[[1]](#footnote-1)

ENG 7854 Design Project Final Report Webcam Whiteboard

Michael Paul Noseworthy (200620524), Mitchell Charles Chaulk (200633006), *Student* Members, IEEE

*Abstract*—This paper describes the design of Webcam Whiteboard; an application developed using techniques learned in ENG 7854 – Industrial Machine Vision, and uses laptop webcams to perform drawing operations on a virtual canvas.

*Index Terms*—Digital cameras, Webcams, Color tracking, Image motion analysis, Object detection, Computer graphics.

# INTRODUCTION

T

HIS document describes Webcam Whiteboard, a particular application of industrial machine vision technology using laptops equipped with webcams. This project used MacBook Pros running MATLAB equipped with iSight cameras to perform image subtraction, image filtering, image thresholding, object recognition, and data plotting operations. Webcam Whiteboard tracks the location of red, green, or blue whiteboard markers. Depending on the distance of the marker from the webcam and the marker’s position, Webcam Whiteboard will draw coloured lines based on the colour of the marker on a virtual canvas.

This application may be used to inexpensively bring whiteboard functionality to users in situations where whiteboards are impractical. For example, a user may use Webcam Whiteboard to quickly draw a diagram while seated on an airplane or train. Webcam Whiteboard also supports the saving of images for later review; this functionality is impossible when using traditional whiteboards, as they must be erased once they become full.

# Initial Project Proposal

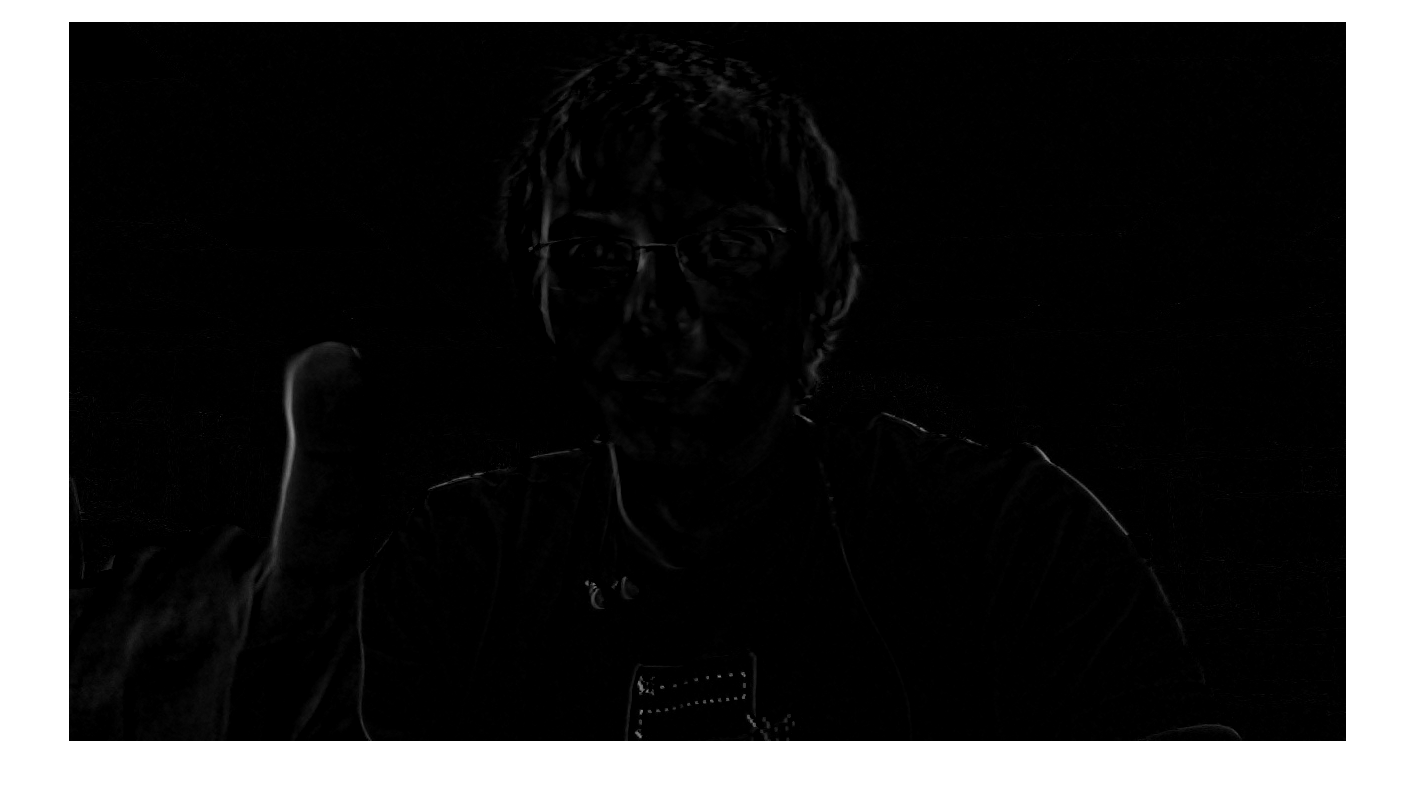


Fig. 1. Results from motion tracking through successive image differencing. Notice the index finger prominently outlined in the foreground left of the image, also notice the noise introduced from the user shifting in his chair.

## Concept Stage

At the beginning of the Spring 2011 semester we were challenged to conceive of and implement a realistic application of machine vision related to industrial inspection, grading, tracking, object recognition, metrology or robotics. A number of potential projects were considered but eventually we settled on the development of an object tracking application. We decided that a virtual whiteboard application would be a challenging and novel application with real world applicability.

It was determined that whiteboards are large, cumbersome, and relatively expensive. However, whiteboards are an extremely effective means of conveying ideas to others using diagrams and figures. We proposed to develop an application that would allow users to take advantage of the functionality of webcams while only being equipped with a webcam and a laptop.

The initial proposal described a system that would use a webcam and a laptop to track a user’s hands to create and edit virtual whiteboard drawings using hand gestures.

## Initial Design Stage

Motion tracking, background subtraction, as well as various colour tracking techniques were considered as potential solutions to track the user’s hands.

1. Motion Tracking

Motion tracking performed a difference operation on two successive frames. The result of this operation was an image which only contained objects that moved position every frame. This method worked well at detecting the direction of motion from a video feed, but when used on a laptop where the webcam is pointed toward the user, significant noise was introduced by the user shifting in their chair or from slight movement in the background. This method also experienced a slower frame-than we would have preferred. Because of this, the technique was also unsuitable in the case of quick movement. Blur greatly decreased the effectiveness of the image differencing technique. Successive image differencing was also limited in that the object recognition relied on movement. If the user stopped moving their finger, the object was lost.

1. Background Subtraction

Background subtraction is similar to successive image differencing, however rather than subtracting two successive images, an initial clean background shot is subtracted from each successive frame. The results from the background subtraction technique were comparable to those obtained from successive image differencing. Noise was generated whenever the camera was displaced, the user moved their position in front of the camera, or if there was any movement in the background. The only advantage this method posed over successive image differentiation was that when the user’s finger stopped moving, the object was not lost.

1. Colour Based Tracking

Using a captured image of a finger, an average colour value for human (Caucasian) skin was calculated. Using the red, green, and blue channels, only pixels coming within 10% of this average value were isolated.

Using this technique resulted in a thresholded image where skin tones were highlighted, but shadowing and background objects added too much noide. In addition to noise, it was computationally expensive when being performed in MATLAB, slowing down drawing to an unacceptable rate.

## Final Design Scope

After researching and experimenting with a number of different techniques for tracking a user’s hands, it was determined that there is not a simple, effective and computationally efficient method of isolating user’s hands within a captured image.

To solve this problem we decided to modify the scope of our project to include the tracking of specifically coloured objects, and to use these objects for drawing on the virtual whiteboard. We discovered that red, green, and blue whiteboard markers were convenient for this purpose. The markers also worked as excellent pointers for increasing immersion within the application since users were familiar with drawing using a whiteboard marker.

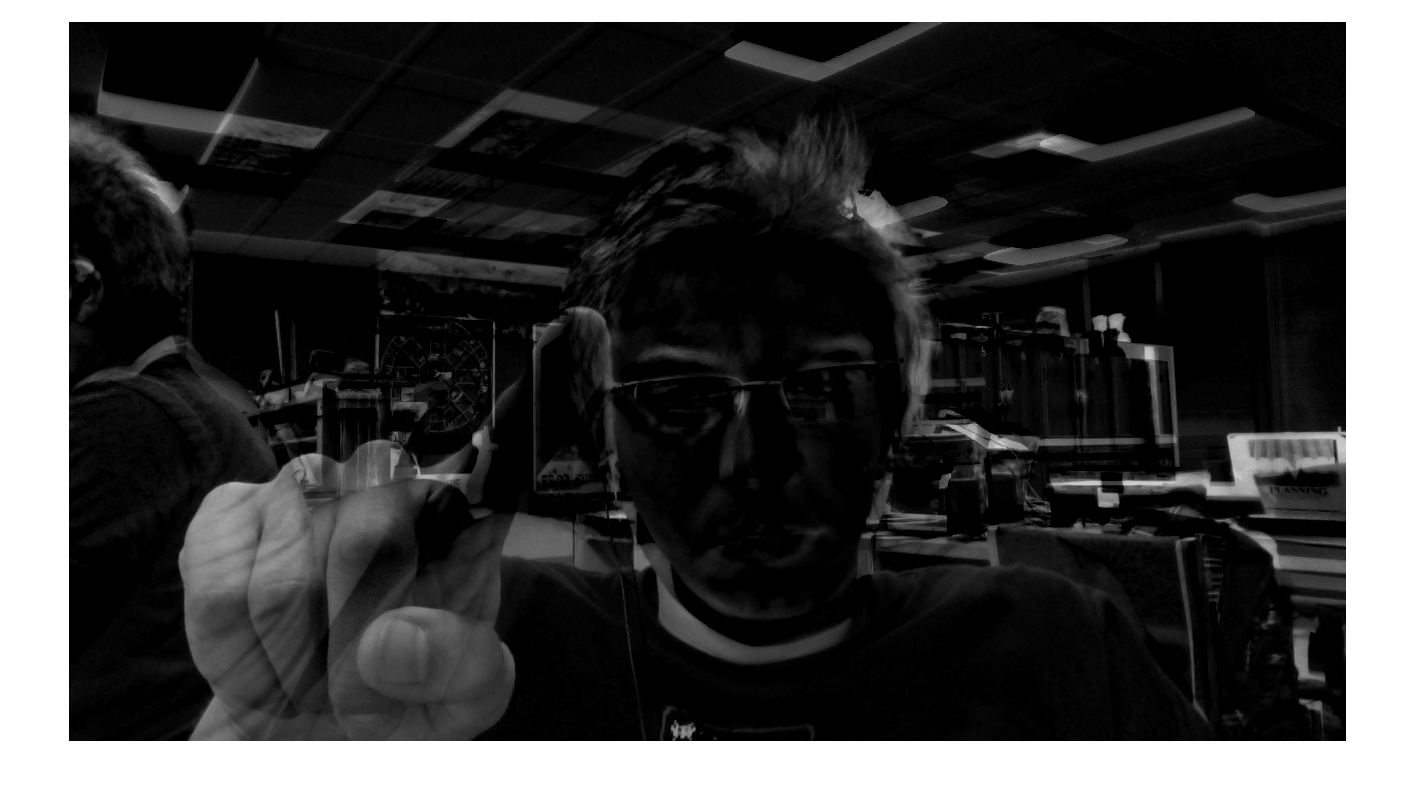


Fig. 2 Resultant image from background subtraction. Notice raised index finger in the foreground left of the image. See also the noise resulting from camera displacement or movement in the background of the image.

# Final Design

After the final scope of the project was determined, application design began. It progressed in a number of phases. Initially, we worked on developing an efficient algorithm for isolating the whiteboard marker colours in the image. Next we proceeded to develop an algorithm to translate marker top centroids to generate lines. We then developed a method for plotting lines onto a virtual canvas.

## Object of Interest Isolation

The object of interest we will be tracking is the brightly coloured top of a whiteboard marker. To do this, we designed an efficient algorithm using MATLAB that extracts a colour component of an image, thresholds the results of the extraction and then performs connected component labeling for blob analysis. Blob analysis is used to determine the object’s centroid, eccentricity, and area.

The first step in this algorithm is the extraction of a particular colour channel (red, green, or blue). Using the MATLAB imsubtract function, we get the difference between the chosen colour channel and a grayscale version of the image. For example; to isolate red objects, you would subtract a grayscale image of the object from the red channel of the original RGB image. The imsubtract function is used because it features optimizations for Intel based architectures. The result of this operation is a grayscale image that is composed of only objects that are primarily of the chosen colour of interest.



Fig. 4 Result of subtraction of grayscale image from red channel. Notice how the red marker tip is the dominant object in the frame.



Fig. 3 Resultant image from skin tone isolation. Notice how the shadows in the image greatly reduce the effectiveness of the filtering, also notice how there is a great amount of noise in the background.

After colour channel subtraction, the frame undergoes median filtering using a 3x3 kernel as to remove any salt and pepper noise from the frame. In practice, this step provides little benefit since the noise in the frame is negligible after the opening method (discussed later), and the median filtering operation can be computationally expensive on large images. Median filtering is included in the final application as an optional feature with the caveat that enabling it would decrease application performance.

Next, to isolate only the marker tip in the grayscale frame, we were required to perform thresholding on the resultant image. Otsu thresholding was attempted as a possible thresholding technique, but the calculation of the threshold on every frame required a significant amount of computing power and greatly decreased the application’s performance. Through experimentation, it was determined that a threshold value of 0.18 (where 0 is black and 1 is white) would provide effective thresholding to isolate the marker top in the frame.

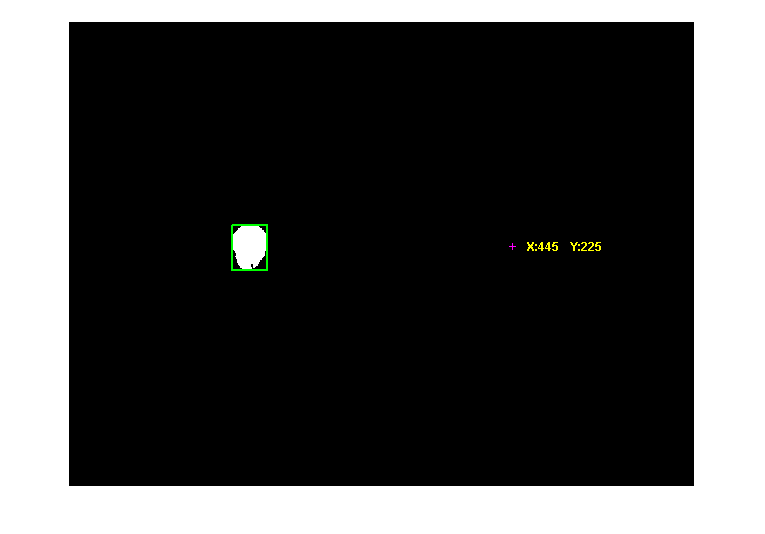


Fig. 7 Bounding box and centroid of the marker tip. The centroid of the object is reflected across the vertical axis of the frame because the camera acts as an observer and not a mirror.



Fig. 5 Image after thresholding at 0.18 was performed. Notice how the marker tip is the dominant feature in the image.

After thresholding, the largest object in the frame in Fig. 5 is the tip of the whiteboard marker. To remove the noise in the image, opening is performed using MATLAB’s bwareaopen function (specifying a minimum area of 500 px2). The result of this operation can be seen in Fig. 6.

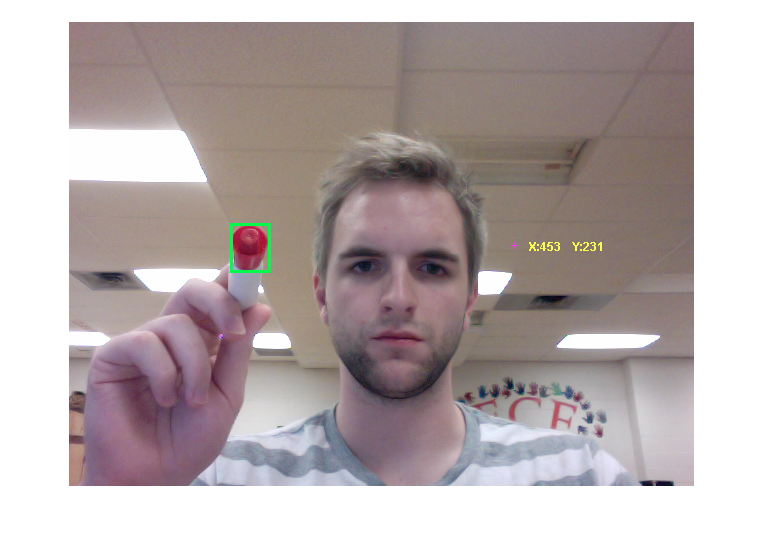


Fig. 8 Bounding box and reflected centroid point are overlaid on the original image. Notice how the bounding box correctly shows the only tracked object in the image is the marker tip.

After the marker tip is isolated in the image, blob analysis is performed to determine the properties of all bright objects in the image. We make use of the MATLAB regionprops function for blob analysis, due to the efficiency optimizations of using MATLAB’s precompiled functions over our own functions. We use regionprops to calculate the blob’s area, centroid, and eccentricity. Though the marker tip should be the only object in the image, there still may be noise present. The marker tip is relatively small in comparison to the image frame, so we ignore any objects in the image that have an area greater than 10 000 px2. Since the marker tip is relatively circular in shape, we also use eccentricity as a criterion to determine which blob is in fact the marker tip. All blobs that have an eccentricity value of greater than 0.55 are ignored. The thresholds for area and eccentricity were determined to be optimal to detect marker tips through experimentation.

Once a blob is determined to be the marker tip, we translate the centroid of the marker tip into points than can be used to plot lines on a virtual canvas.

## Linear Approximation of Points

The centroid points found during blob analysis are calculated during each iteration of the run loop. Even with the centroids being calculated at the fastest possible rate, there will still be breaks between successive points. This is due to the frame rate of the webcam being used, and the time it takes to perform blob analysis. Therefore, it is not practical to simply plot each centroid point calculated, since the result would be an image full of isolated points, not smooth lines. We perform a linear line of best-fit analysis on the centroid points calculated to account for this issue.

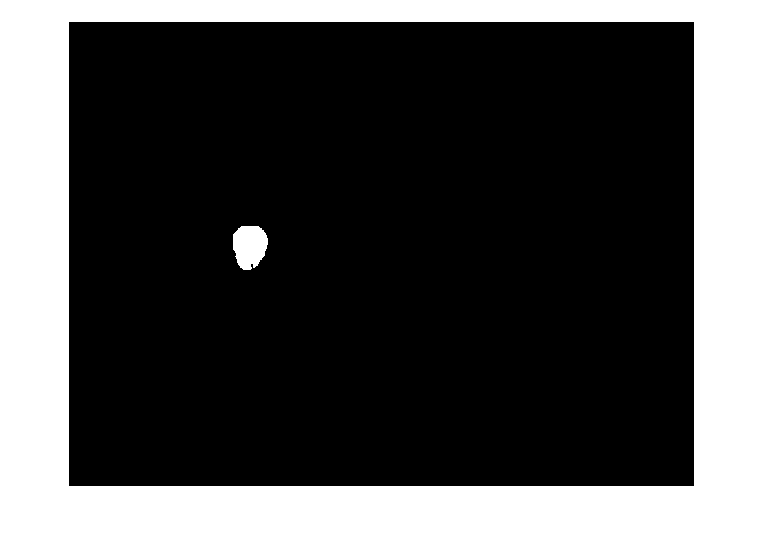


Fig. 6 Image after opening, filtering out images less than 1000 px2. Notice how the only object left in the image is the tip of the whiteboard marker.

We determined that the frame rate of the webcam allowed us to do linear best-fit analysis. Higher order best-fit analysis was not required since the difference between successive centroid points would not be so different that the user could not plot an adequate approximation of a curve.

Least squares analysis was the method originally chosen for performing linear best-fit analysis, but we found it to be unnecessarily complex. We decided that a simple method of determining the slope and y-intercept of a line from two successive points would be an acceptable method for our purposes.

We developed an algorithm using MATLAB that takes two points and calculates integer x and y values for points that connect the two provided points.

## Line Plotting

The final component of the webcam whiteboard application is the virtual canvas. To provide 1:1 representation of the coordinates of the marker position, we use a canvas with the same resolution as the image obtained from the webcam. The canvas is simply a three dimensional matrix representing an RGB image.

The canvas is initially contains only zero values which is a black canvas. We chose to use black as a background colour because it is easier to see the drawn lines with a black background as opposed to white.



Fig. 9 Centroid of the tracked object is represented by the magenta ‘+’. Notice how the user has stopped drawing, but the marker is still being tracked by the application.

Initially we used MATLAB’s plot function to plot each successive centroid point. We found that performance decreased greatly the more the user drew. This was because MATLAB holds the plotted points in a buffer, and redraws the points in this buffer each time the image is updated. Therefore, the more the user drew, the larger this buffer became, and the more performance was impacted. Updating an RGB image matrix, rather than plotting on top of a figure solved this performance issue.

As centroid points are calculated, the x and y points received from our linear approximation algorithm are used to insert full brightness values (255 for an unsigned 8-bit representation) into the matrix. At the end of each iteration of our run loop, this new updated matrix is displayed as an image using MATLAB’s imshow function.

We decided it would be beneficial to be able to track multiple marker colours, and draw using the colour of the marker that the user was tracking. On startup, the user can select what colour marker they intend on using. Once the application is running, there is functionality that allows them to switch between tracking red, green, or blue. These colours are drawn on the canvas by putting full brightness values into the red, green, or blue channels of the matrix – depending on the colour being tracked.

# Final Modifications

After we had a fully functioning prototype, we performed significant testing to improve the usability of the application. We added the ability to stop and resume drawing by lifting the marker away from the camera, and then moving it back towards the screen, respectively. We also added functionality that displays the marker location on the canvas even when the application was not drawing.

We set a threshold on blob area to enable drawing. This provides us with functionality that allows the user to pull the marker away from the screen to stop drawing, allowing them to begin drawing again only when the marker was close enough to the screen. The area threshold value of 2500 px2 was chosen as a lower limit. This allowed users to temporarily stop drawing by pulling the marker away from the webcam. When we were testing this functionality, we found that it was difficult to determine where the marker would begin drawing when the marker was moved closer to the webcam. This was solved by always plotting the centroid of the marker.

Since we removed any blobs that were smaller than 500 px2, and larger than 10 000 px2, and only drew lines when the detected marker area was greater than 2500 px2, there was a significant range where lines were not being drawn, but the marker was still being tracked. We used this gap between tracking and drawing to show users where the marker would begin drawing by plotting the centroid of the marker while it was being tracked. This allowed users to draw more accurately since they knew where the line would start when they moved the marker toward the camera.

# Future Work

MATLAB provides users with powerful tools for image processing and machine vision in their image processing toolbox. However, programs created in MATLAB are not as efficient as compiled programs. Webcam Whiteboard could eventually be ported another programming language or development environment such as C, C++, or Java. Using a compiled computer language would allow us to take advantage of the parallel nature of today’s computer architectures, performing our image analysis faster. The performance enhancements from porting to a compiled language would also allow us to implement further functionality, like the ability to calibrate the system to specify the colour of the object that is to be tracked.

We are going to submit Webcam Whiteboard to MATLAB’s file exchange to allow other MATLAB developers to build onto our existing application.

# Conclusion

We have designed and implemented a practical application that provides whiteboard functionality on a personal laptop equipped with a common webcam. This application utilizes techniques learned in ENG 7854 – Industrial Machine Vision.

Appendix

Appendixes, if needed, appear before the acknowledgment.

Acknowledgment

We would like to acknowledge A. Bhargav Anand who gave us the idea to track objects based on colour channel subtraction from grayscale came. We used his redObjectTrack.m file as a guide in learning how to effectively, and efficiently, track a specific color in MATLAB.

References

1. <http://www.cs.toronto.edu/~smalik/downloads/2503_project_report.pdf>
2. G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
3. W.-K. Chen, *Linear Networks and Systems* (Book style)*.* Belmont, CA: Wadsworth, 1993, pp. 123–135.
4. H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
5. B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
6. E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.
7. J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.
8. C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
9. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].
10. M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.
11. J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility (Periodical style),” *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.
12. S. Chen, B. Mulgrew, and P. M. Grant, “A clustering technique for digital communications channel equalization using radial basis function networks,” *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul. 1993.
13. R. W. Lucky, “Automatic equalization for digital communication,” *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.
14. S. P. Bingulac, “On the compatibility of adaptive controllers (Published Conference Proceedings style),” in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.
15. G. R. Faulhaber, “Design of service systems with priority reservation,” in *Conf. Rec. 1995 IEEE Int. Conf. Communications,* pp. 3–8.
16. W. D. Doyle, “Magnetization reversal in films with biaxial anisotropy,” in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.
17. G. W. Juette and L. E. Zeffanella, “Radio noise currents n short sections on bundle conductors (Presented Conference Paper style),” presented at the IEEE Summer power Meeting, Dallas, TX, Jun. 22–27, 1990, Paper 90 SM 690-0 PWRS.
18. J. G. Kreifeldt, “An analysis of surface-detected EMG as an amplitude-modulated noise,” presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.
19. J. Williams, “Narrow-band analyzer (Thesis or Dissertation style),” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
20. N. Kawasaki, “Parametric study of thermal and chemical nonequilibrium nozzle flow,” M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.
21. J. P. Wilkinson, “Nonlinear resonant circuit devices (Patent style),” U.S. Patent 3 624 12, July 16, 1990.
22. *IEEE Criteria for Class IE Electric Systems* (Standards style)*,* IEEE Standard 308, 1969.
23. *Letter Symbols for Quantities*, ANSI Standard Y10.5-1968.
24. R. E. Haskell and C. T. Case, “Transient signal propagation in lossless isotropic plasmas (Report style),” USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.
25. E. E. Reber, R. L. Michell, and C. J. Carter, “Oxygen absorption in the Earth’s atmosphere,” Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (420-46)-3, Nov. 1988.
26. (Handbook style) *Transmission Systems for Communications,* 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.
27. *Motorola Semiconductor Data Manual,* Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.
28. (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume (issue). Available: <http://www.(URL>)
29. J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: <http://www.atm.com>
30. (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: <http://www.(URL>)
31. R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876–880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar

**First A. Author** (M’76–SM’81–F’87) and the other authors may include biographies at the end of regular papers. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author’s educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year degree was earned. The author’s major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author’s last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author’s title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.

1. Manuscript received July 26, 2011. This work was a requirement for graduation from ENG 7854 – Industrial Machine Vision.

   M. P. Noseworthy is an undergraduate student with the faculty of Engineering and Applied Science at Memorial University of Newfoundland, St. John’s, Newfoundland and Labrador, Canada (michael.noseworthy@mun.ca)

   M. C. Chaulk is an undergraduate student with the faculty of Engineering and Applied Science at Memorial University of Newfoundland, St. John’s, Newfoundland and Labrador, Canada (n74mcc@mun.ca) [↑](#footnote-ref-1)