

*It would be possible to describe everything scientifically,  
but it would make no sense; it would be without meaning,  
as if you described a Beethoven symphony as a variation of wave pressure.  
(Albert Einstein)*

## APPENDIX C

# Robust Motion Estimation with LTS and LMedS

In Chapter 4, we employed the RANSAC algorithm to calculate global-motion parameters for a set of point-correspondences between two images. A robust estimation algorithm is required for this problem, since the input is a mixture of foreground object motions and the camera motion. Additionally, the input data is contaminated with erroneous correspondences.

One disadvantage of the RANSAC algorithm is its dependence on the threshold  $\epsilon$ , which decides on how close a point-correspondence must be to the computed model to be considered an inlier. To relieve from the dependency on this threshold, modifications of the RANSAC algorithm have been proposed that do not require an explicit inlier threshold. We will discuss two algorithms, which are based on a common principle: Least-Median-of-Squares (LMedS), and Least-Trimmed-Squares (LTS). Both algorithms are similar to RANSAC with the only exception that Steps 3 and 4 are modified. Since LMedS and LTS are very similar, we will describe them together in the following.

We start again by drawing a random subset of input data to compute a motion model candidate  $\mathbf{H}$ . After this, we also compute the residuals between all input correspondences and the model  $\mathbf{H}$ . However, the difference is that we do not mark the data with high residuals as outliers, but that we sort the input data according to increasing residual errors. More

specifically, let again  $\mathcal{C} = \{\mathbf{p}_i \leftrightarrow \hat{\mathbf{p}}_i\}$  be the input correspondences, then we impose an ordering  $r(i)$  on the input data such that

$$d(\hat{\mathbf{p}}_i, \mathbf{H}\mathbf{p}_i) \leq d(\hat{\mathbf{p}}_k, \mathbf{H}\mathbf{p}_k) \leftrightarrow r(i) \leq r(k). \quad (\text{C.1})$$

The LMedS algorithm repeats the random sampling process and selects the motion model, for which the median of the residuals

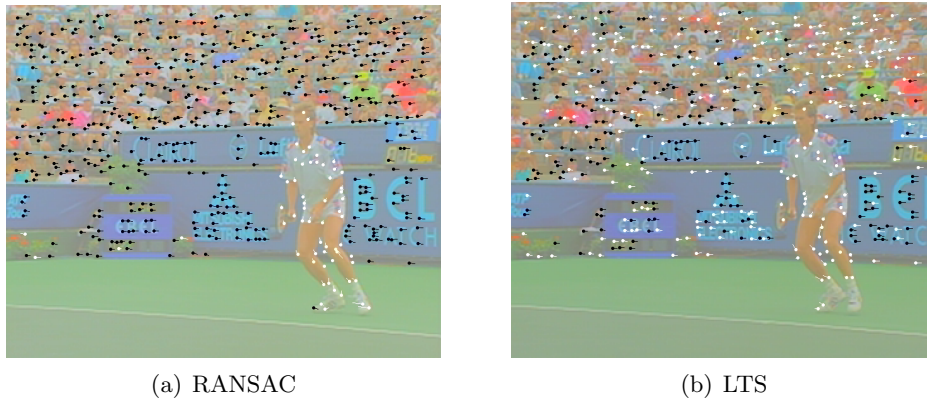
$$d(\hat{\mathbf{p}}_m, \mathbf{H}\mathbf{p}_m) \quad \text{with the median position} \quad m = o(|\mathcal{C}|)/2 \quad (\text{C.2})$$

is lowest. Under the assumption that at least half of the feature-points will be moving according to the global motion, LMedS will select the correct global motion model.

The LTS algorithm [159] computes the same ordering  $r(i)$  over the input data. However, not only the median of the residual errors is considered to select the best transform, but a fixed percentage of the best ranked input data. This subset of the input data is considered the *inliers*, even though they might only be part of the real inlier data. On this inlier data, an additional *compaction step* (CSTEP) is carried out. This is a refinement of the estimated motion model on the selected inlier set, comparable to Step 5 of the RANSAC algorithm. The CSTEP can even be applied several times where the set of inliers is adapted to the refined motion model in each iteration. The complete LTS algorithm can then be summarized as follows.

1. Draw a random subset  $\mathcal{S}$  from the input data as in the RANSAC algorithm.
2. Compute the motion model  $\mathbf{H}_0$  based on the drawn subset  $\mathcal{S}$ .
3. Rank the input data to increasing residual errors and determine the inlier set  $\mathcal{I}_1 = \{c_{r(k)} \mid k \leq p \cdot |\mathcal{C}|\}$  so that it includes a fixed fraction  $p$  of the correspondences  $c_i \in \mathcal{C}$  with lowest error. Compute a refined motion model  $\mathbf{H}_1$  using a least-squares approximation to  $\mathcal{I}_1$ . Let the total residual error be  $Q_1 = \sum_{\mathbf{p}_i \leftrightarrow \hat{\mathbf{p}}_i \in \mathcal{I}_1} d(\hat{\mathbf{p}}_i, \mathbf{H}\mathbf{p}_i)$ . Based on the new motion model, compute a new set of inliers  $\mathcal{I}_2$  and a new motion model  $\mathbf{H}_2$  and residual  $Q_2$ . Iterate this process until  $Q_{k-1} = Q_k$ .
4. Repeat Steps 1-3  $N$  times and choose the motion model, for which  $Q_k$  was lowest.

Comparing the LTS algorithm with RANSAC, we see that no threshold  $\epsilon$  is required any more, but on the other hand, we have to specify a percentage  $p$  which is equal to the minimum fraction of inlier data. The LMedS does not have a comparable parameter, but since the median of the residuals is taken, it implicitly assumes that the inlier fraction is at least 50%.



**Figure C.1:** *Examples of inlier classification for (a) RANSAC, and (b) LTS. Inliers are marked in black, outliers are drawn in white.*

However, it is obvious that the LMedS algorithm can be generalized for other inlier fractions by not considering the median residual but a different position in the ranked list of residuals.

## Evaluation

Compared to the RANSAC algorithm, LTS and LMedS did not show clear advantages. In fact, the accuracy of the obtained motion parameters were even slightly worse than with RANSAC. For LTS, this can be explained as follows. We have seen in the evaluation of the RANSAC algorithm that the accuracy improves with an increasing number of refinement iterations, where the image area which is supported by inliers gets larger in each step. Note that this iterative refinements are comparable to the compaction steps of LTS. However, LTS only includes a fixed percentage of best fitting data into the refinement. The effect is that the covered set of the inlier data does not grow larger than this fraction and the final model estimate is based on a smaller set of input samples. It is a good model for this subset, but not for all the inliers. A similar reasoning also holds for LMedS.

An example comparison between RANSAC and LTS is depicted in Figure C.1. It is clearly visible that RANSAC achieves an accurate separation of background motion and foreground motion. The LTS algorithm does not give this clear separation, but all selected inlier vectors are part of the background motion. Consequently, LTS also successfully locked to the correct background motion.

