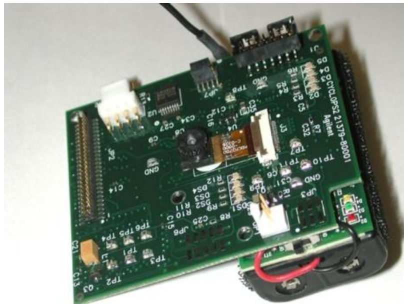
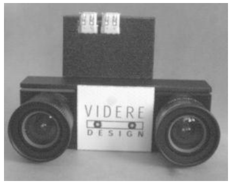
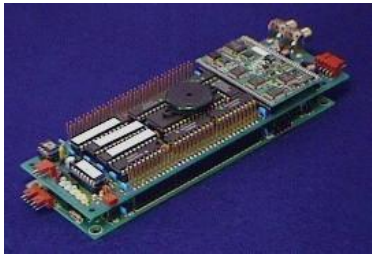
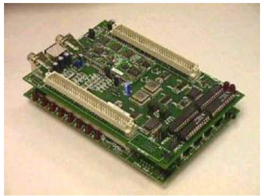
Overview of the Development in Computer Vision with CMUcam









-Brief study of systems dealing with Computer Vision(CV)

-Algorithms involved in implementing CV

-Components required for implementing a CV machine

-Available systems in the current market and academia

-Study of a special class of CV platform

Presented By: Onkar Raut

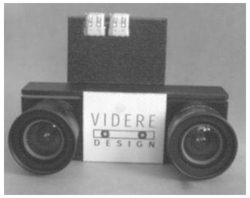
Components required for a CV machine

- Digital Camera
  - Type
  - Shutter Speed
  - Sampling Speed
  - Fill Factor
  - Chip Size
  - Analog Gain
  - Sensing Noise
  - ADC Resolution
- Storage space
  - Amount
  - Refresh rate
- Processing Unit
  - Complexity of calculations required

Available Systems for implementing machine vision

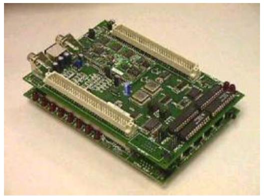
-SRI Simple Vision System

http://www.ai.sri.com/~konolige/svs/



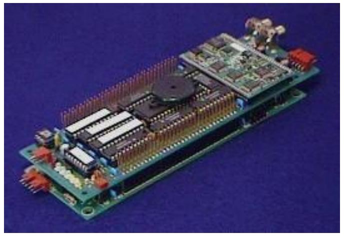
-The MIT Cheap Vision Machine

http://www.ai.mit.edu/people/ceb/cvm.html



-The Cognachrome Color Vision System

http://www.newtonlabs.com/cognachrome/



-CMUcam

http://www.cs.cmu.edu/~cmucam/

-Stanford MeshEye

-UCLA Cyclops

-Bluetechnix Blackfin

Constraints on Embedded Vision

- Computation
- Cost
- Power
- Space

It's all about compromises!!

Goals of the CMUcam project (Year -2002):

- Perform Color Blob tracking at a Frame Rate of 16.7 fps
- Provide high level information to other processors
- Implement simple algorithms used in robot activities
- Keep the design simple and make the project low cost and effective compared to other vision systems

Hardware Selected:

- Omnivision OV6620 CMOS camera
  - A few details
- SX28 microcontroller
  - Configurable Communications Controllers
- Level shifter for the RS232 serial data

Interface of the CMUcam to CPU interface

An example of communication with the CMUcam:

- :cr 18 44 17 2 19 32
- ACK
- :sw 30 60 50 80
- ACK
- :pm 1
- ACK
- :gm
- ACK
- S 150 20 30 5 2 6
- :pm 0
- ACK
- :sw 0 0 80 143
- ACK
- :tc 145 18 24 155 22 36
- ACK
- M 50 80 38 82 53 128 35 98
- M 52 81 38 82 53 128 35 98
- M 51 80 38 84 53 128 35 98

What compromises were made during the development of the CMUcam?

- No frame buffer
- Nonstandard frame rate
- Processing data as it is streamed
- Dropped G component acquisition
- Horizontal RGB resolution is 80 RGB pixels
- Slow microcontroller

Image Processing Algorithms Implemented on the CMUcam

- Color Blob Tracking
  - Enter minimum and maximum values for each RBG or YCrCb
- Color Statistics
  - Keeps a running sum of the individual color channel components
  - Building block for motion detection
- Noise Filtering
  - makes the color tracking algorithm more robust by requiring a valid detection to consist of two horizontally adjacent pixels in the specified color range
- Additional Demo mode
  - The camera acquires the color of the first object it sees upon power up and tracks it using a simple feedback loop to point a servo toward it.

Performance

- Maximum rate 16.7 fps
- Resolution 80x143
- Center of mass tracking: jittering by 0.005 pixels and 0.011 pixels and standard deviation of 0.005 pixels and 0.011 pixels on x and y axis for blue objects and 0.1460 and 0.2900 with standard deviations of 0.146 and 0.216 for green objects
- Robot tracking

CMUCam2

Differences:

- Hardware
  - SX52 communications microcontroller
  - Fourth chip, a frame buffer
- Additional Algorithms implemented
  - Frame Differencing
  - Edge detection
  - Color histogram
- Performance Specs
  - Tracking speed 50 fps
  - Power saving
  - Interfacing to low end microcontrollers
- Cost: $199

- Hardware differences:
  - ARM7 processor
  - Additional SD MMC card slot
  - 4 servo ports
- Architectural Differences
  - SPI communication with flash card
- Additional Algorithms implemented
  - JPEG compression
  - frame differencing
  - color tracking
  - Convolutions
  - Face recognition
  - Polly at 4 fps
  - Spoonbot: follow Colored objects
- Cost: $239

A brief explanation of stereovision

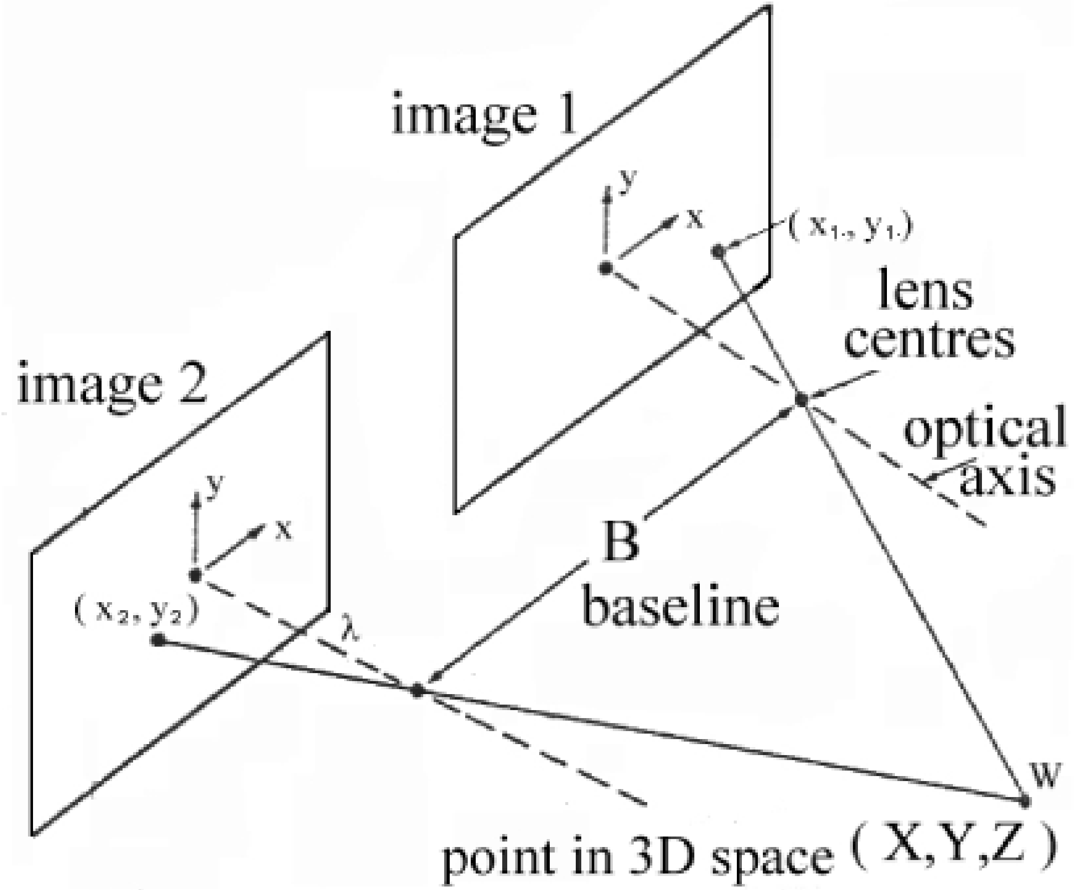


Figure 2. Stereo image geometric model.