Project Thoughts

(There’s another document somewhere….FIND THAT TOO!)

* Motor precision
  + Initial resistance
  + Caster resistance
* Power requirements
* Initial GUI idea
  + Not required – project is essentially prototype on cheaper hardware for more expensive implementation
    - Making use of pre-existing more powerful hardware of desktop computer
    - Explored idea of implementing interface and functionality with JavaScript
      * No existing OpenCV port
      * Not really required because of reason for project
        + Interoperability is achieved through python but ultimately would be self-contained in robot hardware
* Algorithms
  + Appropriate:
    - Boosted Classifiers
      * Limited use:
        + Viola-Jones (2001)
        + Liao (2007)
        + SIFT
      * OpenTLD
      * CMT
    - Strengths and weaknesses and suitability of both
    - SURF!!
  + Other
* Issues
  + Identifying from multiple perspectives
    - Phased detecting
      * E.g. for USB
        + USB on its side
        + USB flat
  + Identifying rotations
    - Rotate image to detect?
  + How to stop
    - Distance – ultrasonic sensor
    - Size of object in image (i.e. if x of object > x of image/90 OR y of object > y of image/90 >
* Likely Improvements
  + Identification in busy environments
  + More efficient search patterns
  + Ability to self-learn/easy way to allow non-technical user to train
  + Improve image stabilisation (hardware/software)
  + Improved hardware
    - Hardware suited for manual control, not so much autonomous
    - More powerful motors
      * With encoders
      * Will require different controller board and power source
    - Better camera
    - Improved light tolerance
    - Tank tracks for better movement (less likely to slip)
    - More powerful processor
    - Enter approx. width/height – box
      * Extra hardware to hold objects at correct distance

# SURF

* Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008
  + Keypoint Detection:
    - Integral Image (Summed Area Table) – introduced by Viola-Jones (2001)
      * Calculation time “independent of size”
        + Allow fast computation of box convolution filter
    - Hessian Matrix Based Interest Points (for detecting points)
      * Matrix consists of convolution of Gaussian second order derivatives
        + Gaussian good for handling scale change
        + Must be cropped and used to approximate in discrete quantity

Less reliable under rotation (odd multiples of pi/4)

Affects all Hessian-based detectors

* + - * SURF uses approximate second order Gaussian derivatives which are evaluated at low computational cost due to integral images being used
  + Keypoint Description/Matching
    - First Haar wavelet responses in x/y directions; use integral image for speed
    - Index based on sign of Laplacian
      * Increase robustness and matching speed
    1. Fix reproducible orientation using information from circular region around point
       - Identify orientation for interest points
         * Calculate Haar wavelet responses in x/y direction within circle of radius 6s, s scale at interest point detected
         * Once calculated and weighted with Gaussian using sigma 2s centred at interest point

Space in horizontal along abscissa and vertical along ordinate

Dominant orientation estimated by calculated sum of responses within sliding orientation window pi/3

Reponses summed yield local orientation vector

Longest vector is orientation of interest point

* + 1. Construct square region aligned to orientation and extract descriptor
    2. Features matched between two images

….

* SURF is still relatively computationally expensive and while suitable for PC for a self-sufficient system like that proposed in project a less computationally expensive algorithm is required (ORB or FAST + BRIEF)

# FLANN Matching

* Fast Library for Approximate Nearest Neighbours

# Finding Homography

* Find transform between matched keypoints in two images
* Apply transform to map points accurately
* Used to map approximate area of image with bounding box

# RANSAC

* Data consists of inliers: distribution explained by model parameters, i.e. set object described by extracted interest points (noise can interfere) and outliers that don’t fit model