientation estimation since BRISK, SURF and SIFT require significant computational and memory requirements. We use BA as the back-end to detect and estimate poses and keypoint matches over the previous frames. We adopted BA for re-localisation and loop closure in order to correct the accumulated drifts over the history of frames. In BA approach the lack of marginalisation means that elements will remain sparsely inter-connected. Fig. 3 shows three SLAMs with camera location, KeyFrames, the local mapping (in red and black dots), loop closure and Covisibility using Monocular Visual Odometry Datasets.

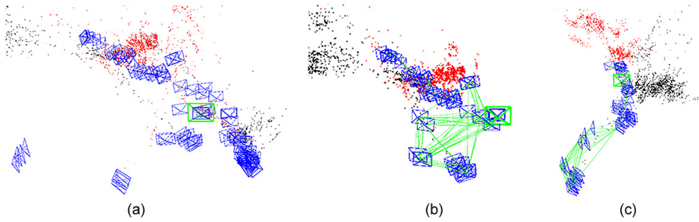

Fig. 3. The trajectories and sparse reconstruction of the sequences with multiple loop closures.

III. Methodology and Implementation

A. Fire detection based on video processing

Various algorithms could be integrated to determine if a motion in a scene is due to flame or to an ordinary moving object. Examples include background subtraction methods, temporal differencing and optical flow analysis ([32] and [33]). As such, detailed and adequate local fire information could be communicated directly to the appropriate fire department. In this study, we have adopted temporally extended normalised covariance descriptors (TENCD) [34] to extract fire features from video sequences. TENCD describes spatiotemporal video segments in that $I(i, j, n)$ shows the intensity of $(i, j)^{th}$ pixel of the n^{th} image frame. Some attribute parameters are used to construct a covariance matrix describing spatial data [35]. The video is split into chunks of size $10 \times 10 \times F_{rate}$ with F_{rate} being the frame rate of the video. To improve the performance of computations we calculated the normalised covariance parameters. Only pixels corresponding to the non-zero values of the mask are used in the selection of blocks.

$$\Phi(i, j, n) = \begin{cases} 1 & \text{for M(i,j,n) = 1} \\ 0 & \text{Otherwise.} \end{cases} \tag{1}$$

Where Φ is the mask, $M(\cdots, \cdots, n)$ is the binary mask obtained from colour detection and moving object detection algorithms. Property parameters are used for each pixel satisfying the colour condition. We have computed the normalised covariance values of the pixel property vectors separately to reduce the computational cost. During the implementation of the correlation method, the first derivative of the image is computed by filtering the image with $[1 \quad 0 \quad 1]$ and second derivative is found by filtering the image with $[1 \quad -2 \quad 1]$ filters, respectively. The pixels and frame-by-frame approach were performed while considering the RGB (Red, Green, Blue), HSV (Hue, Saturation, Value) and YCbCr (non-linear RGB or Luminance, Chrominance) colour space [1], as well as motion information from video sequences since flames, are moving objects. The conditions in YUV (luminance component: Y, chrominance channels U and V) colour space are as follows [35];

$$\begin{cases} Condition1: & Y > T_Y \\ Condition2: & |U - 128| < T_U \ and \ |V - 128| < T_V \end{cases} \tag{2}$$

The threshold values T_Y , T_U and T_V are taken from [36] which are determined experimentally. The luminance component takes values in the range $[0, 255]$ in an 8-bit quantised image and the mean values of chrominance channels, are increased to 128 so that they also take values between $[0, 255]$.

B. Mixed Reality Simulation

Mixed Reality (MR) creates a space in which both physical and virtual elements co-exist [37]. MR enables elements in one world to react directly to elements in the other world via direct and real-time data communication. The robots performance and analysis in the real-time/world is often a cumbersome task. Thus, we propose a system that adopts MR using virtual environment and real-world interaction to provide a safe and simple testbed. Wind effects in both virtual and physical worlds are neglected. The high precision pressure sensor (LPS25H) of the Crazyflie makes it more suitable for the situation with a moderate breeze. The system allows a gradual transition of virtual training into the physical system in robotics or any other fields. The proposed system was tested for fire rescue operation in MR operation, enabling real-time communication, data collection and navigation. It uses the bag-of-words technique to convert an image into a sparse numerical vector and creates an image database. The Unity and ROS systems are interfaced using a yaml-based[4] communications, ROS-bridge and HTTP protocol [38]. ROS-bridge enables data streaming and communication between ROS and Unity. Unity facilitates interaction of the various virtual and physical suits by simulating the required scenarios [39]. We mapped ROS messages to events that are processed as part of the basic rendering loop in the visualised environment. A real-time interaction was obtained using rendering loop, and the virtual goal locations (flame). The virtual world is visualised by the camera in the 3D world, while SLAM is performed using the IMU and camera data independently in the physical and virtual environments. The teleoperation system uses a virtual drone which controls the physical quadrotor's navigation. Our system tracks the quadrotor's pose, velocity and orientation data that are received by the system at frame-rate. The operation enables us to achieve reliable information to match the map points with keypoints on the frame. Each frame j is optimised by minimising the feature re-projection error [30]. The optimisation problem is performed based on the Gauss-Newton algorithm in g_2o (General Graph Optimization) [40].

IV. Experimental Evaluation

To facilitate an online, real-time navigation and fire detection system, ROS is set up on the ground-based station. The

[4]http://www.yaml.org/

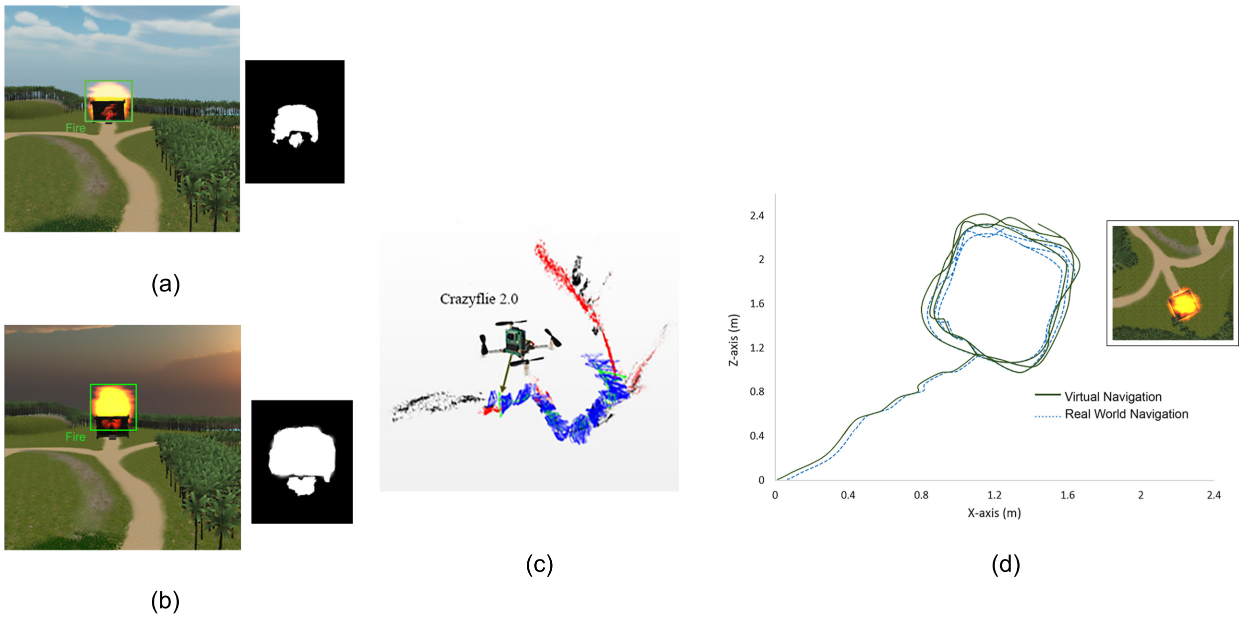

Fig. 4. The fire detection and mask generating algorithm in the 3D simulated environment in two different light condition with BAR=2.181 (a) day and (b) night. (c) The real Crazyflie with the mini ArduCAM in the real world while generating the SLAM. Tele-operation is performed simultaneously over ROS as the virtual quadrotor navigates. (d) The graph of the real and virtual quadrotor's navigation relative to x and z-axis as the quadrotor detected the fire and hovered around the fire.

intelligent fire detection system was adopted to communicate and exchange information via the WiFi ad-hocs with two host machines (a laptop and a desktop computer) running Ubuntu 16.04 LTS. The ground truth analysis (computer vision metric) was conducted for the images taken from virtual camera using segmentation and labelling of images to measure the performance and accuracy of the algorithms. The algorithm was tested in a virtual scene where a wooden hut in flames is set to be the target. The testing was conducted using two scenarios; sun and sunset (Fig. 4-a and Fig. 4-b) to measure the effects of the light on algorithm's performance. The disorder analysis was adopted with the Boundary Area Roughness (BAR) of 2.18. It is determined by relating the perimeter of the region to the square root of the area. In both scenarios, the flame was detected, and masks are generated successfully. Fig. 4-a and Fig. 4-b illustrate the simulated environment in Unity and the mask next to each figure shows the detected fire. Fig. 4-c shows the screenshots of the SLAM generated by the physical quadrotor while being controlled by the virtual drone. The virtual UAV flew inside the 3D world and controlled the real quadrotor in the real-time successfully. Fig. 4-d demonstrates the virtual flying path vs real world navigation from the start to end.

V. Conclusion

In general terms, the proposed system provided a robust real-time performance using MR and fire algorithm. The maximum navigation speed is set to be 0.2m/s, taking off and finally landing above the origin location. The computer keyboard controls the start as well as any emergency stops. The virtual quadrotor avoided obstacles during its navigation in the virtual world and autonomously flew towards the target. The Crazyflie's trajectory estimated independently by its onboard sensors and the SLAM system using ORB-SLAM [41]. The data fits well with the ground truth data. The quadrotor in the virtual world navigated towards the target and hovered around the wooden hut in a counter-clockwise direction for 6 min. The video streams are transferred to the ground station and saved on the hard disk.

The future work will be dedicated to adding more functionality to swarms to effectively accomplish the more complex search and rescue tasks in different environmental conditions.

References

[1] G. Healey, D. Slater, T. Lin, B. Drda, and A. D. Goedeke, "A system for real-time fire detection," in *CVPR*, vol. 93, 1993, pp. 15–17.

[2] D. Kim and Y.-F. Wang, "Smoke detection in video," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 5. IEEE, 2009, pp. 759–763.

[3] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

[4] Z. Liu and A. K. Kim, "Review of recent developments in fire detection technologies," *Journal of Fire Protection Engineering*, vol. 13, no. 2, pp. 129–151, 2003.

[5] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Safety Journal*, vol. 44, no. 3, pp. 322–329, 2009.

[6] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images," *Fire safety journal*, vol. 41, no. 4, pp. 285–289, 2006.

[7] R. Kjeldsen and J. Kender, "Finding skin in color images," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. IEEE, 1996, pp. 312–317.

[8] W. Phillips Iii, M. Shah, and N. da Vitoria Lobo, "Flame recognition in video," *Pattern recognition letters*, vol. 23, no. 1-3, pp. 319–327, 2002.

[9] B. U. Töreyin, Y. Dedeoğlu, and A. E. Cetin, "Wavelet based real-time smoke detection in video," in *Signal Processing Conference, 2005 13th European*. IEEE, 2005, pp. 1–4.

[10] X. Qi and J. Ebert, "A computer vision based method for fire detection in color videos," *International journal of imaging*, vol. 2, no. S09, pp. 22–34, 2009.

[11] H.-C. Muller *et al.*, "A new approach to fire detection algorithms based on the hidden markov model," *NIST SPECIAL PUBLICATION SP*, pp. 129–138, 2001.

[12] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[13] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 759–773.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," *Computer Vision–ECCV 2010*, pp. 778–792, 2010.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571.

[16] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[17] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Computer Vision–ECCV 2006*, pp. 430–443, 2006.

[18] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[20] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2657–2664.

[23] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[24] P.-J. Bristeau, F. Callou, D. Vissiere, and N. Petit, "The navigation and control technology inside the ar. drone micro uav," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1477–1484, 2011.

[25] R. Sim, P. Elinas, M. Griffin, J. J. Little *et al.*, "Vision-based slam using the rao-blackwellised particle filter," in *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, vol. 14, no. 1, 2005, pp. 9–16.

[26] G. Younes, D. Asmar, E. Shammas, and J. Zelek, "Keyframe-based monocular slam: design, survey, and future directions," *Robotics and Autonomous Systems*, vol. 98, pp. 67–88, 2017.

[27] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[28] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, p. 13091332, 2016.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." Georgia Institute of Technology, 2015.

[30] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[31] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[32] B. Ko, K.-H. Cheong, and J.-Y. Nam, "Early fire detection algorithm based on irregular patterns of flames and hierarchical bayesian networks," *Fire Safety Journal*, vol. 45, no. 4, pp. 262–270, 2010.

[33] I. Kolesov, P. Karasev, A. Tannenbaum, and E. Haber, "Fire and smoke detection in video with optimal mass transport based optical flow and neural networks," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 761–764.

[34] S. Verstockt, "Multi-modal video analysis for early fire detection," Ph.D. dissertation, Ghent University, 2011.

[35] A. E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y. H. Habibolu, B. U. Töreyin, and S. Verstockt, "Video fire detection–review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.

[36] B. U. Töreyin, "Fire detection algorithms using multimodal signal and image analysis," Ph.D. dissertation, bIlkent university, 2009.

[37] Z. Pan, A. D. Cheok, H. Yang, J. Zhu, and J. Shi, "Virtual reality and mixed reality for virtual learning environments," *Computers & Graphics*, vol. 30, no. 1, pp. 20–28, 2006.

[38] D. Crockford, "The application/json media type for javascript object notation (json)," 2006.

[39] R. Codd-Downey, P. M. Forooshani, A. Speers, H. Wang, and M. Jenkin, "From ros to unity: Leveraging robot and virtual environment middleware for immersive teleoperation," in *Information and Automation (ICIA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 932–936.

[40] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[41] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.