A Deep Learning Approach to Camera Pose Estimation

Federico Izzo
DISI, University of Trento
Trento, Italy
federico.izzo@studenti.unitn.it

Francesco Bozzo
DISI, University of Trento
Trento, Italy
francesco.bozzo@studenti.unitn.it

***Abstract*—The task of camera pose estimation aims to find the position of the camera in an image within a given environment. While different geometric approaches have already been studied in the literature, the aim of this project is to study the performances of deep learning models for the camera pose estimation problem. In this work, we analyze models for both relative camera pose estimation (MeNet) and absolute camera pose estimation (PoseNet, MapNet). Moreover, we propose a pipeline for the generation of a ground truth dataset based on structure from motion techniques (COLMAP). Finally, we (1) show how the proposed framework has been used to build a dataset of the second floor of the Povo 1 building in the University of Trento, (2) train an absolute pose estimation model with PyTorch, (3) and deploy it through a web dashboard using FastAPI. The deep learning approach could give interesting results in combination with geometric methods, especially for: relocation after lost tracking, closed-loop detection, better dealing with moving objects in the scene.**

***Index Terms*—camera pose estimation, COLMAP, deep learning, vision**

I. Introduction

The *camera pose*, referenced also with *camera extrinsics*, can be expressed as a combination of two components:

1) a tuple of three elements that identifies the absolute coordinates x, y and z in a reference space:

$$x_c = (x, y, z) \quad x, y, z \in \mathbb{R} \tag{1}$$

2) a quaternion of four elements that identifies the rotation of the camera:

$$q_c = (qw, qx, qy, qz) \quad qw, qx, qy, qz \in \mathbb{R} \tag{2}$$

Consequentially, the pose is referred as $p_c = (x_c, q_c)$.

It is important to notice that this is not the only available representation of a pose: other methods are based also on rotation matrices and Euler angles. It is worth specifying that even if Euler angles are the most straightforward and efficient in terms of memory consumption, they suffer from of the Gimbal lock problem. Even if rotation matrices guarantee a good representation, they are more memory expensive (9 values) than quaternions (only 4 values): for this reason the latter form is preferred here.

Given an image I_c captured by a camera C, an absolute pose estimator E tries to predict the $3D$ pose orientation and location of C in world coordinates, defined for some arbitrary reference $3D$ model. The *absolute pose estimation (APE)* problem can be formally defined as the problem of estimating a function E taking as input an image I_c captured by a camera C and as output its respective pose:

$$E(I_c) = (x_c, q_c) \tag{3}$$

Apart from APE, a popular task is also *relative pose estimation (RPE)*. In this kind of approach the estimator takes two images I_c^1 and I_c^2 captured by C and aims to predict the relative pose between them. In this case, the formulation of the function E described in eq. (3) is a little different, since it receives in input two images:

$$E(I_c^1, I_c^2) = (x_c^{rel^2}, q_c^{rel}) \tag{4}$$

where x_c^{rel} is defined as the absolute pose with *coordinates reference system* in I_c^1 or, in an equivalent way, as the translation vector from I_c^1 to I_C^2.

With this work, we show how it is possible to build a deep learning model which is able to learn the function E using a data-driven approach.

II. Related works

In the literature there are many deep learning approaches used to perform RPE and APE: here we focus on MeNet for the first and PoseNet and MapNet for the latter.

APE deep learning models rely mostly on *transfer learning*: the idea is to use SOTA vision models to extract features from images and use them to estimate camera extrinsics. The PoseNet model (link to paper) has been the first to be developed following this idea. The starting network for the knowledge transfer was a GoogLeNet (link to paper), where the softmax classification layer is replaced with a sequence of fully connected layers. Even if the obtained results are decent, the model lacks of generalization when applied to unseen scenes.

In order to solve this problem, other techniques have been developed, which can be classified in:

- *end-to-end* approaches;
- *hybrid* approaches.

Most of the end-to-end proposed models are based on the PoseNet architecture, with the addition of some components, such as *encoder/decoder blocks*, *linear layers*, and *LSTM* *blocks*. The most successful model on this category is MapNet and related variants MapNet+ and MapNet+PGO (link to the paper).

Hybrid approaches instead try to focus on different support tasks with the goal of helping the final pose prediction. Those techniques rely on unsupervised learning, 3D objects reconstruction and other data extracted with external tools: for this reason those methods are under the scope of out work.

III. Dataset generation

A. Tested approaches

The deep learning approaches explained in this document are *supervised learning* techniques that require a labeled dataset. Several paths were tested in order to generate this kind of dataset:

- *IMU sensors*: usage of gyroscope and accelerometer sensors of a smartphone to estimate the position of the camera during a video given a fixed origin point.
- *digital video*: usage of free online 3 dimensional datasets in which video can be recorder in a digital way.
- *motion capture system*: usage of a motion capture system that estimates the camera position following some tracking objects attached to the subject.
- *structure from motion techniques*: techniques that compute a sparse and dense reconstruction from a sequence of images.

The main problem encountered with IMU sensors was the high noise presence during acquisitions, the final signal was very dirty, and the resolution was not acceptable for the dataset generation. A possible solution could have been the usage of a well calibrated hardware used in other kind of contexts.

Most of the 3 dimensional acquisitions available online for free are acquired with *depth sensors* or *LIDAR sensors*, for this reason although the camera pose estimation would not have presented any errors the images would have been at low quality.

The motion capture system is able to follows the position of the tracked objects with extremely precision, the main problematic remains the associated of poses to video captured from the camera held by the tracked subject. Other difficulties involved the calibration of the tool.

The techniques of structure from motion were invented with the goal of generate structures for which a huge amount of photos is available. The overall idea is to feed the algorithm with data in order to extract feature and build a recomposition of the environment. A step required in order to obtain a result is the estimation of the pose of images. These intermediate requirement have been exploited by us to generate a labeled dataset.

B. Pipeline

The implemented pipeline require a video captured by any camera, it is not required any calibration of the sensor. It is composed by several steps:

1) video split: the captured video is split into many frames;
2) structure from motion: images obtained from the previous step are fed into a structure motion tool called *COLMAP*;
3) cross validation dataset: positions obtained during the camera estimation of the reconstruction process are split into three batches: train, validation, test.

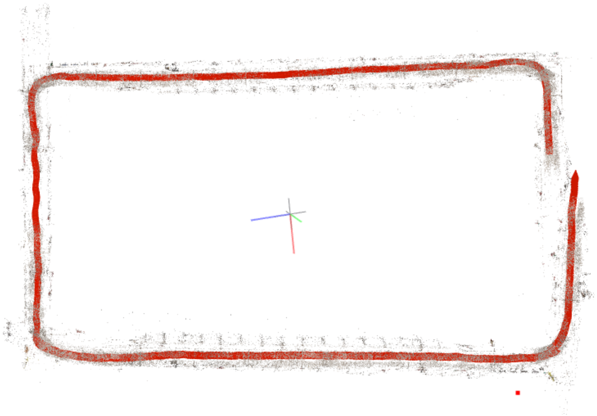

Fig. 1. Trajectory computed by COLMAP

In fig. 1 is presented the trajectory obtained with the structure from motion techinque through COLMAP. The process involes a feature extraction phase, elements obtained are shown in fig. 2.

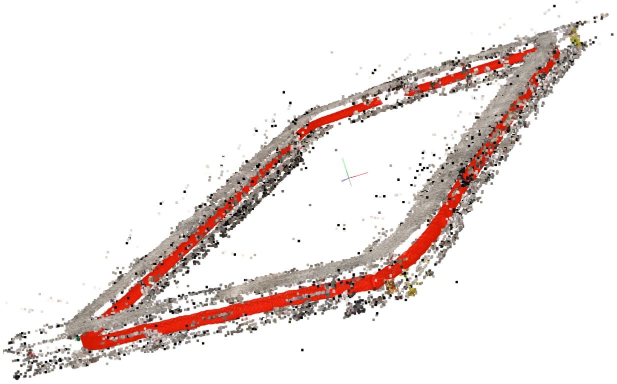

Fig. 2. Features extracted by COLMAP

IV. Models

In this work we took in consideration some models used in the state of the art, also adding small modifications to make them fit better to our use case scenario. In particular, we focused on:

- Menet for RPE;
- PoseNet and MapNet for APE.

A. Menet

The first model we would like to analyze is the MeNet model (fig. 3), which is specifically targeted for RPE. The

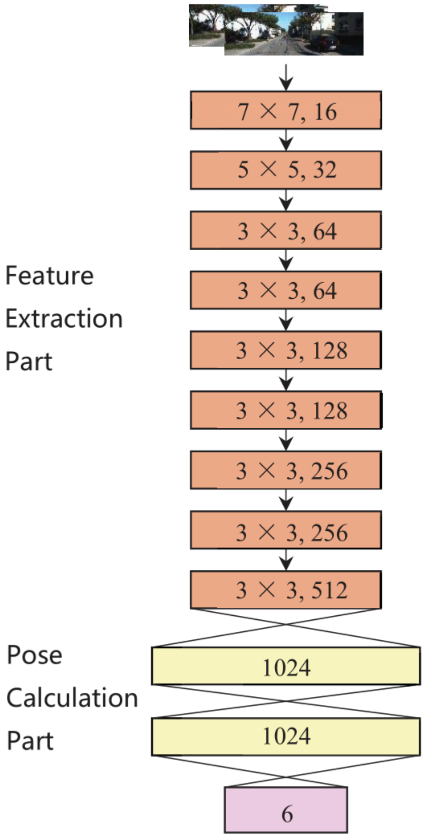

Fig. 3. The architecture of the MeNet model.

input of the network consists in a stack of two images: the goal is to estimate the relative pose of the second image with respect to the first one.

The loss function used is a composition of two Mean Square Errors (MSE) computed separately on the position and rotation. Then they are combined weighting them:

$$Loss(w) = \frac{1}{N} \sum_{i=1}^{N} \left\| P^i - \hat{P}^i \right\|_2^2 + \alpha \left\| Q^i - \hat{Q}^i \right\|_2^2 \tag{5}$$

where the P is the translation, Q the rotation and α the weight for balancing the displacement error and the rotation angle error.

B. PoseNet

The network is based on the ResNet architecture (reference) . . .

C. MapNet

The MapNet model for APE represents an evolution of the PoseNet model: in fact, the model architecture remains actually the same. On the contrary, the main difference between the PoseNet is the loss function used to train the model. In this case, the errors in the prediction of absolute poses are not the only ones which are penalized: also errors in the relative poses are taken in consideration.

The size of the last linear block depends on the dimension of the map that we would like to introduce.

V. Results

A. Posenet

Several pretrained models can be used as features extractor in the PoseNet structure. In table II are presented the most powerful ones for features extraction tested on the same final linear encoder. The overall trend is similar, this highlights that the extracted features are enough for the task independently from the backbone used.

TABLE I
PoseNet backend comparison

Model	Position err.	Rotation err.	Params	Tr.[a] params
GoogleLeNet	0.781	0.119	-	-
ResNet-18	0.635	0.288	11,180,103	3,591
ResNet-34	0.632	0.223	21,288,263	3,591
ResNet-50	0.707	0.191	23,522,375	14,343
ResNet-152	0.594	0.139	58,158,151	14,343
EfficientNet-B7	0.817	0.132	63,804,887	17,927

[a]Trainable

B. MapNet

Several pretrained models can be used as features extractor in the MapNet structure. In table II are presented the most powerful ones for features extraction tested on the same final linear encoder. The overall trend is similar, this highlights that the extracted features are enough for the task independently from the backbone used.

TABLE II
MapNet backend comparison

Model	Position err.	Rotation err.	Params	Tr.[a] params
ResNet-18	0.202	0.0658	14,853,703	3,677,191
ResNet-34	0.187	0.0757	24,961,863	3,677,191
ResNet-50	0.220	0.0969	30,330,951	6,822,919
ResNet-152	0.233	0.0869	64,966,727	3,677,191
EfficientNet-B7	0.210	0.0848	71,658,455	7,871,495

[a]Trainable

Another point that emerges is the importance of the final encoder, it works in a similar way of the *bag of words* used by structure from motion (link al paper). Extracted features are mapped in a space that is used later as a comparison tool for new images for which the pose is asked. For this reason the final encoder was modified from the original one (link al paper) in order to increase the latent space in which data can be stored.

C. Comparison

D. Dashboard

A dashboard was developed with the aim to easily allow users to interact with model inference through a webserver. In fig. 4 is presented the *UI* where red zones are not walkable areas.

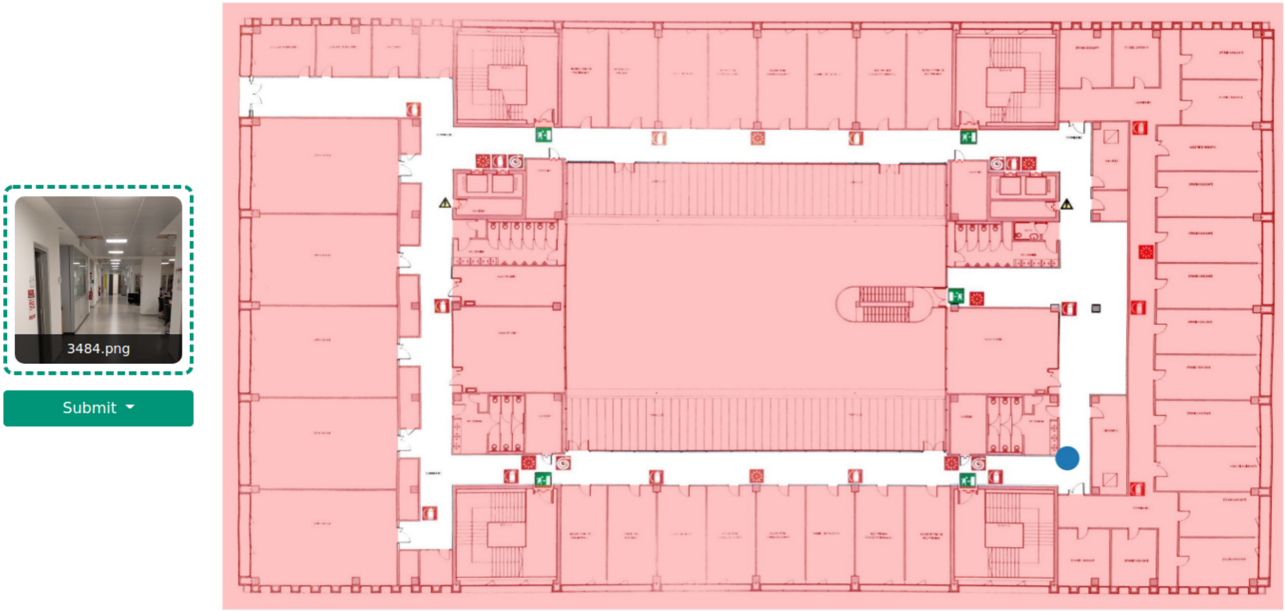

Fig. 4. Inference dashboard

VI. Materials

Every material used in the project have been uploaded respectively:

- the datasets have been uploaded on the Google Drive folder;
- the code is available in the GitHub repository.

The project has been developed in Python 3, using common data science libraries, such as numpy, pandas, PyTorch, matplotlib, scipy, and many others.

A. Repository organization

The repository follows the structure:

- `camera-pose-estimation/`
  - `model/` contains everything related to the deep learning part of the project. It also includes the code used for implementing the web server under `webserver.py` and `static/`.
  - `tools/` contains scripts used for the dataset generation pipeline.
- `config_parser/`: Python package written by us that allows to create configuration files, with the idea of improving reproducibility in our experiments. Each configuration file can be subdivided in sections: for each section you can define variables with the sintax `label=value`, where `value` is a parsable JSON object (boolean, int, float, list, object).
- `notebooks/` contains some Python Jupyter Notebooks that have been used for data exploration, validation, and post-processing of the model predictions.

B. Data organization

For each footage, a folder has been created:

- `imgs/` contains the video frames exported with ffmpeg;
- `processed_dataset/` contains the train, validation, and test datasets that can be reused during different trainings: this helps speeding up the loading procedure from . . . minutes to . . . seconds;
- `workspace/` contains the models generated by COLMAP;
- each of `train.csv`, `validation.csv`, and `test.csv` contains a table for specifying the pose for each image frame. This are the files generated with the `video_to_dataset.sh` script.

VII. Conclusion

VIII. Ease of Use

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

IX. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections IX-A–IX-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: "Wb/m^2" or "webers per square meter", not "webers/m^2". Spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25". Use "cm^3", not "cc".)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{6}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use "(6)", not "Eq. (6)" or "equation (6)", except at the beginning of a sentence: "Equation (6) is . . ."

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [7].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 5", even at the beginning of a sentence.

TABLE III
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

Fig. 5. Example of a figure caption.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[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.

[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.