VBASR: The Vision System

Vision Based Autonomous Security Robot

Bradley University – ECE Department

Senior Capstone Project

Sponsored by Northrup Grumman

May 04, 2010

Student: Kevin Farney

Advisor: Dr. Joel Schipper

BRADLEY UNIVERSITY

NORTHROP GRUMMAN

Presentation Outline

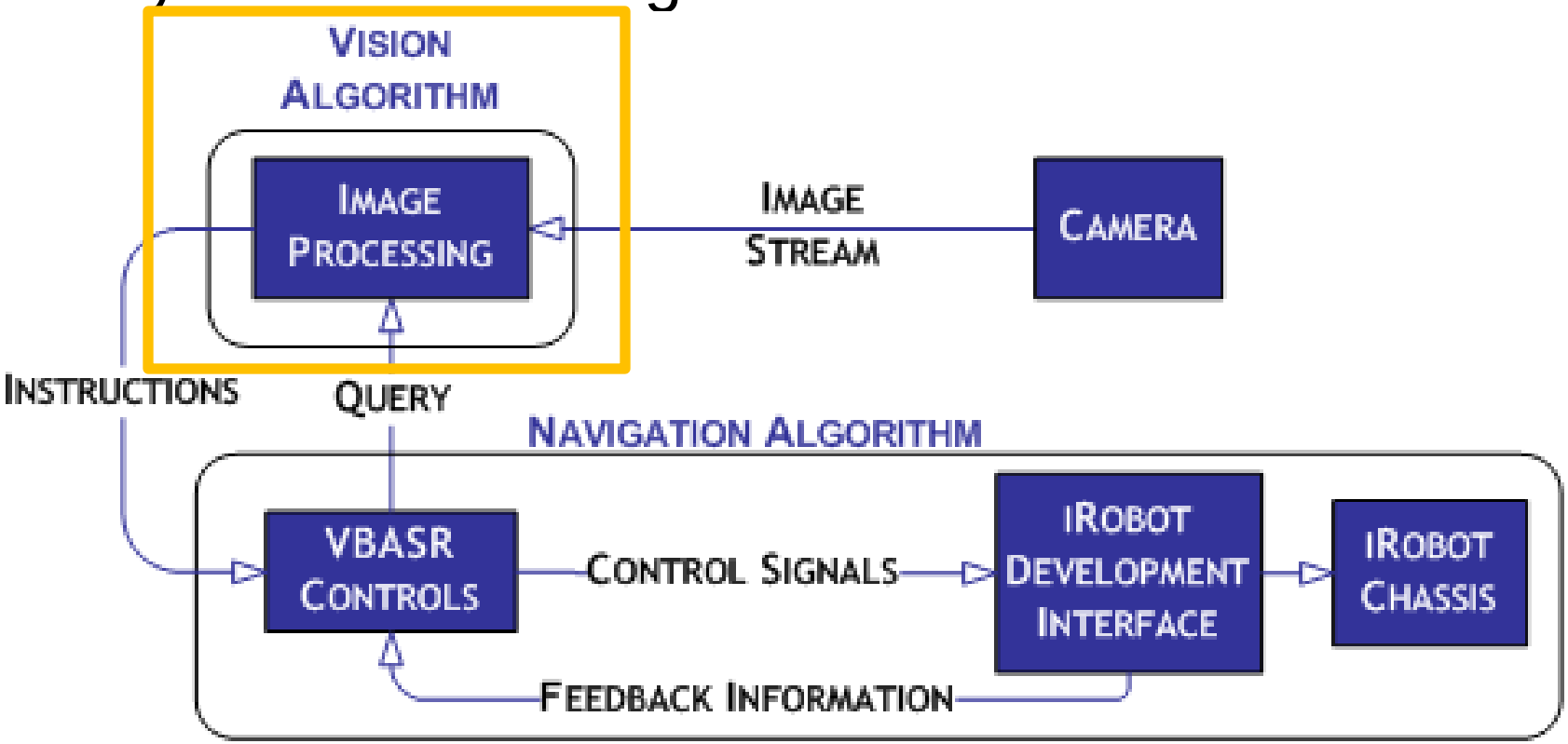
- What the project is…
- What has been completed…
- Results…

Project Summary

- What is VBASR?
  - Autonomous, Mobile, Security Camera
- VBASR is a computer vision project
- Primary Goals – Using Computer Vision
  - Navigation
    - Obstacle Avoidance

Vision Algorithm

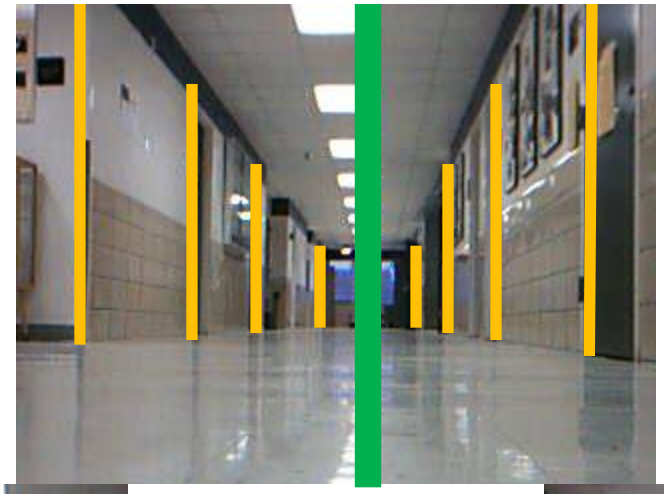
- System Block Diagram

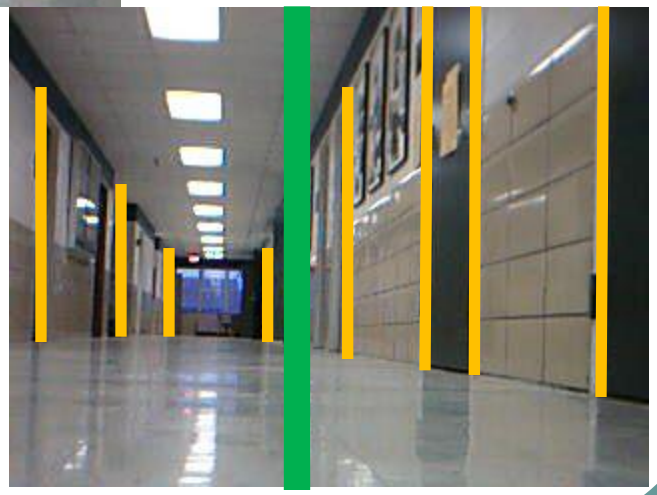


The Platform

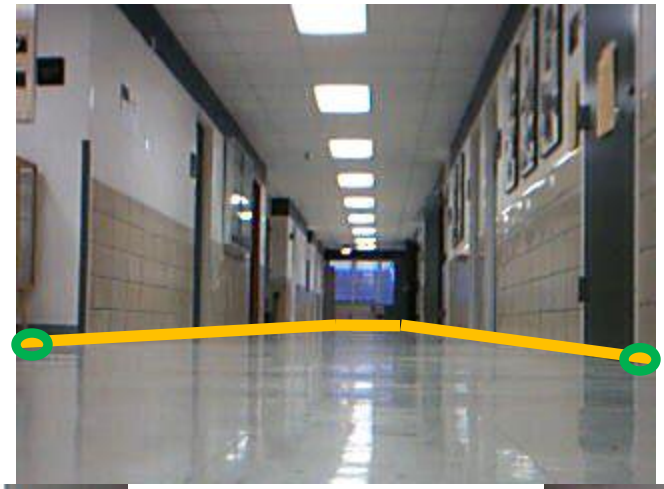
- Hardware
  - iRobot Create
  - Webcam
- Software
  - OpenCV2.0

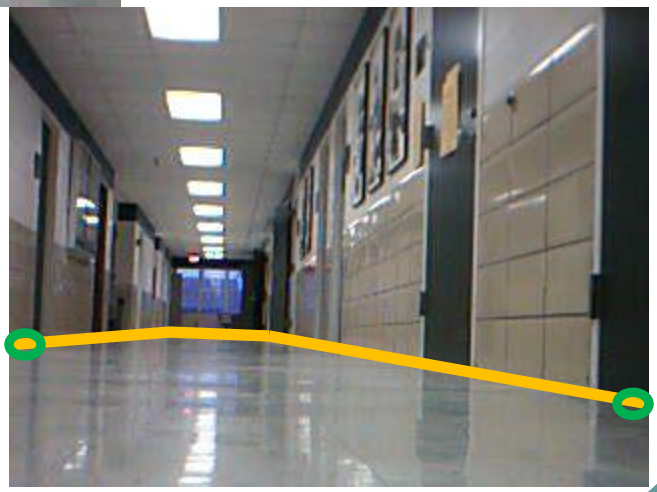
Vision Algorithm – Idea #1



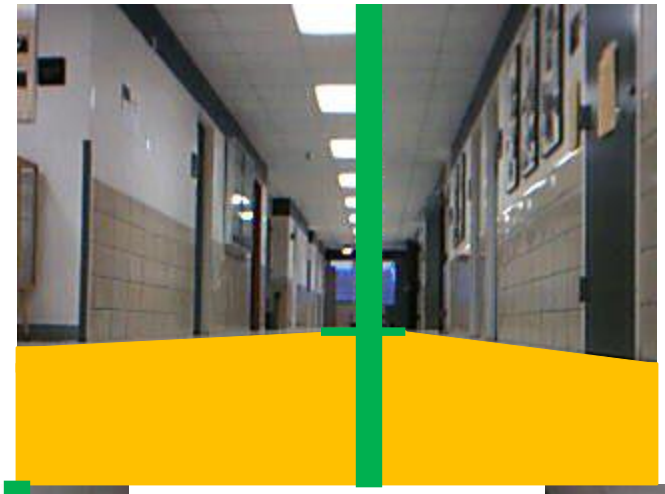


Vision Algorithm – Idea #2

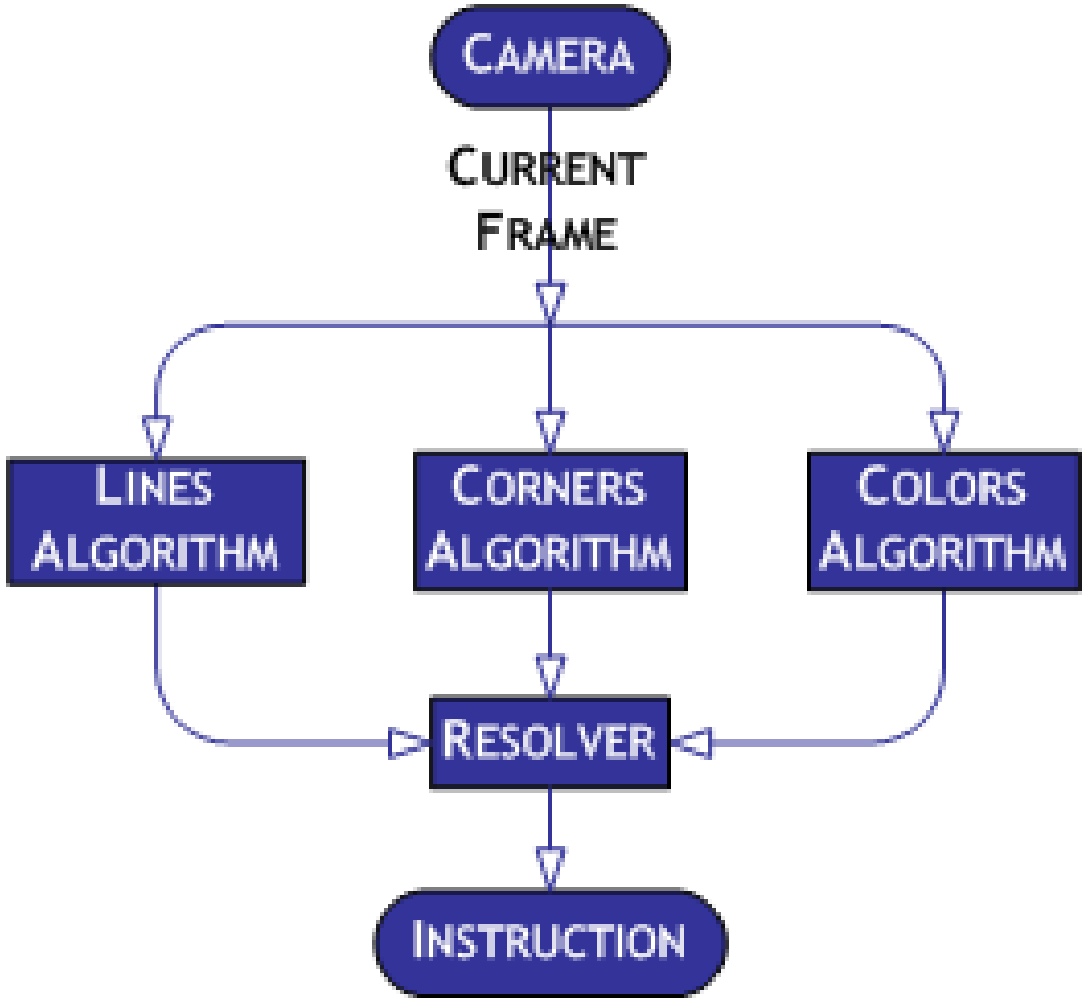




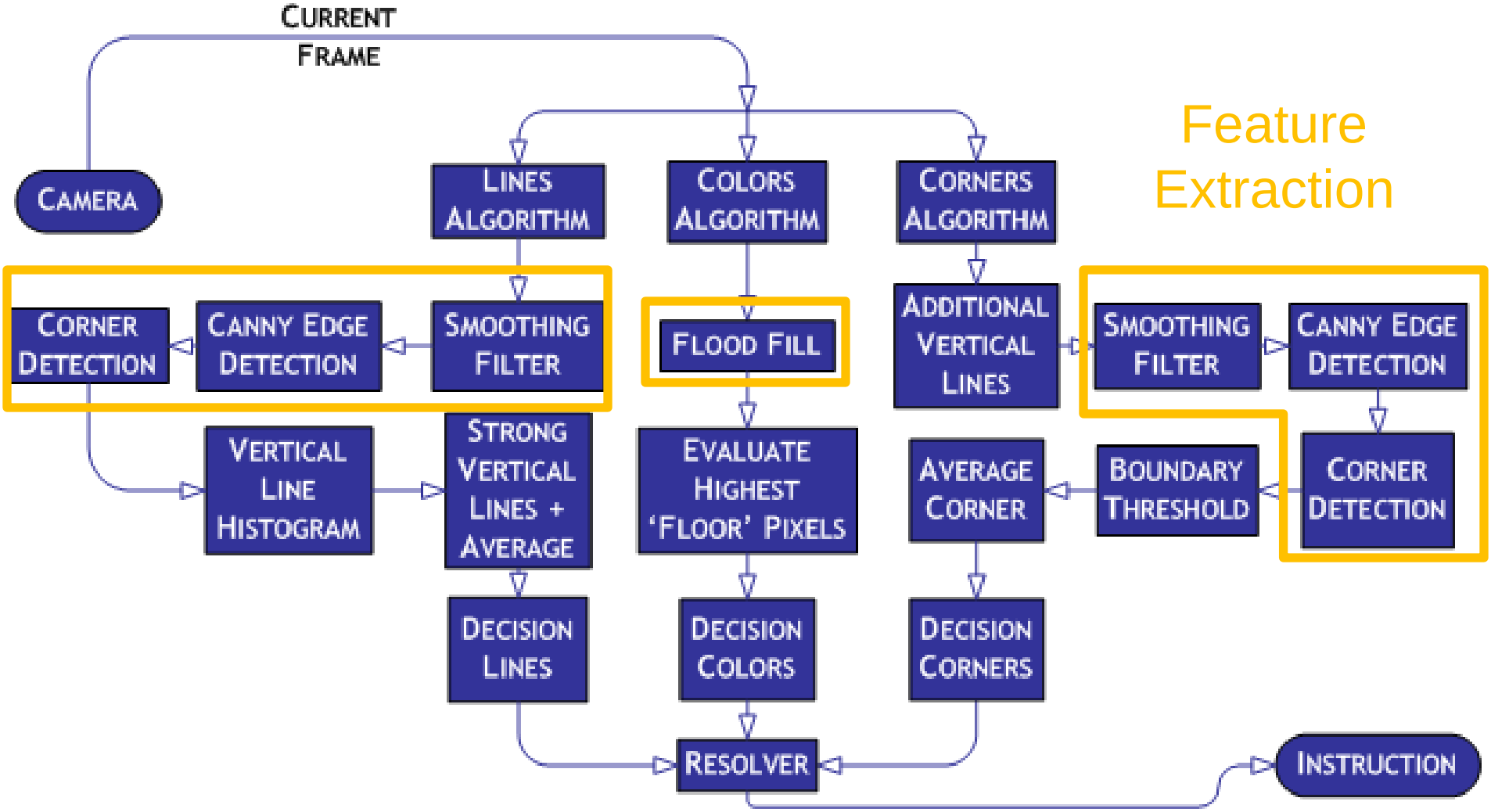
Vision Algorithm – Idea #3



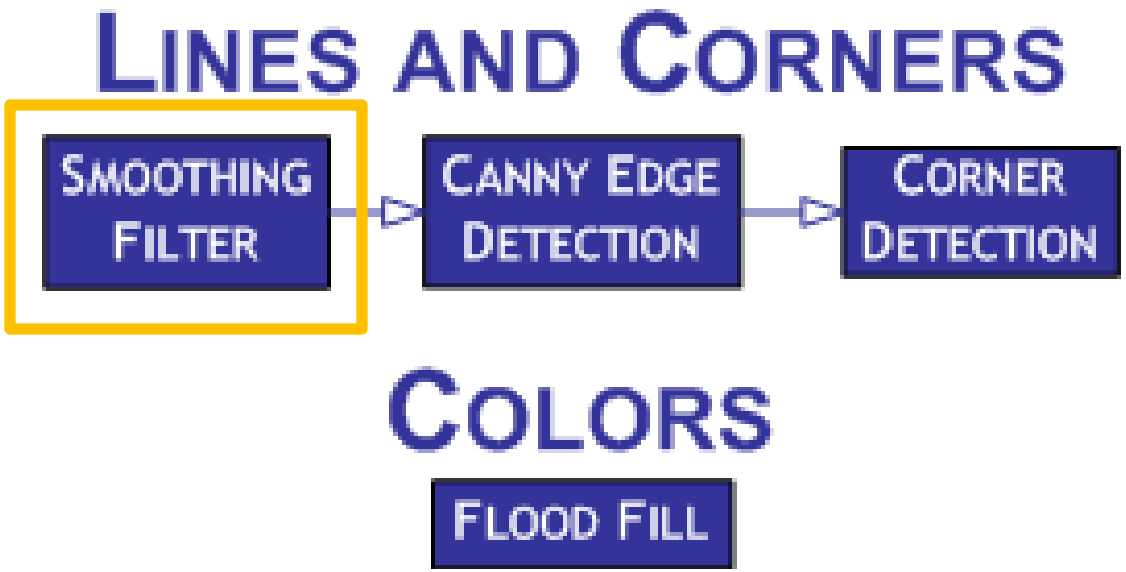
Vision Algorithm – High Level



Vision Algorithm – Detailed



Feature Extraction

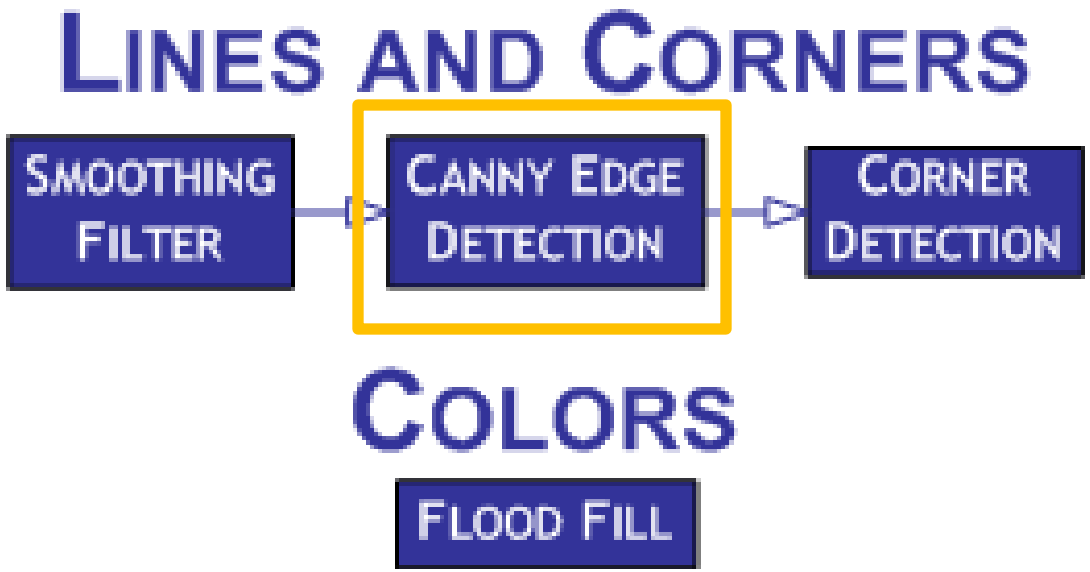


Testing OpenCV - Filters

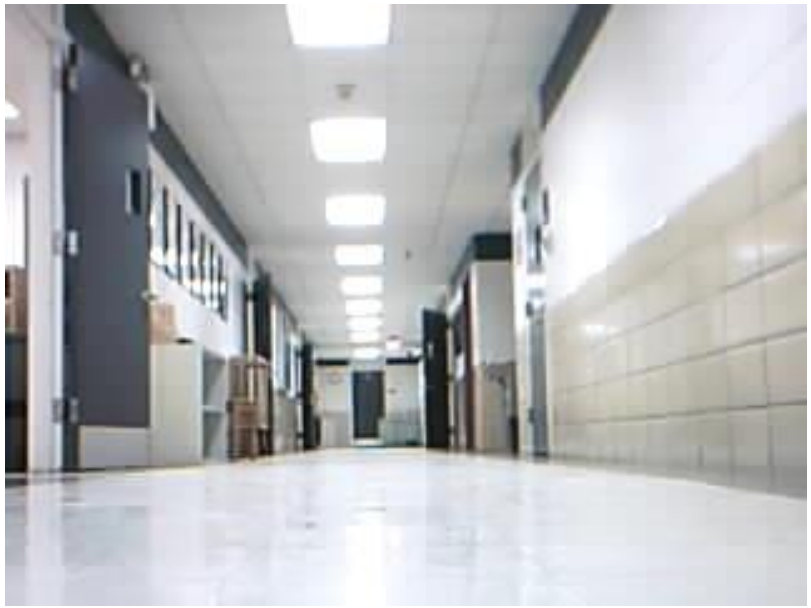
Testing OpenCV - Filters

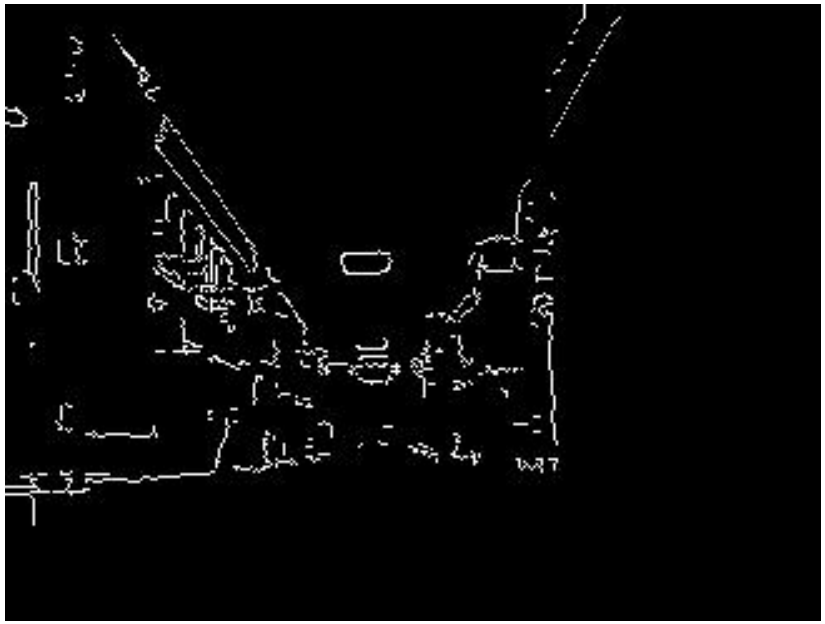
Testing OpenCV - Filters

Feature Extraction



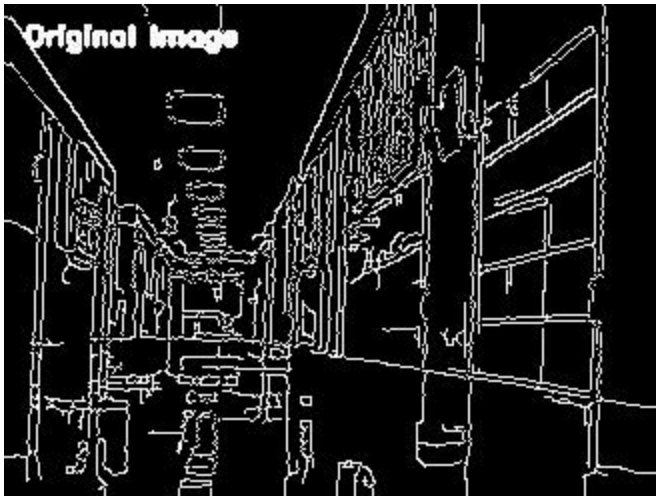
Testing OpenCV - Edge

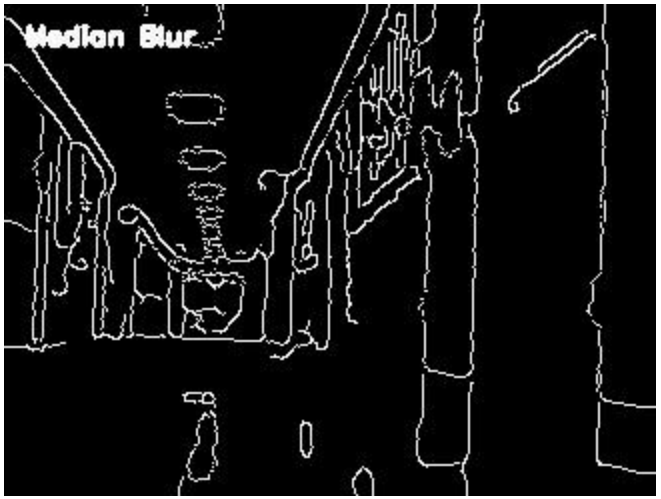




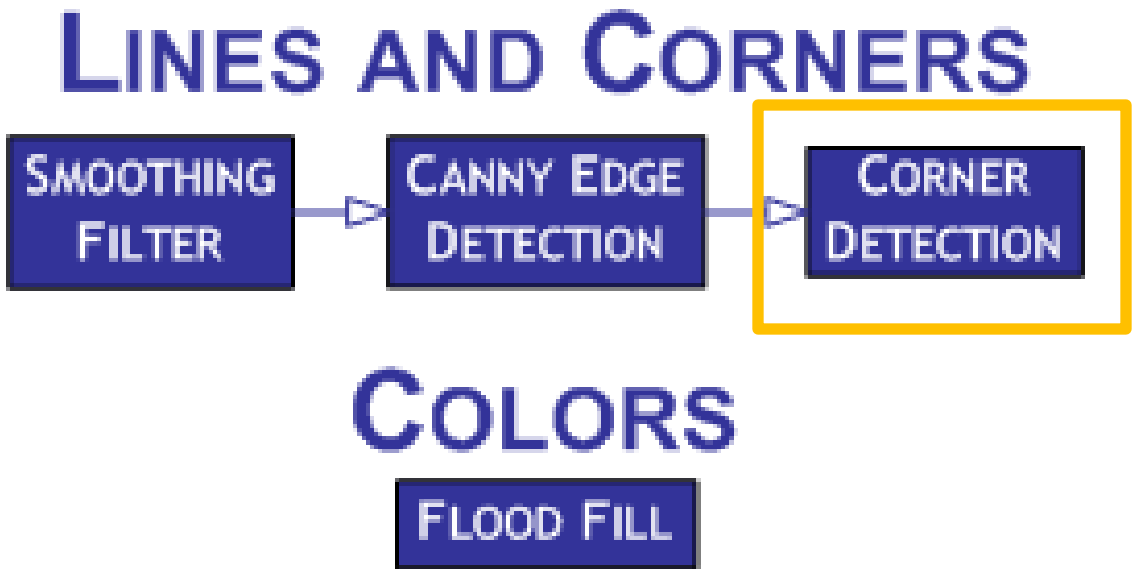
Why Filters?

- Noise Reduction

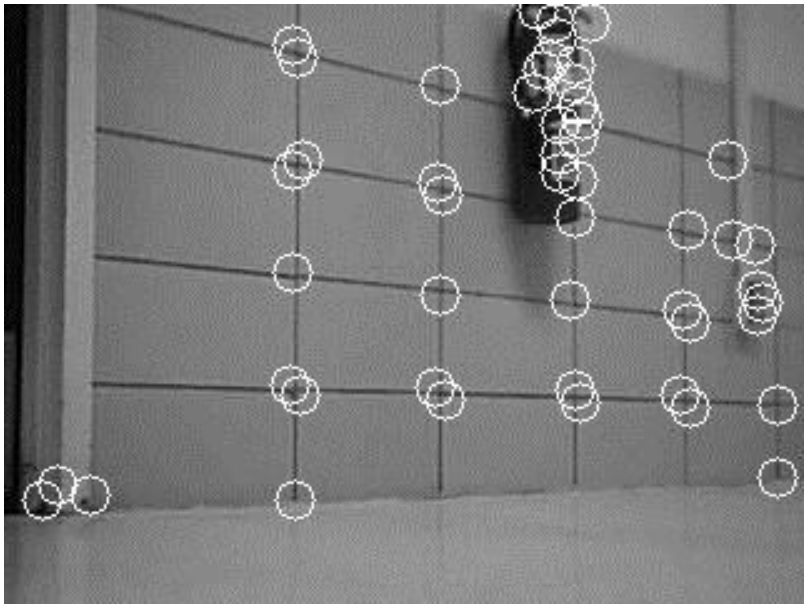




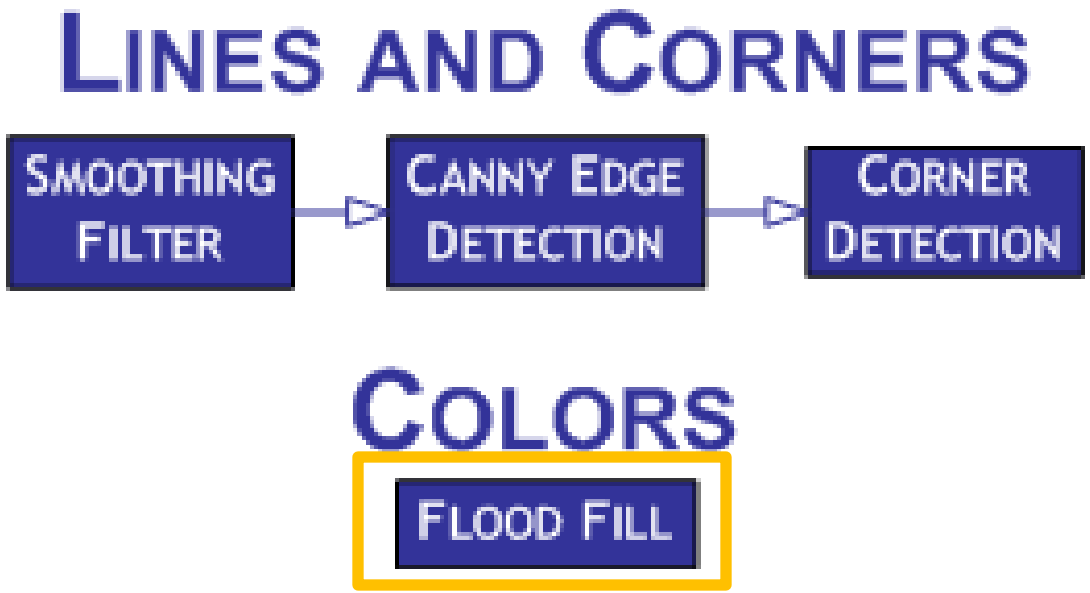
Feature Extraction



Testing OpenCV - Corners

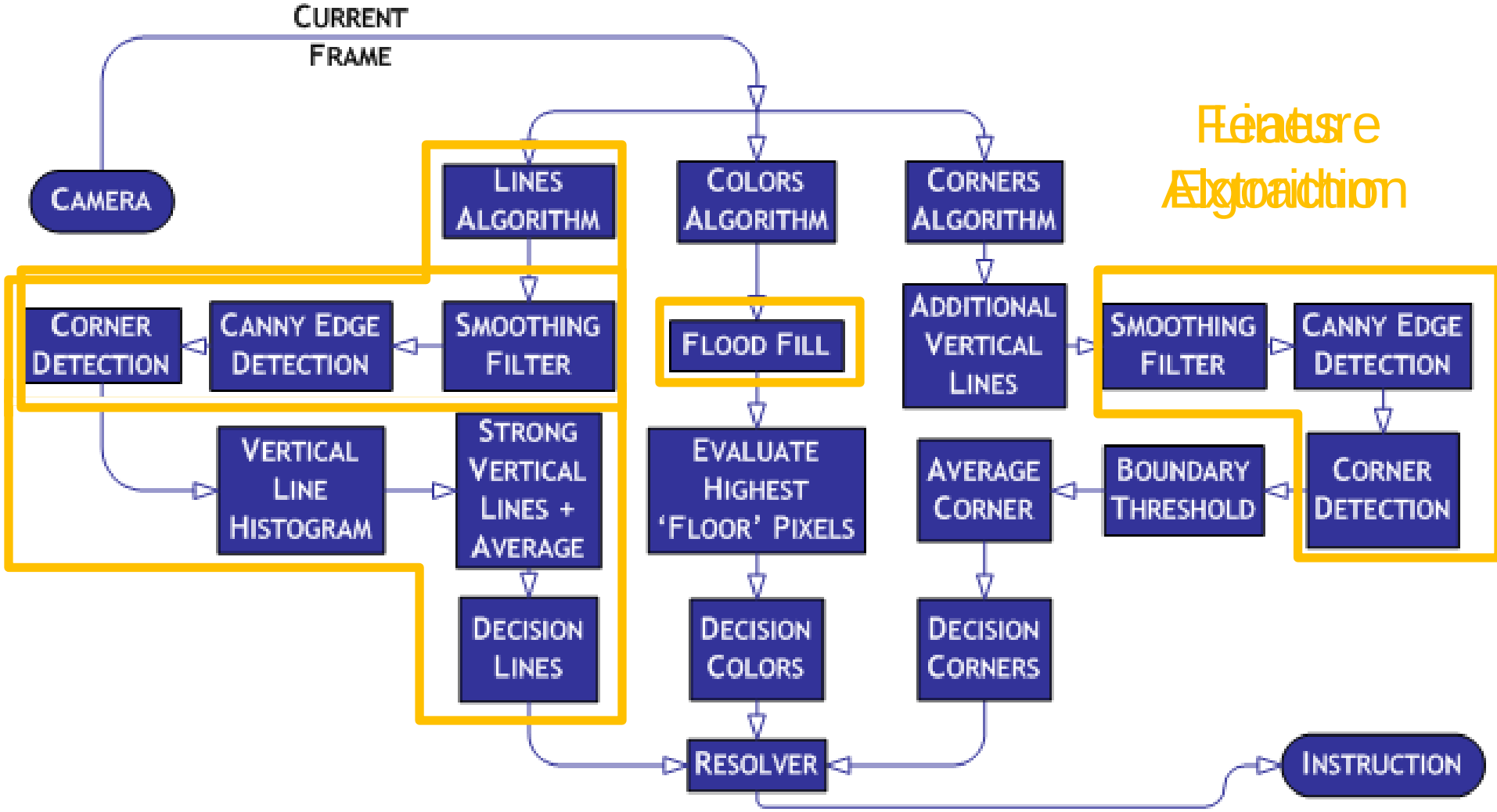


Feature Extraction

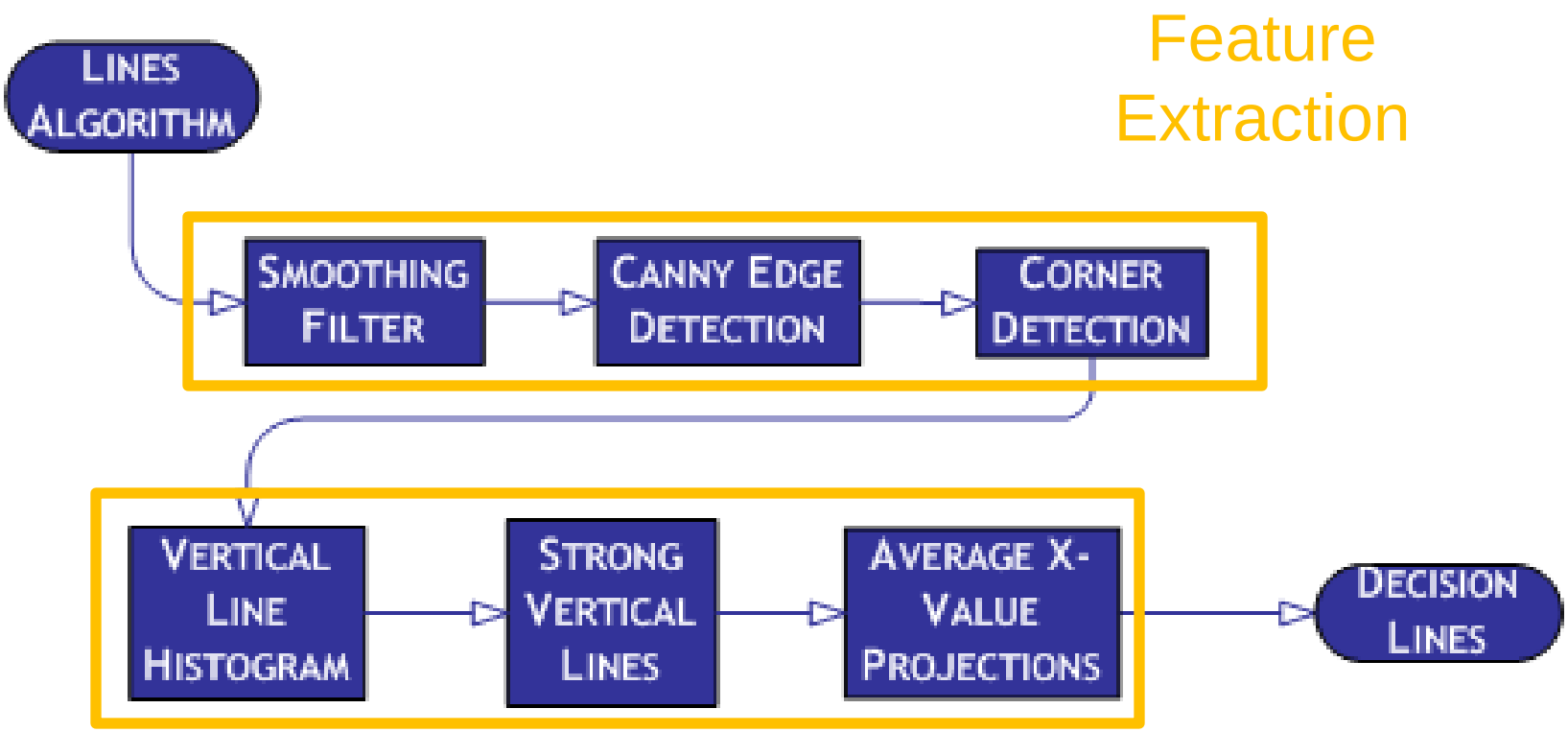


Testing OpenCV – Flood Fill

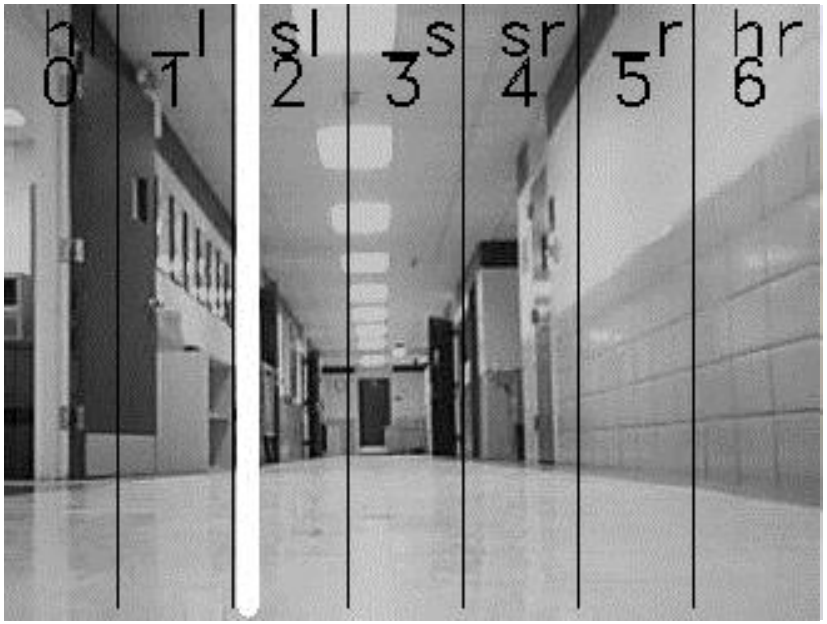
Vision Algorithm – Detailed



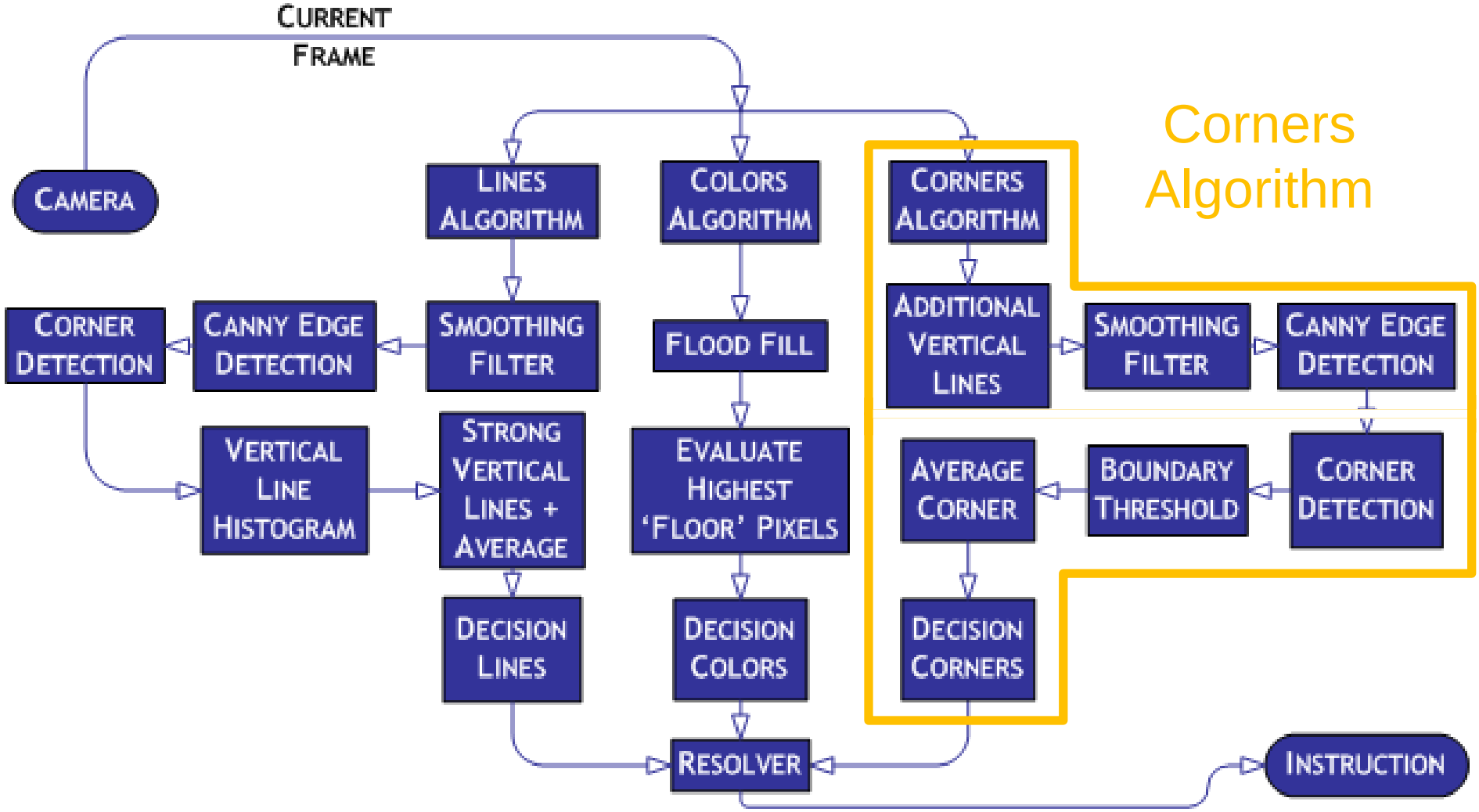
Lines Algorithm



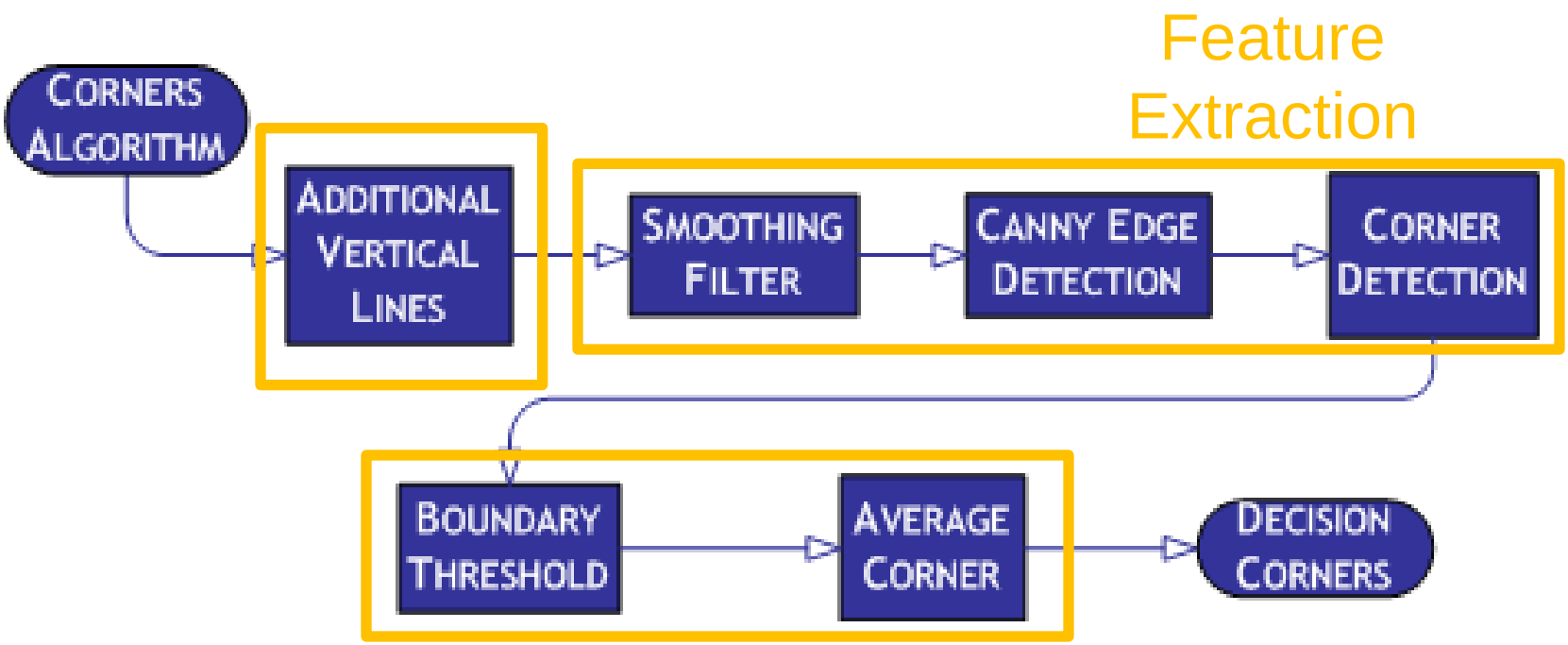
Lines Algorithm



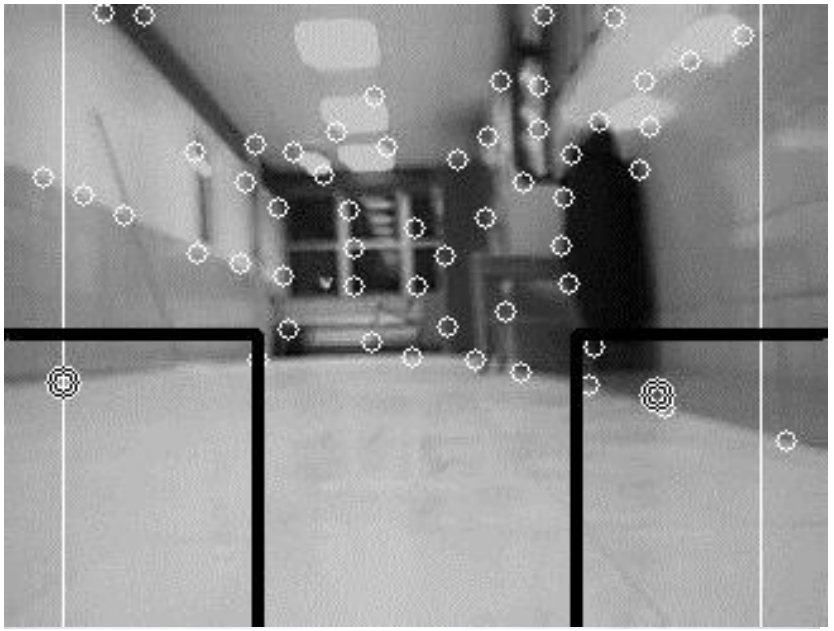
Vision Algorithm – Detailed



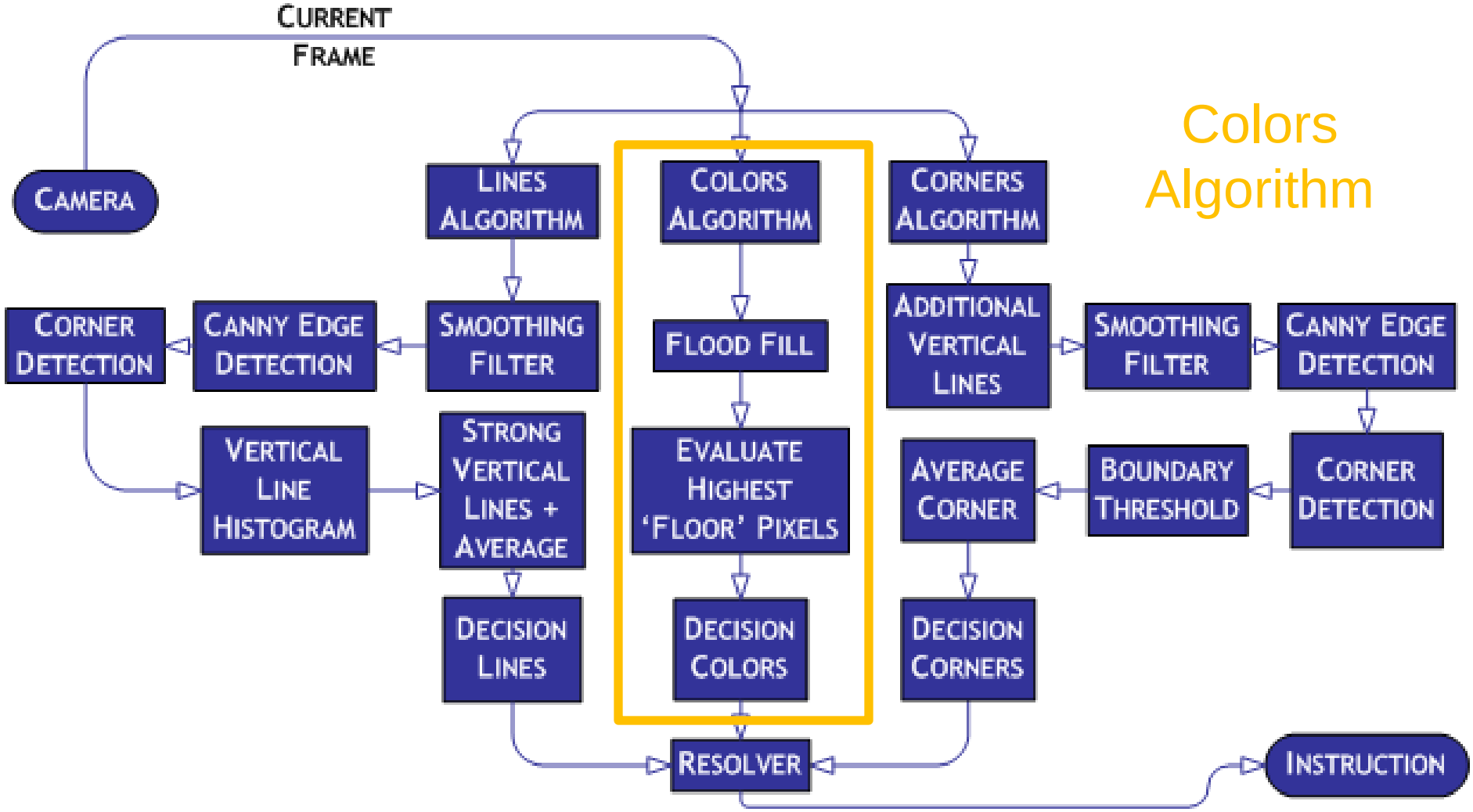
Corners Algorithm



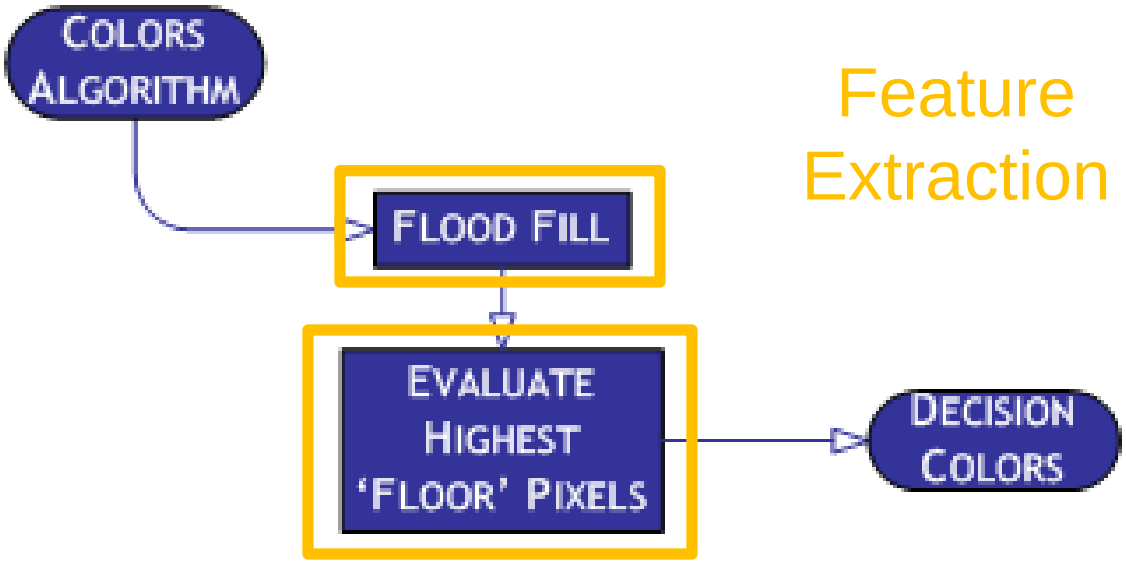
Corners Algorithm



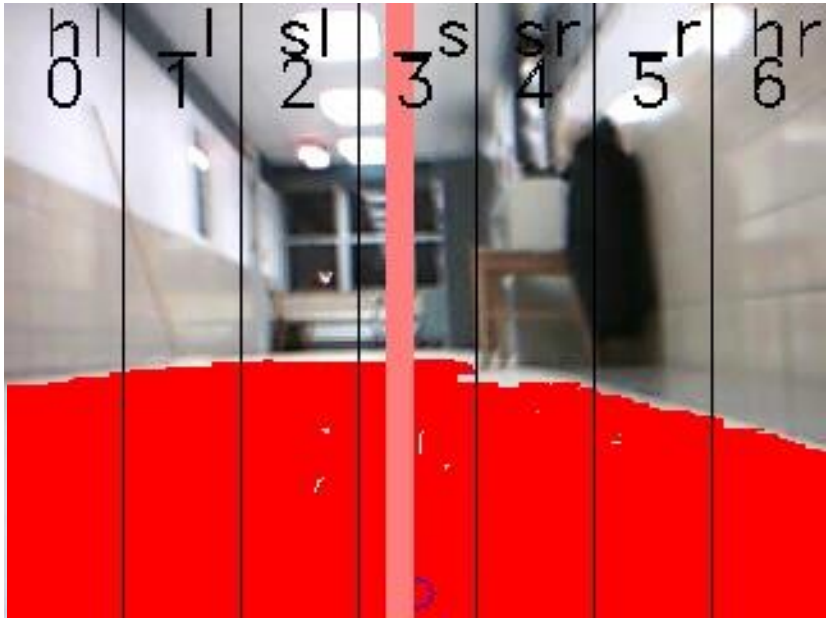
Vision Algorithm – Detailed



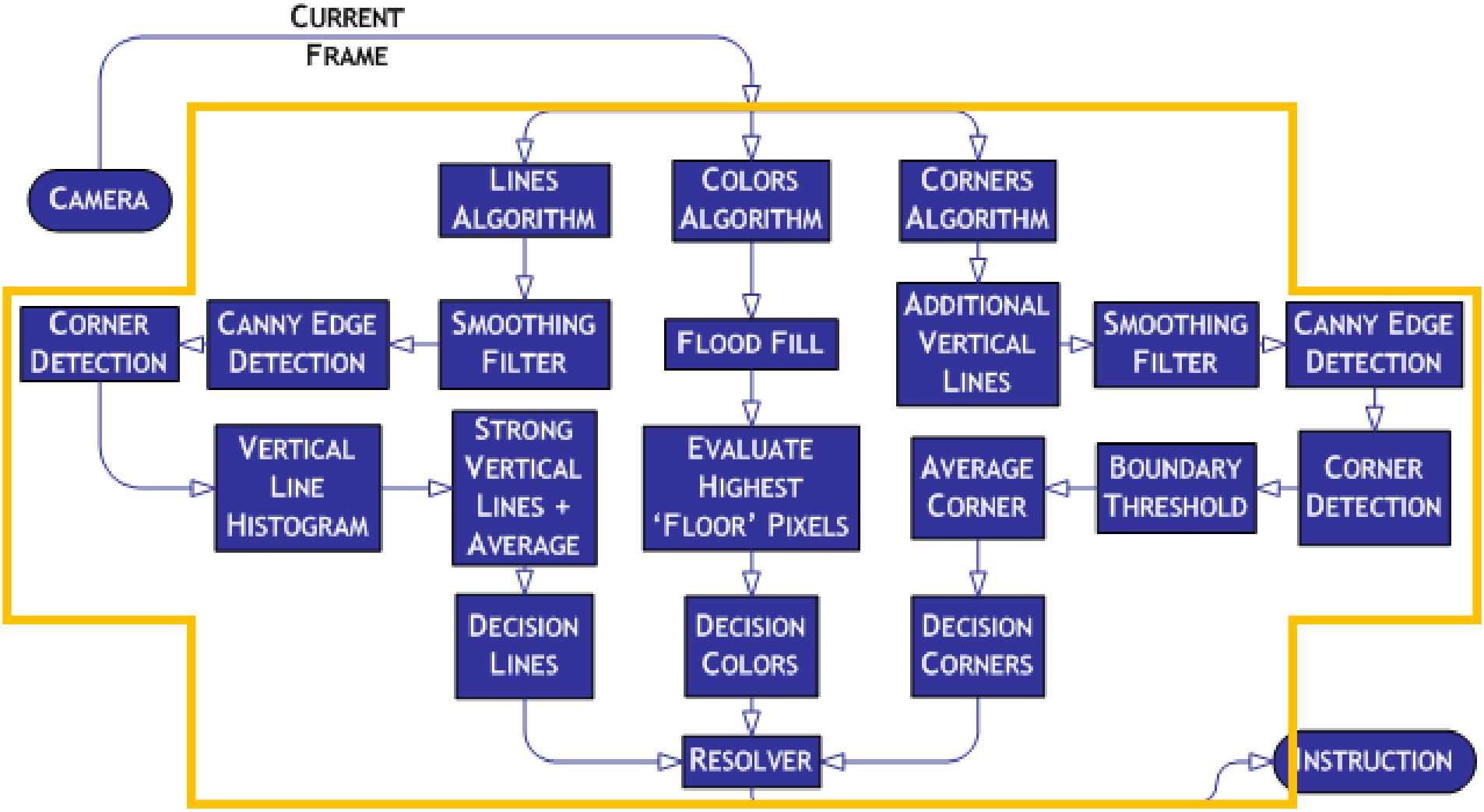
Colors Algorithm



Colors Algorithm

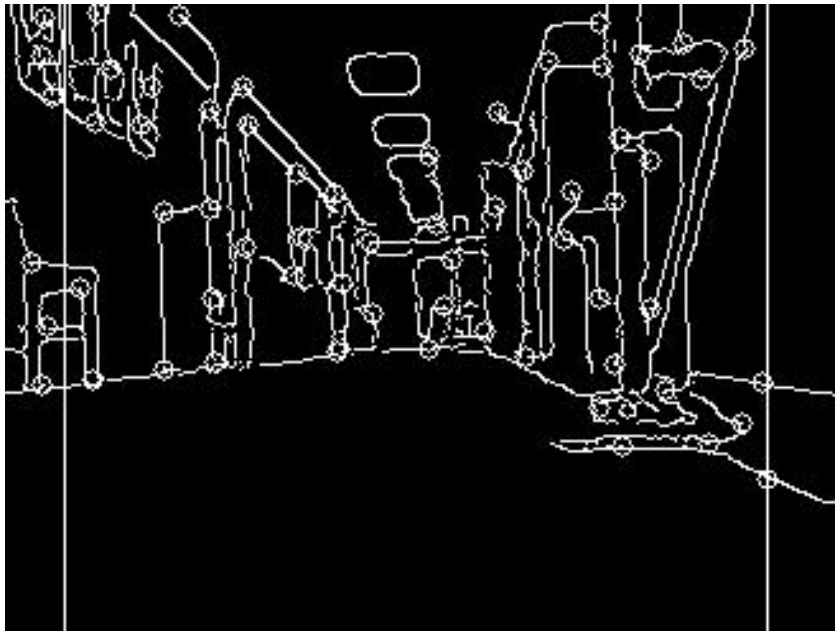


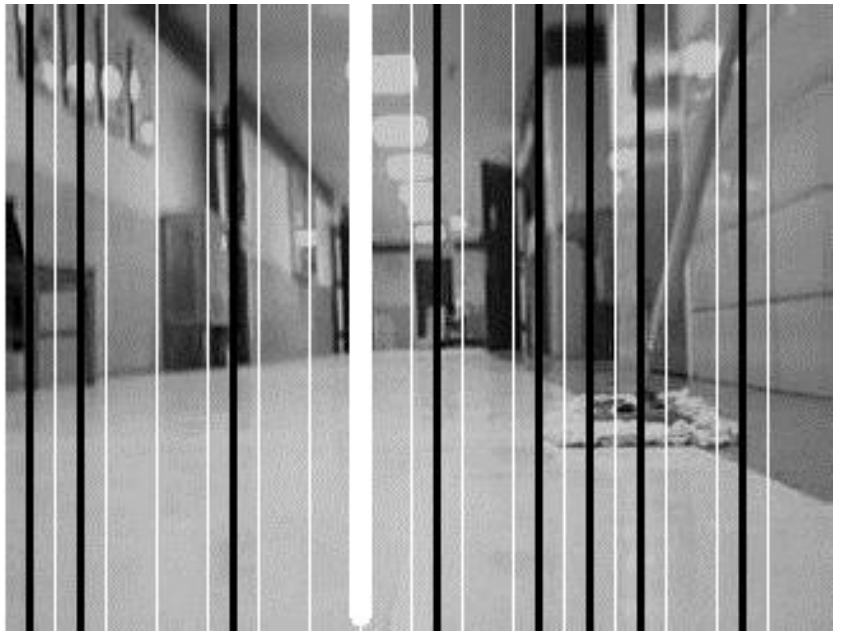
Vision Algorithm – Detailed



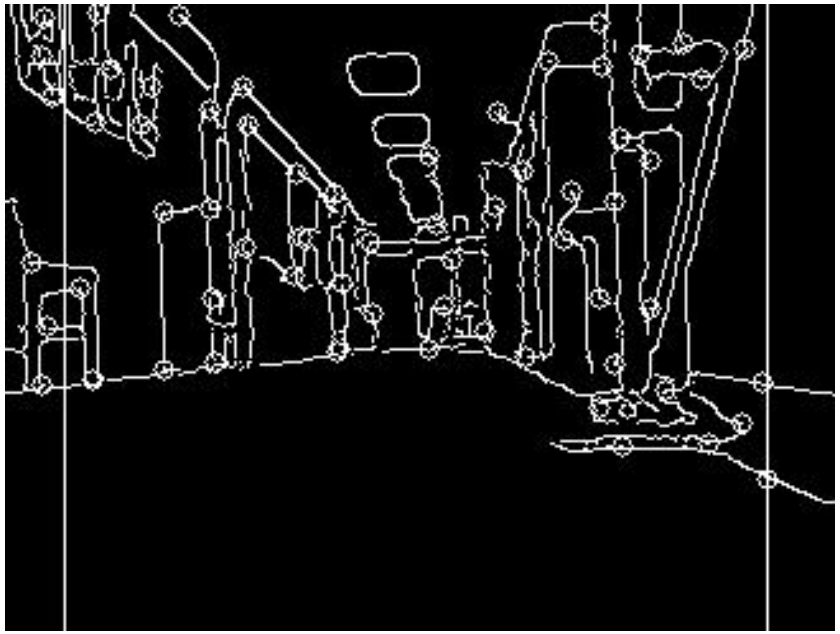
Vision Algorithm - Example One

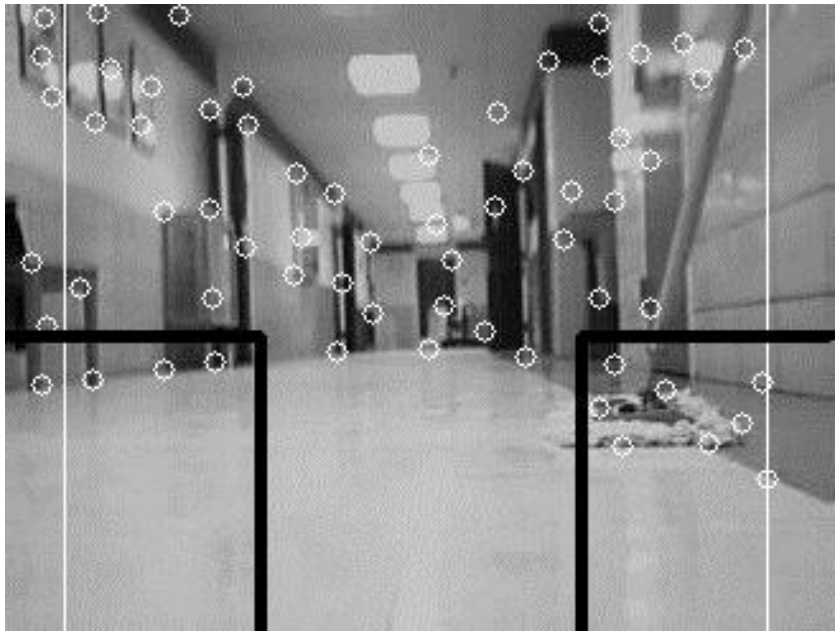
Vision Algorithm - Example One





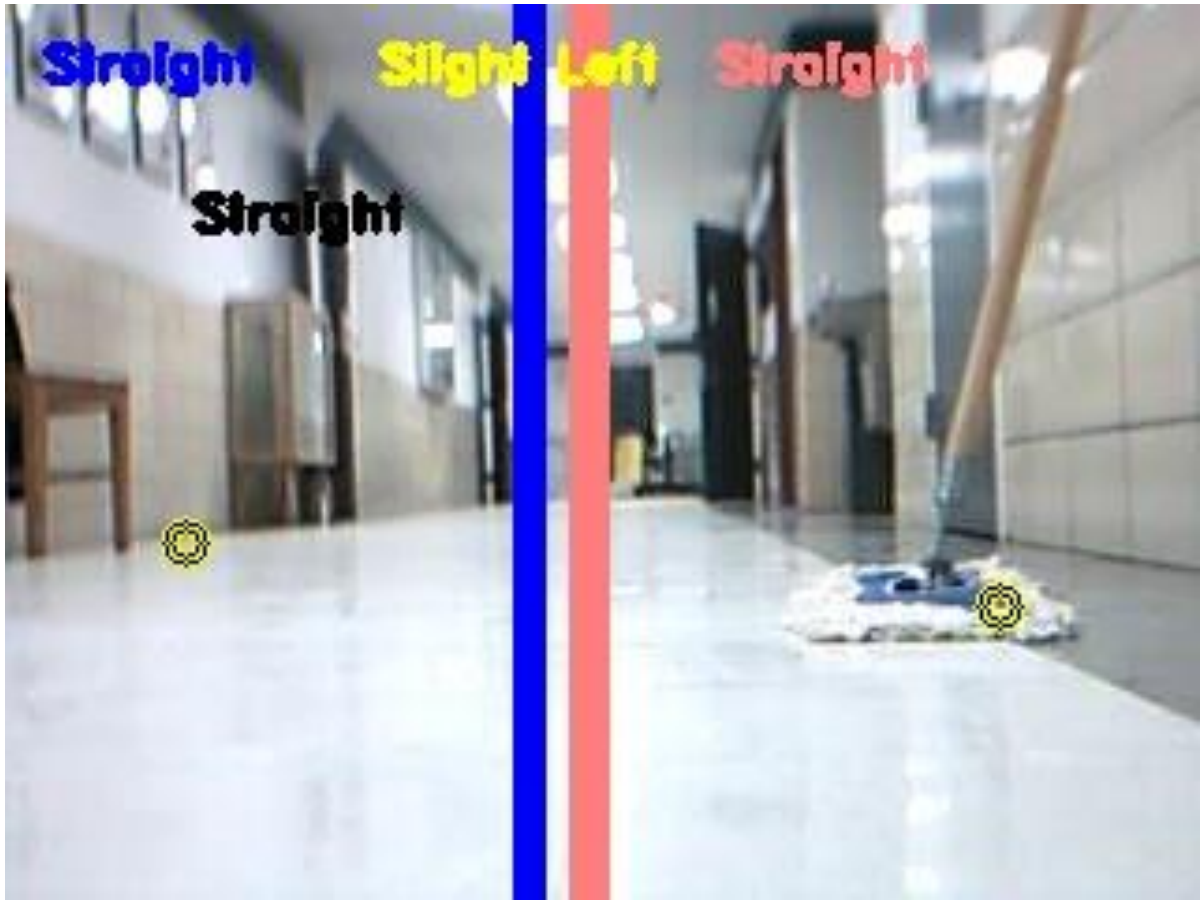
Vision Algorithm - Example One





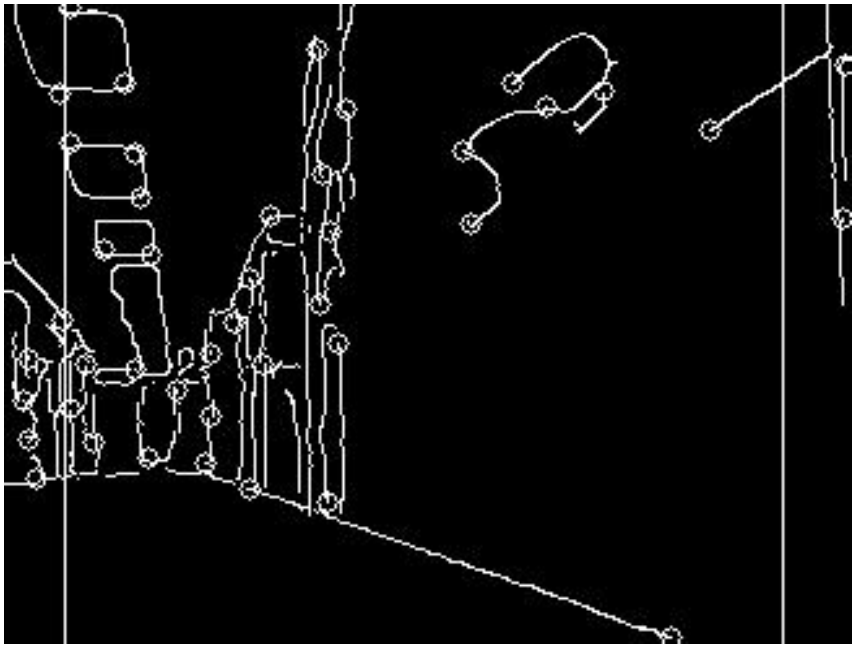
Vision Algorithm - Example One

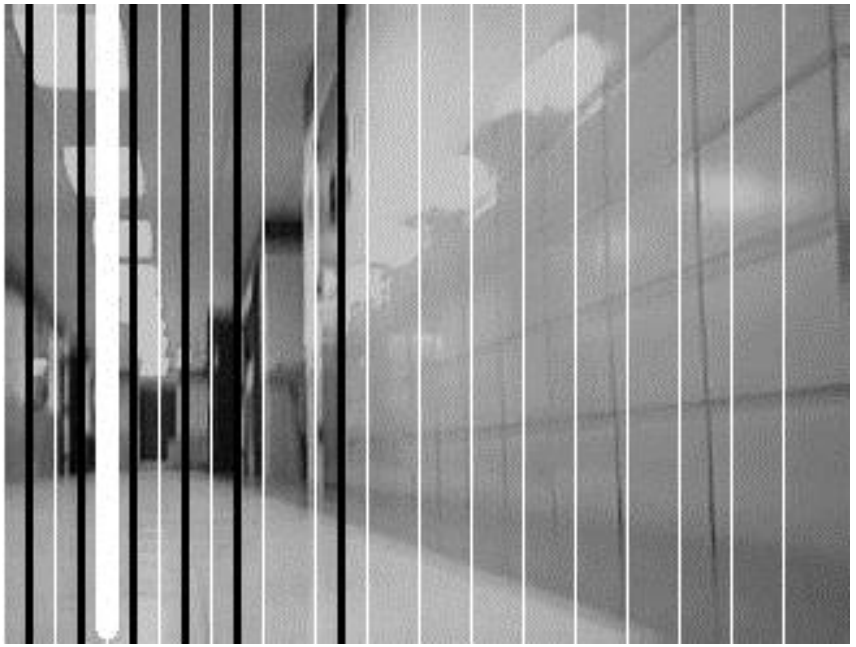
Vision Algorithm - Example One



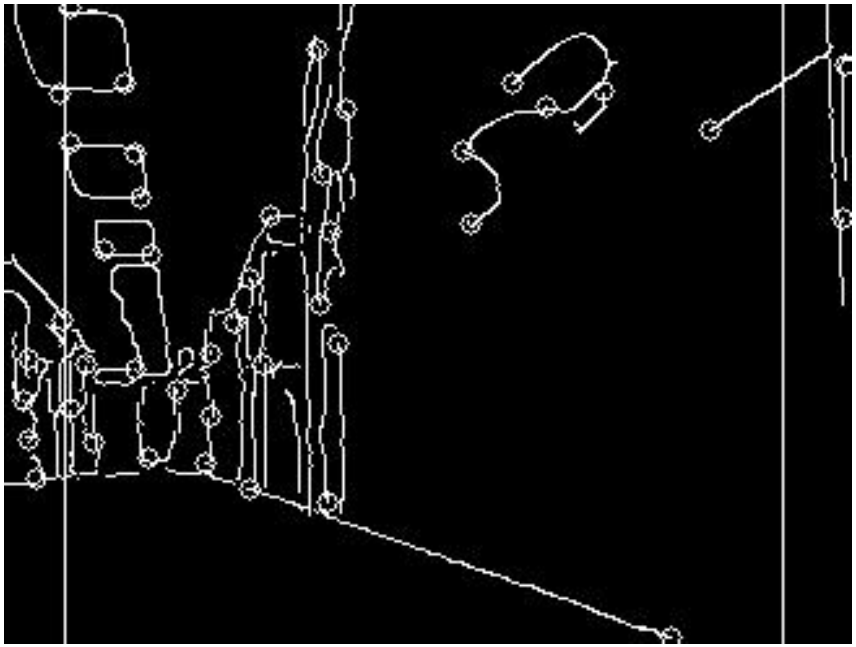
Vision Algorithm - Example Two

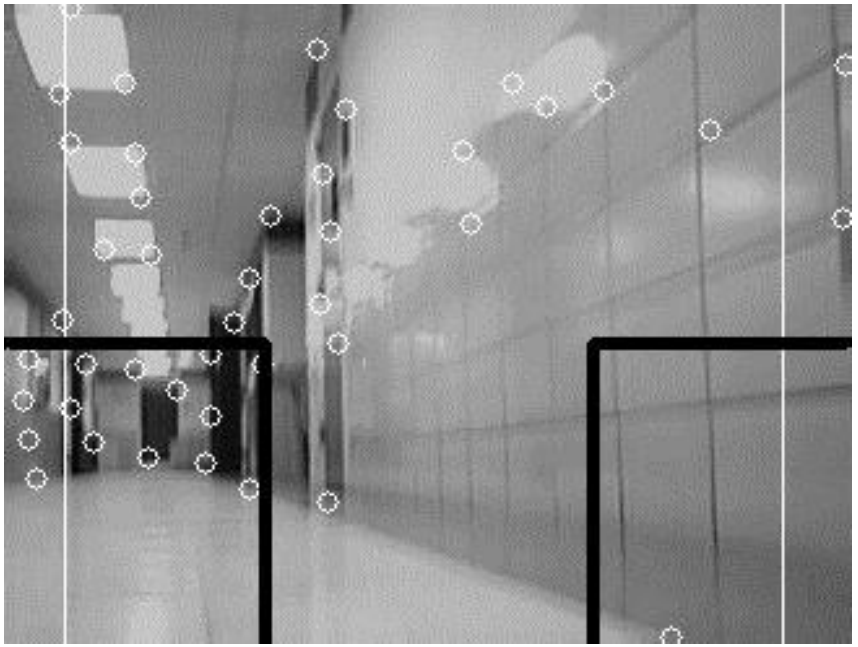
Vision Algorithm - Example Two



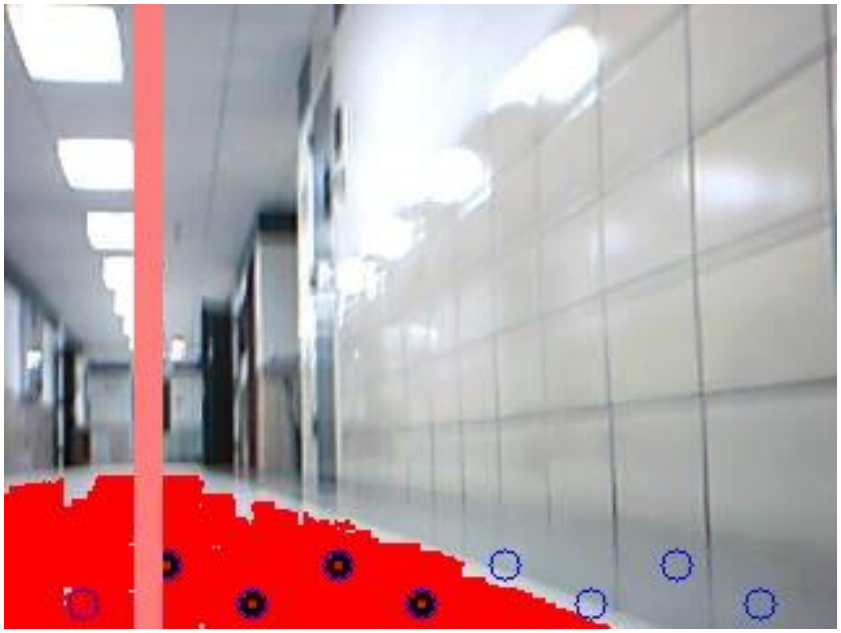


Vision Algorithm - Example Two





Vision Algorithm - Example Two



Vision Algorithm - Example Two

Quantitative Results

	Lines	Corners	Colors	Resolved
Hard Left	33.3	13.3	93.3	93.3
Left	87.9	55.2	93.1	100.0
Slight Left	97.1	28.6	91.4	94.3
Straight	96.4	48.2	96.4	98.2
Slight Right	97.6	29.3	92.7	100.0
Right	57.1	46.9	96.9	95.9
Hard Right	26.3	21.1	100.0	94.7
Totals	70.8	34.7	94.8	96.6

Qualitative Results

- Initial testing yields promising results!
  - Two programs ran independently
    - Vision system
    - iRobot controls
  - Verified quantitative results
  - Exceeded expectations

Questions?

- VBASR by Kevin Farney

References

- Sage, K., and S. Young. "Security Applications of Computer Vision." *IEEE Transactions on Aerospace and Electronic Systems* 14.4 (1999): 19-29. Aug. 2002.
- DeSouza, G. N., and A. C. Kak. "Vision for Mobile Robot Navigation: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (2002): 237-67. Aug. 2002.
- Davies, E. R. *Machine Vision: Theory, Algorithms, Practicalities*. San Francisco: Morgan Kaufmann, 2005.
- Forsyth, D., and J. Ponce. *Computer Vision: a Modern Approach.* Upper Saddle River, N.J.: Prentice Hall, 2003.
- Shapiro, Linda G., and George C. Stockman. *Computer Vision*. Upper Saddle River, NJ: Prentice Hall, 2001.

References

- Scott, D., and F. Aghdasi. "Mobile Robot Navigation In Unstructured Environments Using Machine Vision." *IEEE AFRICON* 1 (1999): 123-26. Aug. 2002.
- Argyros, A. A., and F. Bergholm. "Combining Central and Peripheral Vision for Reactive Robot Navigation." *IEEE CSC Computer Vision and Pattern Recognition* 2 (1999): 646-51. Aug. 2002.
- Shimizu, S., T. Kato, Y. Ocmula, and R. Suematu. "Wide Angle Vision Sensor with Fovea-navigation of Mobile Robot Based on Cooperation between Central Vision and Peripheral Vision." *IEEE/RSJ Intelligent Robots and Systems* 2 (2001): 764-71. Aug. 2002.
- Matsumoto, Y., K. Ikeda, M. Inaba, and H. Inoue. "Visual Navigation Using Omnidirectional View Sequence." *IEEE/RSJ Intelligent Robots and Systems* 1 (1999): 317-22. Aug. 2002.
- Orghidan, R., J. Salvi, and E. M. Mouaddib. "Accuracy Estimation of a New Omnidirectional 3D Vision Sensor." *IEEE/ICIP Image Processing* 3 (2005): 365-68. Mar. 2006.

References

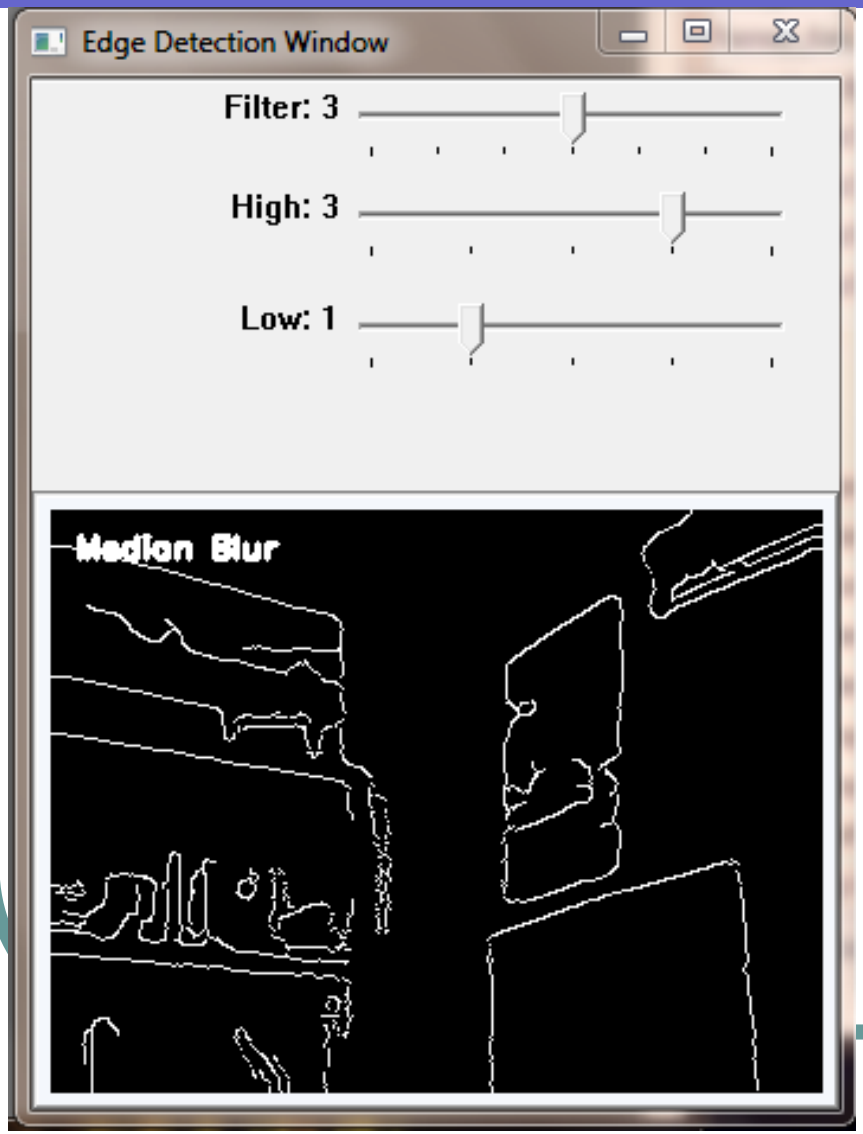
- Kosinski, R. J. "Literature Review on Reaction Time." Clemson University, Aug. 2009. 10 Nov. 2009. <http://biae.clemson.edu/bpc/bp/Lab/110/reaction.htm>
- Canny, J. "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986): 679-98. Jan. 2009.
- Shi, W., and J. Samarabandu. "CORRIDOR LINE DETECTION FOR VISION BASED INDOOR ROBOT NAVIGATION." *IEEE CCECE* (2006): 1988-991. Jan. 2007.
- Marques, C., and P. Lima. "Multisensor Navigation for Nonholonomic Robots in Cluttered Environments." *IEEE Transactions on Robotics and Automation* 11.3 (2004): 70-82. Oct. 2004.
- Ohya, I., A. Kosaka, and A. Kak. "Vision-Based Navigation by a Mobile Robot with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing." *IEEE Transactions on Robotics and Automation* 14.6 (1998): 969-78. Aug. 2002.

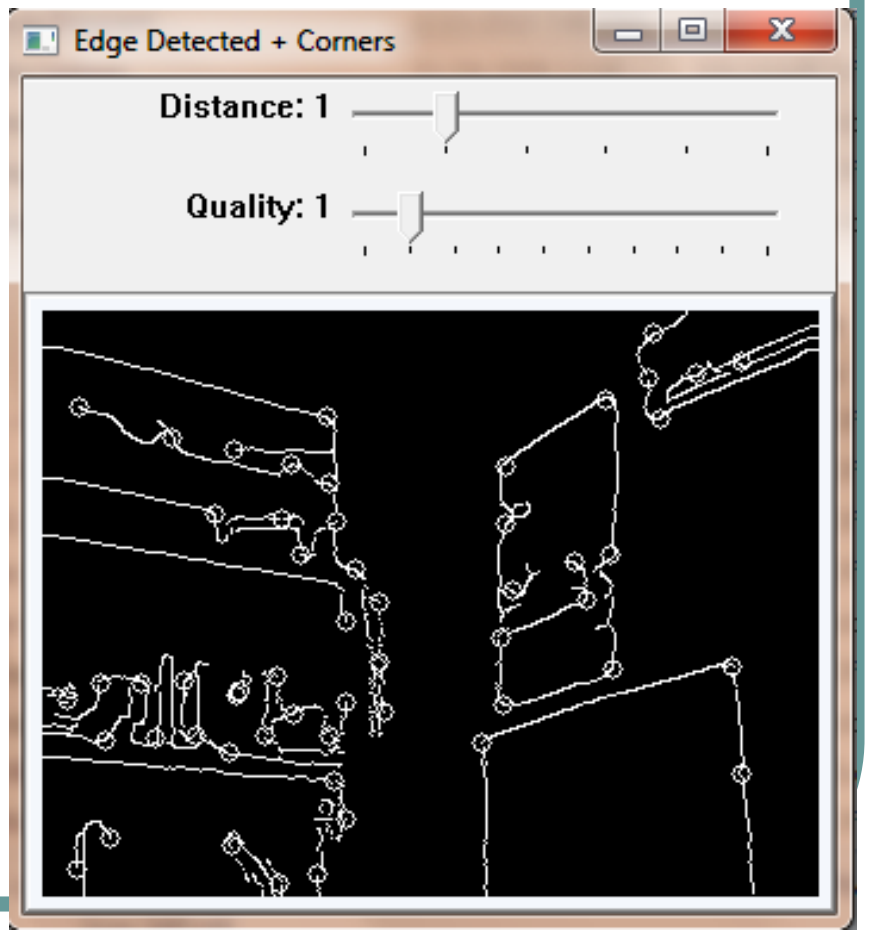
Quantitative Results

	Lines	Corners	Colors	Resolved
Hard Left	73.3	46.7	93.3	73.3
Left	79.3	89.7	93.1	98.3
Slight Left	100.0	68.6	91.4	97.1
Straight	100.0	69.6	96.4	100.0
Slight Right	97.6	51.2	92.7	100.0
Right	45.9	87.8	96.9	90.8
Hard Right	10.5	52.6	100.0	57.9
Totals	72.4	66.6	94.8	88.2

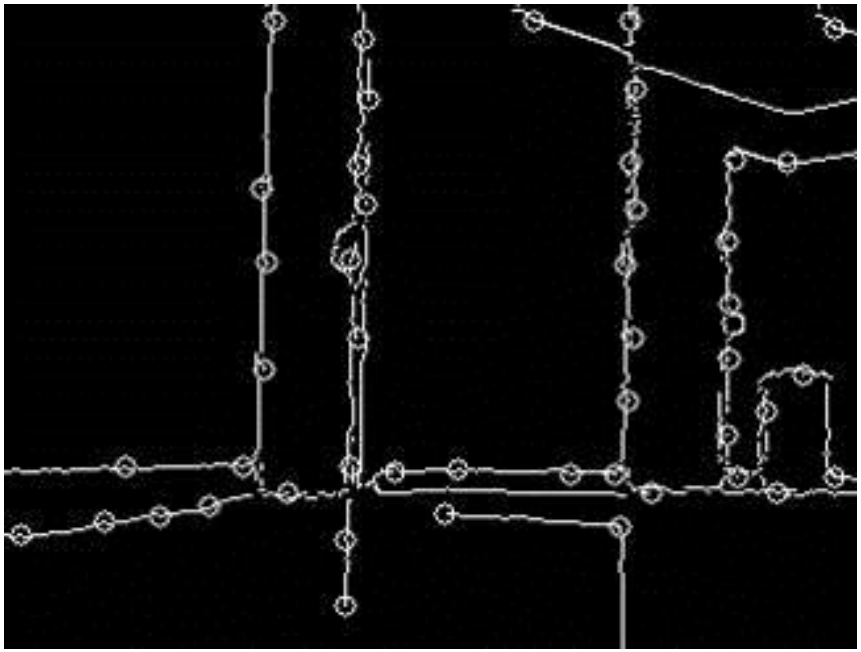
	Lines	Corners	Colors	Resolved
Hard Left	33.3	13.3	93.3	93.3
Left	87.9	55.2	93.1	100.0
Slight Left	97.1	28.6	91.4	94.3
Straight	96.4	48.2	96.4	98.2
Slight Right	97.6	29.3	92.7	100.0
Right	57.1	46.9	96.9	95.9
Hard Right	26.3	21.1	100.0	94.7
Totals	70.8	34.7	94.8	96.6

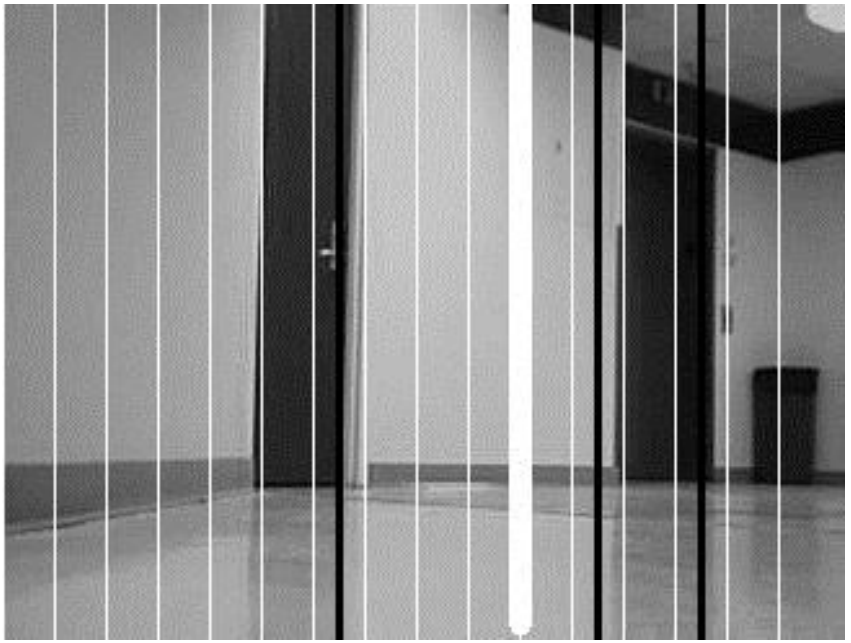
Selecting Parameter Values



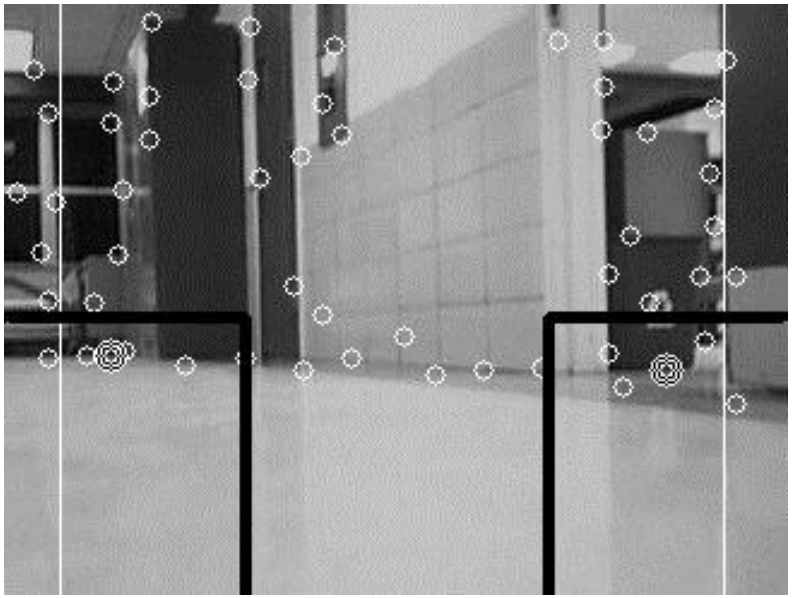


Lines Algorithm - Problems



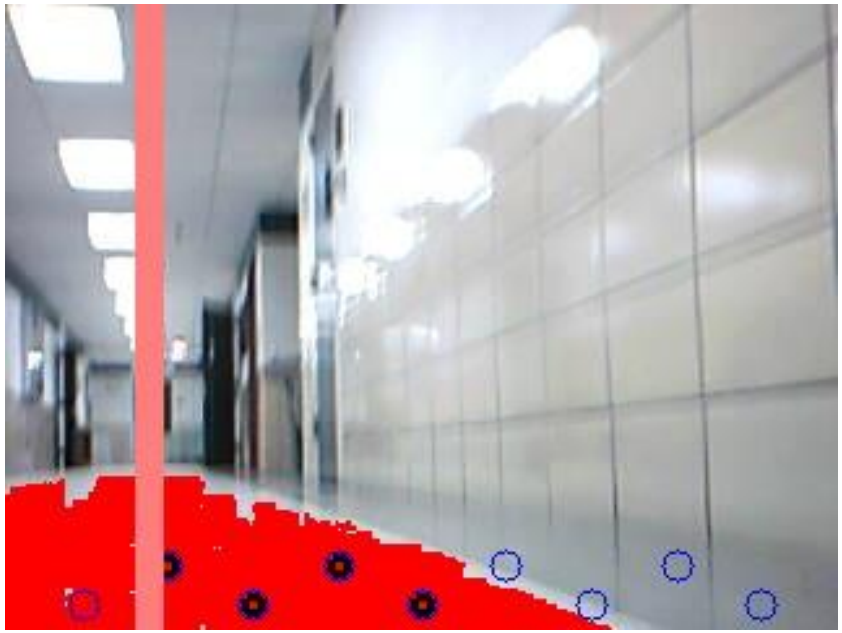


Corners Algorithm - Problems



Colors Algorithm - Problems

Colors Algorithm - Solution



Filters - Normal

- Normal Blur
  - Normalized box filter – summation of pixels over a neighborhood

$$\mathtt{K} = \alpha \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

$$\alpha = \begin{cases} \frac{1}{\texttt{ksize.width*ksize.height}} & \text{when } \texttt{normalize=true} \\ 1 & \text{otherwise} \end{cases}$$

Filters – Gaussian

- Gaussian Blur
  - Convolution of source image with specified gaussian kernel

Matrix of ksize (parameter) x 1 with filter coefficients:

$$G_i = \alpha * e^{-(i-(\mathtt{ksize}-1)/2)^2/(2*\mathtt{sigma})^2},$$

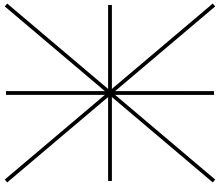
$$\alpha = \sum_i G_i = 1$$

Filters

- Median Blur
  - Returns median of pixel neighborhood into the destination image for each pixel

Canny Edge Detection

- Implements Canny Algorithm
  - First noise-reduction needed (filters)
  - Intensity Gradients $\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$ $\mathbf{\Theta} = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$
    - 8 points



  - Non-Maximum Suppression
  - Hysteresis Thresholding
    - High – discards noisy pixels
    - Low – connects the edges into lines (binary)

Corner Detection

- Good Features To Track
  - Calculates minimal eigenvalue per pixel
    - Covariation Matrix of derivatives
    - Then eigenvalues represent corners
  - Non-maxima suppression (3x3 pixels)
  - Rejection by quality level (parameter)
    - qualityLevel•max(eigImage(x,y))
  - Rejection by distance (parameter)

Price Breakdown

- iRobot Create Premium Development Package
  - $299
- Pioneer 3-DX
  - upwards of $5000
- Microsoft Robotics Developers Studio R2
  - free download
- Visual Studio 2008
  - $500 and up
  - Visual C# editor – free download
- Small Netbook
  - Looking for around $300

Microsoft Robotics Developer Studio

- CCR (Concurrency and Coordination Runtime)
- DSS (Decentralized Software Services)
- VPL (Visual Programming Language)
- VSE (Visual Simulation Environment)