

AdelaideRMF analysis and correction

Image Analysis & Computer Vision – A.Y. 2023/2024

Carlo Fabrizio*, Gabriele Giusti†

M.Sc. Computer Science and Engineering, Politecnico di Milano - Milan, Italy

*Email: *carlo.fabrizio@mail.polimi.it, †gabriele.giusti@mail.polimi.it*

*Student ID: *10747982, †10689818,*

1. Introduction

The general goal of the project is to improve the quality of the AdelaideRMF dataset[13]. The latter is a well known dataset, used for validating Multi Model Fitting solutions, and it is often used as a Benchmark in the field.

However, the dataset notoriously contains some errors and furthermore it is presented in the form of an hard clustering that is not the general case of Multi fitting scenarios [5]. Therefore our work consisted in: cleaning the dataset from the errors and re-propose it in the form of soft clustering to be more aligned with a Multi Model Fitting scenario.

The result of our work is thus a corrected version of the AdelaideRMF dataset, that contains less errors and can be used to validate Multi Model Fitting solutions.

1.1. Mathematical formulation

Multi Model Fitting in general

Let's consider a set of N points or point correspondences $P = \{p_i\}_{i=1}^N$ ($p_i \in \mathbb{R}^d$ or $p_i = \langle x_i, x'_i \rangle$ with $x_i, x'_i \in \mathbb{R}^d$) corrupted by noise and outliers, and given a family of parametric models Θ , the goal of multi-model fitting is to automatically estimate:

- A set of models $\{\mu(\theta_1), \mu(\theta_2), \dots, \mu(\theta_M)\}$ with $\theta_i \in \Theta$
- A set of structures $\{U_1, U_2, \dots, U_M\} \subset \mathcal{P}(P)$.

A structure U_i is defined as the set of inliers for the parametric model $\mu(\theta_i)$. More formally, a parametric model $\mu(\theta_i)$ is defined as the zero level set of a smooth parametric function $f_\mu(x, \theta)$, that is:

$$\mu(\theta) = \{p \in \mathbb{R}^d, f_\mu(p, \theta) = 0\} \quad (1.1)$$

Let's now define the error for datum p with respect to model $\mu(\theta)$:

$$e_\mu(p, \theta) = \min_{p' \in \mu(\theta)} dist(p, p') \quad (1.2)$$

We can now define the structure U_i as the Consensus Set of a model $\mu(\theta_i)$ with respect to a threshold δ as:

$$U_i = CS_\mu(P, \theta_i, \delta) = \{p \in P, e_\mu(p, \theta_i) \leq \delta\} \quad (1.3)$$

In general, it is not true that $U_i \cap U_j = \emptyset$ given $i \neq j, \in \{1, \dots, M\}$, indeed it is possible for a datum to be an inlier for multiple models.

Adelaide Dataset

The AdelaideRMF dataset consists of a set of pairs of images, each pair contains a set of points' correspondences P such that:

$$P = \left(\bigcup_{i=1}^M U_i \right) \cup O \text{ where } U_i \cap U_j = \emptyset; \forall i, j \in \{1, \dots, M\}, i \neq j \quad (1.4)$$

That is, the set of points correspondences is composed by the union of several structures and a set of outliers.

Given this definition, it is pretty clear that this dataset is intrinsically not correct for instances of Multi Model fitting, since the structures are disjoint sets, that is not the case in Multi Model Fitting scenarios. Moreover, the structures provided are not completely correct: there are points that are labeled as inliers but are actually outliers. Practically speaking, our objective with respect to a single image of AdelaideRMF, is to "clean" the labeling.

More formally, given P and the partition $\{U_i\}_{i=1}^M$, we aim to find another structure set $\{\hat{U}_i\}_{i=1}^M$, that does not contain labeling errors and that admits intersections between structures.

2. Outline

The aim of the project is basically to find outliers among a set of points that in principle are inliers. Note that, since those points have "already" been analysed and have been classified as inliers we are actually looking for outliers that are well masked among the correct inliers. Revealing the "impostors", then, requires the adoption of an approach that is as robust as possible.

The two cases, Fundamental Matrix estimation and Homography estimation were tackled in two similar but different ways. The Fundamental matrix scenario gave us some problems due to the intrinsic complexity of epipolar geometry.

The main problem, shared between the two scenarios, is that we have no labels or external validation measures to actually check the correctness of our results, therefore, we came up with a measure based on the concept of influence function[3], in order to evaluate the performance of our methods.

2.1. Homography scenario

In the context of Homography estimation, the estimation of outliers has been done using GC-RANSAC [2] method, notoriously being a very powerful model for Robust Model Fitting. However, GC-RANSAC, as all the RANSAC [4] based methods, requires as input parameter an inlier threshold on the residuals, that is used to distinguish between inliers and outliers. That threshold is specific for each image and for each model and, therefore, it must be estimated accordingly.

The estimation of the inlier thresholds was performed via statistical methods applied on model's residuals. More precisely, LMEDS instances were fitted on each model and the residuals were collected. Then statistical methods were applied on each model's residuals to compute the inlier threshold. The latter threshold was then used as parameter for GC-RANSAC for the specific model.

To choose between which statistical method to use on a given model, we applied a performance based approach, using the Silhouette score[7] as Performance index.

The results of the GC- RANSAC were finally validated using a measure based on the influence function measure.

A general overview of the procedure for a single image is the following:

- 1) Apply LMEDS → LMEDS residuals.
- 2) Change Point Detection on LMEDS residuals → Inlier Thresholds.
- 3) GC-RANSAC with Inlier Thresholds → GC-RANSAC residuals & soft clustering & outliers.
- 4) Validate the outliers using the Influence Function.
- 5) New dataset creation.
 - re-fit GC-RANSAC on inliers from soft clustering and correct the outliers.

2.2. Fundamental Matrix scenario

The context of Fundamental Matrix estimation was more complex to approach due to the complexity of epipolar geometry.

The first issue with this scenario concerned the definition of a suitable distance function for the computation of the residuals, initially we used the Sampson distance (better explained in Paragraph 7.1) that is the standard in the context of Fundamental Matrix, however, we then applied also the Symmetric Epipolar Distance (Sed, better explained in paragraph 7.2) for the computation of residuals and consequent inlier threshold computation, because the merge of the two (better explained later) gave us better experimental results.

The second issue we faced concerned the change point detection for Inlier threshold estimation. In particular, we noticed that all the statistical methods for the inlier threshold computation that are "robust" with respect to outliers (all but the variance based) were too severe resulting in low performances. Therefore we decided to use only the Variance based method for the inlier thresholds computation in this scenario.

The third problem that raised was the fact that GC-RANSAC method, although it is notoriously very powerful and robust, was not performing as expected, returning many "false positives" and "false negatives". To solve this problem we applied several different methods, including GC-RANSAC, and ensembled their results (better explained in paragraph 6), obtaining a better overall performance.

The fourth and last problem we have had in this context, was the fact that, for some images the above procedure returned "insignificant outliers", i.e., outliers whose correction would not be significant. The latter is likely due to the fact that those "insignificant outliers" are probably significant with respect to the other points of the same model, but their value is not significant with respect to other real outliers. To solve this problem we decided to correct only those images whose increase ratio of the influence function was above a threshold.

The procedure is summarized here below:

- 1) Apply LMEDS → LMEDS residuals Sampson & LMEDS residuals Sed.
- 2) Change Point Detection
 - LMEDS residuals Sampson → Inlier Thresholds Sampson.
 - LMEDS residuals Sed → Inlier Thresholds Sed.
- 3) Ensemble models
 - Ensemble with Inlier Thresholds Sampson → Labels Sampson.
 - Ensemble with Inlier Thresholds Sed → Labels Sed.
 - Final Labels = Labels Sed \cup Labels Sampson.
- 4) Validate the outliers using the Influence Function.
- 5) New dataset creation.

- Eliminate the outliers only for the images for which it is worth it (Influence Function Increase Ratio above 30).

In this scenario we preferred not to change the structure of the dataset to turn it into a soft clustering because the task is highly delicate. Adding points to the dataset requires careful verification, and an automatic approach might not be the right solution. Therefore, we chose not to implement it.

2.3. LMEDS for "masked outlier" detection

In order to understand how all the procedure was done, we have to briefly discuss the LMEDS robust estimator. This method is used to identify the "outliers" in the data, that is, points severely deviating from the model. If these points can be discriminated in the projection along the direction of the intercept, the above procedure is optimum.

Rousseeuw proposed the least-median-of-squares (LMEDS[11][8]) method in which the parameters are estimated by solving the nonlinear minimization problem.

$$\min_i \text{med}(r_i^2)$$

That is, the estimates must yield the smallest value for the median of squared residuals computed for the entire data set. The LMEDS minimization problem cannot be reduced to a least-squares based solution, unlike the M-estimators. The least-median-of-squares minimization must be solved by a search in the space of possible estimates generated from the data. Since this space is too large, only a randomly chosen subset of points can be analyzed.

Assume that our data consist of n points from which we want to estimate the p regression coefficients β_j , with $j = 0, \dots, (p - 1)$, of the linear model

$$z_i = \sum_{j=0}^{p-1} \beta_j x_i(j) \quad i = 1, \dots, n \quad (2.1)$$

The intercept value $\hat{\beta}_0(i_1, \dots, i_p)$ is then found by solving the minimization problem

$$\min_i \text{med}_i(r_i^2) \quad (2.2)$$

given $\hat{\beta}_j(i_1, \dots, i_p)$, $j = 1, \dots, (p - 1)$. To solve this problem we must use a mode estimation technique. The mode of a continuous probability distribution is the location of its maximum. In the case of a discrete, ordered sequence, the mode corresponds to the center of the subinterval having the highest density.

Thus, if we define the projected sequence

$$s_i = z_i - \sum_{j=1}^{p-1} \hat{\beta}_j(i_1, \dots, i_p) x_j(i) \quad (2.3)$$

compute its mode, and take it as $\hat{\beta}_0(i_1, \dots, i_p)$, we obtain the solution for the formulated problem (2.2). The breakdown point of the least median squares method is 0.5 because all the median computations are over the whole data set. So we need at least half of the dataset to be inliers to have a good estimation of the model.

3. GC-RANSAC

In the contest of Robust single model fitting the family of RANSAC models is well known for their state-of-the-art solutions.

3.1. Mathematical formulation of the problem

Let's define the following quantities:

- $P = \{p_1, p_2, \dots, p_N\}$ a set of points, with $p_i \in \mathbb{R}^n$
- $\theta \in \mathbb{R}^m$
- $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, a distance function ($\phi(p, \theta)$ = distance of point p to model θ).
- $L \in \{0, 1\}^N$: binary labeling of the N points (0=inlier, 1=outlier).
- $\varphi : \mathbb{R}^n \times \{0, 1\} \times \mathbb{R}^m \rightarrow \mathbb{R}$. Cost function that takes as input a point, its label and the model and returns a cost.
- $E(P, L, \theta)$ the cost associated to the entire set of points P . It is usually a summation of all $\varphi(p_i, \cdot, \cdot)$ for i in $1, \dots, N$.

The objective of RANSAC based models is the maximization of the Consensus set cardinality with respect to an inlier threshold ϵ . The Consensus set of model θ with respect to inlier threshold ϵ is defined as:

$$CS(\theta, \epsilon) = \{p_i \mid \phi(p_i, \theta) < \epsilon\}$$

3.2. Standard RANSAC

The basic RANSAC algorithm is defined in algorithm 1.

Algorithm 1 RANSAC

- 1: **Input:** P , ϵ (inlier threshold), k_{max} max iteration
 - 2: $E_{\{0,1\}}(P, \theta)^* = \infty$
 - 3: **for** $k = 1, 2, 3, \dots, k_{max}$ **do**
 - 4: Randomly select $S \subset P$
 - 5: Estimate model θ on S
 - 6: Compute the labeling L
 - 7: Evaluate the cost $E_{\{0,1\}}(P, L, \theta) = \sum_{p \in P} \varphi_{\{0,1\}}(p, L_p, \theta) = \sum_{p \in P} L_p$
 - 8: **if** $E_{\{0,1\}}(P, L, \theta) < E_{\{0,1\}}(P, \theta)^*$ **then**,
 - 9: $E_{\{0,1\}}(P, \theta)^* = E_{\{0,1\}}(P, L, \theta)$
 - 10: $\theta^* = \theta$
 - 11: **end if**
 - 12: **end for**
 - 13: $\theta^* = \text{Least Square Model fitting on inliers } (p : L_p = 1)$
 - 14: **Return** θ^*, L
-

where it is used a binary cost function $\varphi_{\{0,1\}}$:

$$\varphi_{\{0,1\}}(p, L_p, \theta) = L_p = \begin{cases} 1 & \text{if } \phi(p, \theta) > \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

In general, all the RANSAC variants are based on the same concepts: use a smart sampling strategy, use a smart cost function ϕ and/or improve the so-far-best-model with some optimization techniques. Citing few interesting extensions, the NAPSAC extension uses spatial coherence to impove

the sampling strategy (higher probability of sampling a point that is close to an inlier), the PROSAC model exploits the ordering of the points by their predicted inlier probability and the LO-RANSAC applies a local optimization step on the so-far-best-model in order to improve the quality of the results.

The GC-RANSAC it is a variant of RANSAC that tries to combine NAPSAC, PROSAC and LO-RANSAC. In particular, GC-RANSAC is a locally optimized RANSAC (as the LO-RANSAC) that exploits spatial coherence in the cost function of the local optimization step and uses a smart sampling strategy like the one of PROSAC or NAPSAC.

3.3. Definition of GC-RANSAC

Let's define the following quantities:

- μ = confidence
- $H(|L^*|, \mu) = \frac{\log(\mu)}{\log(1-P_I)}$ (with $P_I = \frac{\binom{|L^*|}{m}}{\binom{N}{m}}$) max iteration
- $K : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ Gaussian Kernel function.
- $\omega_k = \sum_{p \in P} K(\phi(p, \theta_k), \epsilon)$ support of the model.

The overall GC-RANSAC procedure is defined:

Algorithm 2 RANSAC

```

1: Input:  $P$ ,  $\epsilon$  (inlier threshold),  $r$ (sphere radius) ,  $\mu$  confidence.
2:  $\omega^* = 0$ 
3:  $n_{LO} = 0$ 
4:  $A$ = Neighborhood-graph using  $r$ 
5: for  $k = 1,2,3,\dots,H(|L^*|, \mu)$  do
6:   Draw a minimal sample set (using PROSAC or NAPSAC or uniform sampling strategy)  $S_k \subset P$ 
7:   Estimate model  $\theta_k$  on  $S_k$ 
8:   Compute  $\omega_k$  support of the model  $\theta_k$ 
9:   if  $\omega_k > \omega$  then
10:     $\theta^*, L^*, \omega^* = \theta_k, L_k, \omega_k$ 
11:    if Apply Local Optimization then
12:       $\theta_{LO}, L_{LO}, \omega_{LO} = \text{Local Optimization}$ 
13:       $n_{LO} += 1$ 
14:      if  $\omega_k > \omega$  then
15:         $\theta^*, L^*, \omega^* = \theta_{LO}, L_{LO}, \omega_{LO}$ 
16:      end if
17:    end if
18:  end if
19:  if  $n_{LO} = 0$  then
20:     $\theta^*, L^*, \omega^* = \text{Local Optimization}$ 
21:  end if
22: end for
23:  $\theta^*$ =Least Square Model Fitting on inliers
24: Return  $\theta^*, L^*$ 
```

Local Optimization.

The local optimization step aims computing an optimal labeling L^* defined as:

$$L^* = \operatorname{argmin}_L E(P, L, \theta)$$

- . The optimization step is performed in polynomial time considering a graph-cut algorithm.

The quantity $E(P, L, \theta)$ is defined as follows:

$$E(P, L, \theta) = E_K(P, L, \theta) + \lambda E_S(P, L, \theta)$$

where $E_S(P, L, \theta)$ is a cost function that uses spatial coherence, while $E_K(P, L, \theta)$ is a cost function based on a Gaussian Kernel.

In particular:

$$E_S(P, L, \theta) = \sum_{(p,q) \in A} \begin{cases} 1 & \text{if } L_p \neq L_q \\ \frac{1}{2}(K_p + K_q) & \text{if } L_p = L_q = 0 \\ 1 - \frac{1}{2}(K_p + K_q) & \text{if } L_p = L_q = 1 \end{cases} \quad (3.2)$$

where A is the neighbor graph, so (p, q) is an edge of the neighbor graph (p and q are neighbors) and $K_p = K(\phi(p, \theta), \epsilon)$; $K_q = K(\phi(q, \theta), \epsilon)$.

While:

$$E_K(P, L, \theta) = \sum_{p \in P} \varphi_K(p, L_p, \theta) = \sum_{p \in P} \begin{cases} K(\phi(p, \theta), \epsilon) & \text{if } L_p = 0 \\ 1 - K(\phi(p, \theta), \epsilon) & \text{if } L_p = 1 \end{cases} \quad (3.3)$$

The local optimization step is performed only a few number of times in order to avoid wasting time optimizing not promising models. The criterion for determining whether or not performing the local optimization step is based on the relationship between the inlier ratios of the two so-far-best models.

4. Performance Indexes

4.1. Introduction

As anticipated in the outline, we had no labels or ground truth values for validating our results. Therefore, we adopted clustering internal validation measures and the concept of influence function in order to estimate the performance of our models.

In the case of change point detection, considering it as an hard clustering task, a simple Silhouette score can be used to evaluate its performance.

The concept of influence function is based on the idea that, having a large set of inliers and few outliers, removing an inlier lightly changes the model computed, while, removing an outlier will strongly change the model computed. The validation measure we applied to support our results consists in showing that the average influence function score of the points that have been detected as outlier is much higher than the one of the inliers.

4.2. Silhouette

The Silhouette score is a well known internal validation measure for clustering that considers a trade-off between separation and cohesion.

The Silhouette score of a single point p_i is defined as :

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \quad (4.1)$$

where b_i and a_i are respectively the average distance of point p_i with the points in the second closest cluster and the average distance of point p_i with the points in the same cluster.

The Silhouette score of a set of data points is the average of all the silhouette scores of all the points:

$$S = \frac{1}{N} \sum_{i=1}^N s_i \quad (4.2)$$

4.3. Influence function

The core idea behind the influence function is that, suppose to have a large set of "coherent" (inliers) points and a small set of "discordant" (outliers) points, then, the impact on the model estimation of removing a single point from the coherent set will be negligible, while the impact of removing a discordant point will be significant.

An important consideration that must be done is that, the computation of the influence function, requires the adoption of a model that is sensitive to changes in the dataset. Initially we used GC-RANSAC for the purpose, but since it is stable and robust it was not giving us notable results. Finally we switched to the simple Least Square solution, obtaining interesting results.

The computation of the influence function for a single model of a single image is described in algorithm 3.

Algorithm 3 Influence function score

```

1: Input:  $P$ 
2:  $Scores = \emptyset$ 
3:  $Model\_baseline = Least - Square(P)$ 
4: for  $p$  in  $P$  do
5:    $P_i = P - \{p\}$ 
6:    $Model_i = Least - Square(P_i)$ 
7:    $score_i = ||Model\_baseline - Model_i||_2$ 
8:    $Scores = Scores \cup score_i$ 
9: end for
10: Return  $Scores$ 
```

The Results obtained are shown in the table 1. Note that only the pairs image-model where outliers are detected are presents.

Image-Model	Inliers Avg Score	Outliers Avg Score	Increase Ratio
img0 model0	0.054 (50)	2.738 (2)	51.010
img2 model0	0.566 (51)	15.375 (1)	27.167
img3 model0	2.916 (37)	979.683 (1)	336.000
img4 model0	18.867 (41)	122.403 (1)	6.488
img4 model1	5.295 (25)	44.576 (3)	8.419
img5 model0	0.005 (88)	0.111 (2)	22.749
img6 model0	0.689 (74)	117.497 (3)	170.473
img6 model1	0.044 (90)	0.112 (1)	2.534
img6 model2	0.417 (68)	15.149 (1)	36.285
img6 model3	0.267 (57)	17.530 (1)	65.752
img7 model0	8766.788 (16)	54684.886 (4)	6.238
img7 model1	0.689 (74)	117.497 (3)	170.473
img7 model2	0.009 (290)	55.764 (1)	6358.609
img7 model3	0.417 (68)	15.149 (1)	36.285
img7 model6	0.267 (57)	17.530 (1)	65.752
img8 model0	0.043 (106)	19.624 (2)	458.326
img8 model1	1.391 (49)	16.719 (3)	12.022
img9 model0	0.053 (49)	0.923 (1)	17.410
img9 model1	0.309 (45)	2.222 (1)	7.181
img10 model1	0.725 (80)	0.854 (2)	1.177
img11 model0	10.316 (48)	42.286 (1)	4.099
img11 model2	0.111 (71)	4.538 (1)	40.796
img12 model0	0.047 (63)	0.080 (1)	1.714
img12 model2	1.502 (45)	117.213 (1)	78.041
img13 model0	0.010 (91)	0.049 (1)	4.915
img13 model1	0.812 (76)	202.201 (1)	248.954
img14 model0	0.002 (184)	0.606 (1)	332.099
img14 model1	0.184 (68)	2.367 (3)	12.865
img16 model0	0.688 (85)	143.328 (1)	208.186

TABLE 1: Homography estimation - Inliers and Outliers Average Influence function Scores and their ratio in the last column - the number between parenthesis are the number of inliers and outliers

Image-Model	Inliers Average Score	Outliers Average Score	Increase Ratio
img0 model0	0.000 (143)	0.014 (3)	104.498
img1 model0	0.000 (94)	0.000 (3)	30.338
img2 model2	0.000 (53)	0.014 (1)	267.461
img3 model0	0.000 (63)	0.000 (6)	556.287
img3 model1	0.000 (66)	0.086 (2)	1736.593
img3 model2	0.000 (25)	0.000 (4)	17.874
img4 model0	0.000 (103)	0.000 (2)	0.180
img5 model0	0.000 (32)	0.085 (1)	39708.388
img5 model3	0.000 (56)	0.001 (2)	704.229
img6 model0	0.001 (62)	0.008 (1)	13.465
img6 model1	0.000 (99)	0.000 (3)	1.082
img7 model0	0.000 (32)	0.002 (2)	29.231
img7 model1	0.000 (56)	0.001 (1)	23.732
img8 model1	0.000 (57)	0.029 (1)	410982.937
img9 model0	0.012 (35)	0.001 (2)	0.090
img9 model1	0.000 (38)	0.000 (1)	49.830
img9 model2	0.000 (29)	0.009 (5)	10.214
img10 model2	0.000 (52)	0.001 (1)	81.966
img11 model0	0.000 (94)	0.000 (3)	4.154
img12 model0	0.000 (65)	0.000 (6)	18.934
img12 model1	0.000 (48)	0.068 (1)	723.364
img12 model3	1.803 (79)	0.031 (2)	0.017
img13 model0	0.000 (81)	0.000 (3)	1259.259
img14 model0	0.000 (76)	0.002 (2)	599.299
img14 model1	0.009 (66)	0.013 (6)	1.393
img15 model0	0.000 (76)	0.041 (2)	64894.942
img15 model1	0.000 (83)	0.001 (3)	765.715
img15 model2	0.000 (36)	0.000 (5)	28.383
img16 model0	0.000 (60)	0.000 (3)	1.789
img17 model1	0.000 (76)	0.000 (12)	3.011
img18 model0	0.000 (44)	0.000 (1)	8.127
img18 model1	0.000 (68)	0.087 (1)	402.114

TABLE 2: Fundamental matrix estimation - Inliers and Outliers Average Influence function Scores and their ratio in the last column - the number between parenthesis are the number of inliers and outliers

5. Change Point Detection for Inlier Threshold computations

5.1. Introduction

As introduced in paragraph 2 (the outline), we computed the inlier thresholds from the residuals of LMEDS. More precisely, we applied LMEDS on each model of each image and then, we computed the residual of each point of the image with respect to the model.

Let's focus on a single model of a single image, after the application of LMEDS and residuals computation and sorting, in an ideal situation we would obtain something like fig. 1 therefore, the inlier threshold would simply correspond to score value associated with the "knee" of the plot. On the X-axis

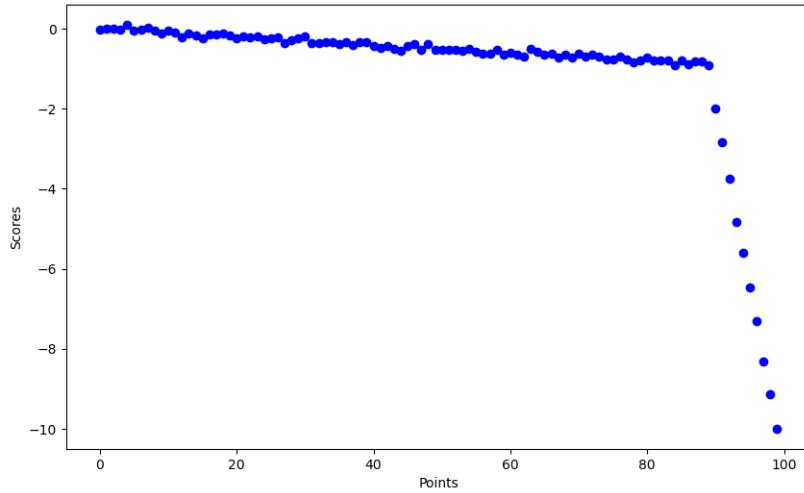


Figure 1: Example of residuals of a single model of a single image

we have the points and on the Y-axis we have *-residual* of the points. We plotted the residuals with the minus just because we preferred to visualize a decreasing function. However, since in reality the plots are more corrupted by noise, we obtained something like what is shown in fig. 2.

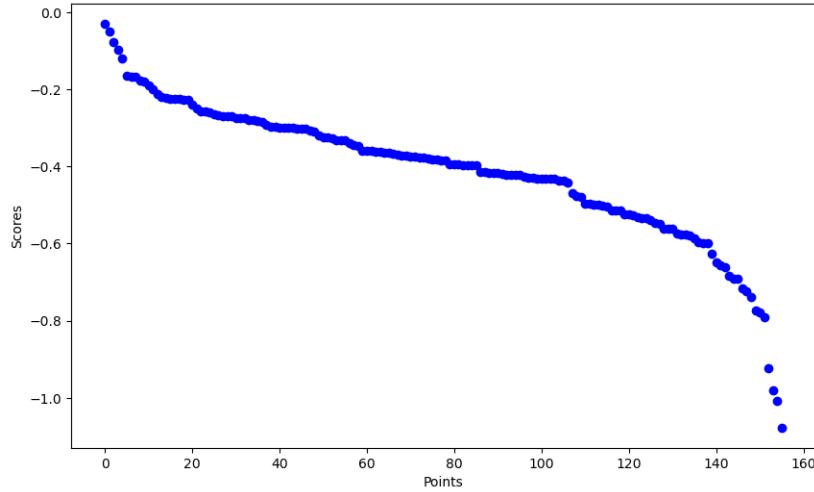


Figure 2: Example of residuals of a single model of a single image

The goal now is to find the specific value of the residuals that leads to the best possible split. The latter problem could be considered as an Anomaly Detection task, i.e. cluster the data into Inliers and Outliers by playing on the inlier threshold.

5.2. Methods

In order to compute the Inlier Threshold, we developed several different methods. All the methods are based on the same concept: estimate the location μ and the spread σ of the distribution of residuals and fix the threshold as $thresh = \mu + \alpha \cdot \sigma$ as a cut-off for inlier-outlier. The methods differs in the way they estimate σ . Most of the methods estimate the location using the Median since it is more robust with respect to outliers compared to the simple Mean. The list of methods and their estimation of the spread is the following:

	Location	Spread	Threshold
Inter-Quantile-Range	Q_2 (median)	$\sigma = Q_3 - Q_1$	$Q_3 + 1.5 \cdot \sigma$
Median Absolute Value	Median	$\sigma = \text{Median Absolute deviation}$	Median + $2.9 \cdot \sigma$
Variance Based Estimator S_n	Median	$\sigma = \text{Variance}$	Median + $1.5 \cdot \sigma$
Estimator Q_n	Median	$\sigma = S_n$	Median + $3 \cdot \sigma$
Forward Search[1]	Median	$\sigma = Q_n$	Median + $3 \cdot \sigma$
		$\sigma = \text{Adaptive Percentile}$	

Figure 3: Methods location and spread estimation

The S_n statistics [11] is computed as:

$$S_n = c \cdot \text{medi}\{\text{med}_j |r_i - r_j|\} \quad (5.1)$$

that is, for every point correspondence residual r_i with $i = 1, \dots, N$ we computed the median of the distances with all the other point correspondences residuals, this yields N numbers. The final statistics is the median of the N quantities.

The Q_n statistics [11] is computed as:

$$Q_n = d \cdot \{|r_i - r_j| ; i < j\}_{(k)} \quad (5.2)$$

that is, the k -th order statistic of the $\binom{n}{2}$ interpoint distances, where $K \simeq \binom{n}{2}/4$.

The values of α for each methods have been decided by visual inspection.

5.3. Selection of the method - Performance Index based approach

For each model of each image it has been selected the best inlier threshold method, among the ones of fig. 3, according to the Silhouette score obtained. Basically, each method returns an hard clustering on the points of the image, so its performance can be measured using internal clustering validation measures. We used the Silhouette score as a performance index of the different methods and for each model we applied the method with the highest performance index. It is like optimizing the Silhouette score by playing on the inlier thresholds but keeping a statistical based approach undercover. This approach is not "new" in the task of automatic estimation of inlier thresholds [12], however, in general, the methods works with the assumption that all the models of the same image have the same inlier threshold, which is not the case of our scenario.

Some of the results are shown in fig. 4 (in purple the detected inlier threshold).

In fig. 5 and 6 are shown some statistics on the estimated inlier thresholds and silhouette scores.

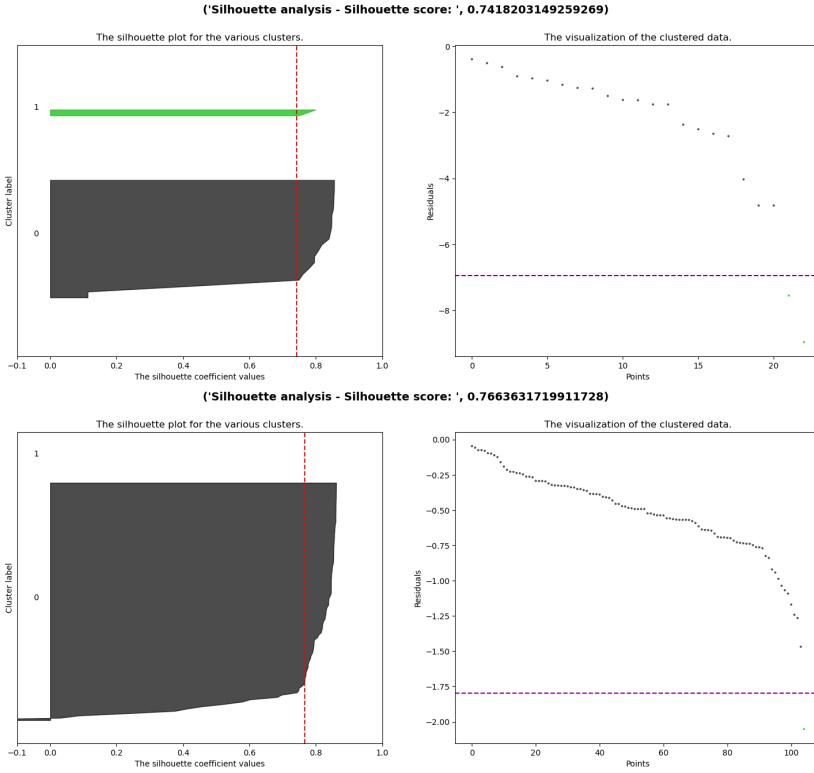


Figure 4: Silhouette Scores one model one image H estimation - inlier threshold

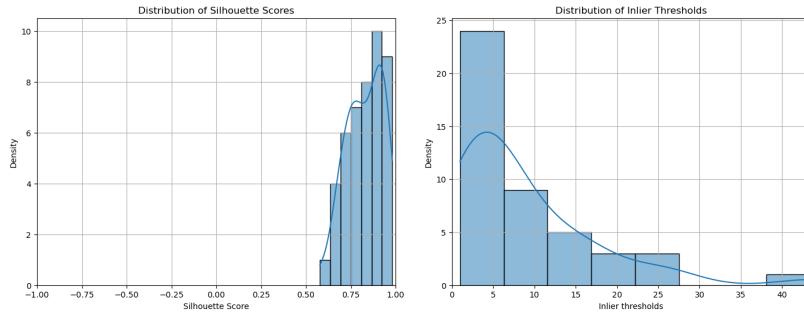


Figure 5: Silhouette Scores - Inlier thresholds distributions for H estimation

6. Ensemble

To enhance the performances of the outliers estimation we chose also to use simple ensembling methods. We implemented an averaging of the prediction scores over the points $s_i = \frac{1}{M} \sum_{j=1}^M r_{ij}$, where s_i is the score of the i -th points calculated as the average of the residual of the M models over that point, (e.g. models estimated by RANSAC, GC-RANSAC, LMEDS). If s_i is greater than the inlier threshold, then the point is classified as an outlier. We noted a more precise behaviour using ensembling rather than using single models to estimate the outliers.

Ensembling performs better both in recognizing false positives, as shown in fig. 7a and fig. 7c, and in finding new outliers that a single model can't detect, fig. 7b and fig. 7d, by combining the strengths of the models and minimizing their weaknesses.

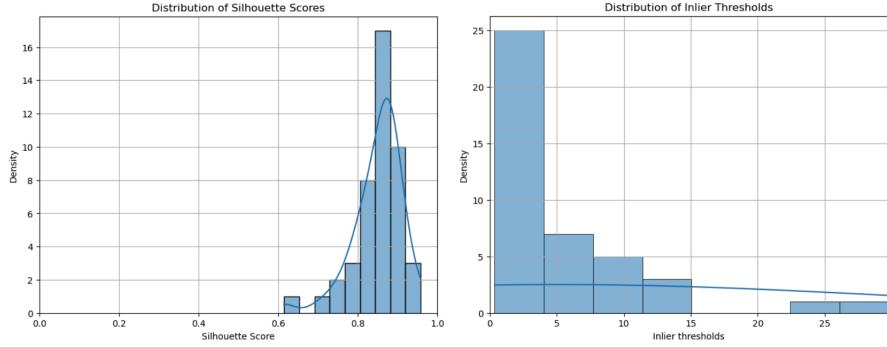


Figure 6: Silhouette Scores - Inlier thresholds distributions for FM estimation

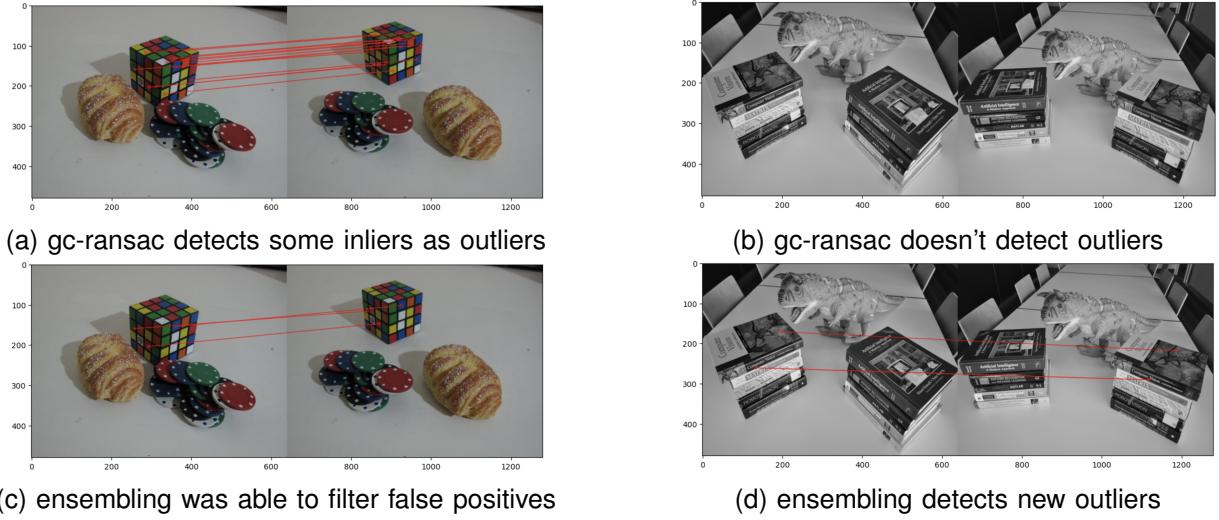


Figure 7: Results found by gc-ransac and ensembling on outlier detection in FM estimation

7. Fundamental matrix estimation

Fundamental matrix estimation deserves a separate section to be revised. Until now we tackled the problem of finding outliers from an estimated homography. This was done by estimating the residuals between the estimated homography mapping of the points and the one given from the dataset, which is the Euclidean distance of the two points $d(p, p') = |p - p'|$. The problem is that this kind of distance is not feasible when using the fundamental matrix, since it encapsulates the the intrinsics of the epipolar geometry, which is "the geometry of the intersections of the image planes with the pencil of planes having the baseline as axis" (Harley & Zisserman, 2004 - [9]). This is because the mapping is no more from point to point, like in homographies, but it is from point to line, which is called epipolar line.

More formally we write the epipolar line as

$$l' = [e'] \times H_\pi \mathbf{x} = F \mathbf{x}$$

, where we define $F = [e'] \times H_\pi$ the fundamental matrix, with e' the epipole and H_π the transfer mapping from one image to another via any plane π . It comes naturally that we have to identify a new type of distance to determine the residuals of this mapping with the already present points.

7.1. Sampson distance

The Sampson distance is the standard when it comes to have a metric that can estimate the reprojection error. It is a first-order approximation of the geometric error, in the case defined by $\mathbf{x}'^T F \mathbf{x} = 0$. It is defined as

$$d(F\mathbf{x}_i, \mathbf{x}'_i) = \frac{(\mathbf{x}'_i^T F \mathbf{x}_i)^2}{J J^T} \quad (7.1)$$

, where J is the partial-derivative matrix $J = \left[\frac{\partial(\mathbf{x}^T F \mathbf{x})}{\partial x}, \frac{\partial(\mathbf{x}^T F \mathbf{x})}{\partial y} \right]$. From the definition of J we obtain

$$J J^T = (F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2 + (F^T \mathbf{x}'_i)_1^2 + (F^T \mathbf{x}'_i)_2^2$$

, where $(F\mathbf{x}_i)_j^2$ represents the square of the j -th entry of the vector $F\mathbf{x}_i$. The cost function is

$$\sum_i \frac{(\mathbf{x}'_i^T F \mathbf{x}_i)^2}{(F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2 + (F^T \mathbf{x}'_i)_1^2 + (F^T \mathbf{x}'_i)_2^2} \quad (7.2)$$

Now that we have a way to compute residuals in this setting we can follow roughly the same procedure depicted in the previous sections.

7.2. Symmetric Epipolar Distance

Equation (6.2) is similar to

$$\sum_i d(x'_i, Fx_i)^2 + d(x_i, F^T x'_i)^2 = \quad (7.3)$$

$$\sum_i (\mathbf{x}'_i^T F \mathbf{x}_i)^2 \left(\frac{1}{(Fx_i)_1^2 + (Fx_i)_2^2} + \frac{1}{(F^T x'_i)_1^2 + (F^T x'_i)_2^2} \right) \quad (7.4)$$

which the symmetric epipolar distance. It measures the geometric distance of each point to its epipolar line.

7.3. Remarks on the usage of different distances

Even though the algebraic expressions of SED and Sampson distance are similar, we cannot conclude that they have the same accuracy properties. A more accurate investigation [6] reveals the discrepancy between SED and SD. Since

$$(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T \mathbf{x}'_i)_1^2 + (F^T \mathbf{x}'_i)_2^2 \geq (Fx_i)_1^2 + (Fx_i)_2^2$$

it follows that

$$\frac{1}{(Fx_i)_1^2 + (Fx_i)_2^2} \geq \frac{1}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T \mathbf{x}'_i)_1^2 + (F^T \mathbf{x}'_i)_2^2} \quad (7.5)$$

Similarly we have

$$\frac{1}{(F^T x'_i)_1^2 + (F^T x'_i)_2^2} \geq \frac{1}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T \mathbf{x}'_i)_1^2 + (F^T \mathbf{x}'_i)_2^2} \quad (7.6)$$

From (6.5) and (6.6) we have

$$\frac{1}{(Fx_i)_1^2 + (Fx_i)_2^2} + \frac{1}{(F^T x'_i)_1^2 + (F^T x'_i)_2^2} \geq \frac{2}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T \mathbf{x}'_i)_1^2 + (F^T \mathbf{x}'_i)_2^2} \quad (7.7)$$

Multiplying both sides by $0.5(\mathbf{x}_i^T F \mathbf{x}_i)^2$ we obtain

$$\frac{1}{2} \left(\frac{1}{(F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2} + \frac{1}{(F^T\mathbf{x}'_i)_1^2 + (F^T\mathbf{x}'_i)_2^2} \right) (\mathbf{x}'_i^T F \mathbf{x}_i)^2 \geq \left(\frac{1}{(F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2 + (F^T\mathbf{x}'_i)_1^2 + (F^T\mathbf{x}'_i)_2^2} \right) (\mathbf{x}'_i^T F \mathbf{x}_i)^2 \quad (7.8)$$

From (6.2) and (6.4) we can write

$$\frac{SED}{2} \geq SD$$

8. Experimental Results

In this section we presents some interesting results obtained. In figure 8, some interesting results obtained on the Homography estimation task: in red the original point correspondence, in green our corrected point correspondence.

In figure 9 some interesting outliers we have detected, in the case of Fundamental matrix the correction was not possible since we should have had a third image for that.

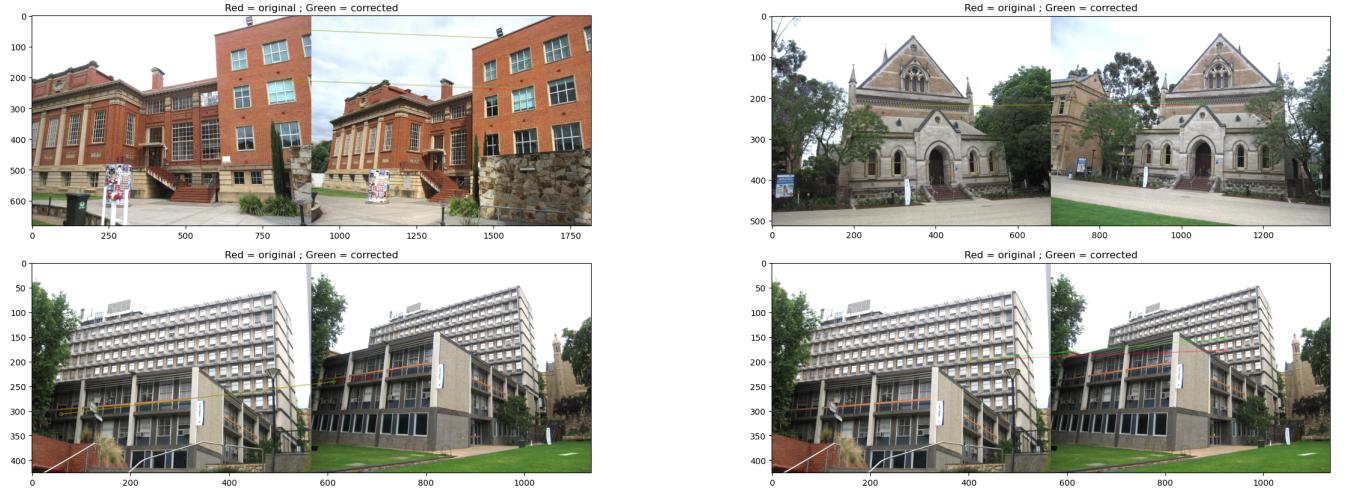


Figure 8: Outliers' correction for Homography - interesting results

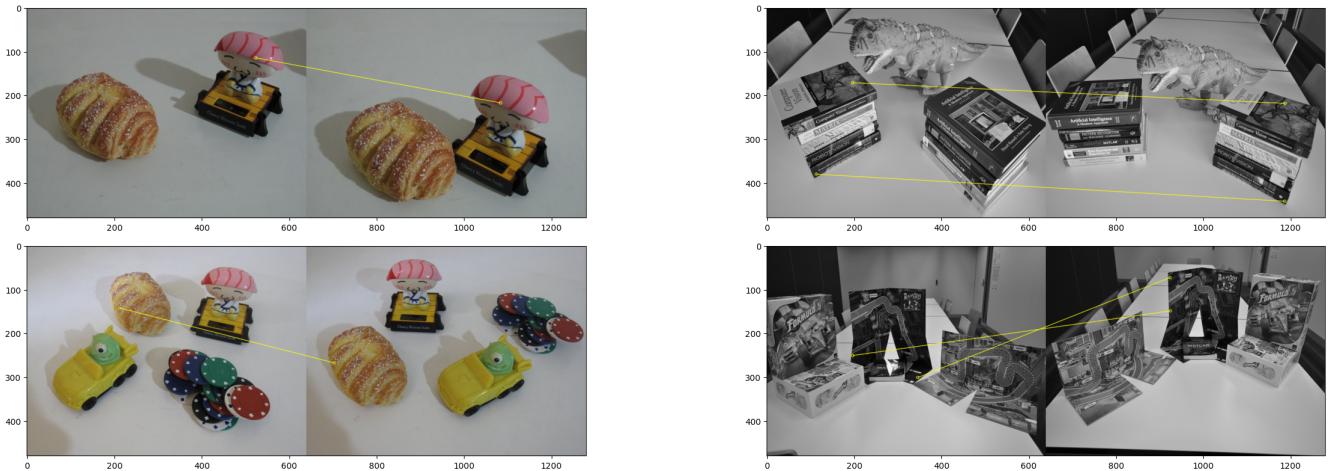


Figure 9: Fundamental Matrix - Outliers detected in the 4 models with the highest influence function increase ratio

9. Other experimental activities

We spent some time trying to define a performance index for soft clustering of instances of Multi Model Fitting. We needed a way to prove that the final dataset obtained after the corrections was actually better than the original one. Moreover, in our opinion, a definition of a metric to assess the performance of a soft clustering solution in the case of Multi Model Fitting, would have a great impact in the field, as it would be technically possible to optimize the clustering in function of such metric.

We started our analysis in this direction with a fuzzy extension of the standard silhouette score, called Fuzzy Silhouette [10]. However, the problem in our scenario is that we had to cluster point correspondences whose coordinates are not really well defined as they are represented as distances from a model. Moreover, we also had to consider the cluster of outliers (otherwise a simple elimination of all but 1 point for model would result in a perfect clustering) and defining the distance of a point from that cluster is not trivial. Basically, a possible visualization of the problem could be the one shown in fig. 10, an instance with only two models can be displayed.

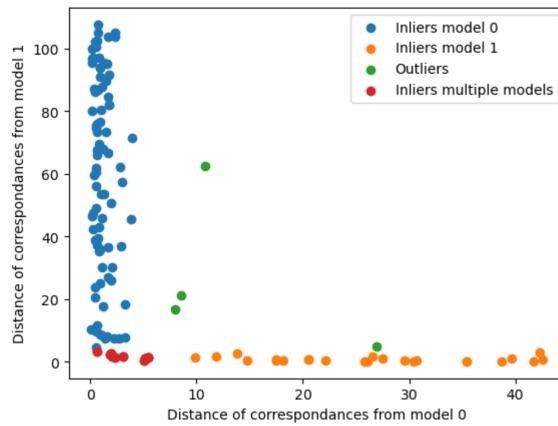


Figure 10: Soft Clustering in Multi Model Fitting scenario

We haven't succeed yet in defining a reasonable and coherent metric for this scenario.

9.1. Fuzzy Silhouette

Let's define the following quantities:

- $P = \{p_1, p_2, \dots, p_N\}$ set of data points with $p_i \in \mathbb{R}^n$.
- $\{C_1, C_2, \dots, C_K\}$ partition of data points into K clusters.
- $\mu_{ij} = [\sum_{l=1}^K (\frac{\text{dist}(p_i, C_j)}{\text{dist}(p_i, C_l)})^{\frac{2}{(m-1)}}]^{-1}$. The latter is an element of the so called Partition Matrix, μ_{ij} is large if point p_i is an inlier to cluster C_j , low otherwise. $m \in]1, \infty[$ is a weighting parameter called "fuzzifier" that controls the softness of the cluster. As m approaches 1 the clustering gets hard ($\mu_{ij} = 0$ or 1). As suggested in the paper, we used $m = 2$.

Given the partition matrix $P = [\mu_{ij}]_{i=1, \dots, N}^{j=1, \dots, K}$, the Fuzzy Silhouette Score FS is defined as:

$$FS = \frac{\sum_{i=1}^N (\mu_{ij}^* - \mu_{ij}^{**})^\alpha s_i}{\sum_{i=1}^N (\mu_{ij}^* - \mu_{ij}^{**})^\alpha} \quad (9.1)$$

where s_i is the standard silhouette score of point p_i and μ_{ij}^* , μ_{ij}^{**} are respectively the first and the second largest element of the i -th row of the partition matrix.

The coefficient α is a user defined parameter (1 by default) that controls the fuzziness of the Silhouette score. When α is 0 the Fuzzy Silhouette is exactly the Standard Silhouette. When α increase, the relative impact of points in overlapping clusters is decreased. Basically, the fuzzy silhouette score of a single point is the standard silhouette score normalized using the values of the partition matrix.

Now we have to define the distances point-point for the different points belonging to different clusters. Let's consider a single image, we have: one cluster for each model and the cluster of outliers.

The distance of point p_i to a model j is simply its residual $r_{i,j}$.

Let's define the distance of a point to the cluster of outliers

$$\text{dist}(p_i, O) = \begin{cases} 0 & \text{if } p_i \in O \\ \min_j \|r_{i,j} - \epsilon_j\| & \text{if } p_i \notin O \end{cases}. \quad (9.2)$$

where $r_{i,j}$ is the residual of point p_i to model j and ϵ_j is the inlier threshold of model j .

If a point p_i belongs to multiple models j and k , we considered the union of the points of the two models as the points of the same cluster of p_i , that is point p_i is in the same cluster with all the points of model j and also with all the point of model k .

Given these considerations we applied the fuzzy silhouette, however we didn't obtained reasonable results so we decided to abandon this direction.

References

- [1] Anthony C. Atkinson and Marco Riani. Robust diagnostic regression analysis, 2000.
- [2] Daniel Barath and Jiri Matas. Graph-cut ransac.
- [3] Ruwan Tennakoon Tat-Jun Chin Alireza Bab-Hadiashar Giang Truong Erchuan Zhang, David Suter and Syed Zulqarnain Gilani. Maximum consensus by weighted influences of monotone boolean functions.
- [4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.
- [5] Luca Magri and Andrea Fusiello. T-linkage: a continuous relaxation of j-linkage for multi-model fitting, 2014.

- [6] Mohammed F. Tolba Mohammed E. Fathy, Ashraf S. Hussein. Fundamental matrix estimation: A study of error criteria, 2017.
- [7] Jr. Mohammed J. Zaki, Wagner Meira. Data mining and machine learning: Fundamental concepts and algorithms 2nd edition - cambridge university press, 2020.
- [8] AZRIEL ROSENFELD DONG YOON KIM PETER MEER, DORON MINTZ. Robust regression methods for computer vision: A review, 1991.
- [9] Andrew Zisserman Richard Hartley. Multiple view geometry in computer vision, 2003.
- [10] E.R. Hruschka R.J.G.B. Campello. A fuzzy extension of the silhouette width criterion for cluster analysis, 2006.
- [11] Peter J. Rousseeuw. Least median of squares regression, 1984.
- [12] Roberto Toldo and Andrea Fusiello. Automatic estimation of the inlier threshold in robust multiple structures fitting.
- [13] Hoi Sim Wong, Tat-Jun Chin, Jin Yu, and David Suter. Dynamic and hierarchical multi-structure geometric model fitting, 2011.