1 Pan-STARRS Library PSLib

1.1 Math Utilities

1.1.1 Sorting

A variety of sorting algorithms exist, with a wide range in speed for different set sizes. Three of the best sorting algorithms are "heapsort", "quicksort", and "mergesort" (see, e.g., knuth, sedgewick, press). These three sorting algorithms all run in a time of $O(n \log n)$ on the average, but have different worse-case run times. Implementations of all three sorting algorithms are described in references above and in various places online (e.g., http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/algoen.htm). The linux `qsort` function uses a heapsort if it can allocate a sufficiently large temporary buffer, and otherwise resorts to a quicksort in-place. The GSL function `gsl_heapsort` uses the heapsort algorithm. Both of these implemenations require a pointer to the comparison function, which may result in a slower code than if the comparison were done within the sort function.

For PSLib, the sorting functions shall be implemented via the system `qsort`. The function `psSort` shall return the sorted result of the input array without over-writing the input array. The function `psSortIndex` shall return an integer index to the sequence of the input array without overwriting the input array. Given the following line of code:

```
out = psSortIndex (NULL,&in);}
```

the elements of the array `out` are in the sequence `in.arr[out->arr[0]]` to `in.arr[out->arr[in.n - 1]]`.

1.1.2 Smoothing: Boxcar and Gaussian

Smoothing may occasionally be perfomed on data. We present the algorithms for two typical versions: boxcar and Gaussian smoothing. In both smoothing techniques, given a series of data values $f_i(x_i)$ where x_i are the values of the corresponding to the center of the bin, the smoothed values $g_i(x_i)$ are determined by calculating a linear combination based on the input data point and its nearest $2N$ neighbors in the form:

$$g_i = \sum_{j=i_{\min}}^{i_{\max}} c_{ij} f_j \tag{1}$$

where the values of c_{ij} determine the filter type. For boxcar smoothing, the values c_{ij} are constant and scaled to maintain the zeroth moment of the data (care must be taken at the ends of the data range to reduce the value of c_{ij} as fewer input data points may be used). For Gaussian smoothing, the crucial parameter is σ_x, the standard deviation. The values of $i_{\min}$ and $i_{\max}$ are functions of the standard deviation: $i_{\min}$ corresponds to the bin in which $x_i - N\sigma_x$ is found; similarly $i_{\max}$ is the bin corresponding to $x_i + N\sigma_x$. The value of N should be chosen to be large enough to sample the Gaussian, $N = 5$. The values of c_{ij} are then just the Gaussian curve:

$$c_{ij} = \frac{e^{\frac{-(x_j - x_i)^2}{2\sigma_x^2}}}{\sqrt{2\pi\sigma_x^2}} \tag{2}$$