Università degli Studi di Milano

Department of Mathematics “Federigo Enriques"
Doctoral School in Mathematical Sciences
Ph.D. in Mathematics and Statistic for the Computational Sciences

Multiple structures recovery via preference analysis in conceptual space

Luca Magri

Advisor: Prof. Andrea Fusiello

Coordinator: Prof. Giovanni Naldi

INF/01, XVIII cycle, 2015

Ph.D. Thesis
Advisor:
prof. A. Fusiello

Università degli Studi di Milano
Department of Mathematics “Federigo Enriques"
Via Cesare Saldini 50, 20133 Milano
Italy

Contents

Abstract

Finding multiple models (or structures) that fit data corrupted by noise and outliers is an omnipresent problem in empirical sciences, including Computer Vision, where organizing unstructured visual data in higher level geometric structures is a necessary and basic step to derive better descriptions and understanding of a scene.
This challenging problem has a chicken-and-egg pattern: in order to estimate models one needs to first segment the data, and in order to segment the data it is necessary to know which structure points belong to. Most of the multi-model fitting techniques proposed in the literature can be divided in two classes, according to which horn of the chicken-egg-dilemma is addressed first, namely consensus and preference analysis. Consensus-based methods put the emphasis on the estimation part of the problem and focus on models that describe has many points as possible. On the other side, preference analysis concentrates on the segmentation side in order to find a proper partition of the data, from which model estimation follows.

The research conducted in this thesis attempts to provide theoretical footing to the preference approach and to elaborate it in term of performances and robustness. In particular, we derive a conceptual space in which preference analysis is robustly performed thanks to three different formulations of multiple structures recovery, i.e. linkage clustering, spectral analysis and set coverage. In this way we are able to propose new and effective strategies to link together consensus and preferences based criteria to overcome the limitation of both. In order to validate our researches, we have applied our methodologies to some significant Computer Vision tasks including: geometric primitive fitting (e.g. line fitting; circle fitting; 3D plane fitting), multi-body segmentation, plane segmentation, and video motion segmentation.

1

The challenges of multiple structures estimation

Computer Vision ultimately aims to mimic and emulate human visual abilities, starting from perception up to understanding and decision-making processes. Although this ambitious goal has inspired research efforts towards the design of automatic systems that can effectively analyze and extract information from the visual environment under almost any operating condition, there is still an unfulfilled need for compact, abstract representations of the visual content in order to bridge the semantic gap that separates automatic perception from human comprehension.

A first step towards this direction is represented by geometric multi-model fitting, a stream of research aimed at recovering geometric models from unstructured data for the purpose of organizing and aggregating visual content in adequate higher-level geometric structures.

To set a general context, let μ be a model – e.g. lines or other geometric primitives – and $X = \{x_1, \dots, x_n\}$ be a finite set of n points, possibly corrupted by noise and outlier. The problem of multiple model recovery consists in extracting κ instances of μ – termed structured – from the data, defining, at the same time, κ subsets $C_i \subset X$, $i = 1, \dots, \kappa$, such that all points described by θ_i are aggregated in C_i. Often the models considered are parametric, i.e. the structures can be represented as vectors in a proper parameter space Θ.

This ubiquitous task can be encountered in many Computer Vision applications. A typical example of this problem can be found in 3D reconstruction, where multi-model fitting is employed either to estimate multiple rigid moving objects and hence to initialize multi-body Structure from Motion [35, 73], or to produce intermediate geometric interpretations of reconstructed 3D point cloud by fitting planar patches and geometric primitives [17, 44, 98]. Other scenarios in which the estimation of multiple geometric structure plays a primary role include face clustering, body-pose estimation, augmented reality, image stitching and video motion segmentation, just to name a few.

In all these cases the information of interest can be extracted from the observed data and organized in semantical significant structure by estimating some underlying geometric parametric models, e.g. planar patches, homographic transformations, fundamental matrices or linear subspaces.

Due to the huge volume of visual data involved in Computer Vision applications, parameter estimation techniques are typically heavily overconstrained, and model fitting problems ought to be solved by least squares methods or, more generally, maximum likelihood estimation techniques. Nonetheless, the peculiar nature of visual data – which are typically affected by arbitrarily large gross measurement errors– impedes the adoption of classical statistical estimators, which are fragile and sensitive to outliers. For these reasons, robust estimators are required. The presence of multiple structures hinders also robust estimation, which, in addition to gross outliers and noise, has to cope with pseudo-outliers, a concept introduced by Stewart [91] for describing those points that do not match a model of interest because they are inliers of a different structure.

Many other issues are afoot, making the multi-model fitting problem a challenging and demanding task. For example, the estimation and the segmentation tasks are two closely entangled aspects that give rise to a "chicken-and-egg" dilemma: points should be segmented based on their geometric proximity to structure whose unknown parameters must be estimated at the very same time. In other words, in order to estimate models one needs to first segment the data, but conversely in order to segment the data it is necessary to know the structures associated with each data point.

In addition the problem of multi-model fitting, is inherently ill-posed, since many different interpretations of the same dataset are possible. Making the problem tractable requires a regularization strategy that constrains the solution using prior information, usually in the form of one or more parameter, such as the number κ of sought structures. Unfortunately estimating this quantity turns to be a thorny problem. Following the spirit of Occam's razor – that one should not presume more things than the required minimum – κ should be kept as low as possible, but finding a correct tradeoff between data fidelity and model complexity (a.k.a. bias-variance dilemma) is an intricate model selection task.

1.1 Consensus and preferences

Among the wide variety of algorithms proposed in Computer Vision to address these challenges, the analysis of *consensus* together with its dual counterpart, the analysis of *preferences*, can be traced as a *fil rouge* connecting the extensive literature on multi model geometric fitting.

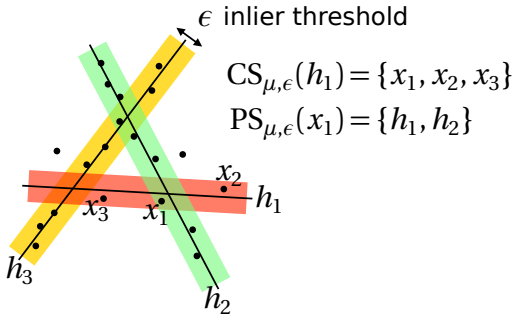


Fig. 1.1: Consensus and preference sets

In order to define these pivotal concepts, it is necessary to introduce a goodness-of-fit measure that evaluates how adequately a point is described by a given structure. To this end two notions, particularly important in statistics, come to play a relevant role: the definition of the residual and the knowledge of the scale. Residuals can be regarded as the deviations from an estimated structure and are modeled by an error function

$$\mathrm{err}_\mu \colon X \times \Theta \to \mathbb{R}^+ \tag{1.1}$$

that associates to every point-model pair $(x,\theta) \in X \times \Theta$ the corresponding residual error $\mathrm{err}_\mu(x,\theta)$. As the scale is concerned, a threshold $\epsilon \in \mathbb{R}^+$, commonly known as *inlier threshold*, is actually used to assess the noise variance. A point x is said to belong to a given structure θ if

$$\mathrm{err}_\mu(x,\theta) \leq \epsilon. \tag{1.2}$$

Thus, the *consensus set* of a model is simply defined as the set of points that fits the model within a certain inlier threshold ϵ:

$$\mathrm{CS}_{\mu,\epsilon}(\theta) = \{x \in X \colon \mathrm{err}_\mu(x,\theta) \leq \epsilon\}. \tag{1.3}$$

Dually, the *preference set* of a point is the set of models having that point as an inlier:

$$\mathrm{PS}_{\mu,\epsilon}(x) = \{x \in X \colon \mathrm{err}_\mu(x,\theta) \leq \epsilon\}. \tag{1.4}$$

Most of the multi-model fitting techniques proposed in the literature can be ascribed to one of these two concepts, according to which part of the chicken-egg-dilemma is addressed first. Consensus-based algorithms put the emphasis on the estimation part and the focus is on models that have to describe has much points as possible. On the other hand, preference approaches concentrate on the segmentation side of the problem, and are aimed at finding a proper partition of the data, hence estimation follows

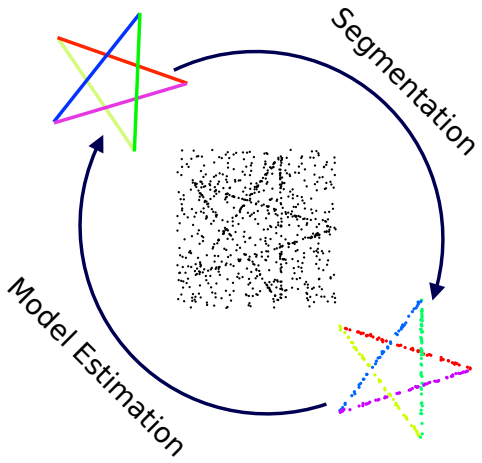


Fig. 1.2: The chicken-and-egg problem

as a consequence. In the next section we attempt to retrace the path that, starting from consensus throughout preference analysis, have been followed to address the challenging issues presented by multiple structures recovery.

1.2 Consensus analysis

Consensus analysis stands out as one of the first attempts to address robust model estimation. The methods belonging to this category follow a common paradigm. At first the space Θ of all the feasible structures is approximated as a suitable finite hypothesis space H in different ways. Then a voting procedure elects the structures in H that best explain the data in terms of consensus set.

Random Sample Consensus

The idea of exploiting consensus is at the core of the celebrated RANSAC (Random Sample Consensus), a method, firstly introduced by Fischler and Bolles [34], aimed at estimating the parameters of a single model in the presence of large amounts of outliers. The goal of minimizing squared residuals, typical of Least Square method, is replaced in RANSAC with the objective of maximizing the size of the consensus set of a structure, provided the inlier threshold as input. RANSAC approximately maximizes this criterion by searching through a pool of putative structures determined by randomly sampling. In particular at each iteration a *Minimum Sample Set* (MSS) – composed by the minimum number, say ζ, of points necessary to instantiate the free parameters of a structure

– is drawn. In this way the estimation problem in the continuous domain Θ is converted into a selection problem in a finite discretized subset $H \subset \Theta$.

For each estimated model the corresponding consensus set is computed counting the residuals below the inlier threshold. This procedure is repeated until a structure having enough supporting inliers is discovered among the data. A number of efforts have been made to improve the RANSAC paradigm. For example, MSAC (M-estimator Sample Consensus) and MLESAC (Maximum Likelihood Estimation Sample Consensus) [100] propose to increase the robustness of the RANSAC paradigm incorporating the use of M-estimator techniques. A lot of other refinements in terms of both accuracy and efficiency have been made [57], for example different sampling strategies have been proposed in the literature to reduce the number of iterations necessary to recovery an inlier structure. A nice survey on all these advancements can be found in [21] or in the more comprehensive overview of recent researches presented in [81] where USAC (Universal Framework for Random Sample Consensus) is derived.

Sequential RANSAC *& Multi-*RANSAC

The RANSAC strategy has been adapted to estimate multiple structures. Its most straightforward generalization is embodied by Sequential RANSAC, an iterative algorithm that executes RANSAC many times and removes the found inliers from the data as each structure is detected. Zuliani et al. [122] noticed some drawbacks of this greedy *esitmate-and-remove* approach, which in fact may happen to be sub-optimal since the quality of the attained solution can be affected by inaccurate estimation of the initial structures.

In order to correct this behavior Zuliani et al. introduced Multi-RANSAC. Remaining tied to the idea of maximizing the consensus set, Multi-RANSAC replaces the sequential scheme with a parallel approach. Rather than looking for a single structure having the largest consensus, κ models having maximal support are searched simultaneously at each iteration. This is done by updating iteratively a collection of κ models with κ new sampled structures using a fusion procedure that enforces explicitly the disjointness of the obtained consensus sets. However as demonstrated experimentally in [96], this method may yields poor results in presence of intersecting structures.

Hough Transform

The popular Hough transform and its randomized version (Randomize Hough Transform [111]) can be regarded as well as consensus-oriented algorithms. In these approaches the parameter space Θ is approximated as a quotient space $H = \Theta/_{\sim}$ in which models are represented as equivalence classes of similar structures. The space H is hence employed to build an accumulator collecting data votes: every point adds a vote

to the bins representing the structures it belongs to. After voting is complete, the accumulator is analyzed to locate the maxima that individuate the desired structures. Differently from RANSAC, where H is a discrete sampled version of Θ, in Hough transform the elements of the hypothesis space provide an exhaustive representation of the parameter space, and tentative models are all considered simultaneously. This however comes at the cost of defining a proper quantization of the space, which rapidly becomes intractable as the degrees of freedom of the models increase. Randomized Hough Transform instead of considering the votes of all the points, exploits random sampling to approximate the accumulator for votes, reducing the computational load.

This strategy can be considered as an instance of a more general approach, that consists in finding modes directly in Θ [93]. In this way the difficulties of the quantization step, are alleviated by mapping the data into the parameter space through random sampling and then by seeking the modes of the distribution with mean-shift [26] .

In all these consensus based methods, alongside the voting phase, the approximation of Θ is a recurring theme and a very delicate step. The key point is that, when multiple structures are hidden in the data, consensus oriented algorithms have to disambiguate between genuine structures and redundant ones, i.e. multiple instance of the same model with slightly different parameter. This crucial difficulty is hence addressed by enforcing several disjointedness criteria implicitly implemented in the different approximations of the solution space.

For instance, Hough transform attempts to handle redundancy by capturing similar structures in the same equivalence class via the problematic quantization of Θ. Along the same line, the bandwidth used in mean shift can be thought as a way to localize and aggregate redundant models. As suggested in [36] also both Sequential RANSAC and Multi-RANSAC enforce disjointedness by avoiding to sample similar models. As regards Sequential-RANSAC, this idea can be individuated in the iterative removal of the discovered inliers and in the subsequent sampling of the hypotheses on the remaining data. In Multi-RANSAC this is more evident, since this algorithm explicitly includes in its parallel approach a disjointedness constraint by directly searching for the best collection of κ disjoint models.

In practice, however, using consensus as the only criterion seems short-sighted, as, in many cases, ground truth models can have mutual intersection grater than redundant ones, as shown in Figure 1.3 and, consequently, the rough consensus fails in discerning authentic structures.

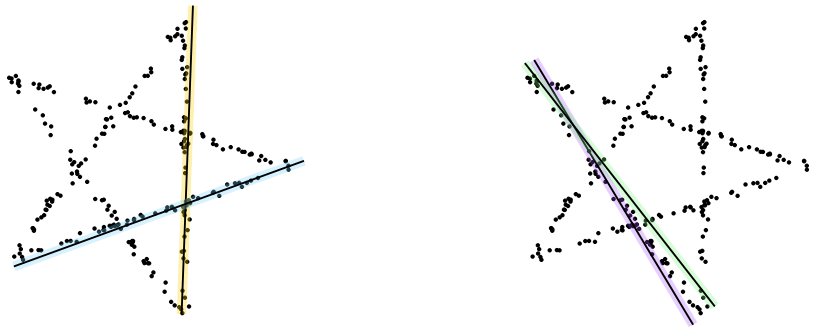


Fig. 1.3: Redundant structures (left) may happen to have smaller intersecting points than genuine intersecting ones (right)

1.3 Preference analysis

In order to overcome the difficulties inherent of consensus methods, it has been proposed to tackle the problem from a different point of view. Instead of exploiting consensus of structures, the role of data and models are reverted: rather than representing models and inspecting which points match them, the preference sets of individual data points are examined.

Residual Histogram Analyisis

This idea can be traced back to Residual Histogram Analysis [119] where the residuals distributions of points, with respect to a set of putative structures randomly sampled, is taken in consideration. In particular, an histogram analysis of the residuals is used to reveal the most significant structures as peaks in the histograms. In addition, the number of models is automatically determined by the median number of modes found over all data points. Even if, in practice, the mode-finding step of this strategy suffers of low accuracy and depends critically on the bin size adopted, this method has the merit to reformulate the model-estimation task in an alternative space where points are described by their residuals.

J-Linkage

J-Linkage algorithm [96] embodies the spirit of preference analysis exploiting a preference based representation of data in order to discover groups of points belonging to the same structures as cluster in a conceptual space.

In particular, at high-level, a two steps *first-represent-then-clusterize* scheme is implemented: at first, data are represented by the votes they grant to a set of model hypotheses, then a greedy agglomerative clustering is performed to obtain a partition of the data. Several trends in common with previous methods can be recognized: an inlier threshold ϵ needs to be provided in advance as in RANSAC and the idea of cast points' votes echoes Randomize Hough Transform. Yet J-Linkage does not work in a quantize space, which is at the root of the shortcoming of Hough Transform, nor in the residual space, which leads to the difficulties of modes estimation, but explicitly introduces a *conceptual space* where points are portrayed by the *preferences* they have accorded to random provisional models. The changes of perspective entailed by preference analysis results in a different approach to the chicken-&-egg dilemma. Structures are recognized as groups of neighboring points in the conceptual space therefore the emphasis is shifted from the estimation to the segmentation part of the problem. More in details J-Linkage can be described in the following way:

Conceptual representation: The method starts generating a space model hypothesis $H = \{h_1, \dots, h_m\}$ by drawing m minimal sets of data points necessary to estimate the model; A $n \times m$ matrix P is build whose (i, j)-th entry is defined as

$$P(i,j) = \begin{cases} 1 & \text{if } x_i \text{ is explained by the } j\text{-th model} \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

Each row P_i can be easily identified with the characteristic function of the preference set $\mathrm{PS}_{\mu,\epsilon}(x_i)$ of a given point x_i, i.e. indicates which models a point has given consensus to. In turn x_i is depicted vector wise in the conceptual space $\{0,1\}^m$. The key idea is that points belonging to the same structure will have similar preference set, in other words, they will cluster in the conceptual space.
The preference representation is extended in a straightforward manner to subsets of data. Let $U \subseteq X$, U is portrayed as the Preference Set of all the common preferences among all the data belonging to it:

$$\mathrm{PS}_{\mu,\epsilon}(U) = \bigcap_{x \in U} \mathrm{PS}_{\mu,\epsilon}(x). \tag{1.6}$$

Clustering: The clustering algorithm proceeds in a bottom-up manner. At first every data is put in its own cluster. The distance between clusters is computed as the Jaccard distance [48] between the respective conceptual representations. The Jaccard distance between two sets A, B is defined as

$$J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{1.7}$$

and measures the degree of agreement between the votes of two clusters and ranges from 0 (identical votes) to 1 (disjoint preference sets).
Starting from singletons, each sweep of the algorithm merges the two clusters with the smallest Jaccard distance. The cut off value is 1. The clustering procedure can be summarized as follows:

1. Put each datum in its own cluster.
2. Define the conceptual representation of a cluster using (1.6).
3. Among all current clusters, pick the two clusters with the smallest Jaccard distance.
4. Replace these two clusters with the union of the original ones.
5. Repeat from step (3) while the smallest Jaccard distance is lower than 1.

The outcome of this procedure is a segmentation of the data in disjoint clusters C_i such that $X = \bigcup_i C_i$ and, if $i \neq j$, $C_i \cap C_j = \emptyset$. Clusters enjoy the following two properties:

1. for each cluster there exists at least one model that is in the preference set of all its points.
2. one model cannot be in the preference sets of all the points of two distinct clusters.

The parameters of the returned structures θ_i are estimated by least squares fitting on each cluster of points C_i. It is worth noting that, if outliers are not present in the data, the number of clusters is automatically detected by this algorithm. This is certainly a remarkable propriety, because the majority of other multi model fitting techniques requires this information as input parameter. Moreover this preference approach is robust to outliers, that can be recognized as observations whose preferences deviate significantly from the rest of the data, and tend to emerge as small clusters.

This method has demonstrated to be very effective in practice, and has been extensively exploited in the literature in many multi-model fitting problems[1], as the somehow unexpected application to cryptography reported in Chapter 6. Nonetheless, the theoretical footing of J-Linkage is still very little explored, and its greedy behaviour is not completely satisfactory (e.g. J-Linkage has been reported to be biased toward under-segmentation [37]), and robustness is gained only *a posteriori* by an *ad hoc* outlier rejection strategy.

Kernel methods

Along the same line of J-Linkage, Kernel Fitting [18] exploits preferences to derive a kernel matrix that encapsulate the order in which models are preferred, (i.e., the order of their residuals). The rationale is that points belonging to the same ground-truth models

[1] For a list of applications of J-linkage see `http://www.diegm.uniud.it/fusiello/demo/jlk/j-parade.html`

should have similar orders of preferred models. Exploiting this information, a transformation is applied to the data points into a space which permits the detection of outliers. The removal of outliers yields to a reduced kernel matrix that, in turn, is used to oversegment the remaining inliers. Finally a merging scheme is used to reassemble these models into the final model estimates.

RCMSA (Random Cluster Model Simulated Annealing) [77] as well takes advantage of the same idea representing data points as permutations on a set of tentative models constructed iteratively, using subsets larger than minimal. Point preferences are organized in a weighted graph and the multi-model fitting task is stated as a graph cut problem which is solved efficiently in an annealing framework.

Higher order clustering

A stream of investigations focused on higher order clustering [1, 41, 49, 117] implicitly adopts a preference based approach. In these works higher order similarity tensors are defined between n-tuple of points as the probability that these points are clustered together exploiting the residual error of the n points with respect to provisional models. In this way preferences give rise to a hypergraph whose hyperedges encode the existence of a structure able to explain the incident vertices. The problem of multi-model fitting is hence reduced to find highly connected component in this preference hypergrpah. In practice, the similarity tensor is properly reduced to pairwise similarity and fed to spectral clustering-like segmentation algorithms.

In summary different perspectives on the multi-model fitting problem have been adopted. Consensus oriented method look at the problem considering some kind of accumulation space – either consisting in individual models, as in RANSAC, or in equivalence classes of structures as in Hough transform – in which votes of points are collected. Structures are hence estimated maximizing consensus. This paradigm has demonstrated to be successful in single model estimation, but it is less effective if multiple structures are present in the data, because consensus does not allow to distinguish clearly between genuine models and redundant ones. When multiple structures recovery is viewed through the lens of preference analysis the attention is shifted to the segmentation part of the problem. Data are represented as points in a high dimensional space or as vertices in hypergraph and clustered together using ad hoc techniques.

It goes without saying that the state-of-the art on multi-model fitting can be also described along other dimensions. For example multiple structures recovery can be seen by an optimization perspective as the minimization of a global energy functional composed by two terms: a modeling error which can be interpreted as a likelihood term, and

a penalty term encoding model complexity mimicking classical MAP-MRF objectives. A survey of multi-model fitting methods form this point of view can be found in [47]. Optimization routines have also been tailored to specific instances of multi-model fitting: a relevant case is subspace segmentation where the use of low-rank and sparsity analysis has produce a solid literature, accurately illustrated for example in [116].

1.4 Outline and contributions

The starting point of this thesis is preference analysis. In particular, in Chapter 2, the first step is to enhance the conceptual representation proposed in J-Linkage by exploiting the use of M-estimators to robustly depict points preferences. In this way we obtain a continuos space, termed Tanimoto space, that is on the basis of three formulations of the multi-model fitting problem: hierarchical clustering, spectral clustering and set cover.
Hierarchical clustering is employed in Chapter 3 where it is used to address two major issues related to J-Linkage: robustness to outliers and scale estimation.
In Chapter 4, we present a robust version of spectral clustering for preference analysis that takes advantage of considerations rooted in consensus. In particular we attempt to disentangle the chicken-and-egg recursive nature of multiple structure recovery reducing it to many single robust model estimation problems.
Finally the perspective is somehow reversed in Chapter 5, when we depart from the clustering preferences and we return back to a discrete formulation mainly focused on consensus. In this framework, based on the notion of cover set, we are able to revisit many classical algorithm in a common framework. Moreover, complementing coverages with preference analysis as side information, we derive a method to deal with intersecting multiple structures and outliers in a principled manner.

2

A lift to Tanimoto space

The binary preference analysis implemented by J-Linkage suffers of the same poor local robustness of RANSAC with respect to MSAC, therefore we propose to enhance it by relaxing the notion of preference set. To this end we borrow from robust statistics the weighting functions adopted by M-estimators and use them as voting function to express robustly point preferences. As a result, we alleviate the influence of outliers and mitigate the truncating effect of the inlier threshold. We hence conceive a continuous conceptual space in which the Jaccard distance is generalized by the Tanimoto distance in order to handle the continuous representations of points.

2.1 Building on the preference trick: the Tanimoto space

In pattern recognition a theoretical framework for conceptual representation was settled by Pekalska and Duin in [74]:

Definition 2.1 (Conceptual representation). *Given two arbitrary sets A and B, let ϕ be a non negative function, expected to capture the notion of closeness between pair of points in $A \times B$, e.g. a similarity or a dissimilarity measure. A conceptual representation of a point $a \in A$ is a set of similarities/dissimilarities between a and the elements of B expressed as a vector*

$$a \mapsto \left[\phi(a,b_1), \phi(a,b_2), \ldots, \phi(a,b_m)\right] \in \mathbb{R}^m \tag{2.1}$$

B is called representation set.

The function ϕ might be non-metric. This definition is very flexible. In the case $A = B$ the conceptual representation is a standard similarity or dissimilarity measure between pair of objects. Allowing B to be an arbitrary set of prototypes [72], several generalizations, recently applied for classification purposes, can be derived. For example [8] exploits hidden markov model to construct a conceptual space for clustering sequential

data, whereas [55] relies on one class support vector machine to represent and aggregate semantically similar images.

The representation step adopted by J-Linkage can be mapped in this framework setting $A = X$, $B = H \subseteq \Theta$, the pool of sampled structure is regarded as the representation set, and choosing as ϕ the similarity measure defined as

$$\phi(x_i, h_j) = \begin{cases} 1 & \text{if } \mathrm{err}_\mu(x_i, h_j) \leq \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

In practice ϕ assess the fitness to x_i with respect to the structure h_j. Note that the image $\phi(X, H)$ is exactly the J-Linkage consensus/preference matrix presented in Equation (1.5). Columns correspond to consensus sets and rows correspond to preference sets. As noted in [74], this construction can be interpreted in statistical sense as the posterior probabilities of the point x with respect to the m classes determined by the consensus set of the putative structures:

$$\left[\mathrm{Prob}\left(x|\,\mathrm{CS}_{\mu,\epsilon}(h_1)\right), \ldots, \mathrm{Prob}\left(x|\,\mathrm{CS}_{\mu,\epsilon}(h_m)\right)\right] \in \mathbb{R}^m \tag{2.3}$$

Seen in this way, this conceptual representation is linked with the stream of research on higher-order clustering where probability are used to defined higher-order affinity between points.

We introduce a continuous relaxation of the binary preference set exploiting the weighting function adopted in the M-Estimator framework (reported in Appendix A). This can be done by defining the similarity $\phi\colon X \times H \to [0,1]$ as

$$\phi(x_i, h_j) = w_c\left(\frac{\mathrm{err}_\mu(x_i, h_j)}{\tau\sigma_n}\right), \tag{2.4}$$

here w_c can indicate any of the weighting functions whose images are contained in the interval $[0,1]$, namely the Huber, Cauchy, Geman, Welsh and Tukey weighting functions reported in Table A.1. The constant c in the expression of w_c in practice plays the same role of the inlier threshold and can be tuned either using this parameter or, under the assumption of gaussian noise, as $c = \tau\sigma_n$ where σ_n is an estimate of the standard deviation of the residuals and τ is chosen to ensure a predefined level of asymptotic efficiency on the standard normal distribution for the specific M-estimator selected. It is straightforward to embed data points from their ambient space to the conceptual one using the vectorial mapping $\phi_H : X \to [0,1]^m$, simply defined as

$$x \mapsto \left[\phi(x, h_1), \ldots, \phi(x, h_m)\right]. \tag{2.5}$$

Every point x is robustly represented as a m-dimensional *preference vector* in the conceptual space whose entries are the robust weights, giving rise to a soft version of preference set. By the preference analysis perspective, the rationale beyond this construction is that the i-th component of this vector expresses with a soft vote in $[0,1]$ the preference granted by x to the tentative structures h_i. Please note how this parallels the difference between RANSAC and MSAC, if consensus sets are considered.

The next step is to introduce in the unitary cube $[0,1]^m$ a suitable metric that generalizes the Jaccard distance. This is accomplished by the Tanimoto distance [95], defined as

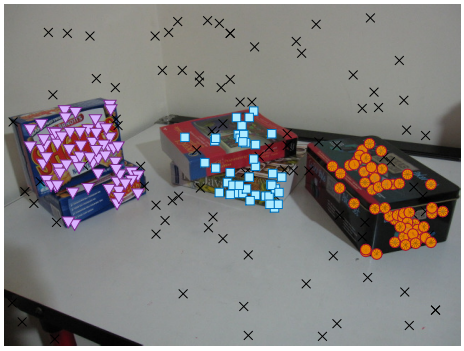
$$d_{\mathscr{T}}(p,q) = 1 - \frac{\langle p,q \rangle}{\|p\|^2 + \|q\|^2 - \langle p,q \rangle} \tag{2.6}$$

for every $p,q \in [0,1]^m$. This distance ranges in $[0,1]$ and equals 0 for preference vectors sharing the same preferences whereas reaches 1 if points have orthogonal preferences, i.e. it does not exist any model in H that can explain both the points p and q. We denote as $\mathscr{T} = ([0,1]^m, d_{\mathscr{T}})$ the metric space endowed with the Tanimoto distance [62]. Please observe that if we confine ourselves to the space $\{0,1\}^m$ the Tanimoto distance coincides with the Jaccard one. The agreement between the preferences of two points in the conceptual space reveals the multiple structures hidden in the data: points sharing the same preferences are likely to belong to the same structures as points matching the same collection of models are likely to belong to the same ground truth model.

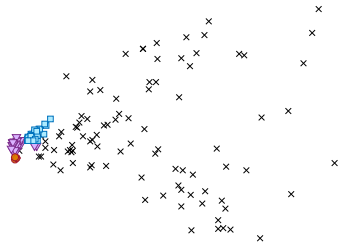
In short, echoing the celebrated “kernel trick", which lifts a non linear problem in an higher dimension space in which it becomes easier, this conceptual representation, shifts the data points from their ambient space to the Tanimoto one, revealing the multiple structures hidden in the data as groups of neighboring points.

Clustering can be thought as the discrete and statistical counterpart of the continuous and geometric problem of finding connected components. With this idea as guide, a geometric analysis of the Tanimoto space can confirm the intuition that points sharing the same preference are grouped together in the conceptual space. To illustrate qualitatively this properties we consider the multi model fitting problem reported in Figure 2.1a, taken from [109]. In this dataset three objects move independently each giving rise to a set of points correspondences in two uncalibrated images: points belonging to the same object are described by a specific fundamental matrix. Outlying correspondences are also present.

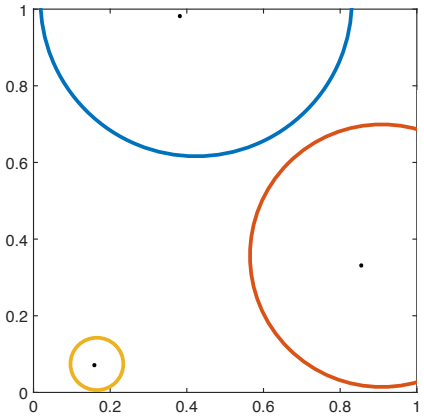
A visualization of the distribution of points in the Tanimoto space can be obtained through multi-dimensional scaling. As can be appreciated from Figure 2.1b, in Tanimoto space points belonging to the same structures are tightly clustered in high density regions. On the contrary outliers, whose votes are underweighted, occupy a region with low density.



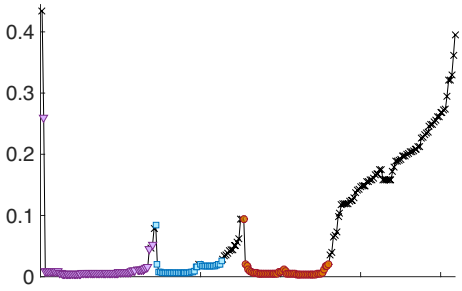
(a) Ground-truth segmentation



(b) Points in $\mathcal{T}$ (with MDS)



(c) Neighbourhoods in $\mathcal{T}$



(d) Reachability plot

Fig. 2.1: Insights on the geometry of Tanimoto space. (a) one frame of the *biscuitbook-box* sequence. Model membership is color coded; black crosses (x) are outliers. (b) conceptual representation of the data in Tanimoto space are projected in the plane using Multi-Dimensional Scaling for visualization purposes. Outliers (x) are recognized as the most separated points. (c) Tanimoto neighbourhoods with the same radius in $[0,1]^2$ have smaller Euclidean diameter if the center lies near the origin. (d) The reachability plot shows the reachability distance of ordered points (model membership is color coded according to the ground truth).

Some insight into the geometrical sparseness of outliers can be reached considering a system of neighbourhoods: Fixed some $\eta \in (0,1)$ and some $p \in \mathcal{T}$ the Tanimoto ball of radius η and center p is denoted by $N_\eta(p)$. As illustrated in Figure 2.1c, the Euclidean diameter of N_η changes accordingly to the position of the center p. In particular this quantity tends to be smaller for points lying near the origin of $\mathcal{T}$, that corresponds to

the region of $\mathscr{T}$ prevalently occupied by outlying points. In fact outliers grant their preferences to very few sampled hypotheses, they have small Euclidean norm and consequently tend to lie near the origin. Hence the probability that two outliers live in the same ball of radius η is significant lower than the probability that two inliers (with higher Euclidean norm) are contained in a ball with the same radius. For this reason outliers can be recognized as the most separated points in $\mathscr{T}$.

2.2 Density analysis

With this perspective as guide, we can examine our conceptual representation through the lens of density based analysis in order to make more explicit these aspects of Tanimoto space. In particular we adopt the multi-scale approach offered by OPTICS (Ordering Points to Identify the Clustering Structure) [3]. OPTICS is a density-based technique which frame the geometry of the data in a reachability plot thanks to the notion of reachability distance. To start with, we tailor the definition of density-connected component proposed in [31] to Tanimoto space:

Definition 2.2. *Given $p, q \in \mathscr{T}$, the cardinality ζ of MSS and $\eta \in (0,1)$*

- *p is said a* core point *if $|N_\eta(p)| > \zeta$;*
- *p is* directly density-reachable *from q with respect to η if $p \in N_\eta(q)$ and q is a core point;*
- *p is* density reachable *from q with respect to η if there is a chain of points $p_1, \ldots, p_\ell$ s.t. $p_1 = p$, $p_\ell = q$ and p_{i+1} is directly density reachable from p_i;*
- *p is* density-connected *to point q with respect to η if there is a point o such that both p and q are density reachable from o.*
- *a* density-connected component *is a maximal set of density-connected points.*

An illustration of these concepts is depicted in Figure 2.2. Density-connectivity is an equivalence relation hence all the points reachable from core points can be factorized into maximal density-connected components yielding the desired segmentation. A crucial advantage of this definition is that it deals directly with outliers which can be recognized as points not connected to any core point. In topological words, outliers can be identified as isolated points, whereas inliers are either internal or boundary points of a density-connected component. A key merit of this notion is that density-connected components may have arbitrary shape. Note that, by definition, a density-connected component must contain at least $\zeta + 1$ points; this is coherent with the fact that at least $\zeta + 1$ points are needed to instantiate a non-trivial model (ζ points always define a model by definition of MSS).

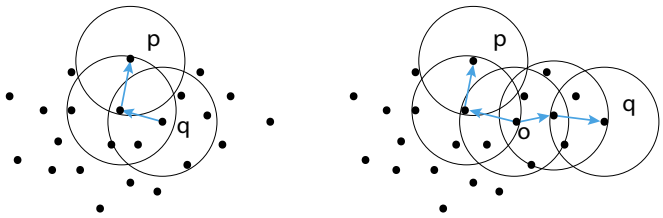


Fig. 2.2: Illustration of reachability. Reachability is not a symmetric relation: in the example on the left p is density reachable from q, but q is not density reachable from p. On the right p and q are density-connected to each other with respect to o.

Definition 2.3. *Given the cardinality ζ of MSS,*

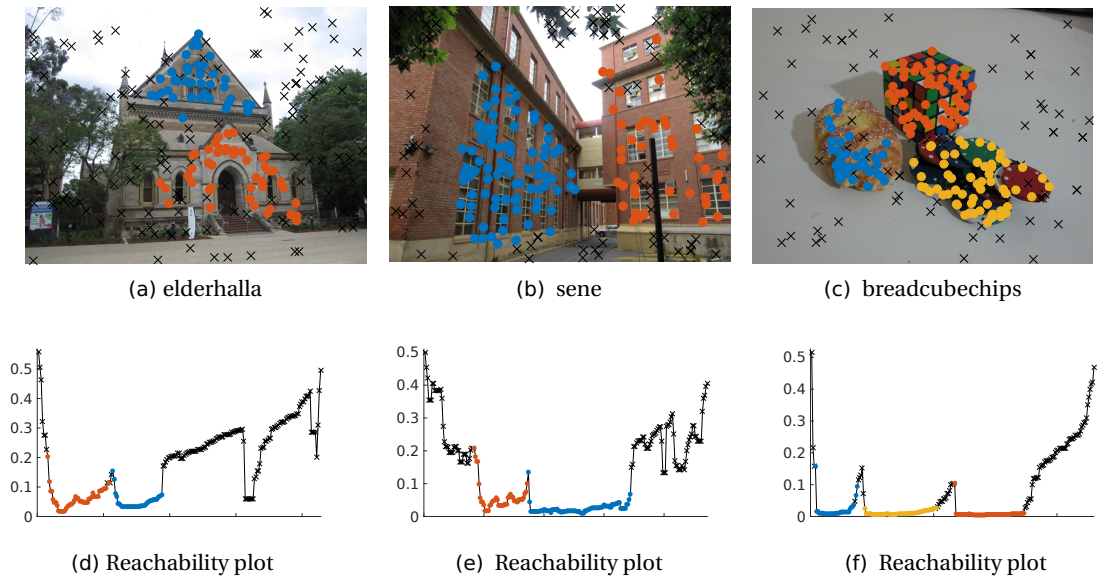
- *if p is a core point, the* core-distance *of p refers to the distance between p and its w-nearest neighbor.*
- *if p is a core point, the* reachability-distance *of a point p with respect to a point q is the maximum between the core distance of p and the distance $d_{\mathcal{T}}(p,q)$.*

After the data have been ordered so that consecutive points have minimum reachability distance, OPTICS produces a special kind of dendrogram, called reachability plot, which consists of the reachability values on the y-axis of all the ordered points on the x-axis. The valleys of this plot represent the density-connected regions: the deeper the valley, the denser the cluster. Figure 2.1d, where the *biscuitbookbox* reachability plot is shown, illustrates this. Outliers have high reachability values, on the contrary genuine clusters appear as low reachability valley and hence are density-connected components in $\mathcal{T}$. Other examples of reachability plots are reported in Figure 2.3.

2.3 Biased Random Sampling in Tanimoto space.

The exploration of the parameter space of models Θ by random sampling straddles all the methods based on either consensus or preferences. Indeed the designing of the pool of tentative models H has a pivotal role as the quality of the embedding ϕ_H is strictly linked to the ability of the sampled space H to adequately represent Θ. In this section we propose a straightforward method to enhance the generation of tentative hypotheses capitalizing the geometric information embodied in the Tanimoto space.

The cues to which models points are likely to belong are somehow latent in the data, for this reason in principle also a simple uniform sampling strategy can capture the



(a) elderhalla (b) sene (c) breadcubechips

(d) Reachability plot (e) Reachability plot (f) Reachability plot

Fig. 2.3: Examples of reachability plots (bottom row). Ground truth segmentation (top row),

hidden multi-modality of a multi-model fitting problem. However this comes at the cost of extensive sampling and increased computational burden, since a large number of trials is required for reaching a reasonable probability of hitting at least a *pure* (i.e., outlier free) MSS per model. The number of required minimal sample sets can be significantly reduced when information about the distribution of the data points is available. This information can be either provided by the user, or can be extracted from the data itself through an auxiliary estimation process. Many strategies have been proposed along this line in order to guide sampling towards promising models both in the case of single-model [22, 23, 71], and in the multiple models scenario [19]. GROUPSAC [71] for example relies on the observation that inliers are often more "similar" to each other, therefore data points are separated into a number of groups that are similar according to some criterion, and intra-group MSS are favorited. A popular choice is the use of Kanazawa sampling [50], neighboring points in the data space are selected with higher probability, thereby reducing the number of hypotheses that have to be generated. However, depending on the application, introducing a local bias in the ambient space of data can be difficult as different structures may obey different spatial distributions of data in the ambient space. Think for example to motion segmentation where different moving objects could have very different shapes or very different sizes due to

perspective effects. Moreover one of the shortcomings of these strategies is that enforcing the spatial proximity requirement can make the estimation prone to degeneracies, and actually specific techniques [120, 121] have been proposed to enforce the opposite condition, i.e. that samples are prevented from incorporating data points that are too close to each other.

In order to overcome this difficulty we propose to sample the hypotheses directly in the conceptual space. This can be easily done in three steps: at first a preliminary uniform sampling of hypotheses is performed, then data are represented in the Tanimoto space according to these putative models, finally a biased sampling in $\mathscr{T}$ is performed. In particular if a point x has already been selected, then a point y such that $x \neq y$ has the following probability of being drawn:

$$\text{Prob}(x|y) = \frac{1}{Z} \exp \frac{d_{\mathscr{T}}\left(\phi_H(x), \phi_H(y)\right)^2}{\beta^2} \quad . \tag{2.7}$$

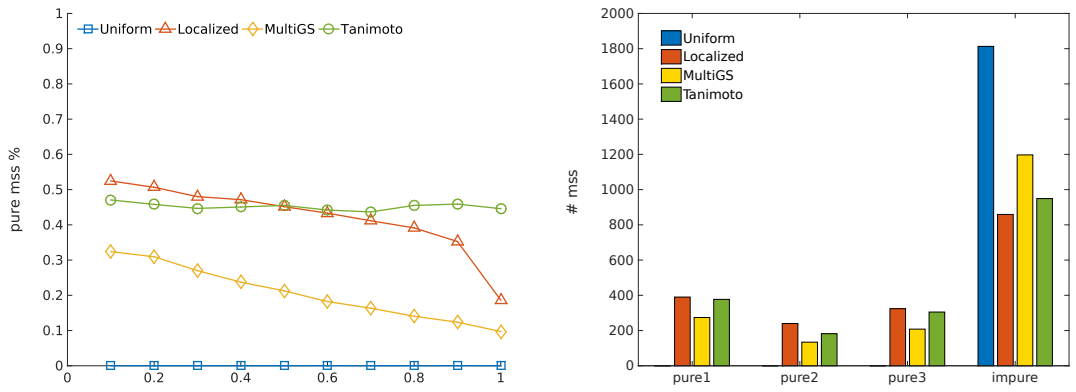
where Z is a normalization constant and β controls the local bias. Tanimoto distances can be then updated on the fly based on the hypotheses already sampled.

We illustrate the effectiveness of this sampling strategy on the *biscuitbookbox* sequence. In Figure 2.4 we compare our biased sampling in Tanimoto space with respect to uniform sampling, localized sampling, and Multi-GS a method proposed in [19], which exploits the intersection kernel proposed in Kernel Fitting. All these methods can be lead back to the conditional sampling scheme presented here, substituting $d_{\mathscr{T}}$ in (2.7) with an appropriate distance function: $d_U \equiv 1$ for uniform sampling, $d_L = \|\cdot\|$ for localized sampling and the intersection kernel d_{GS}. We run these methods with different values of β; in particular we set $\beta = \beta_q$ as the q-th quantile of all these distances, varying $q \in [0.1, 1]$. The experiments demonstrate that our biased sampling provides results comparable with localized sampling for more values of β (Fig. 2.4a) and produces many pure MSS per model (Fig. 2.4b).

This behavior can be motivated in a probabilistic setting considering the lower density of outliers in the Tanimoto space with respect to the inlier distribution. The number m of MSS to be drawn is related to the percentage of outlier and must be large enough so that a certain number (at least) of outlier-free MSS are obtained with a given probability for all the models. As explained in [96], if n_i is the number of inliers of the smaller structure contained in the data, the probability p of drawing a MSS of cardinality ζ composed only of inliers is given by the product

$$p = \text{Prob}(E_1)\,\text{Prob}(E_2|E_1)\cdots\text{Prob}(E_\zeta|E_1, E_2 \ldots E_{\zeta-1}) \tag{2.8}$$

where E_j is the event "extract an inlier at the j-th drawing". It is worth noting that this probability exponentially decrease as ζ increases, therefore, even if in principle



(a) Inliers % vs β

(b) Pure MSS per model (β=$\beta_{0.1}$)

Fig. 2.4: Comparison of guided sampling methods on *biscuitbookbox* sequence. (a) reports the percentage of pure MSS with respect to the local bias parameter β. (b) the number of pure MSS per structures in the data. It is worth observing that uniform sampling struggle in finding genuine MSS.

the space Θ can be explored by instantiating structure on subset with cardinality larger than the minimum as proposed in [80], in practice it is better to keep ζ as low as possible and, if a consensus set can be defined, reestimate the structure via least square or robust technique on its supporting points.

In the case of uniform sampling we have

$$\text{Prob}(E_j|E_1,E_2...E_{j-1}) = \frac{n_i - j + 1}{n - j + 1}. \tag{2.9}$$

In our case we can assume that the first point is sampled with uniform probability, hence $\text{Prob}(E_1) = n_i/n$, while the others are sampled with the probability function (2.7), therefore, after expanding the normalization constant Z, the conditional probability for every $j = 2,\ldots,\zeta$ can be approximated as

$$\text{Prob}(E_j|E_1,E_2...E_{j-1}) = \frac{(i-j+1)\exp\left(-\frac{\alpha^2}{\beta^2}\right)}{(n-n_i-j+1)\exp\left(-\frac{\omega^2}{\beta^2}\right) + (n_i-j+1)\exp\left(-\frac{\alpha^2}{\beta^2}\right)}. \tag{2.10}$$

Assuming that the cardinality of MSS is smaller with respect to the number of inliers, $n_i \gg \zeta$, we have

$$p \approx \delta \left(\frac{\delta \exp\left(-\frac{\alpha^2}{\beta^2}\right)}{(1-\delta)\exp\left(-\frac{\omega^2}{\beta^2}\right) + \delta \exp\left(-\frac{\alpha^2}{\beta^2}\right)} \right)^{\zeta-1}. \tag{2.11}$$

where α represents the average inlier-inlier distance, and ω is the average inlier-outlier distance. Since inlier determine compact cluster with respect to outliers, we have shown that this sampling strategy increases the probability of extracting a pure outlier-free MSS. In order to complete the picture on hypothesis generation, we can observe, following [119], that the probability of drawing at least k outlier-free MSS out of m with a given level of confidence ρ is obtained as:

$$\rho = 1 - \sum_{j=0}^{k-1} \binom{m}{j} p^j (1-p)^{m-j}. \tag{2.12}$$

The better quality of the sampling can be converted into either less samples or into an increase of the quality of the sampled structures (while preserving the same number of MSS).

Said that, drawing of MSS is not more than a computational procedure for approximating the parameter space Θ: any information that can be profitably introduce to make the approximation more accurate can be easily integrated in this step. For example various model verification tests can be adopted in order to enforce desired properties on the sampled structures. A notable example consists in ensuring geometrical non-degeneracy of MSS. A configurations of points is termed *degenerate* with respect to a model if it does not admit a unique solution with respect to that model [102], e.g. collinear triplet of point in case of plane estimation. For instance, in the case of fundamental matrix estimation, a set of correspondence is deemed as degenerate if five or more points lie on the same plane in the scene [25], therefore a specific test aimed to identify MSS where five or more correspondences are related by a homography can be used to prune H from ambiguous structure estimate. Other more general constraints on H depending on the problem at hand can be imposed, for example with respect to fundamental matrix estimation in practice only points in front of the camera are visible, therefore it is possible to enforce chirality constraints via a model checking stage [24].

3

Preference analysis: linkage formulation

In this chapter we investigate how the robust preference trick, introduced in the previous chapter, can be exploited to enhance J-Linkage approach. In particular we tailor the agglomerative linkage clustering to handle continuous representations in the Tanimoto space, so that structures can be recovered as clusters of preferences in the conceptual space. In this setting outliers can be recognized as micro-clusters happened by chance and are filtered out relying on a probabilistic framework. This formulation has the merit to automatically detect the number of models in the data. The only input parameter is the inlier threshold, that can be properly tuned thanks to a scale selection strategy based on consensus clustering.

3.1 Hierarchical clustering

The luxuriant literature on clustering[1] has been organized in different sensible taxonomies according to several criteria; here the distinction that is most relevant to our work is the dichotomy between partitional and hierarchical clustering. In a nutshell, partitional methods directly divides data points in a predetermined number of clusters. On the contrary, hierarchical clustering, rather than defining a static partitioning of the data, aggregates points into a sequence of nested partitions, and exploits the attained hierarchy of subsets to infer the hidden structure of the data. This process can be performed along two directions, namely bottom-up or top-down. In the first case, starting from singleton, a cluster including all the data points is produce by successive merging, vice versa in the latter case the data are sequentially split in several groups. For a data set with n elements, the top-down scheme would start by considering $2^{n-1}-1$

[1] For a short survey on the subject the interest reader is referred to [112]

possible splits of the data , which is computationally expensive, therefore, in practice, bottom-up approaches are usually preferred.

The hierarchy of nested groups is encapsulated in a dendrogram, which depicts the formation of a cluster together with the similarity levels it has been created by merge or split moves. The final segmentation of the data is obtained by cutting the dendrogram at the desired similarity level. Several manners to compute the similarity measure between clusters – called linkage functions – have been proposed in the literature; the most common and popular being:

- Single linkage: where the distance between a pair of clusters is determined by the two closest elements to the different clusters. This procedure tends to generate elongated clusters, which causes the so called chaining effect.
- Complete linkage: In contrast to single linkage, the farthest distance of a pair of objects is used to define inter-cluster distance.
- Average linkage: The distance between two clusters is defined as the average of the distances between all pairs of data points, each of which comes from a different group.

In the next sections we show how hierarchical agglomerative clustering can be tailored to Tanimoto space.

3.2 T-Linkage

In first instance we have to choose a robust estimator in order to frame robustly point preferences by its weighting function. We have seen that J-Linkage uses the step voting function defined as

$$w_{\text{step}}(u) = \begin{cases} 1 & \text{if } |u| \leq 1 \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

with the purpose of emulating its behavior, we pick the Tukey bisquare weighting function because it has a finite minimum rejection points.

$$w_{\text{tukey}}(u) = \begin{cases} (1-u^2)^2 & \text{if } |u| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Thus, provided a pool of m tentative structures H, we can define a conceptual embedding ϕ_H in $[0,1]^m$ using Equation (2.4) by which every point $x \in X$ is depicted as

$$x \mapsto \left[w_{\text{tukey}}\left(\frac{\text{err}_\mu(x,h_1)}{\tau\sigma_n}\right), \ldots, w_{\text{tukey}}\left(\frac{\text{err}_\mu(x,h_m)}{\tau\sigma_n}\right) \right] \in [0,1]^m. \tag{3.3}$$

The tuning constant used to normalize residuals is fixed such that $\tau\sigma_n = \epsilon$, in this way the rejection point of w_{tukey} corresponds exactly to the one used in J-Linkage. The segmentation step follows closely the one of J-Linkage: clustering proceeds in a bottom-up manner. For this purpose we need to define a suitable soft conceptual representation for clusters, extending the preference trick to subset of the data $S \subseteq X$. This is easily done, with a little abuse of notation generalizing ϕ_H from X to the power set $P(X)$, by defining

$$\phi_H(S) = \min_{x \in S} \phi_H(x). \tag{3.4}$$

More precisely a subset S of X is represented as a vector in $[0,1]^m$ whose j-th component expresses the minimum votes granted to h_j among all the points in S, formally:

$$[\phi_H(S)]_j = \min_{x \in S} \phi(x, h_j). \tag{3.5}$$

If we confine ourselves to the binary space $\{0,1\}^m$ we obtain exactly the linkage scheme proposed in J-Linkage. Starting from all singletons, each sweep of the algorithm merges the two clusters with the higher similarity, Tanimoto distances are hence updated and clusters are aggregated until all the distances equals 1. This means that the algorithm will only link together elements whose preference representations are not orthogonal, i.e. as long as there exists in H a structure that received a positive vote from two clusters, they will be merged. This fact explains why we rely on hard descenders and why we adopt the Tukey voting function. Soft descenders indeed are not well suited to fixed cutoff as small preferences, accorded to outlying structures, cause the union of all the points in a unique cluster. On the contrary having set to zero the votes of outliers allows the use of the natural predetermined clustering-cutoff proposed in J-Linkage. Moreover, as a byproduct,

- for each cluster there exists at least one model for which all the points have expressed a positive preference (i.e., a model that fits all the points of the cluster)
- it is not possible that two distinct clusters grant a positive preference to the same model (otherwise they would have been linked).

Each cluster of points defines (at least) one model. If more models fit all the points of a cluster they must be very similar. As a consequence, in principle, it is sufficient to sample every genuine structure once.

The main differences between the conceptual space adopted by T-Linkage and J-Linkage are summarized in Table 3.1.

T-Linkage, as any agglomerative clustering algorithms, fits all the data: bad models must be filtered out *a posteriori* (this aspect will be discussed in the next Section 3.3). Finally, the model for each cluster of points is estimated by least squares fitting.

	T-Linkage	J-Linkage
Voting	Tukey	Hard
Space	$[0,1]^m$	$\{0,1\}^m$
Cluster	$\min \phi_H$	$\bigcap$ PS
Similarity	Tanimoto	Jaccard

Table 3.1: The differences between T-Linkage and J-Linkage

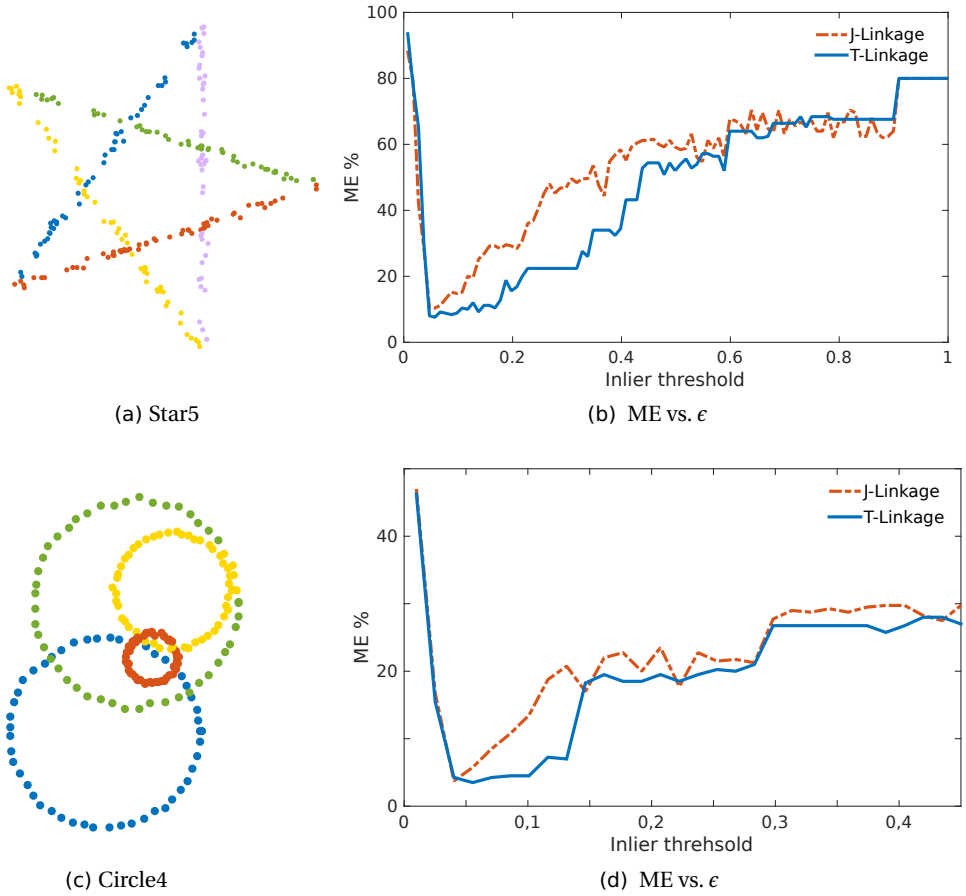
As noted in Section 1, the problem of multiple fitting can be regarded from two alternative points of view usually coexisting: we want to faithfully segment the data and at the same time to obtain an accurate estimate of the underlying models. Each of these two tasks can not be undertaken without the other. T-Linkage is a pure preference based method and concentrates on the first task segmenting the data in the conceptual space and extracting model only at the end via least-squares fitting. However once models have been obtained, optionally it is possible to perform an additional *refinement* step: points are reassigned to their nearest model – if it has distance smaller than ϵ – and finally structures are re-estimated according to this new segmentation. In this way not only the segmentation and the model estimation step can take advantages from each other, but we also gain the benefit of mitigating the greedy behavior of T-Linkage since the final clustering depends less critically on the order in which points were merged together. Under the assumption of gaussian noise, this step can also be viewed as a maximum likelihood estimation, since minimizing the distance of points from the fitted model is equivalent to maximizing their likelihood.

A simple experiment on simulated data with intersecting structures is here conducted in order to characterize the performances of T-linkage with respect to J-Linkage and confirms the benefits of working with continuous values rather than operating with binary preferences. We compare the performances of J-Linkage and T-linkage on fitting lines to the *Star5* data (Figure 3.1c) using the *misclassification error* (ME), defined as follows:

$$\text{ME} = \frac{\text{\# misclassified points}}{\text{\# points}}. \tag{3.6}$$

where a point is *misclassified* when it is assigned to the wrong model, according to the ground-truth.

The results can be appreciated in Figure 3.1 where the corresponding ME is reported as a function of threshold parameters for both J-Linkage and T-linkage on synthetic datasets. The advantages of T-linkage over J-linkage are twofold. On the one hand T-linkage reaches a lower ME, thereby obtaining a more refined clustering. On the other



(a) Star5 (b) ME vs. ϵ

(c) Circle4 (d) ME vs. ϵ

Fig. 3.1: T-Linkage attenuates the sensitivity of ϵ. Left column: segmentations attained by T-Linkage, point membership is color coded. Right column: the ME committed by J-Linkage and T-Linkage on *Star5* (top) and *Circle4* (bottom) datasets is reported as a function of their corresponding inlier threshold parameters. T-Linkage depends less critically on the choice of the inlier threshold.

hand, the threshold parameter integrated in the weighting function is less critical compared to J-Linkage: the error function for T-linkage presents a larger plateau, i.e. a large interval of ϵ the algorithm obtains values near the optimum.

The better result of T-linkage is due to the more expressive representation provided by the continuous conceptual space in proximity of models intersections, since residual information allows to disambiguate more accurately between disputed points. J-Linkage on the contrary has no information to decide to which structure a point in the

intersection of two inlier band has to be assigned. In this perspective we have made a little step toward the solution of intersecting models which caused the poor performances of Multi-RANSAC.

3.3 Dealing with outliers

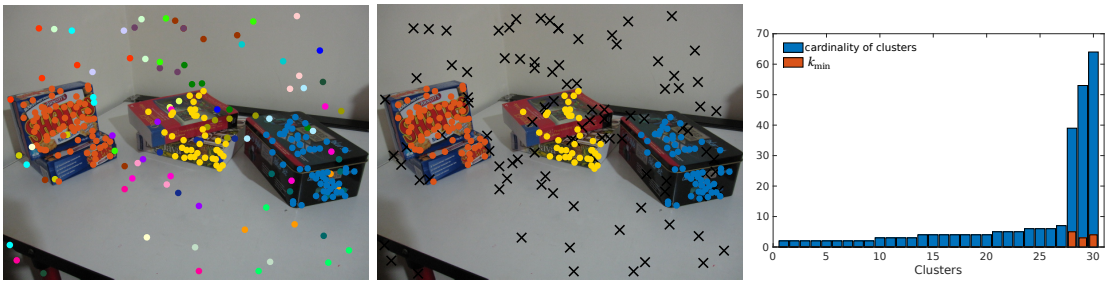
Despite countless efforts spent by the scientific community, there is no universally accepted definition able to capture the elusive nature of outliers. Nevertheless a multitude of approaches have been suggested to characterize outliers; among them we can single out some of the most common assumptions [114]:

- Probability-based : Outliers are a set of small-probability samples with respect to a reference probability distribution.
- Influence-based: Outliers are data that have relatively large influence on the estimated model parameters. The influence of a sample is normally the difference between the model estimated with and without the sample.
- Consensus-based: Outliers are points that are not consistent with the structure inferred from the remainder of the data.

T-linkage is agnostic about the outliers rejection strategy that comes after; depending on the application, different rejection criteria can be adopted. Since the output of T-Linkage is a partition of data points in consensus sets of the estimated structures, a viable solution is to integrate together the approaches based on probability and consensus by analyzing the cardinality of the attained clusters in a probabilistic framework in order to distinguish between good fits from random ones. This solution can be traced back to MINPRAN [90] and PLUNDER [102]. More generally this idea is supported by a stream of research rooted in gestalt theory [30, 67] that provides a formal probabilistic method for testing if a model is likely to happen at random or not. The rationale is the Helmholtz principle [6] which asserts that a strong deviation from a background model is valuable information. In our case the background model is determined by outliers, whereas structures of inliers are regarded as unlikely structure of interest.

First of all we can safely start rejecting all those clusters that have less than $\zeta + 1$ elements since they can be deemed as spurious.

Under the mild assumption that outliers are independently distributed [90], it is possible to easily estimate the probability that a cluster is entirely composed by outliers according to its cardinality and the model it defines. Consequently we retain only the groups with high confidence of being inliers and discard those structures that “happen by chance" and do not reflect an authentic structure in the data.



(a) T-Linkage output (b) Outlier rejection (c) Cluster cardinality

Fig. 3.2: Accuracy: 99.22% False number rate: 0%

In practice, following MINPRAN, at first, the probability p that an outlier belongs to the consensus set of an estimated structure is computed by Monte-Carlo simulation. The value of p can be estimated either in advance for a generic structure, or for every specific model attained by T-Linkage at the end of the clustering. The latter option takes into account the fact that in general models are not all equiprobable and avoids to consider a fixed minimum cardinality. Then the probability that k points belong to the same given model is computed as

$$\alpha(k) = 1 - F(k, n, p), \tag{3.7}$$

where n is the total number of data points, and F is the binomial cumulative distribution function:

$$F(k, n, p) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}. \tag{3.8}$$

For each structure we compute $k_{\min} = \alpha^{-1}(0.01)$ the minimum cardinality necessary to be not considered mere coincidence. If the considered model is supported by less than $k_{\min}$ points is rejected as outlier.

Alternatively, based on the observation that large clusters of outliers are very unlikely, if the number κ of structures is known beforehand, it is sufficient to keep the largest κ clusters as inlier.

In short T-Linkage can be recap as outlined in Algorithm 1.

3.4 Scale estimation by consensus clustering

T-linkage does not have any scale selection strategy and the inlier threshold ϵ has to be manually specified by the user, as in RANSAC. If prior knowledge about the noise in the

Algorithm 1 T-Linkage

Input: the set of data points X, the inlier threshold ϵ
Output: clusters of points belonging to the same structure, and

Conceptual representation step:

Generate by sampling a poll of hypotheses model $H = \{h_1, \ldots, h_m\}$ (uniform sampling and/or guided sampling in $\mathscr{T}$)

Embed each point x in the Tanimoto space, expressing point preferences using the Tukey weighting function as prescribed in Equation (3.3)

Clustering in conceptual space:

Define the preferences of a cluster $S \subset X$ as

$$\phi_H(S) = \min_{x \in S} \phi_H(x),$$

Put each point in its own cluster and compute their conceptual representation
while all the Tanimoto distances are lower than 1 **do**
Find among the current clusters

$$\arg\min_{S_1, S_2} d_{\mathscr{T}}(\phi_H(S_1), \phi_H(S_2))$$

Replace these two clusters with the union of the two original ones and compute the conceptual representation of this new cluster;
end while
Fit structures to cluster.
Outliers rejection.
(Optionally refine estimated structures)

data is available, ϵ can be easily tuned, otherwise the scale turns out to be a sensible free parameter even if the use of a soft weighting function mitigates its criticality.

In this section we develop a method for estimating the scale which results in a novel model selection technique avoiding the classical model selection trade-off of two terms in favor of a single term criterion. In particular, we borrow from the *Consensus Clustering* technique [68] the idea originally outlined in the context of micro-array data, that the stability of the clustering suffices in disambiguating the correct estimate of models. The rationale behind this method is that the "best" partition of the data is the one most stable with respect to input randomization. We translate this principle in the context of geometric fitting, tailoring the Consensus Clustering strategy to T-Linkage.

3.4.1 Choosing the scale in T-Linkage: a model selection problem.

It is important to observe that ϵ plays a crucial role in both the two steps of T-Linkage. At first, in conceptual representation step, the inlier threshold ϵ explicitly defines which points belong to which model (a point belongs to a model if its distance is less than ϵ). If the scale is underestimated the models do not fit all their inliers; on the contrary, if

the scale is overestimated, the models are affected by outliers or pseudo outliers. With respect to the clustering step, points are linked together by T-Linkage until their vectorial representations are orthogonal. Here again, as ϵ controls the orthogonality between these vectors, also the final number of models depends on this parameter.

Given a genuine model, if the true noise variance is known, it is always possible to compute a region containing certain fraction of the inliers. For example, under the typical assumption that the noise for inliers is Gaussian, with zero mean and variance σ^2, the squared point-model errors between an inlier and the uncontaminated model can be represented as chi-square distribution with d degrees of freedom since it is a sum of d squared Gaussian variables, where d is the codimension of the model. For this reason, in order to recover a fraction ρ of inliers, an appropriate threshold ϵ_ρ can be computed as

$$\epsilon_\rho^2 = \chi_d^{-1}(\rho)\,\sigma^2, \tag{3.9}$$

where χ_d^{-1} is the inverse cumulative chi-square distribution, hence it is possible to derive the value of the inlier threshold with a certain level of confidence ρ. Many robust estimators of the noise variance have been proposed, two of the most popular ones are the sample median and the so called MAD (Median Absolute Deviation) which is defined as

$$\text{MAD} = \text{median}_j\,|\,\text{err}_\mu(x_i,\theta) - \text{err}_\mu(x_j,\theta)|. \tag{3.10}$$

Even if both these estimators have 50% breakdown points, they are biased for multiple-mode cases even when the data contains less than 50% outliers.

As a matter of fact, in many real applications selecting the correct scale is a hard problem. In practice many factors hinder scale estimation: the uncertainty of the estimated models has to be taken into account, the presence of high level of contamination due to outliers and multiple structures strains robust estimation and the fact that noise does not always follow gaussian assumption complicates statistical computations. Nevertheless several solutions for automatic scale selection have been proposed. For example this problem is addressed in [20, 82] as regard the case of single model estimation, whereas [32, 66, 107] treat the case of inlier noise estimation for multiple models exploiting elaborated robust statistic. These techniques rely on the idea of simultaneously estimating a structure together with its inlier threshold. Unfortunately, this captivating strategy turns to be impossible of being integrated in T-linkage. As a matter of fact, T-Linkage merges together clusters as long they have a common structure in their preferences. Therefore a single structure for which ϵ have been erroneously over-estimated is sufficient to cause an incorrect aggregation of clusters and to bias the result towards under segmentation. By way of illustration, one can think to the extreme case where all the sampled structures are computed with the correct scale value, but a single inlier

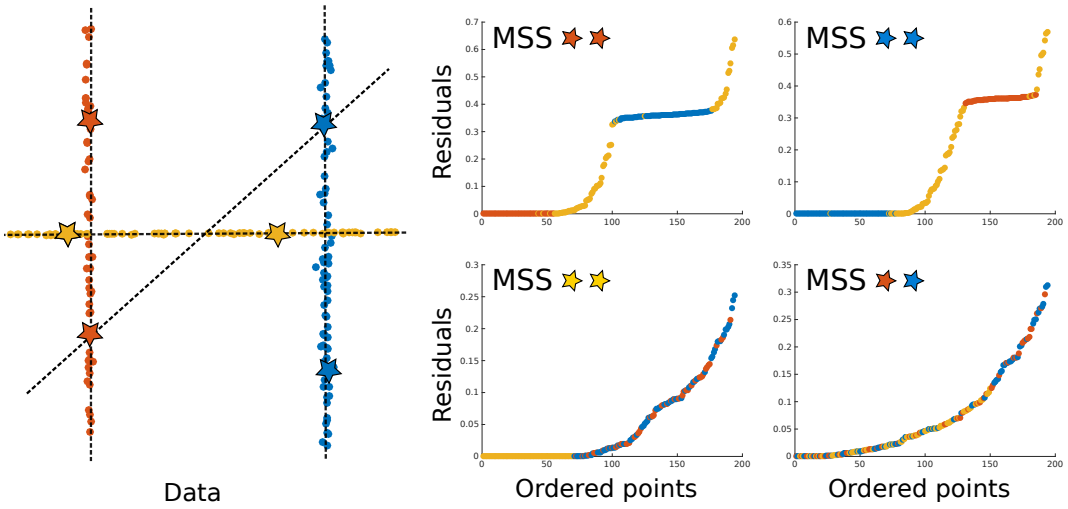


Fig. 3.3: The difficulties inherent to scale estimation for spurious structures. On the left a contrived multi-line fitting example is presented. Data points are sampled, with different level of noise from three ground truth lines (membership to these lines is color coded). Four MSS are drawn, three MSS are pure the fourth is mixed. Analysing the residuals of the corresponding instantiated model (on the left) clearly shows that as regards the pure MSS ordered residuals clearly exhibit the presence of multi-modal population that can be separate by suitable statistical test. On the contrary in the case of the spurious model (bottom-right) residuals do not present any regularity since there are not enough inlier points. As a consequence scale estimation can not produce a reliable result.

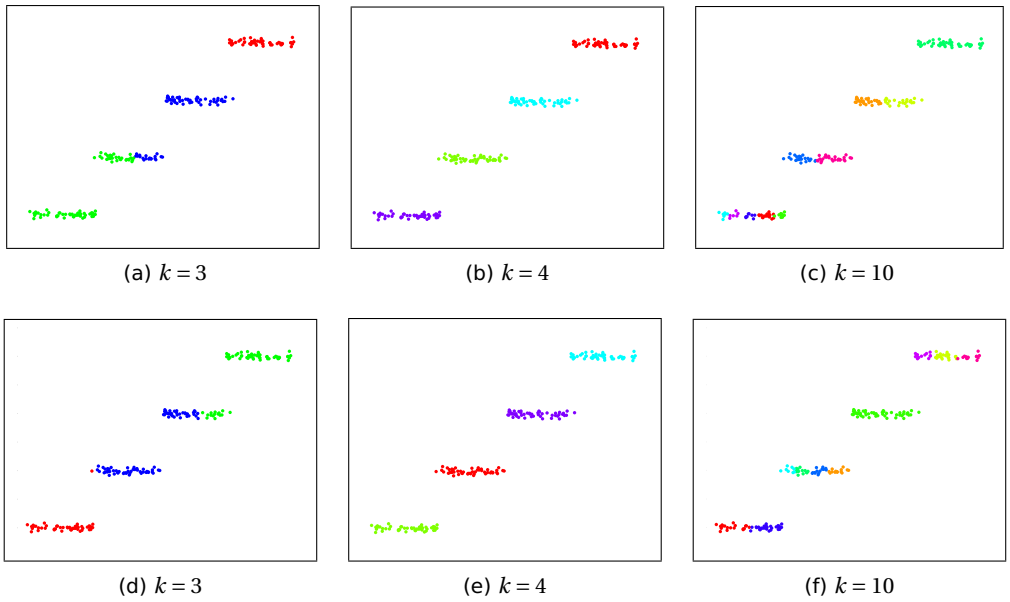
threshold is inaccurately over-estimated in a way that the corresponding consensus set includes all the data point, in this case T-Linkage will return a single cluster.
All the thresholds ought to be estimated accurately in a data dependent fashion. However reasoning about the distribution of inlier residuals is not a viable solution as suggested by Figure 3.3. In first instance all the scale estimators that rely on variants [32, 65] of the MAD, which has a breakdown point of 50%, can not be adopted because they are prone to over-estimation due to the large number of pseudo-outlier in common multi-model fitting scenario. Furthermore, the presence of mixed minimal sample sets thwarts all the approaches with a higher breakdown point such as KOSE [58] and IKOSE [76], which substitute MAD with the k-th ordered absolute residual. In this case the problem is that spurious structures do not have enough supporting points obeying to the statistical assumption made by this kind of estimator. Other approaches, e.g. [5], avoid to estimate the scale using all the data points and exploit

a forward search method [4]: starting from MSS the consensus set is expanded until a statistical test on residual is verified. Also these methods are voted to failure because structures arisen from impure MSS, produce drifting models and, again, over-estimated scales. In short while scale estimator can work reliably for structures close to the ground truth model parameters, the automatic tuning of the inlier threshold of "random" structures is somehow unfeasible. Unfortunately the ideas presented in Section 3.3 can not be used in this context to recognize and discard these spurious structure, since ϵ is required as an input to measure the randomness of a model.

For these reasons we found profitable to tackle the problem by a different perspective. The pivotal observation is that in T-linkage the tuning of ϵ turns to be a typical model selection problem. If ϵ is too small, we are stuck in under-segmentation: multiple similar structures explain the same model in a redundant way. On the contrary, if ϵ is too large, we run into the problem of over-segmentation obtaining fewer structures than necessary that poorly describe the data. We can therefore cast our scale selection problem as a model selection one. The great advantage of this approach is that by tuning the single free parameter ϵ we are able to implicitly balance at the same time between both the complexity of the obtained structures and their fidelity to the data.

Model selection is a thorny pattern recognition problem that appears ubiquitous in multi-model fitting literature (see e.g. [99]). As a matter of fact, following the spirit of Occam's razor, several multi-model fitting methods result in minimizing an appropriate cost function composed by two terms: a modeling error and a penalty term for model complexity. Just to name a few relevant algorithms, this approach is taken in [28, 47, 76, 77, 86] where sophisticated and effective minimization techniques such as SA-RCM [77], ARJMC [76] have been proposed. Several alternatives have been explored for encoding model complexity. PEARL [47] for example, optimizes a global energy function that balances geometric errors and regularity of inlier clusters, also exploiting spatial coherence. In [97], an iterative strategy for estimating the inlier-threshold, the score function, named J-Silhouette, is composed by a looseness term, dealing with fidelity, and a separation one, controlling complexity.

Our starting point is STARSAC [20] in which Choi and Medioni demonstrate that choosing the correct ϵ enforces the stability of the parameter of the solution in the case of a single structure. We extend this result to the multiple structures scenario, reasoning on segmentation rather than on models parameters. The idea of exploiting stability appears in the context of clustering validation. In particular in [68] the authors propose Consensus Clustering, a strategy that succeeds in estimating the number of clusters in the data with a single term model selection criterion based on stability. The next section is devoted to present the Consensus Clustering approach.



(a) $k = 3$ (b) $k = 4$ (c) $k = 10$

(d) $k = 3$ (e) $k = 4$ (f) $k = 10$

Fig. 3.4: Clusters estimation. k-mean is run two times (rows) on subsamples of the same dataset with different values of k (columns). Only for $k = 4$ the attained segmentation is the same. This figure is best viewed in color.

3.4.2 Consensus Clustering

In some cases the thorny problem of correctly tradeoff data fidelity for model complexity (a.k.a. bias-variance dilemma) can be bypassed introducing a different model selection principle based exclusively on the *stability* of models.

The key idea of this approach is that good models should be found among the ones that are stable with respect to small perturbations of the data. This very general principle with the necessary specifications can be applied in many contexts, and can be exploited also in the classical segmentation problem.

For instance, consider the situation illustrated in Figure 3.4. In this case the models to choose are all the possible partitions of points in k disjoint subsets and model selection is employed for choosing the correct value of k. Running k-means several times on subsamples of the same data, with different values of k shows that the resulting clusterings are stable only when k expresses the nature of the data, otherwise they manifest lack of stability. This simple example sustains the intuition that the more stable models represent valid structures in the data.

In [68] the authors develop this idea and present the Consensus Clustering approach to determine the correct number of clusters by maximizing the *consensus*, i.e., the agreement of clustering after perturbation of the data.

More in detail, the Consensus Clustering approach consists in assuming a clustering algorithm, for example k-means, and a resampling scheme (e.g. bootstrapping) in order to perturb the data. Then for each possible cluster number $k = 2,3,\dots,k_{\max}$ the data are subsampled several times and processed by the clustering algorithm. The corresponding results are described for each k by means of a *consensus matrix* M_k which is intended to capture the mutual consensus of attained clusters. The consensus matrix M_k is defined as follows: the element $(M_k)_{ij}$ stores the number of times points i and j are assigned to the same cluster divided by the total number of times both items are selected by the resampling scheme. In other words, the consensus matrix records the proportion of clustering runs in which the two points i, j have been clustered together. For this reason $(M_k)_{ij} \in [0,1]$ and perfect consensus corresponds to a clean consensus matrix with all the entries equal to either 0 or 1[2], whereas a deviation from this case should be explained with lack of stability of the estimated clusters. Exploiting this observation, the k that yields the cleanest consensus matrices according to an ad hoc measure is selected as the optimal estimate of number of model.

3.4.3 T-Linkage with Consensus Clustering

In this section we shall concentrate on a method for automatically fitting multiple models tailoring Consensus Clustering to T-Linkage algorithm without having a priori knowledge of the scale ϵ, thereby conceiving a single term model selection criterion based on consensus stability.

In the case of T-Linkage we do not have to select the number of clusters (that is automatically determined by T-Linkage clustering) but we shall concentrate on the scale ϵ which, as explained in Section 3.4.1, is a sensitive input parameter that implicitly tunes the balance between the complexity of the obtained clusters and their fidelity to the data. If ϵ is too small, we are stuck in under-segmentation: multiple similar structures explain the same model in a redundant way. On the contrary, if ϵ is too large, we run into the problem of over-segmentation obtaining fewer structures than necessary that poorly describe the data.

The outline of our approach is sketched in Figure 3.5. The estimation of ϵ is iteratively laid out as follows. At first the interval search $[\epsilon_L, \epsilon_R]$ has to be defined, ensuring that the correct ϵ belongs to the interval. For this reason a sound choice of ϵ_L is a small

[2] If the data points were arranged so that points belonging to the same model are adjacent to each other, perfect consensus would translate into a block-diagonal matrix

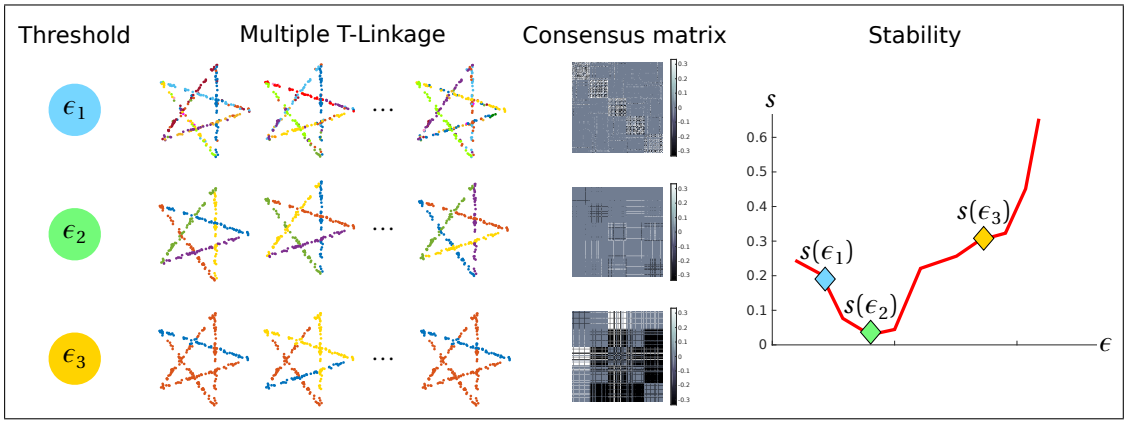


Fig. 3.5: The proposed method in a nutshell. Different ϵ values are used for running multiple times T-Linkage on the perturbed *Star5* dataset. In correspondence of ϵ_1 (which is lower than the ground truth inlier threshold) T-Linkage over-segments the data producing unstable results, when the threshold is ϵ_2 a reliable and stable clustering composed by 5 structures is returned, and finally using the over-estimation of the threshold ϵ_3, data are under-segmented in different ways. The corresponding consensus matrices measure the mutual consensus between the attained segmentations and define the stability index. The most stable clustering, corresponding to ϵ_2, is selected. (Best viewed in color)

scale value that surely over-segments the data, whereas ϵ_R has to give rise to under-segmentation (for example it can be estimated fitting a single model to all the data point and taking the maximum of their residuals). For each ϵ value belonging to the interval search, T-Linkage is run t times $t = 1,\ldots,t_{\max}$ on the data properly perturbed.

Rather than bootstrapping in advance the raw data as in [68], we perturb their representation in the conceptual space inside T-Linkage by bootstrapping the generated hypothesis. After the data have been processed we obtain $t_{\max}$ clustering outputs for each ϵ value. The intuition is that, at the correct scale, there will be consistency between the partitions produced by T-linkage. For each scale the consistency of the partitions is hence tabulated via the consensus clustering matrix M_ϵ introduced in Section 3.4.2.

Now we measure the consensus stability of each matrix boiling down each M_ϵ to a single consensus stability value s per scale. If we were to plot a histogram of the entries of $(M_\epsilon)_{ij}$, perfect consensus would translate into two bins centered at 0 and 1 and, in general, a histogram skewed toward 0 and 1 indicates good clustering. With this idea in mind, consider the following change of variable:

$$F(x) = \begin{cases} x & \text{if } x < 0.5 \\ x-1 & \text{if } x \geq 0.5. \end{cases} \tag{3.11}$$

F redistributes the entries of M_ϵ from the $[0,1]$ range to the interval $[-0.5, 0.5]$. The effect is to rearrange the histogram symmetrically around the origin. In this way stable entries are concentrated around 0 whereas unstable ones are accumulated at the tails of the histogram. For this reason, measuring how far the entries of $F(M_\epsilon)$ are spread out accounts for the consensus stability of a given scale ϵ. For this purpose we propose to employ the variance[3] of the vectorized upper triangular part of $F(M_\epsilon)$ and to define a *consensus stability index* as

$$s(\epsilon) = \text{var}\left(\text{vech}\left(F(M_\epsilon)\right)\right), \tag{3.12}$$

where vech returns the vectorization of the upper triangular matrix it receives in input. Then, assuming to deal with authentic multiple structures, the scale is selected among the ϵ values that segment the data in at least two clusters. Within these ϵ we retain as correct the smallest one obtaining the lower score of s:

$$\epsilon^* = \min\left(\arg\min_{\epsilon:\, \#\,\text{cluster}>1} s(\epsilon)\right). \tag{3.13}$$

The most stable solution (the one obtained with ϵ^*) is then returned.

The procedure can be summarized in Algorithm 2.

With respect to the computational complexity of this method, if c is the execution time of T-linkage, k_1 the threshold values tested and k_2 the number of bootstrapping trials, the total execution time of this method is $k_1 k_2 c$ to which the time needed for computing the consensus matrices has to be added. Even if the number of bootstrap iterations is small ($k_2 = 4$ in our experiments suffices in providing good results), there is space for improvement for example by replacing exhaustive search on the interval $[\epsilon_L, \epsilon_R]$ with a suitable (direct) minimization strategy reducing the number of scale values that are evaluated.

3.5 Experiments

This section is devoted to evaluating the proposed method on both simulated and real data, proving that consensus stability s can be exploited as a single term model selection criterion for automatically fit multiple structures.

[3] We also tested the entropy and other dispersion indices with comparable results.

Algorithm 2 TLCC

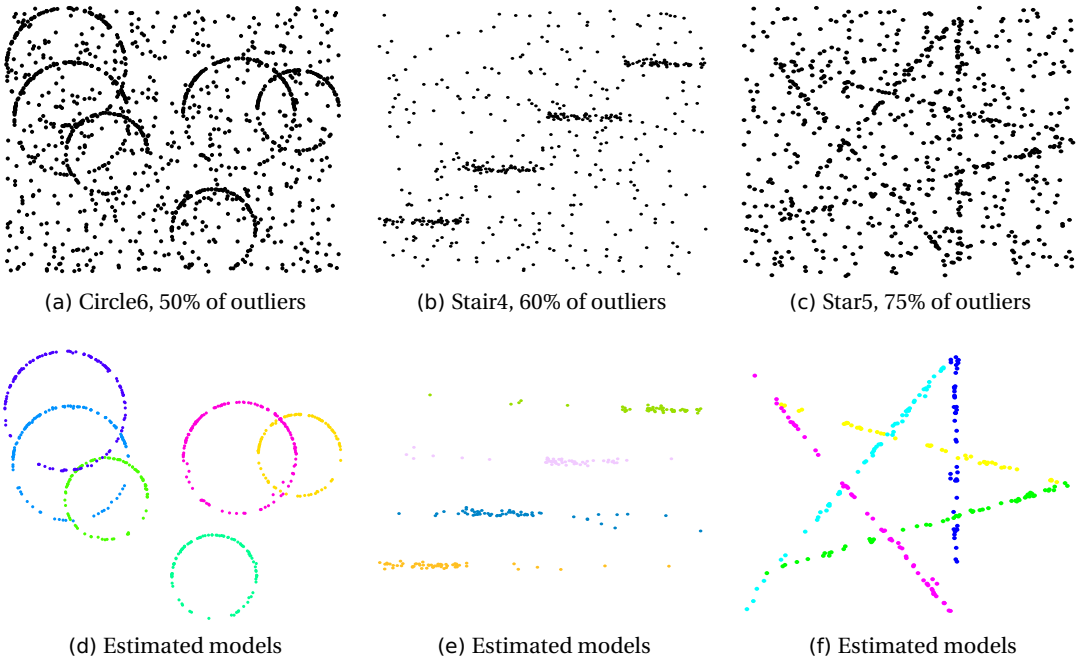
Require: the set of data points X;
an interval search $[\epsilon_L, \epsilon_H]$;
Ensure: scale ϵ^*;
clusters of point belonging to the same model.

Generate hypotheses H;
for $\epsilon \in [\epsilon_L, \epsilon_H]$ **do**
  for $t = 1, \ldots, t_{\max}$ **do**
    $\tilde{H}$ = Bootstrapping(H);
  end for
  C_t = T-Linkage($X, \epsilon, \tilde{H}$);
  Probabilistic outlier rejction
  M_ϵ= Consensus Matrix($C_1, \ldots, C_{t_{\max}}$);
  Compute $s(\epsilon)$;
end for
ϵ^* = min(arg min$_{\epsilon:\ \#\,\text{cluster}>1}\ s$);
C^* = T-Linkage(X, ϵ^*, H);

Some synthetic experiments are carried on in order to qualitatively asses the proposed approach. In particular, as shown in Figure 3.6, we address the problem of fitting circles (Figure 3.6a) and lines (Figures 3.6b, 3.6c) to noisy data contaminated by gross outlier. Since the number of structures is unknown – actually it is determined by the parameter ϵ we want to estimate – we do not rely on this information for rejecting outliers. Therefore we employ the outlier rejection strategy described in 3.3 that discards the structures happened by chance. It is worth to notice that this criterion works properly filtering out bad models with different percentage of outliers.

We validate our approach – henceforth referred to as TLCC (T-Linkage and Consensus Clustering)– on some real datasets. We test our method on image pairs correspondences taken from the AdelaideRMF dataset [109] on both two view motion segmentation and plane experiments. The sequences in this dataset consist of matching points in two uncalibrated images with gross outliers. In the case of plane segmentation the (static) scene contains several planes, each giving rise to a set of point correspondences described by a specific homography. The aim is to segment different planes by fitting homography matrix to subsets of corresponding points. In the second case (*motion segmentation*) the setup is similar, but the scene is not static, i.e., it contains several objects moving independently each giving rise to a set of point correspondences described by a specific fundamental matrix. The aim is to segment the different motions by fitting fundamental matrices to subsets of corresponding points.

First we compare TLCC with T-linkage*, where T-linkage* has an "oracle" that guesses always the optimal scale according to the ME, in the interval search:



(a) Circle6, 50% of outliers (b) Stair4, 60% of outliers (c) Star5, 75% of outliers

(d) Estimated models (e) Estimated models (f) Estimated models

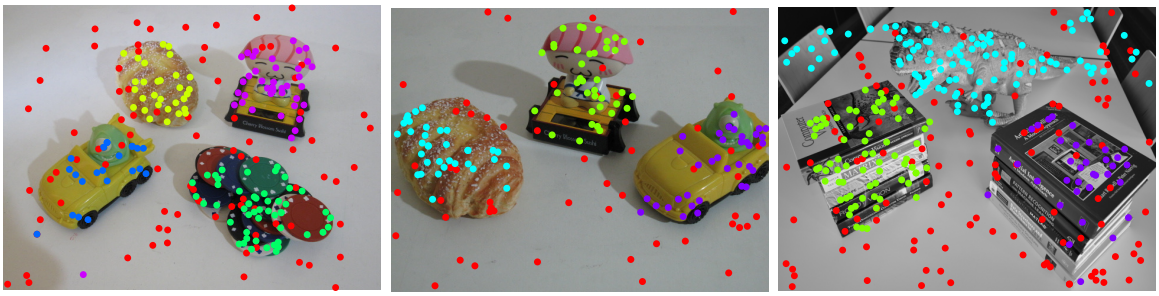
Fig. 3.6: *Synthetic examples*: rough data are reported in the first row, extracted models are shown in the second one. Membership is color coded.

$$\epsilon_{\text{opt}} = \arg \min_{\epsilon \in [\epsilon_L, \epsilon_R]} \text{ME}(\epsilon), \tag{3.14}$$

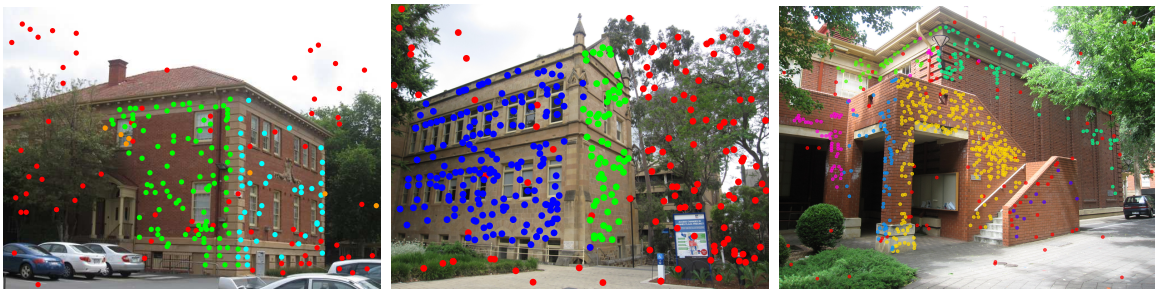
in other words ϵ_{opt} is the global minimum of ME. For each experiments we compare the ME(ϵ^*) achieved by the scale ϵ^* estimated by TLCC with the ME(ϵ_{opt}) of the *optimal* scale.

Using the data reported in [77] we are able to compare indirectly TLCC with other state of the art algorithms inspired to the classical two term model selection approach. For fair comparison with [77], where the parameters of each sequence are individually tuned and the best outcomes out of several trials have been recorded, we adjust the localized sampling parameters per each sequence separately.

As regards fundamental matrix fitting, according to Table 3.2, TLCC succeeds in estimating the optimal ϵ in six cases (marked in bold) and misses the global optimum in two cases, for which we plot the ME and the stability index in Figures 3.8a and 3.8b. It can be appreciated that the profile of the ME is fairly flat near the optimum, and that the minimum of the stability index is fairly close to the optimum of ME anyway.



(a) breadcartoychips (b) breadtoycar (c) dinobooks



(d) ladysymon (e) oldclassicswing (f) johnsonb

Fig. 3.7: Sample results of TLCC in two-view motion segmentation (top row) and planar segmentation (down row). Point membership is color coded, red dots are points rejected as outliers.

Our conjecture for such a behavior is that the models have mutual intersections (or close to), and that ME does not measure properly the quality of a clustering. For instance imagine a point P that lies in the intersection of two models, say A and B, and suppose that, according to the ground truth, it is assigned to A. A clustering that assigns P to B is penalized by ME, whereas it should not. A similar argument applies to points that lie close to two models (without belonging exactly to the intersection): the penalty for assigning a point to the wrong model should be attenuated in such close-to-ambiguous situation.

In all but three cases TLCC achieves the best result, and, if the mean ME is considered, it is the best algorithm. These cases are reported in Figures 3.7a, 3.7b 3.7c where it can be appreciated that the resulting segmentation is reasonable anyway.

On plane segmentation experiments, in five cases (marked in bold) the proposed method estimates an optimal scale according to ME.

For the *johnsonb* image pairs the attained segmentation by TLCC is slightly less accurate than the optimal one, however from Fig. 3.8c, where the ME and the stability

Table 3.2: ME (%) for two-view *motion segmentation.*

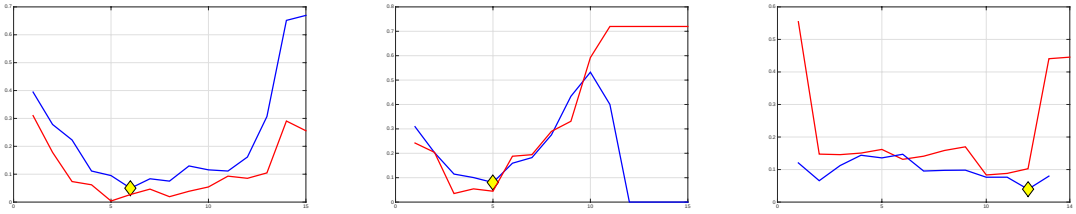
	Two term model selection					Stability	
	PEARL	QP-MF	FLOSS	ARJMC	SA-RCM	TLCC	T-Link*
biscuitbookbox	4.25	9.27	8.88	8.49	7.04	**2.71**	0.39
breadcartoychips	5.91	10.55	11.81	10.97	**4.81**	5.19	5.19
breadcubechips	4.78	9.13	10.00	7.83	7.85	**2.17**	2.17
breadtoycar	6.63	11.45	10.84	9.64	**3.82**	4.27	4.27
carchipscube	11.82	7.58	11.52	11.82	11.75	**1.22**	1.22
cubebreadtoychips	4.89	9.79	11.47	6.42	5.93	**4.46**	3.50
dinobooks	14.72	19.44	17.64	18.61	**8.03**	13.86	13.86
toycubecar	9.5	12.5	11.25	15.5	7.32	**3.03**	3.03
Mean	7.81	11.21	11.68	11.16	7.07	**4.62**	

Table 3.3: ME (%) comparison for *plane segmentation.*

	Two term model selection					Stability	
	PEARL	QP-MF	FLOSS	ARJMC	SA-RCM	TLCC	T-Link*
johnsona	4.02	18.5	4.16	6.88	5.9	**3.12**	3.12
johnsonb	18.18	24.65	18.18	21.49	17.95	**8.83**	8.81
ladysymon	**5.49**	18.14	5.91	5.91	7.17	6.17	6.17
neem	5.39	31.95	5.39	8.81	5.81	**4.78**	4.78
oldclassicswing	**1.58**	13.72	1.85	1.85	2.11	**1.65**	1.65
sene	0.80	14	0.80	0.80	0.80	**0.42**	0.42
Mean	5.91	20.16	6.05	7.62	6.62	**4.08**	

index are shown, it can be appreciated that the value achieved by TLCC corresponds to a plateau of ME. The segmentation produced by TLCC is presented in Figure 3.7f. Notice that the actual global optimum of ME can be conditioned by arbitrary tie-breaking of disputed points between models.

Table 3.3 compares TLCC with state of the art methods (results for all the methods but TLCC are taken from [77]). Our method achieves in all cases, but one, the best ME



(a) biscuitbookbox (b) cubebreadtoychips (c) johnsonb

Fig. 3.8: Stability index s (blue) and ME (red) as a function of the scale ϵ parameter for some image pairs of the *motion segmentation* (3.8a, 3.8b) and *plane segmentation* experiments (3.8c). The estimated scale is marked with a diamond on the s curve.

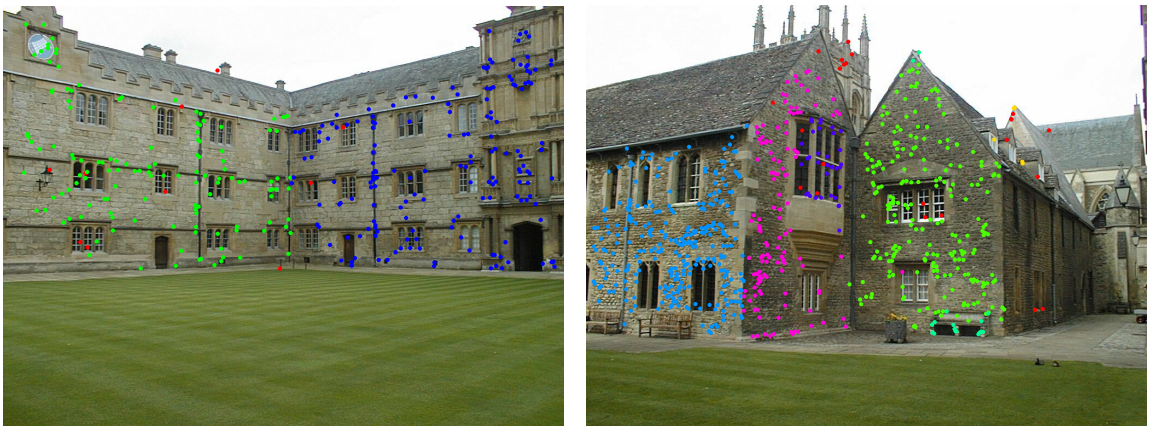
and a reasonable segmentation and it scores first on the average. In Figure 3.9 we also validate TLCC on some plane segmentation experiments taken from the VGG dataset[4].

In summary, results show that TLCC, and *a fortiori* T-Linkage, place in the same range as the state of the art competing algorithm adopting a classical two-term model selection strategy, with a free balancing parameter. Experiments show that this method succeeds in estimating the scale parameter of T-linkage and provides evidence that stability has a minimum in the “right" spot, ideally the same spot where the misclassification error (ME) achieves its minimum.

3.6 Final remarks

In this chapter we have built on the preference trick implemented by J-Linkage, relaxing the notion of preference set in the Tanimoto space. Cluster analysis is hence performed through agglomerative clustering. This segmentation techniques manifest two key advantages. First there is no need to specify the number of sought structures in advance. Second, outliers emerge as small group of points that can be pruned in a probabilistic framework where the reliability of a structure is measured in term of its randomness. The inlier threshold is the only input parameter required by this strategy. The tuning of the scale is reduced, to less extent, by the use of soft representation. Moreover a model selection criterion, which fully takes advantage of the linkage output, has been proposed for automatic scale selection. The experimental results are compelling as shown in the comparison with other methods and demonstrate that by expressing the prefer-

[4] available online at `http://www.robots.ox.ac.uk/~vgg/data/data-mview.html`



(a) Merton College II (b) Merton College III

Fig. 3.9: Qualitative results of TLCC on stereo images from VGG, Oxford (point membership is color coded, red dots are points rejected as outliers)

ences of a point integrating more specific information on residuals, we obtain a more significant representation easing the task of multi model fitting.

4

Preference analysis: spectral formulation

The preference trick coupled with random sampling is a very flexible mechanism that can be applied a wide varieties of scenarios requiring few assumptions on the desired structures. It is sufficient to have at disposal an error function aimed at measuring residuals and then, by means of M-estimator, the structure recovery problem is shifted in the conceptual space where it can be addressed using cluster analysis. In the previous chapter we have seen how hierarchical clustering can be used to recover the latent structures hidden in the data, here we concentrate on partitional clustering based on spectral analysis. In particular we borrow some key concepts from state-of-the-art subspace clustering techniques.

4.1 Subspace estimation: low rank & sparsity

The problem of subspace estimation is a particular instance of multi-model fitting that has a relevant place in Computer Vision, since many applications – from image segmentation to motion segmentation, or temporal video segmentation – can be reduced to fitting a mixture of subspaces to high dimensional data.

A variety of approaches have been proposed. Methods based on matrix factorization were among the first to be introduced; they can be thought as natural extension of Principal Component Analysis (PCA) in which the data matrix X is decomposed as $X = LY$, where L is a low rank matrix and Y is block diagonal and encodes the membership of points to the same subspace. The algorithms of Boult and Brown [12], Costeira and Kanade [27] and Gear [38] belong to this category. Many other techniques have been proposed, for example Local Subspace Affinity [113] is an algebraic method that uses local information around points in order to fit local subspaces and to cluster points making use of a pairwise similarities computed using angles between the local subspaces. Agglomerative Lossy Compression [83] is a bottom up clustering algorithm that

aims at segmenting the data minimizing a coding length needed to fit the points with a mixture of degenerate Gaussians up to a given distortion.

In the last years an emerging stream of research [63, 89] has been concentrating on the use of sparse representation and low rank constraints for segmenting high dimensional vectorial data. The notion of sparsity [14] is straightforward: a vector is sparse if it can be exactly or approximately represented as a linear combination of only a few vectors selected from a predetermined dictionary. This property is encoded by the ℓ_0 "norm" $\|v\|_0 = |\{k\colon (v)_k \neq 0\}|$: a vector admits a k-sparse representation with respect to a dictionary D if it can be written as Dc and $\|c\|_0 = k$. While the reconstruction of a signal from its sparse representation is a simple linear transform, the inverse problem

$$\arg\min_c \|c\|_0 \text{ such that } Dc = v, \tag{4.1}$$

is a non-linear optimization that, in general, is intractable. This fact has motivated the flourishing of many methods in the compressed sensing literature based on the convex relaxation of the ℓ_0-norm: the ℓ_1-norm. ℓ_1-norm, defined as the sum of the absolute values of the entries $\|v\|_1 = \sum_k |(v)_k|$, serves to replace the problem in Equation (4.1) with the following tractable optimization objective:

$$\arg\min_c \|c\|_1 \text{ such that } Dc = v. \tag{4.2}$$

At a high level, the effectiveness of sparsity-oriented approaches can be explained viewing this property as a useful way of constraining the complexity of a vector representation, which can be very generally justified by Occam's razor. Sparse Subspace Clustering (SSC [89]) exploits this principle to derive a segmentation of high dimensional data. The main idea of SCC is to take advantage of the "self-expressiveness" of the input: every points can be expressed as a linear combination of few other points lying in the same subspace. A sparse ℓ_1 optimization program captures this property by defining a collection of vectors of coefficients c_i using as a dictionary the data itself:

$$\arg\min_{c_i \in \mathbb{R}^N} \|c_i\|_1 \quad \text{subject to} \quad x_i = X c_i \quad \text{and} \quad (c_i)_i = 0. \tag{4.3}$$

The constraint $(c_i)_i = 0$ removes the trivial solution that decomposes a point x_i as a linear combination of itself. In this way the sparsest representation of x_i would only select vectors from the subspace in which x_i happens to lie. In matrix notation the problem in Equation (4.3) can be rewritten as

$$\min_C \|C\|_1 \quad \text{such that} \quad X = XC^\top, \operatorname{diag}(C) = 0. \tag{4.4}$$

In case of data contaminated by noise and outlier, instead of expressing a data point as an exact linear combination of other points, it is convenient to introduce a penalty term:

$$\min_C \|C\|_1 + \lambda \|E\|_{2,1} \quad \text{such that} \quad X = XC^\top + E, \operatorname{diag}(C) = 0. \tag{4.5}$$

The $\ell_{1,2}$-norm is defined as $\|E\|_{2,1} = \sum_j \sqrt{\sum_i |E_{i,j}|^2}$. The underlying assumption is that a data point can be written as $x_i = Xc_i + e_i$ where e_i is a sparse vector that models gross outlying entries of x_i.

A related approach, termed Low Rank Representation (LRR) [63], derives a similar convex optimization problem

$$\min_C \|C\|_* + \lambda \|E\|_1 \quad \text{such that} \quad X = DC^\top + E \tag{4.6}$$

where D is the dictionary matrix, either constructed in advance or equal to X, and $\|C\|_* = \sum \sigma_i(C)$ is the nuclear norm that equals to the sum of all the singular values $\sigma_i(C)$ of C.

Both SSC and LRR then use the optimal C to define an affinity matrix. It is quite natural to define a similarity measure between points as $S = |C| + |C|^\top$, because non-zero elements of c_i correspond to points from the same subspace of x_i. This similarity matrix is finally used to feed spectral clustering and a partition of the points is obtained.

The main difference between the two methods is that SSC minimizes the ℓ_1-norm of the representation matrix to induce sparsity while LRR tries to minimize nuclear norm to promote a low-rank structure. Both method however relies on the same idea: taking advantage of the intrinsic redundancy of multi-model data to embed the data point in a discrete metric space to facilitate the segmentation task. Interestingly this *first-represent-then-clusterize* approach is evocative of the preference trick philosophy. If T-Linkage is taken in comparison, the representation matrix C corresponds to the preference matrix, whereas the similarity S plays the same role of the Tanimoto distances. Moreover, as sparsity is concerned, outliers can be recognized in practice as sparse rows of the preference matrix, since the number of sampled structured supporting outliers is considerable smaller than the number of structure supporting an inlier (typically an outlier is explained only by the structures it has happen to generate by the MSS it belongs to). There are also some deep differences, though. The conceptual representation proposed in T-Linkage is not limited to vectorial (and affine) subspace. This comes at the cost of choosing a correct inlier threshold – a parameter that has geometrical meaning but is highly data depended– and of the effectiveness of the sampling scheme adopted to generate hypotheses. On the other hand, SSC and LRR depends on the regularization coefficient λ to handle outlier and noisy data. As the segmentation

is concerned, spectral clustering requires to know in advance the number of models, whereas the greedy linkage strategy, adopted by T-Linkage, can automatically estimate this parameter.

The connection with subspace clustering can be extended to general purpose higher order clustering methods [41, 49] that rely on an implicitly preference based approach paired with low rank constraints. In this line of work the preference matrix is seen as a flattened tensor that encapsulates the probability of ζ-tuple of points to be clustered together. This multi-way order information is properly reduced to a pairwise similarity that, as happen in LRR and SSC, is processed by spectral clustering. In particular Sparse Grassmann Clustering (SGC) [49] exploits the low rank nature of the multiple model fitting problem approximating the multi-way similarity tensor as the Gramian matrix defined by the inner product of points in the preference matrix. It is worth noting that also the Tanimoto distance is defined in terms of inner product of rows of the preference matrix; the normalization factor, however, differs in order to generalize the Jaccard distance. At this point, following the spirit of spectral clustering –which works only with a few eigenvectors of the similarity matrix– the Gramian is projected on its best low rank approximation in a least square sense. This approximation is obtained thanks to Grouse [7] an optimization algorithm that operates on the Grassmann manifold – i.e. the variety of all the subspace of given dimension of a projective space – in order to produce a low rank approximation in the ℓ_2 sense of the input matrix that, at the end, is segmented using k-means.

In summary, starting from the analysis of multiple subspaces estimation throughout higher order clustering, three main recurring themes have emerged: spectral clustering, sparsity and low rank requirements. In the following sections we analyze how these ingredients can be tailored to our general multi-model fitting problem. We start to pave the way to this result concentrating on spectral clustering.

4.2 Spectral clustering

The wide landscape of clustering algorithms can be broadly categorized into hierarchical and partitional methods. In the latter category *spectral clustering* is among the most popular techniques thanks to its ability of dealing with clusters of arbitrary shape (as opposed to k-means, where the clusters are assumed to lie in disjoint convex sets) and its simplicity: indeed it results in a standard and simple-to-solve linear algebra problem, that avoids local minima and initialization issues.

In concrete, spectral clustering can be thought as a way to address the discrete graph-cuts problem in the language of linear algebra: provided a symmetric similarity

matrix $S_{ij} \geq 0$, viewed as the weighted adjacency matrix of a graph, the aim of graph-cut is to create a segmentation of the data into several groups such that points in the same group have high similarity values and different segments are dissimilar to each other (low similarity). In order to avoid trivial solution, this task is pursued by enforcing limits on the number of desired groups as well as by constraining the relative size of the segments. According to the constraints imposed, different fashions of graph partitioning problem can be derived: namely mincut, RatioCut and normalized Ncut. The balance requirements - encoded in the notions of *cut* – turns graph-cut in an NP-hard problem, hence spectral clustering is aimed at solving a relaxed version of graph-cut.

In particular if A and B are disjoint subset of the vertices, the quantity $c(A,B)$ is called *cut* and measures the weighted connectivity of A with the subgraph B:

$$c(A,B) = \sum_{i \in A, j \in B} S_{ij}. \tag{4.7}$$

For a given number κ of desired segments, the *mincut* problem consists in finding a partition of vertices that minimizes

$$\text{cut}(C_1, \ldots, C_\kappa) = \frac{1}{2} \sum_{i=1}^{\kappa} c(C_i, \bar{C}_i). \tag{4.8}$$

The notation $\bar{C}_i$ denotes the complement of the set C_i and the factor $1/2$ prevents from counting each edge twice. Even if mincut problem can be solved in polynomial time through Max Network Flow, in practice it often does not lead to satisfactory partitions since, in many cases, it produces unbalanced partitions where single vertices are disconnected from the rest of the graph. Two alternatives have been proposed to circumvent this issue: the *RatioCut* formulation [16, 42] in which the number of vertices per cluster is considered

$$\text{RatioCut}(C_1, \ldots, C_\kappa) = \frac{1}{2} \sum_{i=1}^{\kappa} \frac{c(C_i, \bar{C}_i)}{|C_i|} = \sum_{i=1}^{\kappa} \frac{\text{cut}(C_i, \bar{C}_i)}{|C_i|} \quad , \tag{4.9}$$

and the the Ncut formulation [88]

$$\text{Ncut}(C_1, \ldots, C_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{c(C_i, \bar{C}_i)}{\text{vol}(C_i)} = \sum_{i=1}^{k} \frac{\text{cut}(C_i, \bar{C}_i)}{\text{vol}(C_i)} \quad . \tag{4.10}$$

where a balance is introduced thanks to notion of the volume of a set $\text{vol}(A) = \sum_{i \in A} d_i$, i.e. the sum of the degree $d_i = \sum_j s_{ij}$ of the vertex in A. Both these objective functions promote even partitions because they take a small value if the clusters C_i are not too small: RatioCut attained its minimum if all the sizes $|C_i|$ coincide, whereas the minimum of Ncut is reached if all $\text{vol}(C_i)$ are equal[1]. If clusters are well separated, all these

[1] This balance requirement is a strong assumption on the distribution of the data per cluster, however as notice in the first chapter, clustering is an ill-posed problem and requires some kind of regularization.

three objectives give very similar and accurate segmentations. However when clusters are marginally separated, Ncut yields accurate results [106].

Normalized spectral clustering [70] solve a relaxed versions of Ncut problem thanks to the notion of (normalized) Laplacian matrix:

$$L = D^{-1/2}(D-S)D^{-1/2} = I - D^{-1/2}SD^{-1/2}, \tag{4.11}$$

D indicates the degree d_i matrix collecting on its diagonal the degree of all the vertices. This matrix enjoys several properties: L is a symmetric and positive semi-definite matrix and the multiplicity of the eigenvalue 0 equals the number of connected components – that in ideal noise-free condition is the number of desired clusters κ. Moreover the top κ eigenvectors corresponding to the zero eigenvalue encapsulate clustering information.

More precisely, if one defines a $n \times \kappa$ matrix H that collects per columns the κ indicator vectors of clustering

$$H_{i,j} = \begin{cases} \mathrm{vol}(C_j)^{-\frac{1}{2}} & \text{if } x_i \in C_j \\ 0 & \text{otherwise,} \end{cases} \tag{4.12}$$

it is possible to rewrite the problem of minimizing Ncut presented in Equation (4.10) as

$$\min_{C_1,\ldots,C_\kappa} (H^\top (D-S)H) \quad \text{subject to} \quad H^\top DH = I, \tag{4.13}$$

exploiting the following three properties of H

$$H^\top H = I, \qquad H^j D H^j = 1, \quad H^j (D-S) H^j = \mathrm{cut}(C_j, \bar{C}_j)/\mathrm{vol}(C_j). \tag{4.14}$$

If we relaxing the condition (4.12) by allowing H to assume real entries, through the substitution $U = D^{1/2}H$, we finally obtain the problem

$$\min_{U \in \mathbb{R}^{n\times\kappa}} \mathrm{trace}(U^\top LU) \quad \text{subject to} \quad U^\top U = I. \tag{4.15}$$

This is a well known trace minimization problem in the form of the standard Rayleigh-Ritz theorem, which is solved by the matrix U that contains the first κ eigenvectors of L as columns. The columns of U can be thought as indicator vectors of the segmentation that solve the Ncut problem. Alternatively the rows of the eigenvector matrix can be considered as a new representation of the data points in a κ dimensional space. In the ideal case of completely separated clusters, points belonging to the same component have exactly the same representation. Nonteheless, in concrete, a common practice is to normalize the rows of U and use k-means as last step to extract the final partition. In this way spectral clustering can be viewed as a projection on the eigenspace spanned by the first κ eigenvectors of the Laplacian matrix followed by a clustering step.

4.2.1 T-Spectral

In this section we explore the performance of concatenating the embedding in Tanimoto space with spectral clustering on two important applicative scenario: video motion segmentation and 3D plane fitting. In particular we feed normalized spectral clustering with the similarity measure derived by the Tanimoto distances as $S_{i,j} = 1 - d_{\mathscr{T}}(\phi_H(x_i), \phi_H(x_j))$ and provide this algorithm, termed T-Spectral, with the number of inlier structures.

Motion segmentation

In motion segmentation the input data consists in a set of features trajectories across a video taken by a moving camera, and the problem consist in recovering the different rigid-body motions contained in the dynamic scene.

Motion segmentation can be seen as a subspace clustering problem under the modeling assumption of affine cameras. In fact it is simple to demonstrate that all feature trajectories associated with a single moving object lie in a linear subspace of dimension at most 4 in $\mathbb{R}^{2F}$ (where F is the number of video frames). As a matter of fact, the image coordinates y_{fi} of the i-th point in the f-th frame satisfies

$$y_{fi} = P_f Y_i, \tag{4.16}$$

where P_f is a projection matrix and X_i collects the 3D coordinates of the p-th point. If we form a matrix containing all the F feature trajectories corresponding to a point per columns, we get

$$\begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & & \vdots \\ y_{F1} & \cdots & y_{Fn} \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_F \end{bmatrix} \begin{bmatrix} Y_1, \ldots, Y_n \end{bmatrix}, \tag{4.17}$$

In brief this can be rewritten as $X = MS^\top$, where M is called motion matrix and S is termed structure matrix. Since $\operatorname{rank}(M) \leq 4$, $\operatorname{rank}(S) \leq 4$, it follows that $\operatorname{rank}(X) \leq 4$. For this reason feature trajectories of a dynamic scene containing κ rigid motion lie in the union of κ low dimensional subspace of $\mathbb{R}^{2F}$ and segmentation can be reduced to clustering data points in a union of subspaces.

We assess the performance of J-Linkage, T-linkage and T-Spectral on the Hopkins 155[2] motion dataset [103]. The dataset consists of 155 sequences of two and three motions, divided into three categories: checkerboard, traffic, and other (articulated/non-rigid) sequences. The trajectories are inherently corrupted by noise, but no outliers are

[2] available online at `http://www.vision.jhu.edu/data/hopkins155`

		Ransac	LSA 4n	ALC 5	ALC sp	SSC	J-Lnkg	T-Lnkg	T-Spec
Checkerboard	mean	6.52	2.57	2.56	1.49	**1.12**	1.73	1.20	1.38
	median	1.75	0.72	**0.00**	0.27	**0.00**	**0.00**	**0.00**	**0.00**
Traffic	mean	2.55	5.43	2.83	1.75	**0.02**	0.70	**0.02**	2.36
	median	0.21	1.48	0.30	1.51	**0.00**	**0.00**	**0.00**	0.52
Others	mean	7.25	4.10	6.90	10.70	0.62	3.49	0.82	**0.53**
	median	2.64	1.22	0.89	0.95	**0.00**	**0.00**	**0.00**	1.22
All	mean	5.56	3.45	3.03	2.40	**0.82**	1.62	0.86	1.82
	median	1.18	0.59	**0.00**	0.43	**0.00**	**0.00**	**0.00**	**0.00**

Table 4.1: Motion segmentation: ME (%) for video sequences with two motions

		Ransac	LSA 4n	ALC 5	ALC sp	SSC	J-Lnkg	T-Lnkg	T-Spec
Checkerboard	mean	25.78	5.80	6.78	5.00	**2.97**	8.55	7.05	4.35
	median	26.02	1.77	0.92	0.66	**0.27**	4.38	2.46	0.79
Traffic	mean	12.83	25.07	4.01	8.86	0.58	0.97	**0.48**	1.51
	median	11.45	23.79	1.35	0.51	**0.00**	**0.00**	**0.00**	1.01
Others	mean	21.38	7.25	7.25	21.08	**1.42**	9.04	7.97	5.45
	median	21.38	7.25	7.25	21.08	**0.00**	9.04	7.97	5.45
All	mean	22.94	9.73	6.26	6.69	**2.45**	7.06	5.78	3.86
	median	22.03	2.33	1.02	0.67	**0.20**	0.73	0.58	0.89

Table 4.2: Motion segmentation: ME (%) for video sequences with three motions

present. In the comparison we take into account also algorithms tailored to subspace clustering: SSC [89] Local Subspace Analysis [113] (in LSA 4n data are first projected to a space of dimension 4 times the number of motions) and Agglomerative Lossy Compression [83] (ALC 5 projected the data to a space of dimension 5, ALC sp uses a projection to a space of dimension 4 times the number of motions)

All methods have been tuned separately on each dataset for best performance.

The average and median misclassification errors are listed in Tables 4.1 and 4.2[3], where it can be appreciated that the results of T-Linkage and T-Spectral are somehow mixed.

On the two-motion sequences T-Linkage, T-spectral and SSC achieves a zero median error. T-Spectral struggles on the Traffic sequences where T-Linkage, on the contrary, achieves the best average error. The overall average misclassification error of T-Linkage is the second best after SSC, and fairly close to it.

On the videos with three moving objects, the advantage of knowing *a priori* the number of subspaces is reflected in the better performance of T-Spectral that score second in overall accuracy. The best performances are yielded by SSC that obtain the lowest median error in all the experiments, corroborating the advantages of integrating sparsity in the formulation of the problem.

3D plane fitting

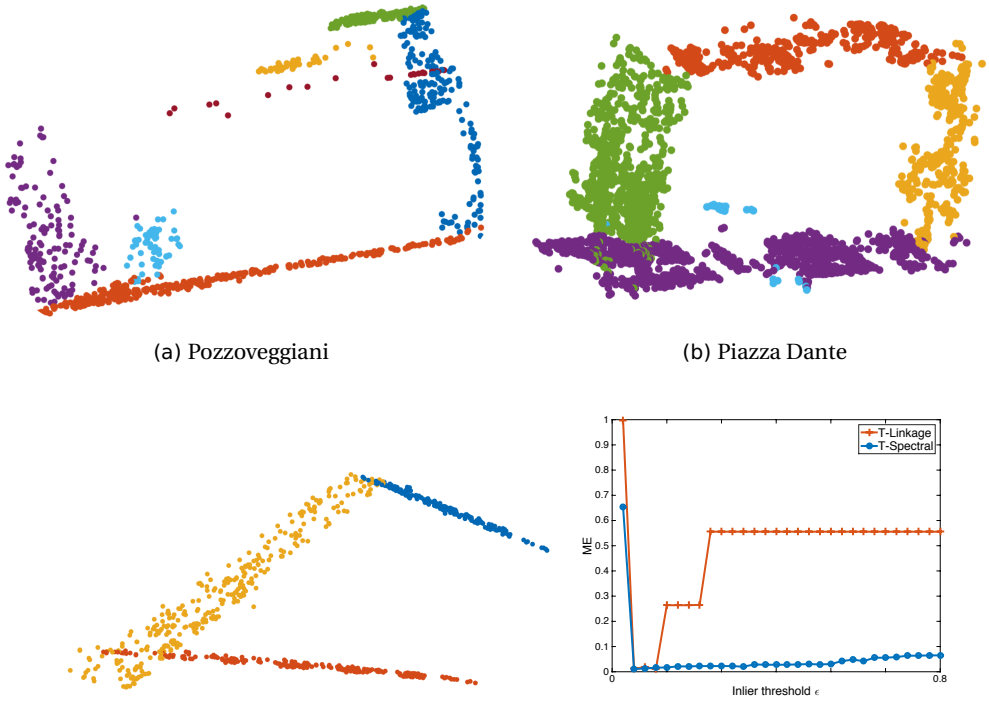
Multiple structure recovery can be fruitfully employed in the contest of automatic architectural and urban modeling from images. In this scenario usually a reconstruction technique produces arbitrarily dense but unstructured points clouds. Fitting multiple geometric primitives to these point clouds is a first step in organizing it in a higher informative semantic level. In this section we asses the performance of T-Spectral on this application, in particular we consider the problem of fitting plane to 3D-point clouds. Figure 4.1 shows some sample results of T-Spectral on three datasets[4].

For the *Castelvecchio* dataset (Figure 4.1c), composed by points lying on three planes, we manually computed the ground-truth and hence we are able to compare the performance of T-Linkage and T-Spectral with respect to the inlier threshold ϵ. Providing both methods with the optimal parameters, we obtain in both cases an accurate segmentation (the ME is 1.06 %). However it is worth to notice that, as shown in Figure 4.1d for T-linkage the inlier threshold is much more a critical parameter with respect to T-Spectral.

This is not surprising since the parameter ϵ in T-Linkage controls the inlier threshold but it also implicitly governs the orthogonality between points represented in the Tanimoto space, and in practice decides the number of attained models. T-Spectral on the contrary, using to good advantage the number of models given in input, is able to split the weighted graph defined by the similarity values in order to obtain the desired number of models. This ability, typical of partitional clustering, however comes at the cost of losing robustness. As a matter of fact, the graph cut formulation does not take

[3] the figures regarding SSC, LSA, and ALC are taken from the site mentioned above

[4] Publicly available from `http://www.diegm.uniud.it/fusiello/demo/jlk/`



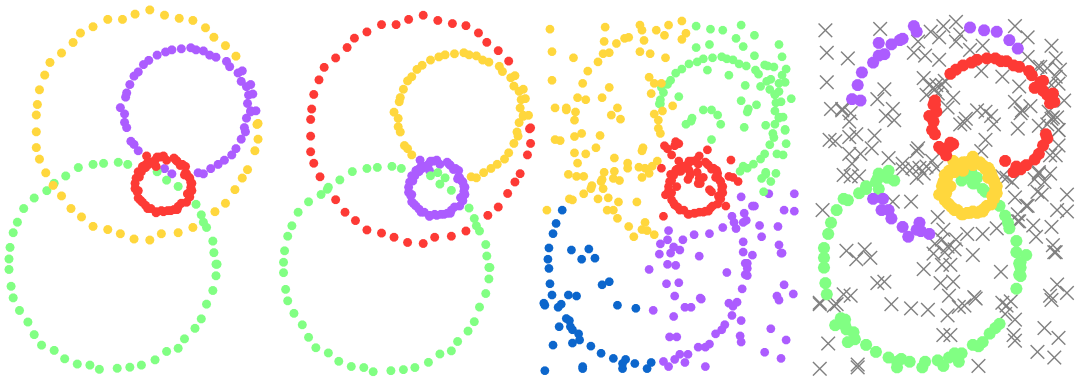
(a) Pozzoveggiani (b) Piazza Dante

(c) Castelvecchio (d) Castelvecchio ME (%)

Fig. 4.1: T-Spectral results on 3D plane fitting. Point membership is color coded.

into account outliers. If gross outliers contaminate the data, they are attached to some of the segments composed by inliers. In this case, as illustrated in Figure 4.2, the clustering produced by T-Spectral is somehow useless, the naive strategy of increasing the number of desired segments in order to collect together outliers is of no help. On the contrary, the presence of outlying data hinders T-Linkage, but the result of hierarchical clustering can be easily paired with proper outlier rejection strategy

In summary T-Spectral is a valuable alternative to T-Linkage if the data are inliers and the number of sought structures is known in advance, otherwise its lack of robustness heavily deteriorate its performances. In order to be able to deal with outliers some criteria of robustness has to be integrated in the spectral clustering framework. In the next section we will see how the ideas encountered in subspace clustering literature – namely low rank constraints and sparisty – can be exploited, together with consensus analysis, to overcome these difficulties.



(a) T-Spectral (b) T-Linkage (c) T-Spectral output (d) T-Link. & outlier rejection

Fig. 4.2: Weakness of T-Spectral. T-Spectral is very effective in dealing with pseudo outliers (top row, left). However it is not aimed at dealing with outliers. When data are contaminated the output of T-Spectral (bottom row, left) can not be easily interpreted for model fitting purposes, even if an additional cluster is required for collecting outliers.

4.3 Robust Preference Analysis

The main idea, pictorially represented in Figure 4.3, is to build on the preference analysis exploiting a soft-descender M-estimator and integrating in this approach decomposition techniques, such as Robust Principal Component Analysis (Robust PCA) and symmetric Nonnegative Matrix Factorization (symNMF). Loosely speaking, this method can be thought as a sort of "*robust* spectral clustering". We have seen that spectral clustering produces accurate segmentations in two steps: at first data are projected on the space of the first eigenvectors of the Laplacian matrix and then k-means is applied. The shortcoming of this approach is that it is not robust to outliers. We propose to follow the same scheme enforcing robustness: the eigen-decomposition step is replaced by Robust PCA on a pairwise affinity matrix, and Symmetric NMF [54] plays the role of k-means. In this way we are able to reduce the multi-model fitting problem to many single-fitting problems which are solved by scrutinizing the product between the matrix produced by Symmetric NMF and the preference matrix, together with the use of robust statistics.

In the previous experiments on the Castelvecchio dataset we have seen that T-Spectral is less sensitive to the choice of the inlier threshold, as the segmentation is affected to less extent by ϵ, being mainly controlled by the number of desired cluster. For this reason we propose to take advantage of this ability and instead of using the

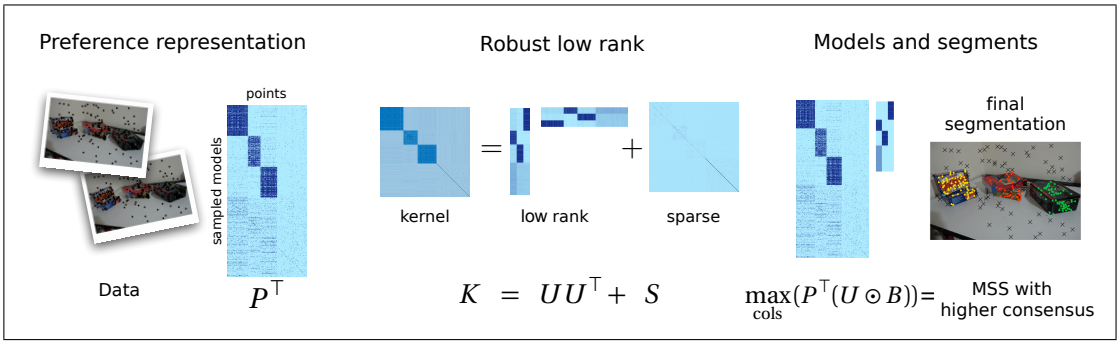


Fig. 4.3: Robust preference analysis in a nutshell: data points are shifted in a conceptual space where they are framed as a preference matrix $P^\top$. A similarity matrix K is defined exploiting agreement between preferences. Robust PCA and Symmetric NMF are used to robustly decompose $K = UU^\top + S$, where S is a sparse matrix modelling outliers, and U is a low rank matrix representing the segmentation. Finally, models are extracted inspecting the product of the preference matrix with thresholded U, mimicking the MSAC strategy. (Points are ordered by cluster for visualization purposes)

Tukey M-estimator, adopted in T-Linkage, we rely on soft descenders to express point votes avoiding hard cutoff. In particular we exploit the Cauchy weighting function. The preference trick consequently becomes:

$$\phi_H(x) = \left[\frac{1}{1+\left(\frac{\text{err}_\mu(x,h_1)}{\tau\sigma_n}\right)^2}, \ldots, \frac{1}{1+\left(\frac{\text{err}_\mu(x,h_m)}{\tau\sigma_n}\right)^2} \right]. \tag{4.18}$$

Rather than using directly Tanimoto distances, we rely on the definition of a a positive semi-definite kernel matrix $K \in [0,1]^{n\times n}$ on $P^\top$:

$$K(i,j) = \exp\left(-\tau(i,j)^2\right) \text{ where } \tau(i,j) = 1 - \frac{\left\langle P_i^\top, P_j^\top \right\rangle}{\|P_i^\top\|^2 + \|P_j^\top\|^2 - \left\langle P_i^\top, P_j^\top \right\rangle}. \tag{4.19}$$

to measure the agreement between preferences.

4.3.1 Clustering

We shall now describe how the affinity matrix can be exploited to segment the data. Consider an ideal affinity $n \times n$ matrix F which encodes point membership to the same

segment: $F_{i,j} = 1$ if x_i and x_j are clustered together and $F_{i,j} = 0$ otherwise. If data belonging to the same segment are arranged as consecutive points, the matrix F exhibits a block structure and therefore has rank κ equal to the number of clusters in the data.

As described in [117, 118] the problem of partitioning a set of data points in κ segments starting from a positive semi-definite affinity matrix K is equivalent to approximating K in a least square sense by means of an ideal affinity matrix F. In formulae, denoting by $\|\cdot\|_F$ the Frobenius norm of a matrix, we are interested in:

$$\min_F \|K - F\|_F^2 \ , \tag{4.20}$$

under conditions on F to be further specified. This problem is usually formulated by introducing a matrix $U \in \mathbb{R}^{n\times k}$ such that $F = UU^\top$, which represents a soft segmentation of the data: the element U_{ij} measures the probability that the i-th point belongs to the j-th segment.

According to the constraints imposed on U, the solution of (4.20) corresponds to different classical clustering algorithms, such as spectral clustering or k-means. More precisely Equation (4.20) can be expanded as:

$$\begin{aligned}
&\min \|K - UU^\top\|_F^2 \\
\iff &\min \operatorname{trace}[(K - UU^\top)^\top (K - UU^\top)] \\
\iff &\min \operatorname{trace}(K^\top K) - 2\operatorname{trace}(U^\top KU) + \operatorname{trace}(I) \\
\iff &\max \operatorname{trace}(U^\top KU).
\end{aligned}$$

The last equation becomes the objective of spectral clustering previously encountered in Equation (4.15) if $-K$ is the Laplacian of a graph and the columns of U are orthogonal. Since, it has been demonstrated [118] that the Laplacian of a similarity matrix can be viewed as the closest double stochastic approximation in relative entropy of the similarity matrix K, balanced partition are implicitly promoted. If K is chosen as the Gramian matrix of the data and if orthogonality, double stochasticity and non negativeness of U are enforced, the considered trace maximization problem corresponds to k-means. Finally, for sake of completeness, we recall that in the case in which $K = XX^\top$ is the covariance matrix of the data, under orthogonality constraint, solving Equation (4.20) is tantamount of doing Principal Component Analysis.

In summary the constraints that are usually imposed on U are: $U \geq 0$, $\operatorname{rank}(U) = \kappa$, $U^\top U = I$ and $UU^\top$ is doubly stochastic. Hard-clustering assignment implies orthogonality; being doubly stochastic represents a balancing condition on the sizes of the clusters; the non negativity of U ensures physical meaning of the entry of U which can be interpreted as the probability of points to belong to a given segment. The last constraint is the most important according to [117, 118], where it is highlighted as the key

ingredients for solving Problem (4.15) are the low-rank nature of both the affinity matrix and U (since since $k \ll n$), together with the non-negativeness of U.

Symmetric NMF (SymNMF) [54], that recently stands out in the clustering literature, enforces exactly these two proprieties. The idea at the basis of SymNMF is to rephrase (4.15) in the equivalent formulation

$$\min_{U \in \mathbb{R}_+^{n \times \kappa}} \|K - UU^\top\|_F^2 \tag{4.21}$$

and hence to find U minimizing (4.21) using an improved Newton-like algorithm that exploits the second-order information efficiently.

Interestingly we can interpret the clustering produced by SymNMF as a dimensionality reduction result: the columns of U form a basis of a latent space of the data, whereas the rows collect the coefficients that express the data as linear combinations of the basis vectors. Since in the preference space the basis over which the points are represented is determined by the sampled structures, the columns of U can be thought as the κ ideal structures that well describe the data.

When data are contaminated by gross outliers K has no longer low rank. For this reason, before applying SymNMF, we search *robustly* for the lowest-rank matrix L and the column-sparsest matrix S such that the data matrix can be decomposed as

$$K = L + S \ . \tag{4.22}$$

This Robust PCA step mimics in a outlier-resilient way the projection of data on the space of κ eigenvectors of the similarity matrix performed in spectral clustering. Moreover it is easy to recognize the formulation of LRR presented in Equation 4.6 when the dictionary A is chosen as the identity matrix. The decomposition (4.22) can be computed with the Augmented Lagrangian Method (ALM) [61], which solves the problem

$$\arg\min \|L\|_* + \lambda \|S\|_1 \qquad \text{s.t} \qquad K = L + S \ . \tag{4.23}$$

The parameter λ has a provable optimal value [15] at $\lambda = \frac{1}{\sqrt{n}}$, where n is the dimension of the square matrix K. In other words, following the ideas in LRR (Equation (4.6)) we are retaining the low rank part of the similarity matrix, rejecting the sparse part of K that corresponds to micro-clusters of outlying preferences. Please note that this approximation differs from the one adopted by SGC [49] where a low rank space is fit to a Gramian matrix in a least square sense. We depart from this model because a least squares fit is reliable as long as the sampled hypotheses are pure, but this property can not be ensured in presence of outliers.

We can now apply the SymNMF machinery to L (instead of K) in order to find a completely positive rank-κ factorization $L = UU^\top$. A segmentation is obtained from U

by considering the matrix B with the same dimension of U that has a one in each row where U achieves its row-maximum, and zero otherwise, i.e. $B_{i,j} = 1$ means that point i belongs to segment j. This last step is similar to the customary k-means that comes at the end of spectral clustering.

At this point, the matrix B represents a provisional segmentation of the points into κ segments containing outliers. The goal of the next section is to refine this segmentation and prune outliers, by solving, within each segment, a robust *single model* fitting.

4.3.2 Pruning outliers

Model extraction maximizing consensus.

Let us first observe that $P^\top \mathbf{1}$ (where $\mathbf{1}$ is a vector of ones) is the sum of the preference vectors of all the points in $P^\top$, so its entries are the votes obtained by each model. Hence finding the maximal entry of $P^\top \mathbf{1}$ is equivalent to doing a sort of MSAC (M-estimator SAmple and Consensus) with the Cauchy weighting function (Eq. 4.18).

We have seen that columns of $B = [B^1, \dots, B^k]$ can be regarded as indicators of the segments. Hence $P^\top B^i$ is the sum of the preference vectors of the points in the segment i, and its maximal entry represents the most preferred model in that segment. Therefore, the maxima over the columns of $P^\top B$ are the indices of the models in $P^\top$ that achieve maximum consensus in each segment. According to the observation above, this is equivalent to running a MSAC within each segment i with preference matrix $(P^\top \operatorname{diag}(B^i))$. The above reasoning can be extended to the matrix $U \circ B$ with entries in $[0,1]$, that corresponds to a soft segmentation in which outliers are under-weighted ($\circ$ denotes the component-wise or Hadamard product).

We found beneficial, prior to this step, to augment $P^\top$ with some pure models by random sampling and to remove "spurious" ones, according to the segmentation represented by B. In particular, we relax the concept of "spurious" to those models that are not contained in a single segment with at least 50% of their points; in other words, we label the points in $P^\top$ according to the segmentation given by B and we remove the columns where no label occurs more than 50% of the times. The new sampling is implemented by drawing random MSS within each segment i with probabilities given by the non-zero entries of $(U \circ B)^i$.

In summary, the maximal entry in each column of $P^\top (U \circ B)$ corresponds to the index of the most preferred model by the points of the segment, hence we choose it as the model that represents the segment. This could be a final result if the goal was to find the correct models. However, having recognized the entangled nature of model fitting and segmentation problems, we will unravel it by iterating between refining the model and updating the segmentation.

Segmentation.

The models computed from $\max_{\text{cols}}(P^\top(U \circ B))$ define a new tentative segmentation by assigning points to the nearest model. Within this segmentation, outliers are singled-out as points with a residual higher than a threshold $T = \tau\,\widehat{\sigma}$ where $\widehat{\sigma}$ is an estimate of the standard deviation of the residuals *of the inliers* and τ is the same tuning constant as in Equation (4.18) (set to 5.0 in our experiments).

The value of $\widehat{\sigma}$ can be obtained in several ways: it can be user provided ($\widehat{\sigma} = \sigma_n$) or can be computed from the residuals themselves, in a robust way. The second solution is to be preferred, as it leaves the choice of σ_n a noncritical step and makes the threshold T data-adaptive. We preferred the S_n estimator proposed in [85]:

$$S_n = c\ \text{med}_i(\text{med}_j(|r_i - r_j|)), \tag{4.24}$$

(where r_i, $i = 1,\ldots,n$ denotes the residual between the data x_i and the considered model) as a valid alternative to the more common median absolute deviation (MAD), which is aimed at symmetric distributions, and has a low (37%) Gaussian efficiency. S_n instead copes with skewed distributions, has the same breakdown as MAD but a higher Gaussian efficiency (58%). The *efficiency* of a robust estimator is defined as the ratio between the lowest achievable variance in an estimate to the actual variance of a (robust) estimate, with the minimum possible variance being determined by a target distribution such as the normal distribution. Asymptotic efficiency is the limit in efficiency as the number of data points tends to infinity.

The factor c can be set to 1.1926 for consistency with a normal distribution, but other distributions require different values (see [85] for details). In our experiments it has been tuned heuristically by analysing the distribution of the residuals of inliers given by the ground-truth. Values are reported in Table 4.3.

We noticed that in some cases most of the outliers are assigned to a single segment, resulting in a contamination greater than 50% that inevitably skews S_n. As a guard against this, S_n is computed only on the residuals smaller than $5.0\sigma_n$.

The model is then refined with a least-squares fit on the inliers, and the threshold T is computed again to determine the final segmentation.

4.4 Experimental evaluation

In this section we assess experimentally the effectiveness of our algorithm, henceforth dubbed RPA. All the code is written in Matlab and is available for download[5]. We used the inexact ALM code [2], whereas the SymNMF implementation is taken from [54].

[5] `http://www.diegm.uniud.it/fusiello/demo/rpa/`

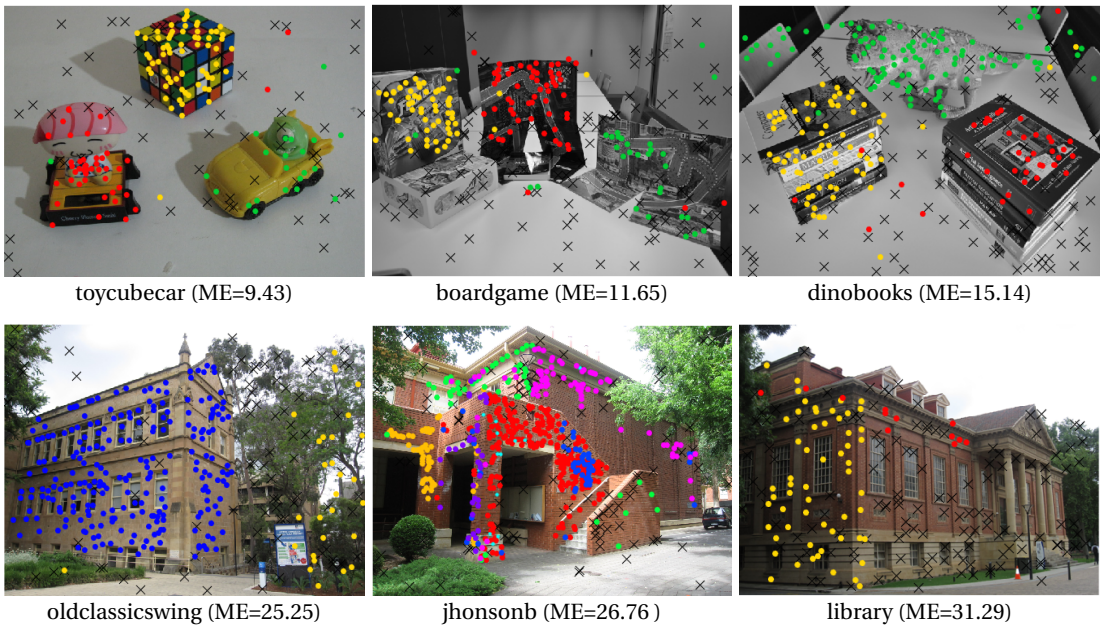


Fig. 4.4: Some of the worst results obtained by RPA on motion segmentation (top row) and planar segmentation (bottom row). Model membership is colour coded, black crosses (×) are outliers.

To start with, we consider two view motion segmentation and plane segmentation on the AdelaideRMF [110] dataset . We compared RPA with T-Linkage, which uses preference analysis and agglomerative clustering, and RCMSA (available at [84]), a robust method which relies on an efficient graph cut clustering based on a label optimization framework.

RPA and T-Linkage shared the same biased sampling in the conceptual space: we drew $3n$ hypotheses by uniform sampling and we used them to instantiate other $3n$ MSSs according to Equation (2.7). In all the experiments β was set to the median of all the Tanimoto distances between data points.

We provided T-Linkage with the inlier thresholds computed from the ground-truth segmentation for each single image pair, and we retained as inliers the largest κ clusters, κ being the correct number of models according to ground-truth. The input parameters of RCMSA and RPA are reported in Table 4.3 and have been kept fixed and equal for all the image pairs in each experiment.

Results are reported in Table 4.4, and demonstrate that our method outperforms its competitors, obtaining the lowest ME in most cases and the best mean and median results overall.

Experiment	σ_n	c	s	β
Motion segmentation	0.005	1.53	0.005	100
Planar segmentation	0.013	2.11	0.005	10

Table 4.3: Parameters used in the experiments. σ_n is the overall standard deviation of the residuals of the inliers, as computed from ground-truth (units refer to normalized image coordinates). c is the value in Equation (4.24) that experimentally provides the best estimate of σ_n from S_n. Parameters s and β refer to [78] and the values are the ones provided by the authors in their implementation.

Some of the worst cases for RPA are reported in Figure 4.4. The top row shows the results of motion segmentation that achieve the highest ME: the quality of the segmentation is nevertheless acceptable. The situation is different in the bottom row – corresponding to homography fitting – where three defective segmentations are shown, and the ME is indeed higher.

In *jhonsonb* the fault is of Symmetric NMF, which fails in finding a correct segmentation of the data, whereas in *library* and in *oldclassicswing* it is the value of σ_n that is respectively too low (over-segmentation) and too high (under-segmentation). While there are no remedies for the first case, the last two can be cured by a better choice of σ_n: for example, the ME drops to 24.53% for *library* and to 0.55% for *oldclassicswing* after assigning to σ_n the standard deviation of the residuals of the inliers *for that specific image pair*.

We also assess the performance of RPA on subspace clustering. In order to evaluate the robustness of the method we add to each sequence of Hopkins155 20% of outlying trajectories which are generated by starting a random walk at a random point in the image and adding to it increments taken from a trajectory (picked at random in the sequence) between consecutive frames (again picked at random). For all the experiments σ_n was fixed to 0.0048 and $c = 1.51$. From the figure reported in Table 4.5 it can be appreciated that our methods is very robust and gives results comparable with SSC which anyway is specifically designed for subspace recovery.

4.5 Final remarks

In this chapter we argued that preference analysis combined with robust matrix decompositions provides a versatile tool for robust geometric fitting which exploit profitably

	κ	%out	T-lnkg	RCMSA	RPA
biscuitbookbox	3	37.21	**3.10**	16.92	3.88
breadcartoychips	4	35.20	14.29	25.69	**7.50**
breadcubechips	3	35.22	**3.48**	8.12	5.07
breadtoycar	3	34.15	9.15	18.29	**7.52**
carchipscube	3	36.59	**4.27**	18.90	6.50
cubebreadtoychips	4	28.03	9.24	13.27	**4.99**
dinobooks	3	44.54	20.94	23.50	**15.14**
toycubecar	3	36.36	15.66	13.81	**9.43**
biscuit	1	57.68	16.93	14.00	**1.15**
biscuitbook	2	47.51	**3.23**	8.41	**3.23**
boardgame	1	42.48	21.43	19.80	**11.65**
book	1	44.32	3.24	4.32	**2.88**
breadcube	2	32.19	19.31	9.87	**4.58**
breadtoy	2	37.41	5.40	3.96	**2.76**
cube	1	69.49	7.80	8.14	**3.28**
cubetoy	2	41.42	**3.77**	5.86	4.04
game	1	73.48	**1.30**	5.07	3.62
gamebiscuit	2	51.54	9.26	9.37	**2.57**
cubechips	2	51.62	6.14	7.70	**4.57**
mean			9.36	12.37	**5.49**
median			7.80	9.87	**4.57**

	κ	%out	T-lnkg	RCMSA	RPA
unionhouse	5	18.78	48.99	**2.64**	10.87
bonython	1	75.13	**11.92**	17.79	15.89
physics	1	46.60	29.13	48.87	**0.00**
elderhalla	2	60.75	10.75	29.28	**0.93**
ladysymon	2	33.48	**24.67**	39.50	**24.67**
library	2	56.13	**24.53**	40.72	31.29
nese	2	30.29	7.05	46.34	**0.83**
sene	2	44.49	7.63	20.20	**0.42**
napiera	2	64.73	28.08	31.16	**9.25**
hartley	2	62.22	21.90	37.78	**17.78**
oldclassicswing	2	32.23	**20.66**	21.30	25.25
barrsmith	2	69.79	49.79	**20.14**	36.31
neem	3	37.83	25.65	41.45	**19.86**
elderhallb	3	49.80	31.02	35.78	**17.82**
napierb	3	37.13	**13.50**	29.40	31.22
johnsona	4	21.25	34.28	36.73	**10.76**
johnsonb	7	12.02	24.04	**16.46**	26.76
unihouse	5	18.78	33.13	**2.56**	5.21
bonhall	6	6.43	21.84	**19.69**	41.67
mean			24.66	28.30	**17.20**
median			23.38	29.40	**17.53**

Table 4.4: Misclassification error (ME %) for motion segmentation (left) and planar segmentation (right). κ is the number of ground truth structures and % out is the percentage of outliers. All figures are the average of the middle 3 out of 5 runs.

the interplay between consensus and preference. We proposed an approach similar in spirit to classic spectral clustering, with the advantage of being robust to outliers. Our strategy was to reduce the multi-model fitting task to many single robust model estimation problems attempting to solve the chicken-&-egg dilemma. In particular, we conceived three levels of protection against outliers. The first one is the adoption of the Cauchy function to model points preferences. The second level appears in the robust low rank approximation, where Robust PCA and SymNMF are used to gives rise to a

		2 motions		3 motions	
		SSC	RPA	SSC	RPA
Checkerboard	mean	8.19	4.53	9.58	6.09
	median	0.32	2.72	2.91	3.77
Traffic	mean	9.89	7.16	12.21	7.88
	median	1.93	4.80	5.87	4.77
Others	mean	17.97	13.68	22.84	19.15
	median	1.07	6.93	22.84	19.15
All	mean	6.33	9.84	10.94	7.24
	median	3.65	0.82	3.68	4.38

Table 4.5: Miscalssification error (ME %) in video motion segmentation (20% of outliers)

soft segmentation where outliers are under-weighted. Robust extraction of models in a MSAC-like framework, together with outlier rejection based on robust scale estimates is our third guard against outliers. The value of σ_n and the number of models κ are the only inputs required from the user. Experiments have provided evidence that our method compares favorably with state of the art competing algorithms.

5

Back to consensus analysis: set cover formulation

In this chapter we explore a different formulation of the multi-model fitting problem returning back to a discrete setting mainly focused on consensus. In this way exploiting the notion of cover set, we are able to deal in a principled manner with intersecting model and outliers, bypassing some limitations typical of cluster analysis. Moreover, having recognized the importance of the interplay between consensus and preferences, we show how it is possible to integrate the information captured by the preference trick in this formulation.

5.1 Introduction

A common trend in all the preference-oriented techniques is the bias towards the segmentation side of the multi-model fitting problem. If this tendency is not balanced taking into account the consensus counterpart of the problem – as indeed happens in RPA where consensus turns to play a crucial role in the final step of the algorithm – solutions produced by working exclusively in the conceptual space risk to inherit some of the disadvantages typical of clustering approaches. Undubitably the preference trick has the great advantage of casting specific multi-model fitting problems in a very general clustering framework. Nevertheless it has been largely recognized by the research community that the segmentation/clustering problem is essentially ill-posed, and it is impossible to decide in favor of a unique general method.

Two theorems corroborate this intuition. The common wisdom that any single clustering method can be optimal only with respect to some specific type of dataset has been demonstrated in the so called "*no free lunch theorem*" [108]. In practice the choice of a clustering scheme remains quite heuristic and mainly depends on the kind of available assumptions on the data – usually in the form of one or more parameters such as number of groups, inlier threshold, segments cardinality, etc...– that can be used

to constrain the solution space. A second theorem [53] confirms that clustering techniques are inherently fraught with ambiguities: Kleinberg conceives an axiomatic theory in which he defines three desirable properties that a clustering scheme ought to satisfy, namely scale-invariance, a richness condition that all partitions are achievable, and a consistency requirement on the shrinking and stretching of distances. In this setting an "*impossibility theorem*" is derived, demonstrating that there is no clustering function satisfying simultaneously all the three properties.

For example, linkage clustering, if a distance based stopping condition is adopted, enjoys the nice theoretical properties of scale-invariance and richness, but consistency is missing. It has also been recognized that single-linkage suffers from the so called chaining effects: since the merging criterion is strictly local, a chain of points can be extended for long distances without regard to the overall compactness of the emerging cluster. Furthermore the greediness of linkage affects the segmentation outcomes: for the sake of illustration a simple example regarding J-Linkage is presented in Figure 5.1. In this line fitting problem the optimal solution– with respect to Occam's razor– is given by two lines supporting three point each, unfortunately during the hierarchical clustering wrong decisions can possibly be made with relatively high probability and the returned solution can consist in three lines supporting two points.

In addition, two other main issues are not satisfactorily handled by clustering techniques. In first instance, the treatment reserved to outliers is not completely sound. For estimation purposes, gross outliers ought to fall in a special group of points, but clustering treats all the segments in the same way. This is the reason why the combination of the preference trick with spectral clustering fails or, more in general, why partitional clustering is not able to enforce robustness by simply requiring an additional group with the hope that outliers will be clustered together. Hierarchical methods in practice are more resilient to outliers, but in principle outliers do not have a specific treatment during the clustering phase: for example in T-Linkage it is necessary to rely on *a posteriori* specific heuristics to ensure robustness.

In second place, classical segmentation approaches are based on hard clustering (i.e. partitioning) and are not suitable for dealing explicitly with intersecting structures. We have seen from Section 1.2 that disjointness of the recovered structures is a delicate and important issue: It was at the root of the critics to Sequential RANSAC, it has motivated the development of Multi-RANSAC and, since relying solely on consensus leads to unsatisfactory performance, it promotes the shifting of the problem in the conceptual space. We have seen how this shifting allows to conceive more accurate methods, but intersecting models are either ignored or dealt indirectly with *ad hoc* post processing refinement on the obtained segmentations.

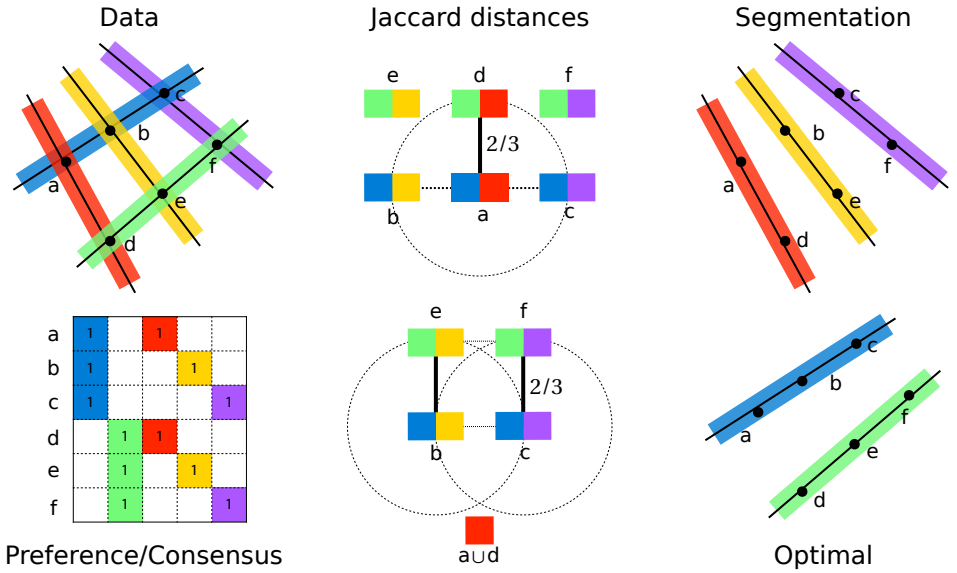


Fig. 5.1: A toy example that demonstrates the disadvantages of the greedy behavior of J-Linkage (best viewed in color). The data configuration captured in the preference matrix (left) gives rise in the conceptual space to the configuration depicted in the middle of the first row. The first sweep of J-Linkage starts merging together the two closest points. Without loss of generality we can assume that one of the chosen points is a. The second point has to be picked among the closest points to a, namely b,c,d which are at Jaccard distance 2/3. If the merged point is b –this happens with probability 1/3– the situation in the preference space becomes the one illustrated in the centre of the second row: b,c,e,f are the vertices of a "Jaccard square", whose edges measure 2/3 and whose diagonals measure 1. If the pairs (b,e) or (c,f) are chosen to be merged together J-Linkage fails in finding the optimal segmentation and returns three clusters instead of two. This happens with probability 1/6 assuming that ties are broken randomly with uniform probability.

In this chapter we outline an alternative strategy to tackle these two problems in a direct and more principled manner. If RPA can be considered a preference strategy strictly complemented by consensus considerations, here the situation is reversed. We return back to a well founded discrete consensus framework in which, however, all the valuable information captured by preferences are easily integrated, being aware that the interplay of preference and consensus is the added value to multi-model fitting solutions.

5.2 Coverages for multi-model fitting

In what follows we will assume a discrete setting in which point votes are expressed by binary values. Incidentally we observe that if one abstracts for a moment from the subtle differences between the implementations of the various multi-model fitting techniques based on either consensus or preferences, a unified view can be readily achieved by looking at the consensus/preference matrix P introduced in Equation (1.5).

The binary matrix P can be interpreted in several ways. It can be regarded as the incidence matrix of an hyper graph where rows correspond to vertices and columns represent hyperedges. We have seen also how rows, identified with preference sets, can be interpreted as representations in high dimensional spaces. In both these cases multi-model fitting boils down to cluster analysis. Changing the perspective, if columns are taken into account we are provided with a collection of consensus sets. RANSAC and Sequential RANSAC simply aim at finding the column or the κ columns respectively having greatest sums. Multi-RANSAC seeks for the κ "most orthogonal" columns. For sake of completeness we would like to mention also a further possible interpretation. The binary preference matrix can be viewed as the biadjacency matrix of a bipartite graph, where one set of vertices represents points and the other one represents structures, an edge links two vertices if the corresponding points belong to the consensus set of the related structures. Maximal bicliques in this graph correspond to biclusters of points and structures. The problem of finding maximal edge biclique can be relaxed to a continuous formulation that in turn is solved by NMF [39]. However, strictly speaking in multi-model fitting we are not interested in maximizing the number of edges of a biclique, as explained in Figure 5.2. Nevertheless this interpretation traces an interesting connection with RPA which relies also on symNMF, a particular instance of NMF.

Now we concentrate on classical consensus method: RANSAC, Sequential RANSAC and Multi-RANSAC in practice do not operate on the whole matrix P as we did for preference analysis. The subtle and decisive difference is that these consensus based techniques estimate the columns of P sequentially to save computational efforts. This principle of parsimony has no drawback in case of single model estimation, but in multiple model scenarios it causes the shortcoming of Sequential RANSAC. Once a structure of inlier is detected, its supporting points are removed and successive hypothesis are sampled exploiting only the remaining of the data, as a consequence inaccurate detection at early stage of the algorithm can heavily deteriorate the results. In addition, points in the intersections do not contribute to the sampling of subsequent structures and this greedy strategy is inherently prone to achieve suboptimal segmentation (see Figure 5.3). As commented in Section 1.2 this *estimate-and-remove* approach is tantamount of enforcing disjointness on the attained structures, which is essentially meant

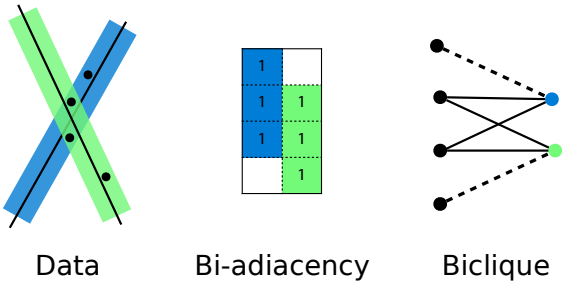


Fig. 5.2: Biclique interpretation. The binary preference matrix can be interpreted as a bi-adiacency matrix of a bipartite graph. The maximal edge biclique induces a segmentation of the data in which a unique cluster is determined in correspondence of the intersection of the two lines. Two points remain unassigned, whereas maximizing consensus leaves a single point unexplained.

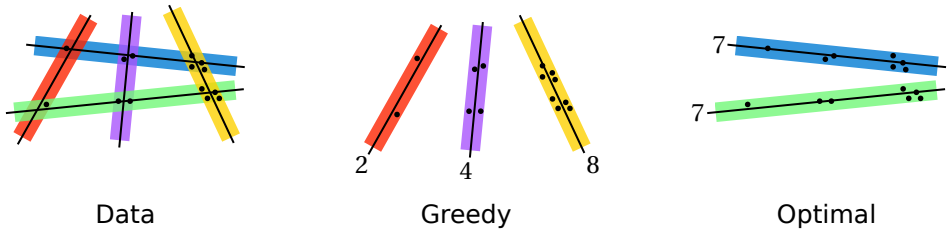


Fig. 5.3: Shortcomings of greediness on a fitting line example. Data points (on the left): three lines supporting each 2,4 and 8 points are intersected by two lines collecting half the total number of points each (7 points). Clearly these two lines suffices at explaining all the data points, therefore the optimal value for the MC problem is opt = 2, the greedy algorithm will pick the 3 remaining lines.

to distinguish between genuine structures and redundant ones. Recognized this fact, the objective of Sequential RANSAC is totally sensible.

For these reasons we decide to rely on the objective of maximizing consensus, but at the same time relaxing the notion of partition, and exploiting sampling on the totality of the data. In particular, at first we concentrate on the case in which all the points are inliers. A natural requirement is to ask that all the points are explained by some structures, in other words, the structures we are interested in determine, by means of their consensus sets, a *cover* of the data, i.e. a family of sets whose union contains X:

$$F = \{S_j : j \in J\} \quad \text{such that} \quad X \subseteq \bigcup_{j \in J} S_j, \tag{5.1}$$

Note that we are not requiring that the extracted sets are disjoint, so we are not limited to partitions and we can handle properly the case of intersecting models. By invoking the Occam's principle, a straightforward formulation is therefore to ask for a cover consisting of a minimal number of consensus set. In this way we are implicitly discouraging redundant models. Thus we are naturally led to the following *set cover* problem.

Definition 5.1 (Set cover problem). *Given a ground set X and a cover $F = \{S_1, \ldots, S_m\}$, the goal of set cover is to find a minimum subfamily in F that also covers X.*

In this formulation, X collects the data points and the family $F = \{S_1, \ldots, S_m\}$ is composed by the consensus sets of the sampled models. The property that F is a cover of X can be easily enforced by requiring that every points of X is sampled at least once. Set cover can be rephrased rigorously using the matrix P in the constraints formulation and introducing m binary variables $z_j \in \{0,1\}$ for each subset S_j. If S_j is selected in the solution then $z_j = 1$, otherwise $z_j = 0$. In this way set cover problem can be rephrased as an Integer Linear programming:

$$\min \sum_{j=1}^{m} z_j \quad \text{subject to} \quad Pz \geq \mathbf{1}. \tag{5.2}$$

The constraint can be expanded as

$$\sum_{j : x_i \in S_j} z_j \geq 1 \quad \forall x_i \in X \tag{5.3}$$

where becomes clear that it is meant to ensures that the solution $\{S_j\}_{j:z_j=1}$ is a cover of X.

If X is corrupted by gross error measurements, we can integrate outliers in the formulation of the problem at the cost of introducing an additional parameter κ measuring the desired number of segments. Instead of trying to find the smallest number of sets that cover all elements, we search for the largest number of points that can be covered by κ sets. This leads to the so called *maximum coverage* problem (MC)

Definition 5.2 (Maximum coverage). *Given a ground set X, a collection of subsets of X $F = \{S_1, \ldots, S_m\}$ and an integer κ, select from F at most κ subsets that cover the maximum number of points in X.*

Note that in MC formulations the collection F does not have to be necessarily a cover. Maximum coverage is translated in an Integer Linear program thanks to a collection of auxiliary variables y_i, such that $y_i = 1$ if x_i belongs to the returned subsets, 0 otherwise:

$$\max \sum_{i=1}^{n} y_i \tag{5.4}$$

subject to

$$\sum_{j=1}^{m} z_j \leq k \tag{5.5}$$

$$\sum_{j:x_i \in S_j} z_j \geq y_i \quad \forall x_i \in X \tag{5.6}$$

$$0 \leq y_i \leq 1 \tag{5.7}$$

$$z_j \in \{0,1\}. \tag{5.8}$$

Condition (5.5) enforces that no more than κ sets are picked and constraint (5.6) ensure that if $y_j \geq 0$ then at least one set S_j is selected.

Set cover and maximum coverage are long known to be NP-hard [52]: not surprisingly, since the inherent complexity of multi-model fitting does not disappear by simply rephrasing it in different terms. Nevertheless these optimization problems are among the oldest, most studied and widespread ones in the mathematical programming literature. Therefore we can reap the benefit of the efforts made by the scientific community in addressing this issues, and enjoy the fruits of several studies focused on approximating the solution of this problems.

For example, the greedy strategy which keeps choosing the set that covers most new points, until they all are covered, can be translated straightforward in the greedy-RANSACOV (Algorithm 3) which embodies the spirit of Sequential RANSAC with the only differences that the hypothesis space is not sampled iteratively and, instead of returning a partition, intersecting segments are allowed. It has been demonstrated by Feige [33] that this greedy strategy is the best possible in terms of approximation ratio. More precisely an approximation of $H(n)$[1] holds in the case of set cover problem, and $1-1/e$ for the MC problem. This result applies effortless to greedy-RANSACOV giving a provable quality measure of the solution.

In practice, a more efficient strategy is to use standard solvers to address the MC problem, such as Integer Linear programming (Algorithm 4). In order to reduce the computational load of the algorithm we find beneficial to perform the following preprocess on the input family of sets. First of all, keeping in mind that our aim is to maximize consensus, we refit a structure to each consensus set via least squares, then we update the structure and its supporting points only if the consensus has increased. The remaining sets are hence ordered by cardinality $S_1, \ldots, S_k$ and a set S_j is discarded if

[1] here we denote by $H(n)$ the n-th harmonic number

Algorithm 3 greedy-RANSACOV

Require: data points X, inlier threshold ϵ, number of structures κ
Ensure: Subsets of X
Generate by random sampling a poll of hypotheses model $H = \{h_1, \ldots, h_m\}$
Instantiate the consensus/preference matrix P
while κ sets are selected or all the points have been covered **do**
pick in F the set that covers the maximum number of uncovered elements
mark elements in the chosen set as covered
end while

Algorithm 4 ILP-RANSACOV

Require: data points X, inlier threshold ϵ, number of structures κ
Ensure: Subsets of X
Generate by random sampling a poll of hypotheses model $H = \{h_1, \ldots, h_m\}$
Instantiate the consensus/preference matrix P
Refine the family F of consensus set defined by P

Solve WMC problem with Integer Linear programming

$$S_j \subseteq \bigcup_{i=1}^{j-1} S_i, \tag{5.9}$$

the rationale of this choice is to keep only those structures that explain at least a new point. Interestingly this step furnished as byproduct assurance on the optimality of the returned solution. In fact this procedure reduces the maximal *frequency* of each elements, i.e. the number of sets that cover a point, and thanks to [105] we are guaranteed to achieve better solutions, since it has been demonstrated that Linear Programming succeeds in approximating the optimum of a factor of f, where f is the maximal frequency among all the points.

Finally, to complete the picture we also cast Multi-RANSAC in the framework of maximal coverage, the resulting method (Multi-MCRANSAC) is presented in Algorithm 5. The strategy is similar to greedy-RANSACOV, the difference is that, after a set is picked, the subsequent ones are searched among the ones having maximal Jaccard distance with the set of currently covered elements. In this way we try to emulate the disjointness constraint enforced in Multi-Ransac.

5.3 Comparison with Facility Location

The closest methods to our in the literature are those casting multi-model fitting as a Facility Location problem: provided a set of potential facilities (which corresponds to the

Algorithm 5 Multi-RANSACOV

Require: X data points, ϵ inlier threshold, κ number of structures
Ensure: Subsets of X
  Generate by random sampling a poll of hypotheses model $H = \{h_1, \ldots, h_m\}$
  Instantiate the consensus/preference matrix P
  Pick the set S_1 that covers the maximum number of uncovered elements
  $U = S_1$
  while κ sets are selected **do**
    Find the set S_j that have maximal Jaccard distance from U
    $U = U \cup S_j$
  end while

pool of tentative structures), Facility Location selects an optimal subset of facilities and assigns customers (i.e. data points) to one facility each, so as to minimize the sum of facility opening costs and the distances between customers to their assigned facilities. This leads to the optimization of a cost function composed by two terms: a modelling error – i.e. customers-facility distances – which can be interpreted as a likelihood term, and a penalty term to encode model complexity – the cost to open the facilities – mimicking classical MAP-MRF objectives. Some authors solves it with ILP [56, 59, 87, 101] while others propose different methods [28, 47, 75, 115].

Although set cover and Facility Location are related (the first is a special case of the second), and ILP has been used to solve both, our ILP-RANSACOV differs from previous work based on Facility Location in many respects.

In first instance, Facility Location needs to guess a correct trade-off between data fidelity and model complexity, in order to strike the correct balance between over and under fitting. Our MC formulation by contrast, eludes this thorny trade-off: instead of balancing two incommensurable quantity in the cost function, it explicitly requires the maximum number of models as a clear, intelligible parameter.

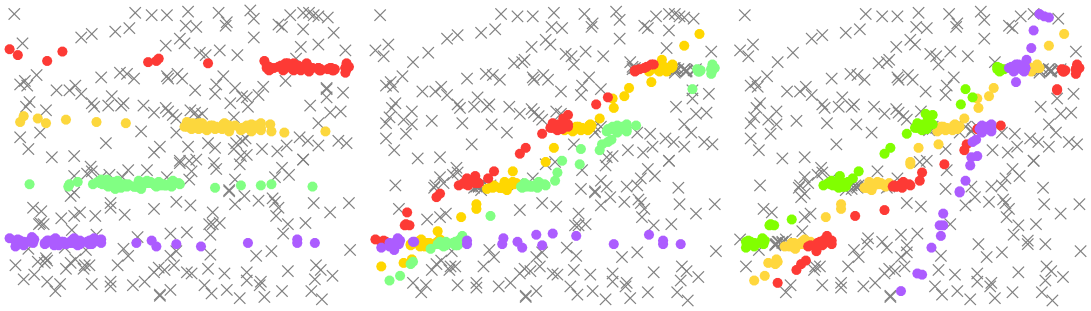
Second, Facility Location minimizes the fitting error on the continuum of residuals, in the same spirit of MLE estimators, while MC gains resiliency to outliers by using an inlier threshold and maximizing the consensus, *à la* RANSAC. The rogue points will be simply left uncovered, whereas Facility Location copes with outliers by introducing a special additional model for which a constant fidelity measure is defined.

Finally, Facility Location produces a partition of the data, whereas set cover inherently caters for intersecting solutions.

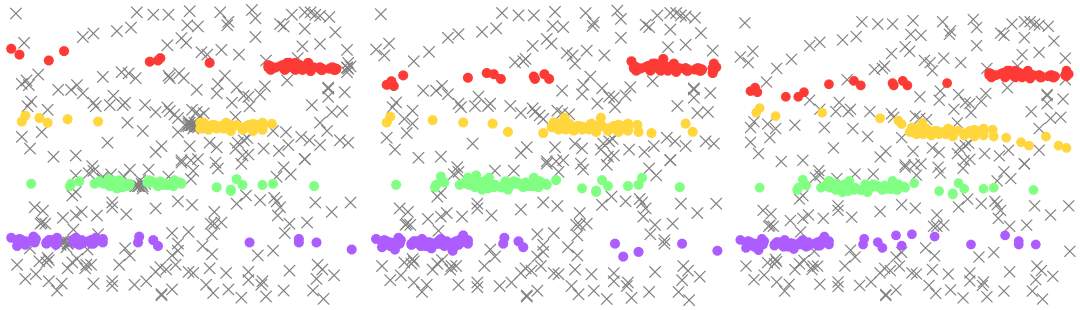
5.4 Experiment on synthetic data

The setup of maximum coverage offers the opportunity of comparing ILP-RansaCov, greedy-RansaCov, Multi-RansaCov, together with J-Linkage and T-Linkage at equal conditions of sampling. In this section we explore the performance of these algorithms on some synthetic datasets. In all the experiments each structure consists of 50 inliers points, contaminated by Gaussian noise and variable outliers percentage. The data sets consist of segments in several configurations: star (*star5*, *stair4* and *star11*) and circles (*circle4*). We provide J-Linkage and T-Linkage with the correct number of structures, the κ retrieved structures supporting more points, among the ones produced by the algorithm, are kept as inliers. We use Matlab's `intlinprog` as the Integer Linear programming solver. The outcomes on different datasets are collected in Figures from 5.4 to 5.7.

First of all we can notice that in the *Stair4* experiment (firstly used in [122] to criticize Sequential Ransac), greedy-RansaCov and Multi-RansaCov perform poorly: the shortcomings of these greedy strategies explained in Figure 5.3 are here afoot: the incorrect selection of the first structure compromises the subsequent interpretation of the data. These two methods attained a suboptimal segmentation also on the *circle4* dataset where one of the four structures is oversegmented at the expense of the smaller circle in the centre. On the *star5* sequence all the methods perform quite similarly with the exception of Multi-RansaCov which is guided by the principle of finding disjoint segments and therefore fails in finding one of the lines that has a consistent intersection with the others inlier structures. The *star11* manifests as well this phenomenon in the results obtained by Multi-RansaCov and by J-linkage which both miss a ground truth segment. While Multi-RansaCov oversegments a structure, for J-Linkage the cause of this behavior can be ascribed to the fact that, during the merging process, some inliers are incorrectly aggregated to spurious outlying model, as a consequence the recovered segment which actually corresponds to a ground truth structure has less points than the necessary to be in the first κ largest models and is deemed as outlier. In general the tendency of loosing inliers during the segmentation step affects J-Linkage (and T-Linkage) also in the other datasets, for example it is particular evident on *circle4* even if on other datasets is less manifest. These artifacts, by the estimation perspective, are certainly unfavorable since the more inliers support a structure the more the fit will be accurate, however in these cases the discovered inliers are enough to recover accurately the corresponding structures. ILP-RansaCov yields reliable segmentations in all the experiments, as it can be appreciated by inspecting its ME values reported in Table 5.1, that are considerable lower than the other algorithms. The reason can be found in the departure from the partition paradigm: as a matter of fact J-Linkage and T-Linkage are

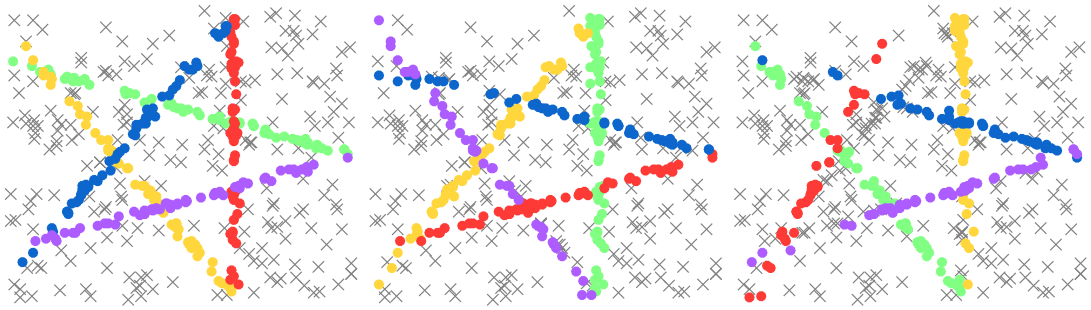


(a) Ground truth (b) greedy-RansaCov (c) Multi-RansaCov

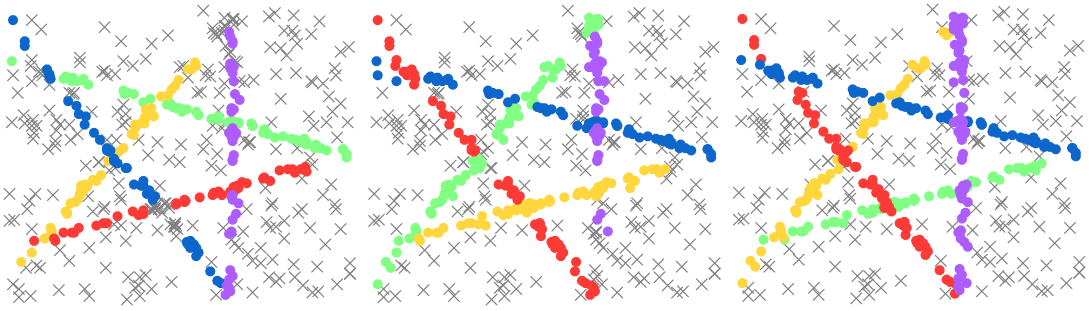


(d) J-Linkage (e) T-Linkage (f) ILP-RansaCov

Fig. 5.4: Comparison on stair4 (50% of outliers x)

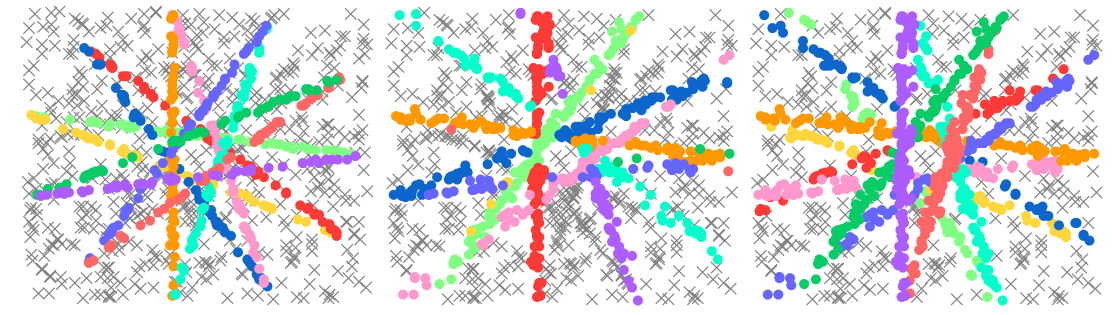


(a) Ground truth (b) greedy-RansaCov (c) Multi-RansaCov

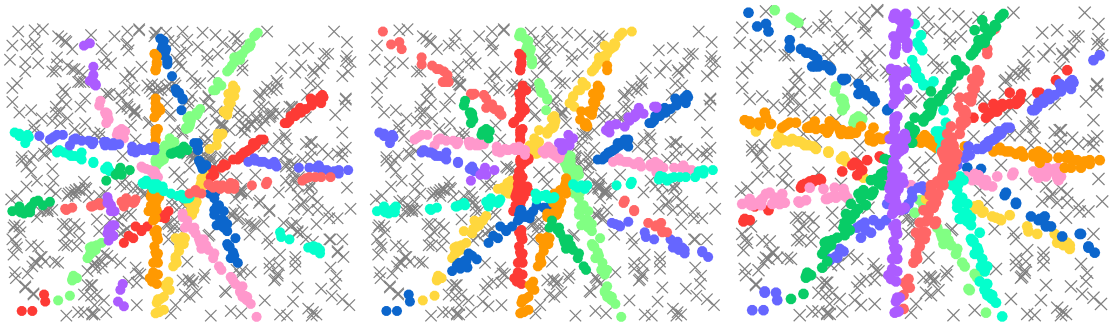


(d) J-Linkage (e) T-Linkage (f) ILP-RansaCov

Fig. 5.5: Comparison on star5 (60% of outliers x)

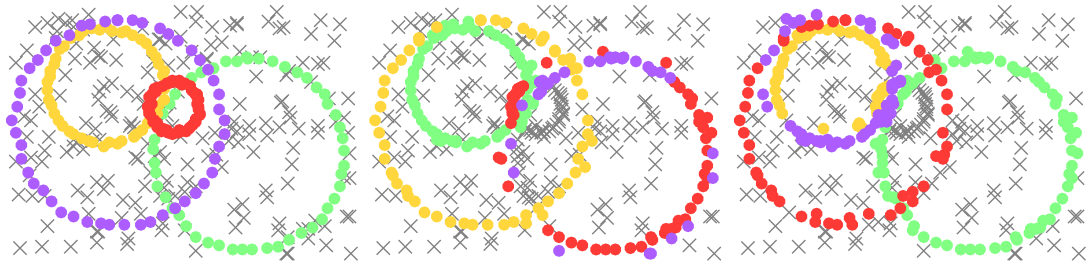


(a) Ground truth (b) greedy-RansaCov (c) Multi-RansaCov

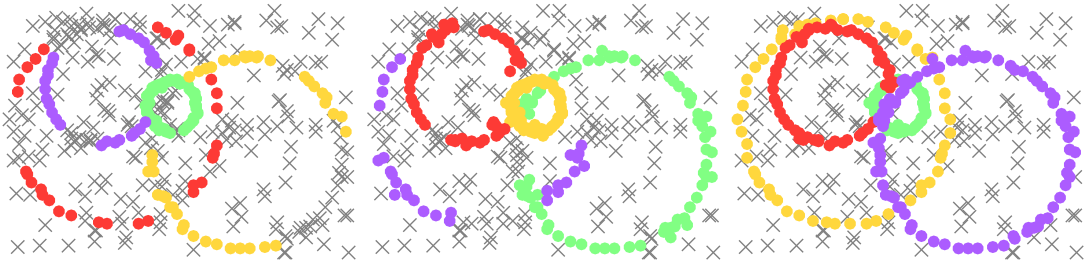


(d) J-Linkage (e) T-Linkage (f) ILP-RansaCov

Fig. 5.6: Comparison on star11 (50% of outliers x)



(a) Ground truth (b) greedy-RansaCov (c) Multi-RansaCov



(d) J-Linkage (e) T-Linkage (f) ILP-RansaCov

Fig. 5.7: Comparison on circle4 (50% of outliers x)

	greedy-RansaCov	Multi-RansaCov	J-Linkage	T-Linkage	ILP-RansaCov
Stair4	39.20	54.80	10.20	10.00	12.00
Star5	10.40	24.40	15.20	14.40	3.80
Star11	32.36	39.27	35.00	33.09	25.18
Circle4	30.25	35.00	26.50	23.00	11.25
mean	28.05	38.37	21.73	20.12	13.06

Table 5.1: Misclassification error (ME %) on the experiments from Figures 5.4 to 5.7.

not aimed at dealing with intersecting models. This can be sensed in *Stair4*, where there are not intersecting structures and the performance of J-Linkage and T-Linkage are in the same range of ILP-RansaCov.

5.5 Experiments on real data

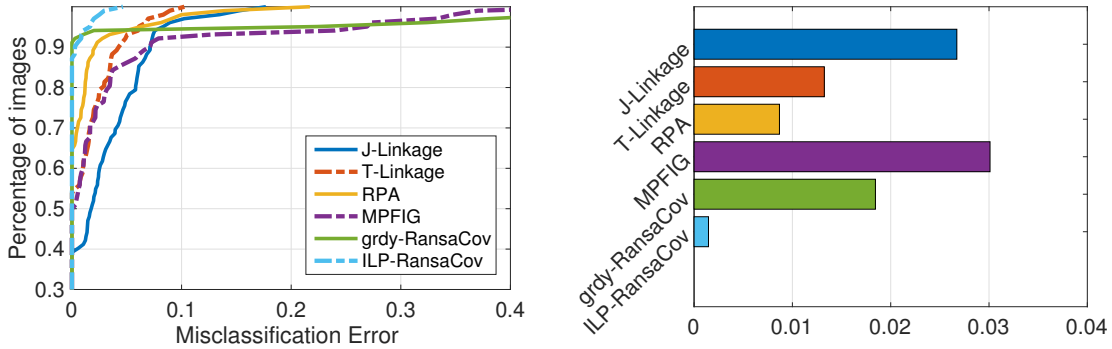
In this section, we demonstrate the performance of ILP-RansaCov on two classical Computer Vision applications, namely: vanishing point detection, and video motion segmentation. In all these scenarios we compare ILP-RansaCov with J-Linkage, T-linkage and RPA. In addition, one reference method have been added to the comparison for each specific scenario, namely: MFIPG [75] in the vanishing point experiments, SSC [89] for video motion segmentation.

Vanishing point detection.

In this experiment we compare the performances of ILP-RansaCov with MFIPG on vanishing point detection using the York Urban Line Segment Database [29], or York Urban DB in short, a collection of 102 images of architectural Manhattan-like environments (i.e. scenes dominated by two or three mutually orthogonal vanishing directions). Annotated line-segments that match with the 3-d orthogonal frame of the urban scene are provided with the ground-truth, no outliers are present in the data. The aim is to group the supplied segments in order to recover two or three orthogonal vanishing points.

MFIPG (Model-Fitting- with-Interacting-Geometric-Priors) is a recently proposed method that builds on the PEARL [28] algorithm adding high-level geometric priors. In particular, in this application, an additional term expressing interaction between vanishing points is included into the Facility Location formulation, to promote the extraction of orthogonal vanishing points. The global input parameters recommended in the

Fig. 5.8: A sample of the worst ILP-RANSACOV results on YorkUrbanDB (**vanishing point detection**). Line membership is color-coded.



(a) Cumulative ME

(b) Area above the cumulative ME

Fig. 5.9: Results on YorkUrbanDB. (a) is the cumulative distributions of the errors per sequence; (b) shows the area above the curve (the smaller the better).

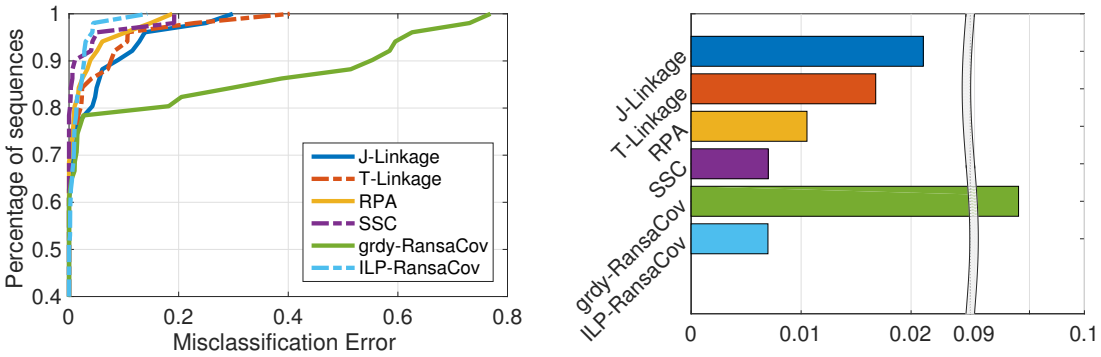
	J-Lnkg	T-Lnkg	RPA	MFIGP	Grdy-RansaCov	ILP-RANSACOV
Mean	2.85	1.44	1.08	3.51	2.38	**0.19**
Med	1.80	**0.00**	**0.00**	0.16	0.00	**0.00**

Table 5.2: Misclassification error (ME %) on YorkUrbanDB.

original paper have been optimized individually for each single image to enhance the results.

Figure 5.8 shows three images where ILP-RANSACOV achieved the worst ME, which are nevertheless qualitatively correct. Figure 5.9(a) reports the cumulative distribution of the ME per sequence, i.e. the value on the ordinate corresponds to the percentage number of sequences where the algorithm achieved a ME lower than the abscissa. The differences among the methods can be better appreciated by plotting the area above

Fig. 5.10: A sample of the worst ILP-RANSACOV results on Hopkins155 (**video motion segmentation**). Point membership is color-coded.



(a) Cumulative ME

(b) Area above the cumulative ME

Fig. 5.11: Results on Hopkins155. (a) is the cumulative distributions of the errors per sequence; (b) shows the area above the curve (the smaller the better).

the cumulative distribution of ME (Fig. 5.9(b)) or by analysing the average and median ME, collated in Tab. 5.2. These quantitative results confirms that ILP-RANSACOV achieves the most accurate performance, followed by RPA. Please note also that greedy-RANSACOV, a proxy of the vilified Sequential RANSAC, performs better than other sophisticated methods, in this task.

In this experiments we use the 51 *real* video sequences from the Hopkins 155 dataset, each containing two or three moving objects, with no outliers. Rather than considering the whole trajectories, following [94] in order to better deal with degenerate motions – typical of these kind of videos –, we project the data onto an affine 4-dimensional space where the rigid-body segmentation is translated in a 3-d plane fitting problem.

Figure 5.10 reports some sample results, in particular three sequences belonging to *Traffic 2* and *Others 3* subsets, respectively, where ILP-RANSACOV achieves suboptimal segmentations. Figure 5.11 and Tab. 5.3 provide a comparison of the performances in terms of ME: ILP-RANSACOV places in the same range of SSC and achieves the best

		J-Lnkg	T-Lnkg	RPA	SSC	Grdy-RansaCov	ILP-RansaCov
Traffic 3	Mean	1.58	0.48	**0.19**	0.76	28.65	0.35
	Med	0.34	0.19	**0.00**	**0.00**	1.53	0.19
Traffic 2	Mean	1.75	1.31	0.14	**0.06**	7.48	0.54
	Med	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Others 3	Mean	6.91	5.32	9.11	**2.13**	14.89	**2.13**
	Med	6.91	5.32	9.11	**2.13**	14.89	**2.13**
others 2	Mean	5.32	6.47	4.41	3.95	8.57	**2.40**
	Med	1.30	2.38	2.44	**0.00**	0.20	1.30
All	Mean	2.70	2.47	1.42	1.08	10.91	**0.98**
	Med	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0,00**

Table 5.3: Misclassification error (ME %) on Hopkins155.

overall results. In this case the advantage of solving the MC problem with a global approach is afoot, since the greedy strategy of greedy-RansaCov, sampling being equal, fails in recovering accurate segmentations.

5.6 Weighted version

The Integer Linear formulation can be straightforward generalized to the *weighted maximum coverage* problem (WMC):

Definition 5.3 (Weighted maximum coverage). *Given a ground set X, an integer κ and a collection of subsets F. Non negative weights c_i are associated to the elements of X, the aim is to select at most κ sets from F so as to maximize the overall weight of covered points.*

The Integer Linear programming formulation of the problem changes accordingly, replacing the objective function of MC expressed in Equation (5.4) with the sum of weights of the covered points:

$$\max \sum_{i=1}^{n} c_i y_i. \tag{5.10}$$

MC can be derived from WMC defining all the weights equal to one. The weighted version allows to take advantage of any kind of available prior information by the specification of suitable weights in order to promote the coverage of certain points. By the

perspective of multi-model fitting, this possibility provides a profitable occasion to integrate in the maximum coverage framework the information furnished by the preference analysis we have discussed in the previous chapters, since the results on Tanimoto space hold in particular for space endowed with the Jaccard distance.

On that account our aim is to exploit the preference trick to discourage the covering of outliers. Since a distinctive attribute of outliers is their sparsity in the conceptual space, we decide to rely on this feature to downweight outlying elements, as proposed in [13]. In Section 2.2 we have introduced the concept of reachability distance (Definition 2.3) and we have presented OPTICS [3], a technique that encapsulates the local density of points in a reachability diagram, a sort of dendrogram in which the points are ordered and scored according to their mutual reachability distance. We recall that in this context locality is simply given by ζ nearest neighbors, where ζ is the cardinality of MSS. In practice, a reachability diagram summarizes a wealth of information on points vicinity, displaying segments of inliers as valleys and outliers as peaks with high reachability value. Tanimoto distances are bounded in $[0,1]$ so by taking

$$c_i = 1 - \mathrm{rd}(x_i) \in [0,1] \tag{5.11}$$

we can define a simple and very general measure of "outlier-ness", where $\mathrm{rd}(x_i)$ indicates the reachability distance of the i-th point. More sophisticated measures can be as well integrated, however this one has proven to be effective as demonstrated by the following experiments.

5.6.1 Experiments

We validate the weighted version of ILP-RANSACOV on the Adalaide datasets, dealing with motions and planes segmentation problems. Additionally we compare WMC with greedy-RANSACOV, Multi-RANSACOV and J-Linkage. All these methods are given the inlier threshold computed from the ground truth[2]. From the figures reported in Table 5.4, in which T-Linkage and RPA are also reported, we can appreciate that the accuracy of this method is comparable with RPA on fundamental matrices estimation and improves the overall performances on multi-homography fitting. RPA and ILP-RANSACOV demonstrate to be the two best methods, corroborating the advantages of the interplay of consensus and preference. Certainly the departure from the partition paradigm has beneficial effects on the value of the ME of ILP-RANSACOV. Some mixed results are reported in Figure 5.12. In Figure 5.12a we present an example of undersegmentation

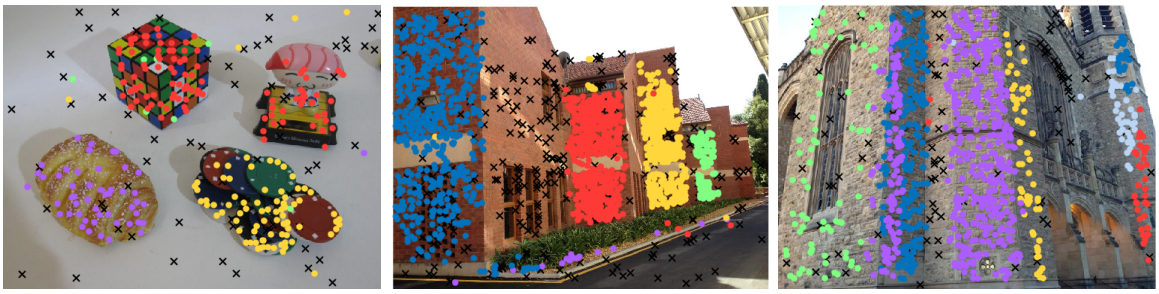
[2] Which indeed is not always very reliable since often the segmentation is biased to be semantic: outliers or pseudo outliers happen to be closer to inlier to a given model. Recognized that, correcting the ground truths however is a very delicate and elusive task, therefore we prefer not to alter the ground truth for comparison purposes

	Seq. MCR	Multi MCR	J-Link.	T-Link.	RPA	WMCR
biscuitbookbox	23,64	31,78	17,05	3,10	3,88	**2,33**
breadcartoychips	35,93	21,65	14,29	14,29	**7,50**	9,09
breadcubechips	18,26	34,35	33,91	**3,48**	5,07	6,52
breadtoycar	29,27	28,66	9,15	9,15	**7,52**	9,15
carchipscube	38,41	23,17	6,50	**4,27**	6,50	**4,27**
cubebreadtoychips	33,76	18,47	8,92	9,24	**4,99**	16,88
dinobooks	44,84	22,42	23,89	20,94	15,14	**13,86**
toycubecar	33,33	32,32	30,30	15,66	9,43	**3,54**
biscuit	1,88	1,88	19,12	16,93	1,15	**0,94**
biscuitbook	4,99	12,02	26,10	**3,23**	**3,23**	4,40
boardgame	40,98	26,69	25,94	21,43	**11,65**	19,17
book	7,03	7,57	3,24	3,24	2,88	**1,62**
breadcube	37,34	**3,00**	49,36	19,31	4,58	**3,00**
breadtoy	17,99	22,66	3,96	5,40	**2,76**	5,76
cube	**2,03**	4,75	9,49	7,80	3,28	**2,03**
cubetoy	4,60	**1,26**	6,69	3,77	4,04	**1,26**
game	**1,30**	1,74	2,61	**1,30**	3,62	1,74
gamebiscuit	7,41	24,69	15,43	9,26	**2,57**	5,86
cubechips	23,10	5,42	6,14	6,14	4,57	**3,25**
mean	21,37	17,08	16,43	9,37	**5,49**	6,04
median	23,10	21,65	14,29	7,80	4,57	**4,27**

	Seq. MCR	Multi MCR	J-Link.	T-Link.	RPA	WMCR
unionhouse	12,95	7,34	33,13	48,99	10,87	**7,06**
bonython	3,11	**2,59**	15,03	11,92	15,89	**2,59**
physics	**0,00**	**0,00**	17,48	29,13	**0,00**	**0,00**
elderhalla	18,69	**0,93**	12,62	10,75	**0,93**	**0,93**
ladysymon	**5,29**	19,38	28,63	24,67	24,67	**5,29**
library	2,36	28,77	30,66	24,53	31,29	**1,42**
nese	38,17	49,38	24,48	7,05	**0,83**	3,32
sene	1,27	10,17	14,83	7,63	**0,42**	0,85
napiera	44,86	16,44	22,95	28,08	**9,25**	16,10
hartley	**5,71**	15,56	13,97	21,90	17,78	17,14
oldclassicswing	19,28	22,59	14,60	20,66	25,25	**12,67**
barrsmith	**28,09**	31,06	33,62	49,79	36,31	31,06
neem	**18,70**	42,17	49,79	25,65	19,86	23,04
elderhallb	**11,43**	39,59	28,16	31,02	17,82	22,86
napierb	32,49	54,01	25,74	**13,50**	31,22	24,05
johnsona	23,51	37,11	12,46	34,28	**10,76**	22,95
johnsonb	21,47	31,73	**19,23**	24,04	26,76	32,37
unihouse	29,96	47,68	33,13	33,13	**5,21**	12,34
bonhall	32,81	53,59	54,01	21,84	41,67	**9,28**
mean	18,43	26,85	25,50	24,66	17,20	**12,91**
median	18,70	28,77	24,48	24,53	17,78	**12,34**

Table 5.4: Misclassification error (ME %) for motion segmentation (left) and planar segmentation (right).

since a unique fundamental matrix explains both the cube and the toy segments. In Figure 5.12b the algorithm fails in estimating the homography that describes the second wall of the building from the left. The reason can be ascribed to the weighting mechanism: there is a perceptible difference in the number of supporting point of this missed wall and the other ones. This density discrepancy in the data space results first in fewer sampled hypothesis corresponding to this structure, second in a less dense cluster in the Jaccard space, third in high reachability values and finally in lower weights. As a consequence the corresponding structure is not selected. In Figure 5.12c we can appreciate the advantages of dealing with coverage rather than partitions: the points on the corner of the facade of the building lie near the intersection of two estimated homographies, multiple-membership takes into account this fact without affecting accuracy.



(a) cubebreadtoychips (b) unionhouse (c) bonhall

Fig. 5.12: Sample result of Weighted Maximum Coverage applied to the Adelaide dataset (best viewed in color, outliers are marked as black crosses x)

5.7 Final remarks

In summary the discrete combinatorial setting of ILP-RansaCov has demonstrated to be a viable alternative to RPA. In some respects these two methods are orthogonal. RPA produces a partition of the data, exploits a soft preference analysis harmonized with consensus based considerations and integrates several levels of protection against outliers. For obtaining reliable results RPA needs to sample many times genuine models, but, as a consequence, is less sensitive to the choice of the inlier threshold (the input values of RPA are fixed per each dataset). On the contrary ILP-RansaCov enjoys a much more simple formulation based on consensus, it can deal with intersecting models and the preference aspects of the problem can be easily added as side information. The discrete nature of ILP-RansaCov guarantees that, in principle, sampling every genuine structure once is sufficient to retrieve an exact segmentation. The other side of the coin is that the inlier threshold is a more sensitive parameter, since a single inaccurate tentative structure in the hypothesis space can heavily affect the final result.

6

An unexpected application: a cryptographic attack

Fault attacks are among the most effective techniques to break real implementations of cryptographic algorithms. They usually require some kind of knowledge by the attacker on the effect of the faults on the target device, which in practice turns to be a poorly reliable information typically affected by uncertainty. This chapter is devoted to address this problem by softening the a-priori knowledge on the injection technique needed by the attacker in the context of Differential Fault Analysis (DFA). We conceive an original solution, named J-DFA, based on translating the stage of differential cryptanalysis of DFA attacks into terms of fitting multiple models to data corrupted by outliers. Specifically, we tailor the preference trick implemented by J-Linkage to the fault analysis. In order to show the effectiveness of J-DFA and its benefits in practical scenarios, we applied the technique under different attack conditions.

6.1 Differential Fault Analysis

The use of hardware faults to attack a cryptographic system, was originally presented by Boneh et al. in 1997 [11]. This attack was applied to recover the secret key of an RSA implementation. Subsequently, the idea of applying faults to attack implementations of cryptographic algorithms was extended to symmetric ciphers [9]. The main technique introduced against block ciphers is referred to as Differential Fault Analysis (DFA) and consists of analyzing the difference between correct and faulty ciphertexts in order to get information on the secret key. Several DFA attacks have been presented and successfully applied against symmetric encryption schemes and in particular against the AES algorithm [10, 40, 69, 79, 104]. Each one of these attacks relies on a fault model implied by the attacker. Depending on the target cryptosystem and the specific attack, the a-priori knowledge of the fault model by the attacker can fundamentally impact the effectiveness of the attack.

As explained in [60], from the point of view of the information theory, every fault provides information about the secret key. Such information depends on the precision of the injection technique but also on the knowledge that the attacker has on the induced effect. All DFA attacks work when all the specific instances of executed faults perfectly match the a-priori knowledge of the attacker about the possible effects of the injections. In practice, however, some more general situations should occur. Firstly, the attacker can be forced to consider a wider set of possible effects of the fault injections. Due to this uncertainty, the efficiency of the attack lowers as the information provided by each fault lowers too. In this case the attack still works, but the number of required faulty ciphertexts increases. Moreover, it is possible that the effects of some faults fall out of the considered set of models. In this case the attack either terminates with a wrong key or with no solution, because of the wrong a-priori hypothesis.

In this chapter we introduce a novel DFA approach, based on the application of J-Linkage, with the aim of increasing the robustness of the attack and softening the requirements on the a-priori knowledge by the attacker. Originally proposed for geometric model fitting in Computer Vision, J-Linkage is used here for the first time to derive a new DFA approach, called J-DFA. Thanks to the inherent properties of J-Linkage, J-DFA will result in a versatile tool that not only can be easily used to quickly replicate many classical DFA attacks, but also produces reliable solutions in a wider range of practical attack scenarios. In order to show the effectiveness of the proposed approach we apply J-DFA to the specific case of faults injected in the last round of AES and we successfully compare our results with the classical approach in different attack conditions.

The chapter is organized as follows. In Section 6.2 we briefly illustrate common DFA attacks against AES with particular attention to the details of a class of them which will be used as example. In Section 6.3 we describe our J-DFA technique and we explain how to map a specific known DFA attack on the J-Linkage settings. In Section 6.4 we show the results of several experiments and we highlight the benefits introduced by the new approach.

6.2 DFA against last round of AES

The vast majority of fault attacks presented against AES falls within the class of DFA, and are performed in two steps. The first stage requires to actively perturb the intermediate state of AES by faulting the execution and to collect the corresponding faulty ciphertext, together with the correct ciphertext (i.e. not-altered). The second stage applies techniques of *differential cryptanalysis* in order to derive information on the secret key from the pair of correct and faulty ciphertexts.

From an information theoretical point of view, the information about the injected fault translates in an equivalent amount of information about the secret key. In particular, the smallest the *model* is (namely, the smallest the set of possible faults is), the smallest the set of key candidates is.

As a reference, we describe the family of classical attacks based on the injection of the fault at the beginning of the last AES round. This class of attacks will be used throughout the chapter to detail the application of our approach and to compare the results with well known attacks.

The considered fault model includes any possible alteration of the AES state just before the SubBytes operation of the last round of AES. Since the MixColumns operation is not performed in the last round, each byte of the AES state affected by the fault can be considered independently. For sake of simplicity the following description focuses on the case where a single byte is affected per injection, without loss of generality. Indeed, due to the AES structure, it is easy to deduce the number of perturbed bytes by simply observing the pair of correct and faulty ciphertexts. Therefore, in case of multiple affected bytes, the attacker can either consider separately the bytes in the second stage of the attack, or just ignore the observations involving more than a single byte.

The first instance of this class of attacks was introduced by Giraud in [40]. In that case, the specific fault model considered for the injection was a single bit flip in the AES state at the beginning of the last round of AES. Note however that the same attack procedure can be trivially extended to different fault models.

According to Giraud's attack, the attacker can compute the corresponding byte of the last RoundKey by an exhaustive search in the following way.

Let (C, C^*) be an experiment, namely a pair of respectively correct and faulty ciphertexts generated on the same plaintext using the same key. Let us denote by $\hat{s}$ the byte where the fault occurs at the beginning of the last round, by c and c^* the corresponding bytes in C and C^* respectively, and by $\hat{k}$ the corresponding byte of the last RoundKey K^{10}.

By definition of AES, we have

$$c = \text{SubBytes}(\hat{s}) \oplus \hat{k} \tag{6.1}$$

and

$$c^* = \text{SubBytes}(\hat{s} \oplus \hat{\epsilon}) \oplus \hat{k} \tag{6.2}$$

where $\hat{\epsilon}$ denotes the injected fault.

In general, $\hat{\epsilon}$ can belong to any subset of the 255 possible faults that can be induced on a byte. In the specific case described in [40], $\hat{\epsilon}$ belongs to the following fault model

$$E = \{\texttt{0x01, 0x02, 0x04, 0x08,}\\ \texttt{0x10, 0x20, 0x40, 0x80}\}. \tag{6.3}$$

For all possible values k of the key byte, the attacker computes

$$\begin{aligned} s &= \text{SubBytes}^{-1}(c \oplus k) \\ s^* &= \text{SubBytes}^{-1}(c^* \oplus k) \end{aligned} \tag{6.4}$$

and subsequently

$$s \oplus s^* = \text{SubBytes}^{-1}(c \oplus k) \oplus \text{SubBytes}^{-1}(c^* \oplus k) = \epsilon. \tag{6.5}$$

For different experiments, the attacker checks whether ϵ satisfies the fault model or not, i.e. if it belongs to the set in Eq. (6.3). If this is the case, the value k is a possible candidate for $\hat{k}$, otherwise the specific k is not compatible with the assumed model and it is discarded. Observe that in a sense, each experiment (C, C^*) induces relationship between faults ϵ and corresponding key values k described by

$$f_{(C,C^*)}(k) = \text{SubBytes}^{-1}(c \oplus k) \oplus \text{SubBytes}^{-1}(c^* \oplus k). \tag{6.6}$$

After testing all values, the set of possible key byte candidates is downsized with respect to the initial set of 256 possible elements. In other words, the attack discards all the candidates of the byte of the key which correspond to a fault that is not included among the faults considered by the model. The size of the resulting set of candidates depends on the size of the fault model.

In the case of the model described in [40], on average only 8 candidates are left. Therefore two experiments on average are enough to univocally identify the correct value of the byte of the key. Namely, with a second pair of correct and faulty ciphertexts, with fault induced in the same byte, the attacker obtains another set of candidates for k and the intersection of this set with the first one contains the correct value for the key.

In general, the procedure must be iterated until only a single candidate for the byte of the key is left in the intersection. In particular, the more precise the fault injection is, the less experiments are necessary and vice versa.

This process must be repeated independently for each of the 16 bytes of the key. Note that the analysis allows to retrieve the last RoundKey and that the secret key can be obtained by simply applying the inverse KeySchedule operation on it.

It is clear from the description that the knowledge of the fault model by the attacker is of fundamental importance for the success of the attack. In particular it is worth underlining the fact that including all the 255 possible faults in the set E is not a viable option for the attacker. In fact such model leads in never discarding key candidates and then

never converging to a solution for the byte of the key. Therefore the attacker is forced to reduce the set of considered faults, by characterizing the injection technique on the specific target device, in the same way the choice done in [40] has been motivated. Such knowledge on the fault model must be obtained a-priori by the attacker and it cannot be derived from observations involving the unknown secret key.

In practice, the most critical aspect of the classical DFA approach, is the fundamental need of guaranteeing that all the considered experiments have been generated by faults belonging to the fault model assumed by the attacker. The presence of few (even a single) experiments that fall out of the model compromises the overall success of the attack. This is due to the fact that every single experiment has the power of discarding the correct candidate for the key, and this condition cannot be recovered by other experiments. In real setups it is hard for the attacker to completely prevent the existence of such bad experiments.

6.3 J-DFA: J-Linkage for DFA

In this section we describe a novel technique for DFA called J-DFA, with the aim of softening the requirement on the a-priori knowledge needed by the attacker to exploit the faults in practice. The main idea of our approach is to map the stage of differential cryptanalysis of DFA attacks into the problem of fitting multiple models to data corrupted by outliers.

Like all classical DFA attacks, J-DFA is performed in two stages. The first stage consists in actively manipulating the target device in order to corrupt the computations and collect a set X of experimental data. Each data $x \in X$ is an experiment, i.e. a pair of correct and faulty ciphertexts (C, C^*) generated on the same plaintext using the same key. F represents the set of all the possible effects of the fault that may happen as a consequence of the fault injection. In practice, among all the possible faults, only a subset of them occurs. We denote with $E \subseteq F$ such subset. The faults belonging to E vary depending on the specific technique used for the fault injection (e.g. glitches on clock, laser beams, ...) and on the target device. This stage works exactly in the same way as for classical DFA and it may include a step where the meaningful experimental data are extracted among all the experiments. Of course such a step only allows to extract experiments that produce a peculiar pattern which can be identified by comparing the correct and faulty ciphertext. For instance, a single affected byte at the beginning of the last round results in a single faulted byte in the ciphertext. Therefore such data can be easily distinguished from faults injected in rounds earlier than the last one. Still, this step does not exclude outliers (as defined, faults not belonging to the model considered by the attacker) that cannot be identified by simply observing the corrupted ciphertexts.

The second stage is the application of techniques of differential cryptanalysis in order to translate the information obtained on the ciphertexts into information on (a portion of) the secret key κ. In this second stage the J-Linkage tool is introduced.

In general the DFA is based on the fact that for each possible fault, every experiment (C, C^*) is *compatible* only with a (small) set of values of the involved portion of the key. In the classical view, this fact is used to remove inconsistent values from the set of the possible candidates for that portion of the key. The J-DFA approach aims at relaxing this hypothesis, by replacing the concept of compatible-incompatible key values (with regards to a specific experiment) with the concept of key values *voted* by a specific experiment.

In the J-DFA view, each experiment can be represented in a conceptual space as characteristic function of the pair(s) (fault, key) = (ϵ, k) preferred by that experiment. Such pairs (fault, key) = (ϵ, k) represent putative models for the J-Linkage clustering technique. The clustering algorithm aggregates together experiments with similar preferences. At the end the experiments are split in clusters, where each cluster refers to one (or more) specific pairs (ϵ, k). Highly-populated clusters include experiments with similar votes, while experiments that are poorly compatible with others are left in lowly-populated clusters.

In a successful J-DFA the correct candidate for the portion of the key is the one corresponding to the most-populated clusters and thus the one that has been voted the most.

In detail, the J-DFA technique involves a sequence of five distinct steps which are also represented in Figure 6.1:

- *Mapping*: Each experiment $x = (C, C^*)$ induces a relationship associating faults ϵ and corresponding key values $k \in K$ which we denotes as f_x. K represents the set of all the possible values for the portion of the key involved in the attack (i.e. 256 values if one byte of the key is involved).
- *Space of hypotheses*: The attacker selects an hypothesized fault model $H \subseteq F$ including only the faults considered likely to occur. This selection is based on the assumption derived from the injection technique and the target device. From the set H, the whole *space of hypotheses* is generated, that is $H \times K$, the Cartesian product between H and K. The space of the hypotheses includes all the pairs (ϵ, k) that are considered possible by the attacker. Such pairs correspond to putative models within the J-Linkage framework. Please observe here the slightly misleading terminology: the term *model* is used to indicate both J-Linkage *putative models* and DFA *fault models*. It is worth noting that most of the classical DFA only consider the case where the fault model is known a-priori, namely $H = E$. Instead in J-DFA we are not making such assumption.

- *J-Linkage conceptual representation*: The preferences are built by assuming that a model (ϵ, k) is preferred by an experiment x if the particular mapping of x associates ϵ and the key k, namely $f_x(k) = \epsilon$. The preference matrix is implemented as a matrix with $m = |H| \times |K|$ columns and $n = |X|$ rows. Recall that each column represents a model (ϵ, k) whereas each row indicates for each experiment x which are the preferred models.
- *J-Linkage clustering*: Experiments are split in clusters by J-Linkage. Each cluster U_i is representative of one or more models (ϵ, k). Experiments belonging to the same cluster means that they have at least one preference (i.e. model) in common. Vice versa experiments split in different clusters means that they did not have any common preference. Note that even if the experiments can be split in different clusters representative of different pairs (ϵ, k), still these models can share a common key (but referring to different faults). More formally we say that a cluster U_i is *k-compatible*, if it exists at least one hypothesized fault model ϵ associated with k by all the experiments in U_i.
- *Ranking of keys*: The candidates of the key $k \in K$ are ranked based on the size of the clusters U_i obtained from the previous step, by defining for each k a weight

$$w(k) = \sum_{U_i \;\; k-\text{compatible}} |U_i|. \tag{6.7}$$

  The recovered key κ is the one with highest weight, i.e.

$$\kappa = \arg\max w. \tag{6.8}$$

Under the correct hypothesis on the fault model, J-DFA guarantees that the correct key is among the preferences of the most populated clusters. Similarly to classical DFA, when a sufficient amount of experiments are provided to J-DFA, the most preferred candidate is the correct key. However, differently from classical DFA, the clustering approach of J-Linkage makes J-DFA a *robust* technique and this is the main rationale behind the interest for J-Linkage applied to DFA.

We now explain this concept more in detail. First of all J-DFA is robust against *outliers*. In the context of fault attacks an outlier can be defined as an experiment which has been produced by a fault that does not belong to the fault model H assumed by the attacker. The effectiveness of classical DFA is heavily compromised in presence of outliers, leading to either no solutions or a wrong solution for the key. In classical DFA the fault model H cannot be simply set equal to F to avoid the presence of any outlier, because such condition would never converge to a solution. Instead, J-DFA nicely manages the outliers and it leads to the correct solution provided that it is fed with enough

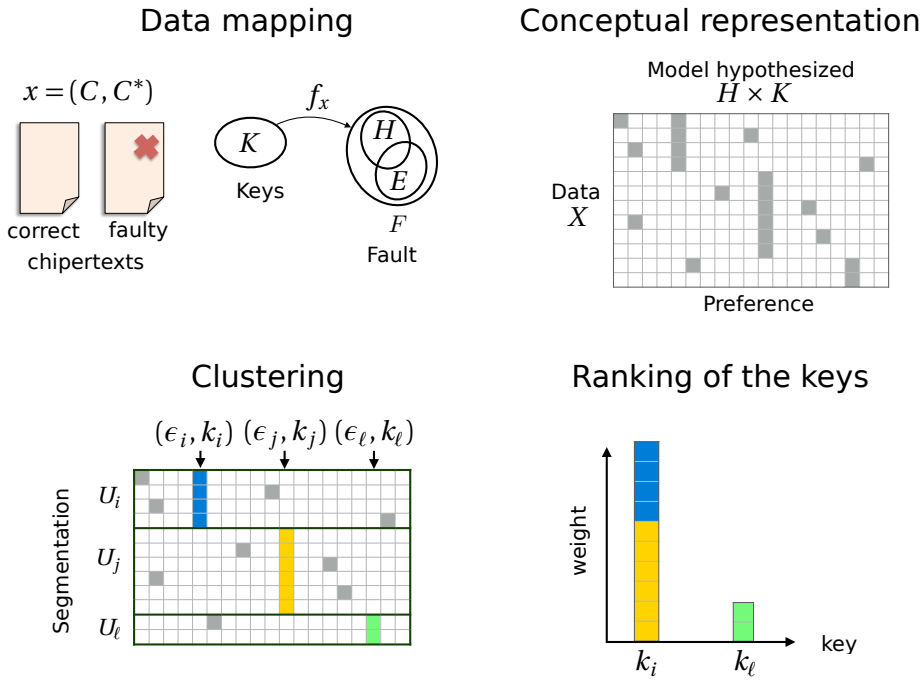


Fig. 6.1: J-DFA in a nutshell. *Data mapping*: An experiment $x = (C, C^*) \in X$ defines a map f_x between the possible key values K and the set of possible faults F. E represents the faults that really occur in the experiments, H consists in the faults hypothesized by the attacker. *Conceptual representation*: The preference matrix is built, representing every datum by the votes (gray cells) it grants to the set of putative models $(\epsilon, k) \in H \times K$. *Clustering*: J-Linkage segments the preference matrix in clusters U_i, U_j, U_ℓ (data are arranged such that consecutive data belong to the same cluster for sake of visualization only). It is hence possible to extract the most preferred models per cluster $(\epsilon_i, k_i), (\epsilon_j, k_i), (\epsilon_\ell, k_\ell)$. Note that the same key may appear as preferred by several clusters. *Ranking of the keys*: Finally votes are aggregated with respect to keys and the most preferred one is retained. (Figure best viewed in color)

coherent experiments. Furthermore J-DFA manages the case with $H = F$ (i.e. all the possible faults are valid) without any special adaptation. In this way, J-DFA is also robust in the choice of the fault model H. Finally, the linkage formulation does not require to know in advance the number of models for the faults that the specific injection technique in place generates. This means that in the extreme case in which the attacker has no knowledge a-priori, the fault model can simply be assumed to $H = F$. Even in this

scenario, which is very interesting in practice, J-DFA successfully leads to the correct solution. We will detail the benefits of this property of J-DFA in the next section.

6.4 Evaluation results

In this section we show the results of the application of the J-DFA technique under different attack conditions. All the presented attacks focus on faults injected at the beginning of the last round of AES. As explained in section 6.2, for the sake of simplicity we consider experiments affecting only a single byte at a time. Different conditions for the attack means that case by case we evaluate different hypothesized fault models H assumed by the attacker, while keeping fixed the set of actual faults E.

Since J-DFA relies on the same approaches of classical DFA for what regards the injection stage, we focus on the cryptanalysis stage for our evaluation. For this reason, similarly to many DFA works described in literature, we simply generated the experiments through simulations. Namely, we used a software implementation of an unprotected AES modified in order to be able to induce the desired fault at the beginning of the last round. In this way we easily collected a large amount of couples (C, C^*), to feed J-DFA.

For the same reason we do not consider any countermeasure against faults in our analysis. In fact most countermeasures aim at making hard for the attacker to successfully apply the first stage of the DFA, which consists in collecting the observations. Examples of such kind of countermeasures are shielding, sensors, redundancy, including multiple executions of the same operation (see [51]). Still novel DFA attacks are of interest because such countermeasures are not able to cover every possible progress in the injection techniques. Like many previous works on DFA, we focus only on the differential cryptanalysis stage and take for granted that the attacker is able to obtain some observations, since J-DFA applies to the second stage of the attack.

We first show the results of J-DFA applied to the very same conditions described in [40]. Then we extend the analysis to different practical attack scenarios.

6.4.1 J-DFA with profiling

In order to show in detail how J-DFA can be concretely used, we first apply our approach in the classical DFA scenario. DFA attacks assume that the attacker has some kind of a-priori knowledge of the effects produced by the injection technique on the target device. This translates in the fact that the fault model H considered by the attacker perfectly matches the set of faults E that occurs in practice, namely $H = E$.

Giraud's Attack

As a running example, the J-DFA technique is applied to Giraud's DFA against AES using the same fault model assumed in [40]. In this case the relationship that binds fault values ϵ to key values $k \in K$ is the same used in [40] and f_x coincides with the function reported in Eq. (6.6). Assuming 8 possible faults in H, the space of hypothesis provided to J-DFA consists of 8×256 available models, one for each combination of (ϵ, k), with $\epsilon \in H$ and $k \in K$.

In order to validate the effectiveness of J-DFA, we applied the attack as described above, by feeding it with one experiment at a time, until a single candidate for the key is found. We repeated the test 100 times on different experiment datasets, obtaining in all the cases the correct candidate κ for the key. On average, 2.1 experiments are necessary to obtain only a single candidate. This result confirms the value provided in [40] and shows that J-DFA is as effective as classical DFA when applied in the same conditions.

This setup, like for the original DFA, is not really computationally intensive. When few experiments are used (i.e. < 20), our implementation of J-DFA performs each attack in a negligible time (i.e. < 1 s). Furthermore, when $H = E$ and the number of experiments is comparable with (or lower than) the size of the set H, the attack does not really benefit of the clustering (since likely each cluster will be populated only by a single experiment).

Extended fault models

Although [40] only explicitly considers the fault model described in (6.3), the same attack can be trivially extended to other fault models injected at the beginning of the last round of AES. In the same way J-DFA applies by simply changing the set of hypothesis H accordingly to such extended models. Once setup J-DFA, several attacks with different fault models can be easily performed, since they all share the same mapping function that binds faults and key values. Indeed such map only depends on the point of injection of the fault, namely the beginning of the last round. With this regards, J-DFA can be conveniently used as a tool for the analysis of classical DFA in different conditions. In order to further validate the J-DFA, we tested it under some representative conditions.

A particular scenario could be case in which the attacker is able to inject always the fault in a fixed position, for instance a bit flip on the least significant bit. In this case the attacker has a perfect a-priori knowledge of the fault model, which is $H = E = \{\texttt{0x01}\}$. The hypothesis space consists of the 256 possible values of the key associated with that single possible fault $\epsilon = \texttt{0x01}$. Similarly, different scenarios could consider the attacker able to fault only a fixed portion of the byte. In case the fault affects only the least significant half of the byte, the fault model would include all the 15 possible combinations

of 4 bits from 0x01 up to 0x0F, where 0x00 is excluded because it represents no-fault. Otherwise, the case in which the attacker is able to affect by fault all the bits except the most significant bit, the fault model would comprise all the 127 possible combinations of 7 bits (again the value 0x00 is excluded). Under even different attack conditions the injection could either affect a single bit or a couple of bits. In this case the fault model is represented by all the combinations of 8 bits with Hamming weight equals 1 (i.e. 8 faults, the same assumed by Giraud) or Hamming weight equals 2 (i.e. 28 faults). In total the fault model would include 36 different faults.

J-DFA has been applied in all the attack conditions listed above, 100 times each. Table 6.1 reports how many experiments are needed on average to obtain the correct value κ as single candidate.

Hypothesized fault model ($H = E$)	AVG number of required experiments
{0x01}	1.9
Giraud's fault model of Eq. (6.3)	2.1
{0x01, 0x02, ..., 0x0F}	2.3
{0x01, 0x02, ..., ..., 0x7F}	210.3
HammingWeight(ϵ) = {1, 2}	13.4

Table 6.1: Average number of experiments (over 100 trials) required in order to obtain the correct key value under different fault models.

It is worth noting that all these tests assume some a-priori knowledge in order to have $H = E$, thus J-DFA can be considered as a tool to quickly replicate classical attacks.

6.4.2 J-DFA without profiling

Besides being able to replicate classical DFA, J-DFA becomes particularly interesting in cases where the a-priori knowledge of the fault model is poor or even completely absent. Indeed, thanks to the inherent robustness of the J-Linkage clustering technique introduced in section 1.3, J-DFA can be applied in a wider range of practical conditions compared to the classical attacks. In particular, the tool converges to the correct key even if the fault model H does not perfectly match the actual set of injected faults E, provided that $H \cap E \neq \emptyset$ and that enough experiments are available.

In order to show the robustness of J-DFA in practice, we performed several attacks where the set of injected faults is fixed for all the tests while the fault model varies. For all the tests the injected faults are the 8 single-bit-flip considered in [40], i.e. E is defined

Hypothesized fault model ($H \subset E$)	AVG number of required experiments
{0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40}	2.2
{0x01, 0x02, 0x04, 0x08, 0x10, 0x20}	2.5
{0x01, 0x02, 0x04, 0x08, 0x10}	3.5
{0x01, 0x02, 0x04, 0x08}	3.6
{0x01, 0x02, 0x04}	4.8
{0x01, 0x02}	6.3
{0x01}	9.5

Table 6.2: Average number of experiments (over 100 trials) required in order to obtain the correct key value under different hypothesized fault models belonging to the case $H \subset E$.

as in Eq. (6.3). The set of faults assumed by the attacker, namely H, varies starting from a single fault (i.e. $H = \{0x01\}$), up to covering all the possible 255 faults on a single byte. A total of 255 different conditions are tested, differing each other for the amount of faults included in the fault model. Note that for the purpose of the tests, the faults in H are selected in order to maximize the intersection $E \cap H$.

The setup of the tests just described leads to 3 different kinds of conditions:

(i) $H = E$. This condition represents the case described in [40] and explored in section 6.4.1. The attacker perfectly knows a-priori the set of possible faults injected in practice.

(ii) $H \subset E$. This condition represents the case in which the attacker underestimates the faults that occur in practice and therefore she assumes a fault model that includes only some of the actual faults, but not all.

(iii) $H \supset E$. This condition represents the case in which the attacker overestimates the faults that occur in practice and therefore the fault model includes some faults that never occur in practice.

We do not treat the case $H \cap E = \emptyset$ since in this case no meaningful information can be extracted from the attack and then the method fails. The more general condition $H \cap E \neq \emptyset$ follows from (ii) or (iii). The first case has already been explored in section 6.4.1. Tables 6.2 and 6.3 show the amount of experiments needed on average to get from J-DFA the correct key κ as single candidate for different sizes of the hypothesized fault model H (belonging to the case $H \subset E$ or $H \supset E$ respectively). Each value is averaged over 100 trials on different experiment datasets.

The results show that J-DFA converges to the correct solution in all conditions. In particular it successfully works even in case the whole set of possible faults is assumed

Hypothesized fault model ($H \supset E$)	AVG number of required experiments
{0x01, 0x02, ..., 0x10}	2.5
{0x01, 0x02, ..., ..., 0x20}	4.1
{0x01, 0x02, ..., ..., ..., 0x40}	8.4
{0x01, 0x02, ..., ..., ..., ..., 0x60}	10.7
{0x01, 0x02, ..., ..., ..., ..., ..., 0x80}	11.5
{0x01, 0x02, ..., ..., ..., ..., ..., ..., 0xA0}	13.4
{0x01, 0x02, ..., ..., ..., ..., ..., ..., ..., 0xFF}	16.9

Table 6.3: Average number of experiments (over 100 trials) required in order to obtain the correct key value under different hypothesized fault models belonging to the case $H \supset E$.

in the fault model (i.e. $H = F$, described in the last row of Table 6.3). This case is particularly interesting in practice because it does not require any a-priori knowledge by the attacker on the effects of the injection technique. We recall that the classical approach of DFA cannot manage such condition. In practical scenario the attacker could successfully attack an AES implementation without the need of any characterization of the injection technique. She can generate several experiments (couples of correct and faulty ciphertexts). Then she extracts only the ones involving a single byte of the ciphertext (where likely the injection technique did affect only a single byte in the last round), independently from the kind of faults. Finally she applies J-DFA with $H = F$ and obtains the correct key.

The results of the tests also show that the most efficient (in term of number of experiments) condition for the attacker is the case where she perfectly predicts the set of faults that can occur. And this is the condition implicitly assumed in most of classical DFA described in literature.

Another information that the results exhibit is how the efficiency of the attack decreases when the model H and the reality E differ. There are two trends:

- When $H \subset E$ the amount of needed experiments grows linearly, due to the fact that among all the experiments provided to J-DFA some of them are generated by faults not included in H and then they do not provide information about the key. Of course the more occurring faults do not belong to H, the less experiments are meaningful among all.
- When $H \supset E$ the amount of needed experiments grows sub-linearly, due to the fact that even if all the experiments do provide some information about the correct key, such level of information per experiment decreases.

This means that in practice the best choice for the attacker is to place herself in the case $H = E$, but this requires perfect knowledge on the effects of the injection technique. When the attacker has some uncertainty about the injection, it is preferable to overestimate H rather than underestimate it. And since in a practical scenario the attacker may not know in advance the number of different occurring faults, a larger H is a safer choice, up to the point to simply use $H = F$ and do not rely on any prediction on the fault model.

It is worth noting that, due to its robustness, J-DFA requires different amounts of experiments in different conditions, but always leads to the correct solution. Instead, classical DFA in case of $H \subset E$ tends to produce no solution (i.e. none of the key candidates are compatible). This effect becomes stronger (i.e. more probable) when the difference between H and E increases. When $H \supset E$, classical DFA are still able to converge to the correct key provided that enough experiments are available. But they can suddenly reject the correct value of the key when outliers come into the picture. It is fundamental for a successful DFA to have the guarantee that none of the experiments falls out of the assumed fault model H. In practical scenarios it can be hard to ensure such condition, since often different faults have different probabilities to occur, but rarely the attacker is completely certain about the effects of the faults. And we recall the fact that these outliers cannot be discarded by simply observing the couple (C, C^*). The usual way to manage such practical condition, besides having a strong a-priori knowledge, is to enlarge the fault model H as much as possible. The concrete issue with this approach in classical DFA is twofold. First, a wide fault model H requires a number of experiments that grows with the size of H, which increases the probability to include an outlier. Second, if any outlier is present among the experiments, with a wide fault model becomes more probable to obtain a wrong key candidate rather than just converge to no solution. And since the attacker can only test the correctness of the whole 16 bytes of the key and not each byte separately, even few wrong candidates per byte can lead to an unfeasible search of the correct key. This explains why is so desirable the property of robustness to the outliers that J-DFA brings in practical attack scenarios.

Besides the most preferred candidates for the key, J-DFA also identifies the most preferred models (ϵ, k). This means that it is possible to understand from J-DFA the actual set of faults E that occurred in the experiments used for the attack. Such information may be used to enhance the overall efficiency of the attack, for instance by getting E while attacking the first byte with a wide fault model and then set the refined model (i.e. $H = E$) for the other bytes (assuming that the injection technique affects all the bytes in the same way). Otherwise that property may be exploited to use J-DFA as an analysis tool (rather than for attacks), to characterize the fault models for different injection techniques.

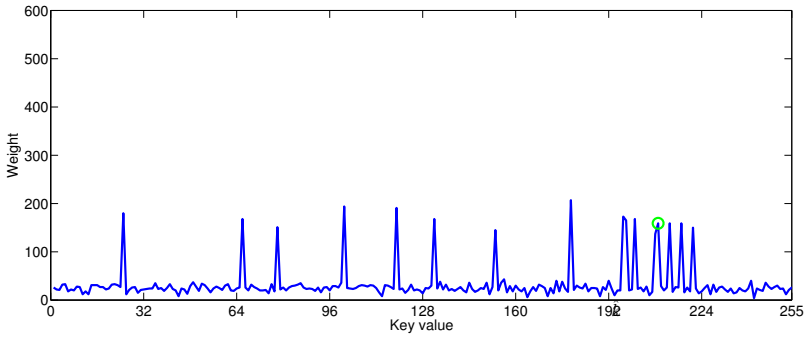


Fig. 6.2: Preferences (weights) obtained for each candidate of the key, using 4000 experiments. The correct key κ is dotted in green.

Worst case scenario

Like the classical DFA, also J-DFA becomes ineffective when all the faults occur (i.e. $E = F$) with the same probability. This is due to the fact that the injection of a fault does not provide any kind of information and consequently the attack is useless.

That said, if there is even a small bias in the effects of the injection technique, for instance at least one of the faults is slightly less probable than the others, then J-DFA can be still applied. In this demanding scenario however the computational overload becomes considerable and even using a large amount of experiments, the results can still be affected by uncertainty. This because all the experiments are compatible with high probability with all the faults but one, and consequently the greedy segmentation used by J-Linkage fails in finding few predominant keys.

As a reference, we applied J-DFA in the case where all the possible effects occur with the same frequency except one, namely $\epsilon = 0$xFF never occurs. We set the space of the hypotheses $H = E$, then assuming that the attacker knows a-priori which is the fault that does not occur. Our implementation of J-DFA fed with 4000 experiments, takes about 23 hours to provide the solution, and it returns 16 candidates for the key that are compatible with all the experiments. Figure 6.2 shows the preferences obtained for each candidate of the key. There are 16 peaks, among which the correct key κ, that have similar weights (much higher than all the others). The test on 4000 experiments is shown here to provide an indication about how heavy becomes the computation to converge to a single solution. With less than 4000 experiments, the compatible candidates are still much more than 16. Instead we did not tried by further increasing the number of experiments because we already consider 23 hours of computation a substantial effort. Rather than pursuing in that way, we look for an alternative approach to tackle this extreme case.

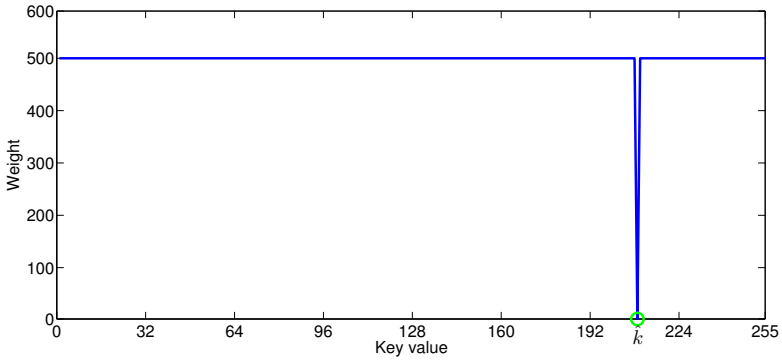


Fig. 6.3: Preferences (weights) obtained for each candidate of the key using $\tilde{P}$ as preference matrix instead of P, using 1000 experiments. The correct key κ is dotted in green.

Therefore in this case, rather than considering the preferences of the experiments, it is convenient to reverse the point of view and consider the negation of the preference matrix:

$$\tilde{P}(i,j) = \begin{cases} 0 & \text{if } x_i \text{ is explained by the } j\text{-th model} \\ 1 & \text{otherwise.} \end{cases} . \tag{6.9}$$

The rationale is that by feeding J-Linkage with the negated preferences, the tool will lead to a wrong set of extracted keys which collects the votes of the majority of data, but the correct key can be singled out as the one that does not receive any vote. We propose to apply J-DFA on $\tilde{P}$ and to change Equation (6.8) in

$$\kappa = \arg\min w. \tag{6.10}$$

It is worth noting that in this complemented case we are using J-Linkage in a non conventional way, from the perspective of the clustering techniques. In fact it is highly probable that J-Linkage will not separate data, but rather it will put all the experiments in a single cluster. Still the procedure is meaningful for the fault attack application, because the correct key is among the models that are not preferred by any cluster.

We applied J-DFA on $\tilde{P}$ in the same conditions of the previous test: $H = E$ including all the possible faults with the same frequency except one. In this case we fed J-DFA with 1000 experiments, which required only 3200 seconds of computation.

Figure 6.3 shows the preferences for each candidate. As explained before, the result must be interpreted differently; in fact we expect the correct candidate to be among the least preferred keys. There are 255 candidates with high level of preference and only a

single candidate which has a much lower level of preference equals to zero. This candidate coincides with the correct key κ, suggesting that it is more efficient to derive information about the secret key by observing models that are *not compatible* with the experiments, rather than the preferred ones.

The test reveals that using $\tilde{P}$ is a viable solution to successfully tackle some conditions that are usually hard for the attack.

6.5 Final remarks

In this chapter we have presented J-DFA: a novel approach for DFA which exploits a robust clustering algorithm tailored to fault analysis. We argue that the benefit yielded by J-DFA is twofold. First, it is a versatile tool that can be easily used to quickly replicate many classical DFA attacks unified in a common framework. A peculiar result of J-DFA is that, besides the preferred candidate for the key, it also provides the preferred models for the fault. This is a quite remarkable ability because it furnishes precious information which can be used to analyze, compare and characterize different specific injection techniques on different devices.

The second benefit is that, thanks to its robustness, J-DFA produces reliable solutions in a wider range of practical scenarios, even if the a-priori knowledge of the attacker is poor or completely absent.

Even if we deal only with faults injected in the last round of AES, our approach could be extended to different positions for the faults (e.g. the attack described in [79]) or even different algorithms, taking advantage of the generality of the J-Linkage conceptual representation. From the theoretical point of view, the only step of the procedure that requires to be adapted is the mapping, which must represent the different injection position or the different algorithm. In practice, attacking a different step of AES (e.g. the second-last round) may lead to a huge amount of models to be considered. One possibility to tackle this, could be to exploit the fact that J-DFA is able to converge to a solution even if the fault model does not include all the faults that occur, then intentionally keeping the fault model limited. However it is worth noting that in a practical scenario, considering an injection point different than the last round (e.g. the second-last round) is of interest only when the fault cannot be injected in the last round (e.g. a countermeasure protecting only the last round but not the second-last). Due to the way we constructed J-DFA and its inherent robustness to outliers, the set of fault models can even include and mix fault generated by injections in different stages of the execution (i.e. mixing different mapping), such as faults coming from the last round and from the second-last round.

7

Conclusions

In this thesis we have described several improvements to the current state of the art in the context of geometric multi-model fitting, making some steps towards the tantalizing prospect of simple and practical methods able to automatically recover the geometric structures hidden in visual data. In particular we elaborated the preference approach in term of performances and robustness, building on both the representation and the segmentation steps.

More in details, at first we concentrated on the conceptual representation of data: we investigated the "*preference trick*" to propose a continuous relaxation of the binary approach followed by J-linkage. In this way we lifted the problem of structures recovery in the Tanimoto space, providing a more general framework, in which we were able to integrate the use of M-estimators to robustly depict data preferences. Density-based techniques have been employed to analyze the geometry of the Tanimoto space, showing that points belonging to the same model are clustered in high density region, whereas outliers can be characterized as the most separated points. We suggested how to exploit this properties to guide random sampling towards promising tentative structures and to downweight outliers, which indeed is a delicate task from which it depends the effectiveness of the conceptual representation.

A related aspect, intrinsically linked to sampling, is the specification of the inlier threshold. In practice, if a tight estimation of the scale is not available, it is better to rely on the use of soft-descenders, and to compensate the looseness on the scale putting more effort on sampling, so that many pure model hypotheses are sampled. RPA can be used to achieve the desire segmentation of the data in this case. On the contrary, if the inlier threshold can be easily specified, it is enough to ensure that one single hypothesis per ground truth model is selected, in this case, ILP-RANSACOV or T-Linkage are able to accurately recover the models.

Provided that the solution space is adequately approximated, the preference trick enjoys an extreme versatility: several kinds of problems – not necessarily coming form Computer Vision, as the cryptographic application described in the last chapter – can be fruitfully framed in a common preference analysis framework.
This flexibility furnishes auspicious basis for further advancements. For example, beside applying preference analysis to solve the "classical" multi-model fitting problem, where all structures are instances of the same model, it would be interesting to adventure into the area of hybrid multi-model fitting, in which multiple instances of different models (e.g. homography vs. fundamental matrix) are sought.

As the segmentation is concerned, three main directions have been explored: hierarchical clustering, spectral analysis and set cover. Having recognized the chicken-and-egg recursive nature of the multi-model fitting problem, in all these three formulations we attempted to integrate the consensus and preference perspectives, exploiting as much as possible the strengths of these approaches while trying to overcome the limitations of both.

In first instance preference analysis was performed through agglomerative clustering in Tanimoto space. This clustering scheme has the merit of automatically discover the number of structures hidden in the data, furthermore it treats rogue points as micro-cluster that, in turn, can be pruned in a probabilistic framework where the reliability of a structure is measured in term of its randomness. The resulting algorithm enjoys a straightforward implementation. In addition only a global scale is required, therefore, if consensus clustering is integrated in this approach, it is possible to estimate this parameter given a proper interval search. Thank to these features T-Linkage is ideal for multiple structure recovery "in the wild" when minimal to none prior knowledge of the data is available.

The second line of investigation concentrates on partitional clustering and explicitly erects the bridge between consensus and preferences approaches. Following the tread of spectral clustering, we ended up to study the connections of this algorithm with low-rank and sparse approximation techniques, which recently sprouted out in data mining literature. In particular we conceive a robust version of spectral clustering, characterizing outliers as sparse vectors in Tanimoto space and using symmetric Nonnegative matrix factorization. Interestingly this technique can be interpreted as a dimensionality reduction of the preference space in which the data are projected on the directions corresponding to the structures that explains better the data. Once these models have been recovered, the chicken-and-egg dilemma is disentangled and the multi-model fitting problem is reduced to many single-fitting problems that can be solved, with the help of robust statistics, maximizing consensus. In practice RPA is a more ductile technique than T-Linkage and can deal with situations where tuning the inlier threshold

may become a tricky problem or a global scale is not reliable – e.g. structures with different noise levels. This comes at the cost of specifying in advance the number of desired models; it will be interesting to explore in future work if this kind of information can be derived by exploiting low rank-estimation techniques.

The interplay between consensus and preference reappeared in the coverage formulation, where an orthogonal strategy, rooted in the discrete setup of set-cover, is explored. Reverting the order followed in RPA, the main focus in this setup is on maximizing consensus, whereas the local properties of points captured by the Tanimoto embedding are profitably integrated as side information. The main advantage is the departure from the segmentation paradigm based on partition. In this way it is possible to revisit in a common unifying framework several classical multi-model fitting algorithms and to handle intersecting multiple structures and outliers in a sound manner.

All the proposed methods have been validated and compared with other state-of-the-art techniques on several multi-model fitting problems on both synthetic and real public datasets – including geometric primitive fitting (e.g. line fitting; circle fitting; 3D plane fitting), multi-body segmentation, plane segmentation, and video motion segmentation – providing accurate and convincing results.

Publications

The following papers, which contain material used in this thesis, have been published.

1. L. Magri and A. Fusiello. T-Linkage: a continuous relaxation of J-Linkage for multi model fitting. In *Conf. on Computer Vision and Pattern Recognition*, pp. 3954–3961, 2014.
2. L. Magri and A. Fusiello. Density based analysis in Tanimoto space for multi-model fitting. In *International Conf. on Image Analysis and Processing*, 2015.
3. L. Magri and A. Fusiello. Scale estimation in multiple models fitting via Consensus Clustering. In *International Conf. on Computer Analysis of Images and Patterns*, 2015.
4. L. Magri and A. Fusiello. Robust Multiple Model Fitting with Preference Analysis and Low-rank Approximation. In *British Machine Vision Conf.*, 2015.
5. L. Magri, S. Mella, F. Melzani, P. Fragneto and B. Rossi. J-DFA: a Novel Approach for Robust Differential Fault Analysis. In *Fault Diagnosis and Tolerance in Cryptography*, 2015.
6. F. Arrigoni, L. Magri, B. Rossi and P. Fragneto, A. Fusiello. Robust Absolute Rotation Estimation via Low-Rank and Sparse Matrix Decomposition. In *International Conf. on 3D Vision*, pp. 491–498, 2014.

I am extremely grateful to my advisor Andrea which is sincerely acknowledged together with my coauthors and colleagues in Figure 7.1. I need to thank many other people for they fundamental support and this will be done in a more adequate context.

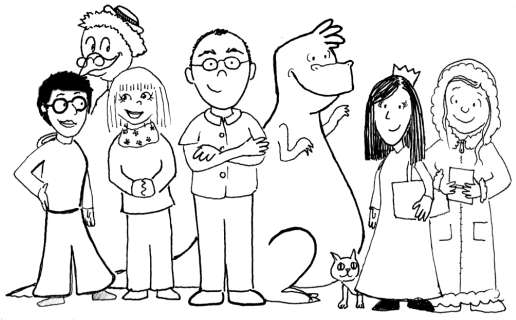


Fig. 7.1: Thanks!

A

M-estimators

This appendix is devoted to briefly sketch out the main ideas at the root of robust M-estimators. Standard least-squares methods estimate a parametric structure by solving the following optimization problem

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} \mathrm{err}_{\mu}(x_i, \theta)^2. \tag{A.1}$$

Since solving (A.1) turns to be unstable if there are gross outliers present in the data, M-estimators have been proposed [46] attempting to reduce the effect of outliers by replacing the squared residuals errors in (A.1) by another function of the residuals:

$$\min_{\theta \in \Theta} \sum_{i} \rho\left(\mathrm{err}_{\mu}(x_i, \theta)\right). \tag{A.2}$$

provided that $\rho : \mathbb{R} \to \mathbb{R}$ is a symmetric, positive-definite subquadratically growing loss function with a unique minimum at zero. In addition $\rho(|u|)$ should be monotonically nondecreasing with increasing $|u|$.

Let $\psi = \rho'$ the derivative of the robust loss function, ψ is called *influence function* [43] and intuitively measures the change in an estimate caused by insertion of outlying point as a function of the distance of the data from the uncorrupted estimate. For instance, the influence function of the least squares estimator is simply proportional to the distance of the point from the estimate. In real applications the variance of residuals has to be taken into account. For this reason, instead of working directly with $\mathrm{err}_{\mu}(x_i, \theta)$, residual are properly rescaled:

$$r_i = \frac{\mathrm{err}_{\mu}(x_i, \theta)}{\tau \sigma_n}, \tag{A.3}$$

σ_n is an estimate of the standard deviation of the error term and τ is a default tuning constants which gives coefficient estimates that are approximately 95% as statistically

efficient as the ordinary least-squares estimates, provided the response has a normal distribution with no outliers. Decreasing the tuning constant increases the downweight assigned to large residuals, on the contrary increasing the tuning constant decreases the downweight assigned to large residuals. Equation (A.2) becomes

$$\min_{\theta \in \Theta} \sum_i \rho(r_i). \tag{A.4}$$

If $\theta = (\theta_1, \ldots, \theta_p)^\top \in \mathbb{R}^p$ is a p-dimensional vector, for all $j = 1, \ldots, p$, the solution satisfies

$$\sum_i \psi(r_i) \frac{\partial r_i}{\partial \theta_j} = 0. \tag{A.5}$$

A more convenient form can be derived introducing a *weight function* $w \colon \mathbb{R} \to \mathbb{R}$ such that

$$w(u) = \frac{\psi(u)}{u}. \tag{A.6}$$

Thank to w we obtain

$$\sum_i w(r_i) \frac{\partial r_i}{\partial \theta_j} r_i = 0. \tag{A.7}$$

which leads to a system of p equations that can be solved by a process known as *iteratively reweighted least squares* (IRLS) [45]. Given an initial guess of θ, this procedure alternates between two steps: calculating weights $w_i = w(r_i, \theta/\sigma_i)$ using the current estimate of θ and solving (A.7) to approximate a new θ with the weights fixed.

M-estimators can be categorized into three types according to the behavior of $\psi(u) = \rho'(u)$. Monotone M-estimators have non decreasing, bounded $\psi(u)$ functions which provide robust estimates when the outliers have low leverage values. Hard descenders force $\psi(u) = 0$ for $|u| > c$ ($c \in \mathbb{R}$ is termed *rejection point*) and allow the most aggressive rejection of outliers Soft descenders do not have a finite rejection point and force $\psi(u) \to 0$ as $|u| \to \infty$. Several ρ functions have been proposed which reduce the influence of large residual values on the estimated fit.

As observed by Stewart [92] also RANSAC and Hough transform can be seen as particular M-estimators. In fact the objective of maximizing consensus can be rephrased as the equivalent problem of minimizing the number of outliers, which may then be viewed as a binary robust loss function that is 0 for small absolute residuals, 1 for large residuals, and has a discontiniuty in correspondence of the inlier threhsold ϵ.

$$\rho(u) = \begin{cases} 0 & \text{if } |u| \leq \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{A.8}$$

	loss function ρ	influence function ψ	weighting function w
Huber $\begin{cases}\text{if } u \le c \\ \text{if } u > c\end{cases}$	$\begin{cases}u^2/2 \\ c(\lvert u\rvert - c/2)\end{cases}$	$\begin{cases}u^2/2 \\ c(\lvert u\rvert - c/2)\end{cases}$	$\begin{cases}1 \\ c/\lvert u\rvert\end{cases}$
Cauchy	$(c^2/2)\log(1+(u/c)^2)$	$\frac{u}{1+(u/c)^2}$	$\frac{1}{1+(u/c)^2}$
Geman-McClure	$\frac{u^2/2}{1+u^2}$	$\frac{u}{(1+u^2)^2}$	$\frac{1}{(1+u^2)^2}$
Welsh	$(c^2/2)[1-\exp(-(u/c)^2)]$	$u\exp(-(u/c)^2)$	$\exp(-(u/c)^2)$
Tukey $\begin{cases}\text{if } u \le c \\ \text{if } u > c\end{cases}$	$\begin{cases}c^2/6(1-(1-(u/c)^2)^3) \\ (c^2/6)\end{cases}$	$\begin{cases}u[1-(u/c)^2]^2 \\ 0\end{cases}$	$\begin{cases}[1-(u/c)^2]^2 \\ 0\end{cases}$

Table A.1: M-estimators

This loss function is no longer continuous and does not have a unique minimum. However interestingly, both RANSAC and Hough transforms, by virtue of the inlier threshold, in practice tolerate globally more outliers than half of the data. A cost of this is that small structures happen by chance can also be found, implying that careful post processing analysis of the discovered structures and outlier rejection heuristic are necessary.

As observed in [64], the use of the binary loss function (A.8) yields very poor local robustness properties, therefore several variants were introduced in which this zero-one loss function is replaced by a smooth once. For instance in [100] Torr et al. adopt the skipped mean

$$\rho(u) = \begin{cases} u & \text{if } |u| \le \epsilon \\ \delta & \text{otherwise} \end{cases} \tag{A.9}$$

in which the inliers are scored according to their fitness to the model, while the outilers are given a constant penalty weight δ. This approach is called MSAC (M-estimator Sample and Consensus) and always yields benefits compared to RANSAC with absolutely no additional computational burden.

References

1. Sameer Agarwal, Jongwoo Lim, Lihi Zelnik-manor, Pietro Perona, David Kriegman, and Serge Belongie. Beyond pairwise clustering. In *Conf. on Computer Vision and Pattern Recognition*, pages 838–845, 2005.
2. `http://perception.csl.illinois.edu/matrix-rank/sample_code.html`.
3. Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM, 1999.
4. Anthony Atkinson, Marco Riani, et al. *Exploring multivariate data with the forward search*. Springer Science & Business Media, 2013.
5. Alireza Bab-Hadiashar and David Suter. Robust segmentation of visual data using ranked unbiased scale estimate. *Robotica*, 17(06):649–660, 1999.
6. Alexander Balinsky, Helen Balinsky, and Steven Simske. On the helmholtz principle for data mining. *Hewlett-Packard Development Company, LP*, 2010.
7. Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of Allerton*, September 2010.
8. Manuele Bicego, Vittorio Murino, and Mário AT Figueiredo. Similarity-based classification of sequences using hidden markov models. *Pattern Recognition*, 37(12):2281–2291, 2004.
9. Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 1997.
10. Johannes Blömer and Jean-Pierre Seifert. Fault based cryptanalysis of the advanced encryption standard (aes). In Rebecca N. Wright, editor, *Financial Cryptography*, volume 2742 of *Lecture Notes in Computer Science*, pages 162–181. Springer, 2003.
11. Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter Fumy, editor, *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 1997.
12. Terrance E Boult and Lisa Gottesfeld Brown. Factorization-based segmentation of motions. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 179–186. IEEE, 1991.
13. Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.
14. Alfred M Bruckstein, David L Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.
15. Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3), 2011.
16. Pak K Chan, Martine DF Schlag, and Jason Y Zien. Spectral k-way ratio-cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 13(9):1088–1096, 1994.
17. Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1261–1268. IEEE, 2010.

18. Tat-Jun Chin, Hanzi Wang, and David Suter. Robust fitting of multiple structures: The statistical learning approach. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 413–420. IEEE, 2009.
19. Tat-Jun Chin, Jin Yu, and David Suter. Accelerated hypothesis generation for multistructure data via preference analysis. *Trans. Pattern Anal. Mach. Intell.*, pages 533–546, 2012.
20. Jongmoo Choi and Gérard G. Medioni. Starsac: Stable random sample consensus for parameter estimation. In *Proc. Conf. on Comp. Vis. and Patt. Rec.* IEEE, 2009.
21. Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of ransac family. In *British Machine Vision Conference (BMVC)*, 2009.
22. O. Chum and J. Matas. Randomized ransac with $T_{d,d}$ test. In *Image and Vision Computing*, volume 22, pages 837–842, 2002.
23. Ondrej Chum and Jiri Matas. Matching with PROSAC - progressive sample consensus. In *Computer Vision and Pattern Recognition*, pages 220–226, 2005.
24. Ondřej Chum, Tomáš Werner, and Jiří Matas. Epipolar geometry estimation via ransac benefits from the oriented epipolar constraint. 2004.
25. Ondřej Chum, Tomáš Werner, and Jiří Matas. Two-view geometry estimation unaffected by a dominant plane. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 772–779. IEEE, 2005.
26. Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.
27. João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.
28. Andrew Delong, Olga Veksler, and Yuri Boykov. Fast fusion moves for multi-model estimation. In *Proc. Europ. Conf. Comp. Vis.*, pages 370–384, 2012.
29. Patrick Denis, James H. Elder, and Francisco J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European Conf. on Computer Vision*, pages 197–210, 2008.
30. Agnès Desolneux, Lionel Moisan, and J-M Morel. *From gestalt theory to image analysis: a probabilistic approach*, volume 34. Springer Science & Business Media, 2007.
31. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.
32. Lixin Fan and Timo Pylvänäinen. Robust scale estimation from ensemble inlier sets for random sample consensus methods. In *Proc. Europ. Conf. Comp. Vis.*, pages 182–195, 2008.
33. Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
34. Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
35. Andrew W Fitzgibbon and Andrew Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *Computer Vision-ECCV 2000*, pages 891–906. Springer, 2000.
36. David F Fouhey. *Multi-model Estimation in the Presence of Outliers*. PhD thesis, Citeseer, 2011.
37. David F. Fouhey, Daniel Scharstein, and Amy J. Briggs. Multiple plane detection in image pairs using j-linkage. In *International Conf. on Pattern Recognition*, 2010.
38. Charles William Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998.
39. Nicolas Gillis and François Glineur. Nonnegative factorization and the maximum edge biclique problem. *arXiv preprint arXiv:0810.4225*, 2008.
40. Christophe Giraud. Dfa on aes. *IACR Cryptology ePrint Archive*, 2003:8, 2003.
41. Venu M. Govindu. A Tensor Decomposition for Geometric Grouping and Segmentation. *Conf. on Computer Vision and Pattern Recognition*, 1:1150–1157, 2005.
42. Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, ieee transactions on*, 11(9):1074–1085, 1992.
43. Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

44. Christian Häne, Christopher Zach, Bernhard Zeisl, and Marc Pollefeys. A patch prior for dense 3d reconstruction in man-made environments. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 563–570. IEEE, 2012.
45. Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.
46. Peter J Huber. *Robust statistics*. Springer, 2011.
47. Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 97(2):123–147, 2012.
48. Paul Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.
49. Suraj Jain and Venu Madhav Govindu. Efficient higher-order clustering on the grassmann manifold. In *International Conference on Computer Vision*, 2013.
50. Yasushi Kanazawa and Hiroshi Kawakami. Detection of planar regions with uncalibrated stereo using distribution of feature points. In *British Machine Vision Conf.*, pages 247–256, 2004.
51. Dusko Karaklajic, Jˆrn-Marc Schmidt, and Ingrid Verbauwhede. Hardware designer's guide to fault attacks. *IEEE Trans. VLSI Syst.*, pages 2295–2306, 2013.
52. Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.
53. Jon Kleinberg. An impossibility theorem for clustering. *Advances in neural information processing systems*, pages 463–470, 2003.
54. Da Kuang, Sangwoon Yun, and Haesun Park. Symnmf: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, pages 1–30, 2014.
55. Carmen Lai, David MJ Tax, Robert PW Duin, ELŻBIETA PĘKALSKA, and Pavel Paclík. A study on combining image representations for image classification and retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(05):867–890, 2004.
56. Nevena Lazic, Inmar E. Givoni, Brendan J. Frey, and Parham Aarabi. FLoSS: Facility location for subspace segmentation. In *Proc. Int. Conf. Comp. Vis.*, pages 825–832, 2009.
57. Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized ransac–full experimental evaluation. In *British Machine Vision Conference*, pages 1–11, 2012.
58. Kil-Moo Lee, Peter Meer, and Rae-Hong Park. Robust adaptive segmentation of range images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(2):200–205, 1998.
59. Hongdong Li. Two-view motion segmentation from linear programming relaxation. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
60. Yang Li, Shigeto Gomisawa, Kazuo Sakiyama, and Kazuo Ohta. An information theoretic perspective on the differential fault analysis against aes. *IACR Cryptology ePrint Archive*, 2010:32, 2010.
61. Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.
62. Alan H Lipkus. A proof of the triangle inequality for the tanimoto distance. *Journal of Mathematical Chemistry*, 26(1-3):263–265, 1999.
63. Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 171–184, 2013.
64. Peter Meer. Robust techniques for computer vision. *Emerging topics in computer vision*, pages 107–190, 2004.
65. James V Miller and Charles V Stewart. Muse: Robust surface fitting using unbiased scale estimates. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 300–306. IEEE, 1996.
66. Sushil Mittal, Saket Anand, and Peter Meer. Generalized projection based m-estimator: Theory and applications. In *Proc. Conf. on Comp. Vis. and Patt. Rec.*, 2011.
67. Lionel Moisan, Pierre Moulon, and Pascal Monasse. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line*, 2:56–73, 2012.
68. Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.

69. Amir Moradi, Mohammad T. Manzuri Shalmani, and Mahmoud Salmasizadeh. A generalized method of differential fault attack against aes cryptosystem. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 91–100. Springer, 2006.
70. Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.
71. Kai Ni, Hailin Jin, and Frank Dellaert. Groupsac: Efficient consensus in the presence of groupings. In *International Conference on Computer Vision*, pages 2193–2200, 2009.
72. Mauricio Orozco-Alzate, Robert PW Duin, and Germán Castellanos-Domínguez. A generalization of dissimilarity representations using feature lines and feature planes. *Pattern Recognition Letters*, 30(3):242–254, 2009.
73. Kemal Egemen Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1134–1141, 2010.
74. Elżbieta Pękalska and Robert PW Duin. *The dissimilarity representation for pattern recognition: foundations and applications*. Number 64. World Scientific, 2005.
75. Trung Thanh Pham, Tat-Jun Chin, Kaspar Schindler, and David Suter. Interacting geometric priors for robust multimodel fitting. *Transactions on Image Processing*, 23(10):4601–4610, 2014.
76. Trung-Thanh Pham, Tat-Jun Chin, Jin Yu, and David Suter. Simultaneous sampling and multi-structure fitting with adaptive reversible jump mcmc. In *Neural Information Processing Systems*, pages 540–548, 2011.
77. Trung-Thanh Pham, Tat-Jun Chin, Jin Yu, and David Suter. The random cluster model for robust geometric fitting. In *Proc. Conf. on Comp. Vis. and Patt. Rec.*, 2012.
78. Trung-Thanh Pham, Tat-Jun Chin, Jin Yu, and David Suter. The random cluster model for robust geometric fitting. *Pattern Analysis and Machine Intelligence*, 36(8):1658–1671, 2014.
79. Gilles Piret and Jean-Jacques Quisquater. A differential fault attack technique against spn structures, with application to the aes and khazad. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *CHES*, volume 2779 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2003.
80. Pulak Purkait, Tat-Jun Chin, Hanno Ackermann, and David Suter. Clustering with hypergraphs: the case for large hyperedges. In *Computer Vision–ECCV 2014*, pages 672–687. Springer, 2014.
81. Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jose Matas, and Jens Frahm. Usac: a universal framework for random sample consensus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):2022–2038, 2013.
82. Rahul Raguram and Jan-Michael Frahm. Recon: Scale-adaptive robust estimation via residual consensus. In *International Conference on Computer Vision*, pages 1299–1306, 2011.
83. Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.
84. `http://cs.adelaide.edu.au/~trung/lib/exe/fetch.php?media=rcmsa_robust_fitting.zip`.
85. Peter J. Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.
86. K. Schindler, D. Suter, and H. Wang. A model-selection framework for multibody structure-and-motion of image sequences. *Int. J. Comp. Vis.*, 79(2):159–177, 2008.
87. Konrad Schindler and David Suter. Two-view multibody structure-and-motion with outliers through model selection. *Pattern Analysis and Machine Intelligence*, 28(6):983–995, 2006.
88. Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.
89. Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. Robust subspace clustering. *Ann. Statist.*, 42(2):669–699, 04 2014.
90. Charles V. Stewart. MINPRAN: A new robust estimator for computer vision. *Pattern Analysis and Machine Intelligence*, 17(10):925–938, 1995.
91. Charles V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(8):818–833, 1997.
92. Charles V Stewart. Robust parameter estimation in computer vision. *SIAM review*, 41(3):513–537, 1999.
93. Raghav Subbarao and Peter Meer. Nonlinear mean shift for clustering over analytic manifolds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1168–1175. IEEE, 2006.

94. Yasuyuki Sugaya, Yuichi Matsushita, and Kenichi Kanatani. Removing mistracking of multibody motion video database hopkins155. In *British Machine Vision Conf., BMVC 2013*, 2013.
95. T Tanimoto. Technical report, IBM Internal Report, 17th Nov. 1957.
96. R. Toldo and A. Fusiello. Robust multiple structures estimation with J-Linkage. In *European Conference on Computer Vision*, volume 5302, pages 537–547, 2008.
97. Roberto Toldo and Andrea Fusiello. Automatic estimation of the inlier threshold in robust multiple structures fitting. In *Proc. Int. Conf. Image An. Proc.*, pages 123–131, 2009.
98. Roberto Toldo and Andrea Fusiello. Image-consistent patches from unstructured points with j-linkage. *Image and Vision Computing*, 31(10):756–770, 2013.
99. P. H. S. Torr. An assessment of information criteria for motion model selection. *Proc. Conf. on Comp. Vis. and Patt. Rec.*, pages 47–53, 1997.
100. P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, (1):138–156, 2000.
101. Philip HS Torr and David W Murray. Stochastic motion clustering. In *European Conf. on Computer Vision*, pages 328–337. Springer, 1994.
102. Philip HS Torr, Andrew Zisserman, and Stephen J Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 1037–1042. IEEE, 1995.
103. Roberto Tron and René Vidal. A benchmark for the comparison of 3D motion segmentation algorithms. In *Conf. on Computer Vision and Pattern Recognition*, 2007.
104. Michael Tunstall and Debdeep Mukhopadhyay. Differential fault analysis of the advanced encryption standard using a single fault. *IACR Cryptology ePrint Archive*, 2009:575, 2009.
105. Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
106. Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
107. Hanzi Wang and David Suter. Robust adaptive-scale parametric model estimation for computer vision. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, pages 1459–1474, 2004.
108. David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *Transaction on evolutionary computation*, 1(1):67–82, 1997.
109. H. S. Wong, T.-J. Chin, J. Yu, and D. Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *International Conf. on Computer Vision*, 2011.
110. H. S. Wong, T.-J. Chin, J. Yu, and D. Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *International Conference on Computer Vision*, 2011.
111. L. Xu, E. Oja, and P. Kultanen. A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.
112. Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.
113. Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *European Conference on Computer Vision*, pages 94–106, 2006.
114. Allen Y Yang, Shankar R Rao, and Yi Ma. Robust statistical estimation and segmentation of multiple subspaces. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 99–99. IEEE, 2006.
115. Jin Yu, Tat-Jun Chin, and David Suter. A global optimization approach to robust multi-model fitting. In *Proc. Conf. on Comp. Vis. and Patt. Rec.*, 2011.
116. Luca Zappella et al. *Manifold clustering for motion segmentation*. PhD thesis, Universitat de Girona, 2011.
117. Ron Zass and Amnon Shashua. A unifying approach to hard and probabilistic clustering. In *International Conf. on Computer Vision*, volume 1, pages 294–301, 2005.
118. Ron Zass and Amnon Shashua. Doubly stochastic normalization for spectral clustering. In *Neural Information Processing Systems*, pages 1569–1576, 2006.
119. Wei Zhang and Jana Kosecká. Nonparametric estimation of multiple structures with outliers. In *European Conf. on Computer Vision*, volume 4358, pages 60–74, 2006.

120. Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27(2):161–195, 1998.
121. Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1):87–119, 1995.
122. M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, 2005.