Design and Implementation of an Indoor PM Concentration Measurement and Mapping System

Andrey Ivanov*, Stefan Hensel†, Marin Marinov*, Borislav Ganev*
* Technical University of Sofia, Faculty of Electronic Engineering and Technology
Sofia, Bulgaria, {andrivanov, mbm, b_ganev}@tu-sofia.bg
†Offenburg University of Applied Sciences. Institute for Machine Learning and Analytics
Offenburg, Germany, stefan.hensel@hs-offenburg.de

Abstract — This paper presents the design and implementation of a particulate matter mapping system using an unmanned aerial vehicle (UAV) equipped with a monocular camera, an optical particle counter (OPC) and backed by a visual simultaneous localization and mapping algorithm (Visual SLAM, or VSLAM). Introduction

Indoor air quality (IAQ) monitoring has come to the fore in the last few years, with an emphasis on identifying the sources of pollutants, studying their movement and the influence of external factors on it. A major problem related to indoor air quality is the limited air movement, which leads to the accumulation of pollutants at certain points or frequently used spaces - offices, classrooms or laboratories, eating places, etc.

The purpose of this publication is to verify the feasibility of using an unmanned aerial vehicle equipped with a monocular camera and an optical particle counter to create a map of indoor air pollution using a visual mapping and localization algorithm.

The algorithm allows the use of video footage to create a map and extract the trajectory of the UAV.

To meet the needs of the system developed within this publication, the particulate matter sensor must comply with a set of requirements: it has to be portable, as small in size and weight as possible, able to operate in unstable environments (mounted on an unmanned aerial vehicle), and its power supply requirements must be in line with the other components of the system.

I. HARDWARE

A. *Parrot Anafi 4K*

The Parrot Anafi 4K Unmanned Aerial Vehicle was chosen as the basis for the system. It is equipped with a high-resolution camera and an electronic stabilization system and vertical mobility within 180°. A 32 GB SD card is used to record videos and images.

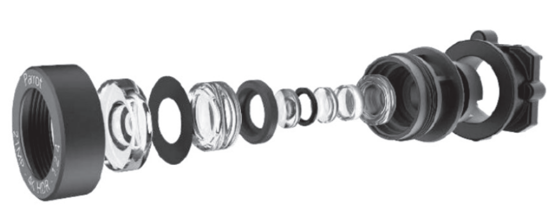
fig

The aircraft is relatively small, making it suitable for indoor flight. It is controlled by a controller that connects to the wireless network created by the drone when it is launched. It creates a web server on which all its parameters can be monitored (status, speed, last command set, battery level, etc.), as well as a complete record of all events during flight (Flight Record).

The web server also supports a REST API which can be used to access recorded media files (captured videos, photos). A list of all files is maintained in a JSON file.

Access to all drone functionality is provided by the so-called Olympe Framework. Also called the "Ground Software Development Kit (SDK)", it is a library written in the Python programming language and allows control of the device via Python scripts after connecting a computer running the Linux operating system to the UAV's wireless network".

Parrot Anafi 4K's camera features a 1/2.4" CMOS sensor, an f/2.4 aperture, and an aspherical lens architecture featuring six lenses to minimize the level of parasitic light while providing temperature stability across a wide range. A picture of its construction is shown on [].



fig

In terms of resolution, the UAV can record video at resolutions of 4096x2160 pixels at 24 frames per second (FPS), 3840x2160 pixels at 24, 25 or 30 FPS, or 1920x1080 pixels at 24, 25, 30, 48 and 60 FPS. [Anafi whitepaper]

Video stability is a high priority for SLAM algorithms because it leads to stability of detected feature points between consecutive frames. To ensure image stability during flight, the camera is mounted on a gimbal. The gimbal allows mechanical stabilization of the camera along 2 axes, the roll axis and the pitch axis. In addition to mechanical stabilization, Electronic Image Stabilization (EIS) is available in all three axes - roll, pitch and yaw. The EIS algorithm makes it possible to eliminate the wide-angle distortion caused by the lens, as well as reducing the influence of vibrations caused by the operation of the rotors. The operation of both types of stabilization, on a total of 5 axes, in tandem facilitates the implementation of SLAM algorithms on video captured by Parrot Anafi 4K.

The components of the sensor system the UAV has include an inertial measurement unit with 9 degrees of freedom, an ultrasonic module for measuring altitudes up to 5 meters and a barometer for flight altitudes above 5 meters, a GPS module, as well as a low-resolution vertical camera and an algorithm for measuring horizontal velocity using its optical flow.

B. *OPC-R1*

The OPC-R1 sensor, manufactured by Alphasense, was used to measure the fine particulate matter concentration. It is an optical particle counter with integrated temperature and humidity sensors. The sensor uses a 4-8 mW semiconductor laser operating with at a wavelength of 639 nm. The sensor uses the light reflected by particles passing through an air volume as an indicator of their size. From the resulting mass concentrations, a histogram is obtained which describes their size distribution.

fig

All particles, regardless of shape, are treated as spherical and their equivalent spherical size is calculated. The relationship between it and the reflected light is described by Mie theory of reflection from a sphere and the index of refraction. The air velocity is determined by the fan speed. The fan speed is monitored for possible variation and adjusted dynamically by the firmware to increase the reliability of the measurements by eliminating the dependence of the values on the flow velocity. The OPC-R1 also has 10 sets of weighting factors for each of the subscales. The classification rate for particles passing through the air volume is 10000 particles per second, assigning the measurement for each particle to one of the subcategories (0.35 ÷ 12.4 μm). To allow the measurement of organic particles, e.g., pollen, a higher reading limit is required – this is achieved by switching between different values of the gain factor.

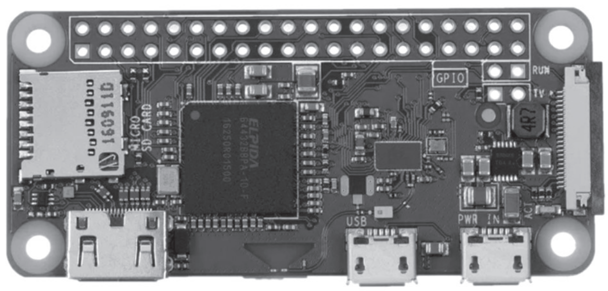
In a previous study, a sensor from the same Alphasense family with similar electrical and measurement parameters, in particular the OPC-N3, was successfully used to measure PM while travelling [high resolution ….]. The sampling rate and resistivity to environmental influences are suitable for obtaining granular results. However, its relatively large size and weight make it unsuitable for mounting on a UAV.

The small size and weight (~30 g) of the OPC-R1, its appropriate features and its software compatibility with the controller used make it suitable for use in a low payload unmanned aerial vehicle like the Parrot Anafi.

C. *Raspberry Pi Zero W*

A Raspberry Pi Zero W single board computer with WiFi connectivity was used to read, process and write the information obtained from the OPC-R1. Its small size and weight (~9 g) as well as compatible logic levels and supply voltage make it suitable for the purpose of the system.

The device supports the Raspbian operating system, which is based on the Debian distribution of Linux. It runs in the so-called headless mode, i.e., without a graphical desktop environment or user peripherals (mouse, keyboard, monitor) to work with it.



fig

Access to the single-board computer is via SSH (Secure Shell) directly from a Linux terminal, or software such as PuTTY for Windows operating systems. To create the script for the connection to the sensor, it is necessary to use a text editor that can run in a terminal - in this case Nano. Execution of the script is done from the terminal, and the resulting csv file with the measurements is downloaded using the scp (secure copy) command invoked from the user's computer.

II. SOFTWARE

Simultaneous Localization and Mapping (SLAM) algorithms offer a way for a system to localize itself in an unfamiliar environment by incrementally building a map of its surroundings.

Thanks to recent advances in CPU and GPU technologies, implementing the necessary complex algorithms in real time is no longer an insurmountable problem. In fact, a variety of solutions have been proposed using different visual sensors, including monocular, stereo, omnidirectional, time-of-flight (TOF), and combined color and depth (RGB-D) cameras.

Visual SLAM algorithms use visual data to detect feature points and track their movement between consecutive images to record trajectories and create feature point maps of the environment the camera has recorded. Data acquisition involves extracting feature points, or features, from the acquired images. Features of interest range from simple point features such as angles to more complex features such as edges and blobs and even complex objects such as doors and windows.

Feature tracking is the process of finding correspondences between such features in adjacent frames and is useful when small changes in motion occur between two frames. From another perspective, feature matching is the process of individually extracting features and matching them across multiple frames. Feature matching is particularly useful when significant changes occur in the appearance of features due to their observation over longer sequences. In the following sections, we briefly describe the most common feature extraction techniques, also called key points, regions of interest (ROIs) or points of interest (POIs), that are used in mobile robotics applications.

A. *OpenVSLAM, StellaVSLAM*

OpenVSLAM is a simultaneous mapping and localization algorithm designed to be as easy as possible to use as part of a larger system and to be invoked by third-party applications. It incorporates functionality allowing automatic trajectory calculation and recording in a number of formats. In addition, maps that are created by OpenVSLAM can be saved as msgpack files and then used by other applications.

Due to a license issue and the subsequent discontinuation of OpenVSLAM development, a fork of the original development relying on public support, called StellaVSLAM, is used in this thesis. Everything said about OpenVSLAM applies to the fork.

There are many similar algorithms - in particular, ORB-SLAM, LSD-SLAM and DSO [] represent the main approaches considered as de-facto standards of visual SLAM algorithms having achieved state-of-the-art results using benchmarking data, such as EuRoC, TUM and KITTI. However, they are not suitably designed in terms of usability and extensibility as visual SLAM libraries. Thus, researchers and engineers need to make great efforts to implement these SLAM systems in their applications. In other words, it is inconvenient to use existing opensource visual SLAM software as the basis of applications arising from 3D modeling and mapping techniques, such as autonomous control of robots and drones.

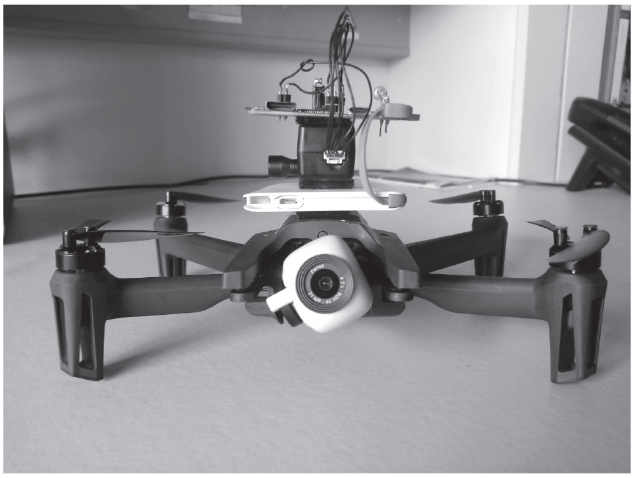
One of the notable features of OpenVSLAM is that the system can handle different types of camera models, such as perspective, fisheye, and equirectangular. At the same time, fisheye cameras are often mounted on drones and robots for visual SLAM and scene understanding as they have a wider field of view (FoV) than perspective cameras. OpenVSLAM can be used with nearly equal implementation between the perspective and fisheye camera models.

Most of the visual SLAM systems cannot store and load card databases. Localization based on a pre-built map is important from a practical point of view for many applications. Accordingly, it is clear that the ability to store and load created maps improves the usability and extensibility of a visual SLAM framework. Therefore, map database input/output functions have been implemented in OpenVSLAM.

III. SYSTEM SETUP

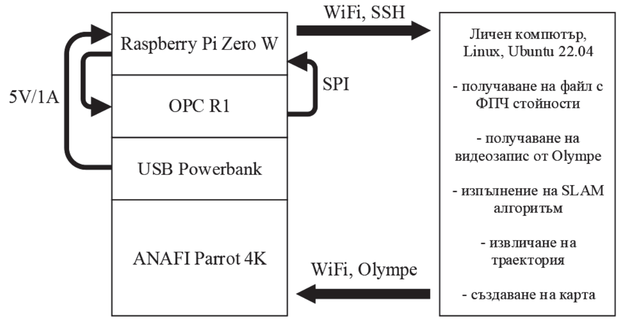
A block diagram of the experimental setup is presented in Fig. The aircraft is used as the basis, as the element with the largest dimensions. On top of it, above the center of gravity and at a sufficient distance from the propellers, a USB Powerbank with a size of 95x60x5 mm was placed. The OPC-R1 sensor and the Raspberry Pi device, which read and write the data, respectively, are positioned above it. Additional consideration is given to the weight of the individual components, as the Parrot Anafi 4K, like most hobby aircraft, is not designed to transport cargo.

During the development of the system, an experiment was carried out to establish the maximum payload capacity of the aircraft, as well as the optimal way to attach the system components to it. The aircraft was loaded gradually until an undesirable effect was obtained - during one of the tests, due to uneven load distribution on the aircraft, it failed to reach the planned altitude and deviated sharply from the desired course. The subsequent collision of the UAV caused a minor failure requiring propeller replacement.



fig

A weight of 210 grams was selected as the maximum load, at which the aircraft had difficulty changing its flying height but remained controllable. A textile Velcro fastener in the form of adhesive tape was used to mechanically connect the individual blocks. A picture of the system is shown on fig. Software ….



fig

IV. REFERENZEN

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (references)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

THIS PAGE IS INTENTIONALLY BLANK

Mapping of Indoor Environments Using UAVs

Andrey Ivanov
Technical University of Sofia, Faculty of Electronic Engineering and Technology
Sofia, Bulgaria, andrivanov@tu-sofia.bg

Abstract — Mapping of indoor environments is the process of creating a digital representation of the layout of a room or building. The resulting map can then be used by an autonomous system, such as an unmanned aerial vehicle (UAV) or a ground-based unit like a rover, to operate within the confines of the mapped space while avoiding collision with the environment. This paper presents a system combining a UAV, the Parrot Anafi 4K, with StellaSLAM, a Visual Simultaneous Localization and Mapping algorithm, to create feature point maps and record trajectories within a closed space using videos recorded by the UAV's built-in monocular camera.

I. Introduction

Indoors operation almost completely eliminates GPS as a reliable method of localization and therefore other means must be pursued. The past few decades have seen a surge of growth in the development of algorithms for simultaneous localization and mapping, or SLAM. Such algorithms create maps and use them to tell the system where it is, based on where it has been before while the algorithm was running. Typically, SLAM uses different types of sensors, such as sonars, laser scanners and infrared sensors. Recently, Visual SLAM has been of great interest, due to the widespread availability of cheap visual sensors and their implementation in our daily lives. VSLAM algorithms use visual data to detect feature points and track their movement between consecutive images to record trajectories and create feature point maps of the environment the camera has recorded.

Feature point maps represent the environment in the form of geometric shapes, such as points and straight lines. Each feature is described by a set of parameters, such as location and geometric shape. Localization in such an environment is performed by observing and detecting feature points and comparing them with those already stored on the map. Since this approach uses a limited number of objects to represent the map, the computational cost can be kept relatively low.

II. Hardware – The UAV

The Parrot Anafi 4K Unmanned Aerial Vehicle was chosen as the basis for the system. It is equipped with a high-resolution camera and an electronic stabilization system and vertical mobility within 180°. A 32 GB SD card is used to record videos and images.

The aircraft is relatively small, making it suitable for indoor flight. It is controlled by a controller that connects to the wireless network created by the drone when it is launched. It creates a web server on which all its parameters can be monitored (status, speed, last command set, battery level, etc.), as well as a complete record of all events during flight (Flight Record).

The web server also supports a REST API which can be used to access recorded media files (captured videos, photos). A list of all files is maintained in a JSON format file.

Access to all drone functionality is provided by the so-called Olympe Framework. Also called the "Ground Software Development Kit (SDK)", it is a library written in the Python programming language and allows control of the device via Python scripts after connecting a computer running the Linux operating system to the UAV's wireless network.

A. *Camera setup*

Parrot Anafi 4K's camera features a 1/2.4" CMOS sensor, an f/2.4 aperture, and an aspherical lens architecture featuring six lenses to minimize the level of parasitic light while providing temperature stability across a wide range. A picture of its construction is shown on [].

In terms of resolution, the UAV can record video at resolutions of 4096x2160 pixels at 24 frames per second (FPS), 3840x2160 pixels at 24, 25 or 30 FPS, or 1920x1080 pixels at 24, 25, 30, 48 and 60 FPS.

From the listed features it can be concluded that the capabilities of the camera far exceed the requirements of the SLAM algorithm. It is possible to reduce the resolution multiple times without loss of information and hindering the system's functionality.

Video stability is a high priority for SLAM algorithms because it leads to stability of detected feature points between consecutive frames. To ensure image stability during flight, the camera is mounted on a gimbal. The gimbal allows mechanical stabilization of the camera along 2 axes, the roll axis and the pitch axis. In addition to mechanical stabilization, Electronic Image Stabilization (EIS) is available in all three axes - roll, pitch and yaw. The EIS algorithm makes it possible to eliminate the wide-angle distortion caused by the lens, as well as reducing the influence of vibrations caused by the operation of the rotors. The operation of both types of stabilization, on a total of 5 axes, in tandem facilitates the implementation of SLAM algorithms on video captured by Parrot Anafi 4K.

The components of the sensor system the UAV has include an inertial measurement unit with 9 degrees of freedom, an ultrasonic module for measuring altitudes up to 5 meters and a barometer for flight altitudes above 5 meters, a GPS module, as well as a low-resolution vertical camera and an algorithm for measuring horizontal velocity using its optical flow.

B. *Olympe Framework*

Olympe is an SDK that provides a Python controller programming interface for Parrot ANAFI drones. It can be used to connect to and control a drone from a remote Python script running on a Linux based computer.

Some of its features include:

- Allowing communication to the aircraft during flight,
- Transmission of commands for movement in four directions as well as automated take-off and landing,
- Modifying video settings and parameters, such as framerate and resolution,
- Starting and stopping video recording, accessing video stream.

III. Software

Simultaneous Localization and Mapping (SLAM) algorithms offer a way for a system to localize itself in an unfamiliar environment by incrementally building a map of its surroundings.

Thanks to recent advances in CPU and GPU technologies, implementing the necessary complex algorithms in real time is no longer an insurmountable problem. In fact, a variety of solutions have been proposed using different visual sensors, including monocular, stereo, omnidirectional, time-of-flight (TOF), and combined color and depth (RGB-D) cameras.

Visual SLAM algorithms use visual data to detect feature points and track their movement between consecutive images to record trajectories and create feature point maps of the environment the camera has recorded. Data acquisition involves extracting feature points, or features, from the acquired images. Features of interest range from simple point features such as angles to more complex features such as edges and blobs and even complex objects such as doors and windows.

Feature tracking is the process of finding correspondences between such features in adjacent frames and is useful when small changes in motion occur between two frames. From another perspective, feature matching is the process of individually extracting features and matching them across multiple frames. Feature matching is particularly useful when significant changes occur in the appearance of features due to their observation over longer sequences. In the following sections, we briefly describe the most common feature extraction techniques, also called key points, regions of interest (ROIs) or points of interest (POIs), that are used in mobile robotics applications.

A typical visual SLAM workflow includes the following steps:

1. Input data processing. In visual SLAM, this mainly refers to the acquisition and preprocessing of images from the camera. For a mobile robot, this would also include the acquisition and synchronization with the motor encoders, IMU sensors, etc.

2. Visual Odometry (VO). The task of VO is to estimate camera motion between adjacent frames and generate a coarse local map. VO is also known as front-end.

3. Back-end filtering/optimization. It involves obtaining the camera poses at different time intervals from the VO and the loop closure results, then applying optimization to generate a fully optimized trajectory and map. Since it is connected after the VO, it is also known as the backend.

4. Loop Closure determines whether the system has returned to its previous position to reduce accumulated drift. If a loop is detected, it will provide information to the backend for further optimization.

5. Reconstruction. The last step constructs a map based on the estimated camera trajectory.

A. *StellaVSLAM*

OpenVSLAM is a simultaneous mapping and localization algorithm designed to be as easy as possible to use as part of a larger system and to be invoked by third-party applications. It incorporates functionality allowing automatic trajectory calculation and recording in a number of formats. In addition, maps that are created by OpenVSLAM can be saved as msgpack files and then used by other applications.

Due to a license issue and the subsequent discontinuation of OpenVSLAM development, a fork of the original development relying on public support, called StellaVSLAM, is used in this thesis. Everything said about OpenVSLAM applies to the fork.

There are many similar algorithms - in particular, ORB-SLAM, LSD-SLAM and DSO [] represent the main approaches considered as de-facto standards of visual SLAM algorithms having achieved state-of-the-art results using benchmarking data, such as EuRoC, TUM and KITTI. However, they are not suitably designed in terms of usability and extensibility as visual SLAM libraries. Thus, researchers and engineers need to make great efforts to implement these SLAM systems in their applications. In other words, it is inconvenient to use existing opensource visual SLAM software as the basis of applications arising from 3D modeling and mapping techniques, such as autonomous control of robots and drones.

One of the notable features of OpenVSLAM is that the system can handle different types of camera models, such as perspective, fisheye, and equirectangular. At the same time, fisheye cameras are often mounted on drones and robots for visual SLAM and scene understanding as they have a wider field of view (FoV) than perspective cameras. OpenVSLAM can be used with nearly equal implementation between the perspective and fisheye camera models.

Most of the visual SLAM systems cannot store and load card databases. Localization based on a pre-built map is important from a practical point of view for many applications. Accordingly, it is clear that the ability to store and load created maps improves the usability and extensibility of a visual SLAM framework. Therefore, map database input/output functions have been implemented in OpenVSLAM

OpenVSLAM uses ORB for feature extraction. The software is roughly divided into three modules - tracking, mapping and global optimization modules. The tracking module estimates the camera pose for each frame that is sequentially input into OpenVSLAM by key-point matching and pose optimization. This module also decides whether to insert a new keyframe (Keyframe, KF) or not.

When a keyframe is deemed suitable for a new KF, it is sent to the mapping and global optimization modules. In the mapping module, the new 3D points are triangulated using the inserted KFs; i.e., a map is created and expanded. Furthermore, local bundle adjustment (BA) is performed in this module. In the global optimization module, contour detection, pose graph optimization and global BA are performed.

The system is intended to run on Linux operating systems with Ubuntu distribution. To run the algorithm 3 components are needed:

- A video recording in MP4 file format
- A configuration file in YAML format describing the settings and intrinsic parameters of the camera used for the recording
- Bag-of-Words dictionary in FBoW or DBoW2 format provided with the algorithm installation

The trajectory is extracted from a particular video using key frames (KF). It can be recorded in the so-called TUM (Technical University of Munich) format. It is a comma-separated value (CSV) file with each line describing a single camera pose by the following values:

- t – elapsed time according to UNIX standard
- X, Y, Z – the position of the optical centre of the camera relative to the coordinate system defined by the moving system

- QX, QY, QZ, QW – the orientation of the optical centre of the camera in the form of a quaternion with three imaginary and one real part

The trajectory recording is enabled before the video recording is loaded with an additional parameter to the command invoked from the terminal, specifying a directory for recording the resulting files.

IV. REFERENZEN

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (references)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.