Online Resource 9

Sample Implementation – Discussion of Constant Selection

While the sample implementation given in Section 4 is only valid for the defined application, it is a valid and useful exercise to examine the selection of the values and constants used. The following is a list of the constants and methods used, and a brief description of their selection and tuning. The method used to select each value is described. These methods, being primarily based on trial-and-error, are not formal, and thus were not directly included in the paper. They can, however, be used as a starting point when selecting and tuning these constant values for a different application.

Stage L1 – Gaussian Blur (M, σ)	A Gaussian Blur was added to remove pixel-level noise. To select the blur parameters, it is necessary to characterize pixel noise by taking a series of images of a static scene from each camera. The average and standard deviation for each color channel of each pixel can be calculated, and trials of the Gaussian filter applied until a desired SD is reached. Typically, one would fix M to be some standard value (a good starting value is 3), and then select σ conservatively. If, later, lack of image detail causes tracking loss in Stage L3 onward, the value of σ can then be iteratively reduced.
Stage L1 – Distortion Model	The complete system calibration method used is described in detail in [32]. As part of this process, a sample data set, comprised of true world coordinate locations and measured pixel locations for a set of features is recorded. Successively higher distortion model orders are applied to this sample set, wherein the true world coordinate locations are projected through the calibration matrices and compared to the measured pixel locations. The average pixel-distance error can then be calculated for the complete set of points. A distortion model order is selected which minimizes the average error without over-fitting. For example, the following average pixel-length errors were found for a sample calibration of camera 1 in the given experimental setup: 1st order: 5.62 2nd order: 1.46 3rd order: 0.35 4th order: 0.09 5th order: 0.10 In this case, one would select a 4th order distortion model ($k_{c_5} = 0$), as a fifth-order model produces over-fitting. If the average error is still large (> 0.1 in our implementation), then pixel shear can be added ($\alpha \neq 0$), or a different distortion model employed.
Stage L1 – Active Tracking Area Interest Filter ($I_{e_{min}}$)	The value of $I_{e_{min}}$ is selected to yield a minimum interest value for this filter that is close to the minimum level of interest for other filters. By examining the range of output values produced by the other two filters, a value of 0.1 was selected as a rough division point between pixels that are of-interest and not-of-interest for the other two filters. For reference, both other filters produced a range of values continuous from 0 – 1, with about 90-95% of pixels that a human (given the same task) would mark as

	'of-interest' having an interest level greater than 0.1. The most direct method to determine this value is to simply apply the filter to a selection of real-world images and inspect visually the ellipse size produced by the choice of $I_{e_{min}}$. This initial value can be iteratively adjusted later.
Stage L1 – Gradient-Based Edge Detection	The size of the box blur was chosen based on the average feature size (in pixels), given the chosen camera resolution and the average size of a feature in the image. This was mainly performed through visual inspection of a number of sample images. The size of the blur was initially set to a small fraction of this size (~1/20 to 1/50). For this trial, a blur of size 5 (from an average feature size of about 100) was chosen, which did not need to be changed later. Again, iterative adjustment after initial selection can be performed.
Stage L1 – Predictive RoI (W_{prd})	This value was set to a constant value of '1' for all trials, and did not need to be adjusted. In cases where the user finds significant over- or under-estimation of interest caused specifically by this filter, this value could be selected as a lower constant value, or as a value proportional to the estimated quality of the pixel velocity estimate.
Stage L2 – Delay	One can estimate the delay margin needed (given f) by simply benchmarking the average synchronization time on the target platform. For this implementation, the time was measured to be about $\frac{1}{200}f$, which was doubled to provide a margin of safety.
Stage L3 – Detection / Tracking	The selection of detection and tracking methods strongly depends on the subject or object to be tracked, and on the features that will commonly be available in 2-D. The selected methods (PCA search, OF+LK tracking) were derived from a detailed feasibility study of tracking methods, as applied to the selected environment and subject. However, these methods are known to be (through inherent design) highly adaptable to a wide variety of 2-D applications. Thus, it is recommended to start with these two methods as a baseline, and move to more advanced methods if 2-D detection or tracking performance is found to be a limiting factor in overall system performance. Alternatively, other methods can be used directly if desired. For example, the basic 2-D detection and tracking can be implemented using an existing system or framework, such as the Kinect system [34].
Stage L4	All unknown values are inherently produced by the complete calibration process used [32].
Stage L5 – GM Parameters	The values of σ for the GM estimator were determined through detailed analysis of multiple sample datasets. For each sample dataset, the GM algorithm was executed, and the correctness and number of iterations were measured (given known world coordinates for the 3-D points being solved). Maximum and minimum values for σ were then selected, and a binary search was performed manually to find a value of σ which minimized the number of iterations, while producing the desired selectivity (as measured by the correctness of the solution).
Stage L6	The initial values of C_1, C_2, and C_3 were determined purely through inspection of the equations for Stage L6 form recovery

	to select roughly equal weightings to each component of uncertainty initially. Other selections were not tested, as this selection was sufficient for the operation of the sample implementation.
Stage L7	The second-order set of state variables were selected instead of a linear model purely to allow for some curvature of the path. Also, over the relatively small time between updates, the difference between this choice and a third-order model was found not to have a significant effect on the overall system performance (through direct comparison). Thus, the second-order model was used to minimize calculation effort.
Stage L8	For simplicity, the basic visibility metric calculation is similar to that used in previous work [32]. Details for the determination of the weights $W_{area} = W_{dist} = W_{angle} = 1$, as well as other constants related to V and ϕ are given in detail in this paper.
Stage L10	For consistency, the second-order KF was used to produce and maintain the fall-back poses. The window was selected to be half of the pipeline depth. In general, these constants should have very little effect on normal operation, as one inherently selects other parameters to minimize their use. As such, these nominal values were sufficient for the sample implementation.