COMBINATORIAL CLUSTERING AND ITS APPLICATION TO 3D POLYGONAL TRAFFIC SIGN RECONSTRUCTION FROM MULTIPLE IMAGES

B. Vallet [a, *], B. Soheilian [a] , M. Brédif [a]

[a] Université Paris-Est, IGN/SRSIG, MATIS, 73 avenue de Paris, 94165 Saint-Mandé, FRANCE
(bruno.vallet, bahman.soheilian, mathieu bredif)@ign.fr
recherche.ign.fr/labos/matis/∼(vallet, soheilian)

WG III

KEY WORDS: 3D reconstruction, mobile mapping, urban areas, clustering

ABSTRACT:

The 3D reconstruction of similar 3D objects detected in 2D faces a major issue when it comes to grouping the 2D detections into clusters to be used to reconstruct the individual 3D objects. Simple clustering heuristics fail as soon as similar objects are close. This paper formulates a framework to use the geometric quality of the reconstruction as a hint to do a proper clustering. We present a methodology to solve the resulting combinatorial optimization problem with some simplifications and approximations in order to make it tractable. The proposed method is applied to the reconstruction of 3D traffic signs from their 2D detections to demonstrate its capacity to solve ambiguities.

1. INTRODUCTION

Traffic signs are important road features that provide navigation rules and warnings to drivers. Their detection and identification in image is largely investigated by the ADAS (Advanced Driver Assistance System) research community since the 90s. Many different pattern recognition techniques are applied for this purpose. A recent state-of-the art in this field can be found in the paper presented by (Fu and Huang, 2010). Most of the traffic sign extraction systems deal with detection and classification of signs in single images. Some systems integrate the detection and classification in a tracking mode (Fang et al., 2003, Lafuente-Arroyo et al., 2007, Meuter et al., 2008). Consequently, more reliable decisions can be made using multi-frame rather than single-frame information. Overall, there is a large amount of research work on detection and classification of traffic signs using single or multi-frame image data. In contrast, fewer authors investigated 3D reconstruction and localization of traffic signs, although, it is required for several applications:

- Road inventory and update,
- Visibility analysis in urban areas,
- Virtual reality and 3D city modeling,
- Vision based positioning using visual landmarks.

1.1 Previous works

The problem of 3D traffic sign reconstruction from multiple images acquired by a Mobile Mapping System (MMS) was first investigated by (Timofte et al., 2009). Plausible correspondence between 2D detections of traffic signs on different images are computed based on geometric and visual consistency criteria. A 3D traffic sign hypothesis is then generated for each consistent pair of signs. Finally, a Minimum Description Length (MDL) approach is used to select an optimal subset of 3D traffic sign hypotheses. The method reaches 95% of correct reconstruction rate with an average localization accuracy of $24cm$, which can deteriorate to $1.5m$ in some cases.

An inventory system based on stereo vision and tracking was also developed for traffic sign 3D localization (Wang et al., 2010). Pairs of corresponding 2D detections of traffic signs are deducted from tracking them in successive images. Two stereo-based approaches called single-camera (stereo from motion) and dual-camera (rigid stereo base) are studied. Traffic sign localization accuracy varies from $1 - 3m$ for single-camera to $5 - 18m$ for dual-camera.

The localization accuracies obtained by aforementioned methods ($24cm$ to a few meters) may be sufficient for road database generation where traffic signs should be associated to road sections. However several applications such as vision based positioning using road landmarks (Li and Nashashibi, 2010) and high scale 3D city modeling (Früh and Zakhor, 2004, Cornelis et al., 2008) may require higher accuracies. To tackle this issue (Soheilian et al., 2013) developed an image based system for traffic sign reconstruction reaching subdecimetric accuracy. The method is efficient even in dense urban areas where traffic signs are not necessarily in standard positions in relation to road axis. However, in such dense areas, ambiguities may occur in the clustering of 2D detections (grouping the 2D detections corresponding to the same 3D sign). An example of such ambiguity is given in Figure 1 where a bad clustering can cause both a decrease in reconstruction accuracy and an omission. The aim of this

[*] Corresponding author.

doi:10.5194/isprsannals-II-3-165-2014

paper is to propose a generic framework to handle such ambiguities. The scope thus goes way beyond the application to traffic sign as our methodology can be applied more generally to disambiguate the 3D reconstruction of similar objects (motion capture trackers, crowds, animal swarms, particle clouds...) from 2D detections.

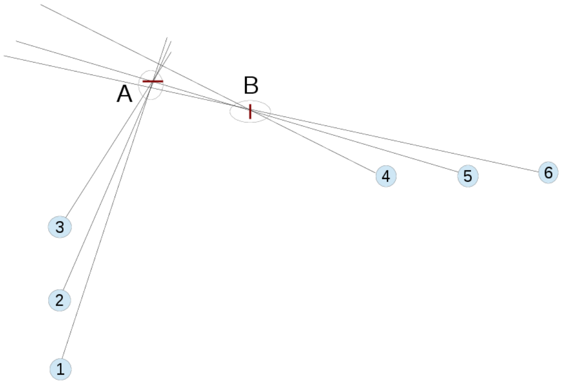

Figure 1: Clustering ambiguity: Two 3D signs (A, B) and six 2D detections (1-6). The correct clustering is (1,2,3) (4,5,6) but a greedy approach will usually give (1,2,3,5,6) (4) leading to a poor reconstruction of sign A and an omission of sign B that cannot be reconstructed from a single image.

1.2 Positioning

In the aforementioned 3D reconstruction approaches, this clustering problem is handled differently:

- The MDL approach of (Timofte et al., 2009) is limited to pairwise matchings, in which case a global optimum can be obtained by Integer Quadratic Programming (Leonardis et al., 1995). Conversely, we are trying to exploit as many views as possible in order to increase the accuracy of 3D reconstruction.
- (Wang et al., 2010) track the detections in image sequences. If the same sign is seen by different cameras of the mobile mapping system or if the tracking fails (because of occlusions for instance) it might be reconstructed multiple times.
- (Soheilian et al., 2013) iteratively groups the 2D detections in a greedy manner then ensures unicity of attribution (a 2D detection cannot be used to reconstruct more than one 3D sign). This heuristic does not ensure global optimality as (Timofte et al., 2009) does, but allows grouping more than 2 signs which helps enhancing the precision of the 3D reconstruction.

The aim of this paper is thus to propose a generic approach in a well justified theoretical framework to solve this clustering problem by exploiting the geometric consistency though a dissimilarity metric. Section 2. poses this clustering problem and proposes a method to solve it. Section 3. presents the application of our combinatorial clustering to the problem of 3D reconstruction of similar objects from their detections then specifies it to polygonal traffic signs. Finally we will conclude and open perspectives in Section 4.

2. COMBINATORIAL CLUSTERING

Combinatorial clustering is the generic tool that we introduce to solve the problem of reconstructing similar objects. The major lock is that it is very hard to know if two 2D detections correspond to the same 3D object, so we need a measure on the compatibility of 2D detections that is not a simple aggregation of a pairwise measures.

2.1 Problem formalization

In this section, we will call:

- $\mathbb{N}^n = \{1, ..., n\}$ the first n positive integers
- $\mathcal{O} = \{o_i\}_{i \in \mathbb{N}^n}$: the n objects o_i that we wish to cluster. For traffic signs, objects are the 2D detections and clusters corresponds to the 2D signs used to reconstruct a single 3D sign.
- $Part(\mathcal{O})$ the set of parts of $\mathcal{O}$: $Part(\mathcal{O}) = \{S | S \subset \mathcal{O}\}$
- $\mathcal{P}(\mathcal{O})$ the set of partitions of $\mathcal{O}$: $\mathcal{P}(\mathcal{O}) = \{\{S_1, ..., S_m\} | S_i \in Part(\mathcal{O})/\emptyset$ and $S_i \cup S_j = \emptyset$ and $\bigcup S_j = \mathcal{O}\}$
- $|S|$ the number of elements in the set S (cardinal).

Let $D : Part(\mathcal{O}) \to \mathbb{R}^+$ be a measure of the dissimilarity of a subset of objects. D should satisfy:

1. $\forall i \in \mathbb{N}^n, \quad D(\{o_i\}) = 0$: Singletons (individual objects) are not dissimilar.
2. $S_1 \cap S_2 = \emptyset \Rightarrow D(S_1 \cup S_2) \geq D(S_2) + D(S_2)$: the dissimilarity of a set is larger that the dissimilarity of any of its partition (or at best equal).

These two conditions imply that D increases when adding a new object to a set:

$$D(S \cup \{o_i\}) \geq D(S) + D(\{o_i\}) = D(S)$$

Most measures typically used for clustering satisfy these two conditions. D implies naturally a measure over $\mathcal{P}(\mathcal{O})$:

$$D(\{S_1, ..., S_m\}) = \sum_j D(S_j)$$

Making no assumptions on the number of clusters, clustering should find the partition that minimises D over $\mathcal{P}(\mathcal{O})$. To avoid the trivial solution of making only singletons (in which case the total dissimilarity is 0), we should however also minimize the number of clusters in the partition, such that we pose the problem of combinatorial clustering as finding:

$$argmin_{P \in \mathcal{P}(\mathcal{O})} E(P) = |P| + D(P) \qquad (1)$$

We will refer to E as the energy to be minimized. D should be chosen or scaled such that merging two subsets S_1 and S_2 should be favoured if D increases by less than 1 ($D(S_1 \cup$

$S_2) < D(S_1) + D(S_2) + 1$) because in this case the energy will decrease:

$$E(\{S_1 \cup S_2, S_3, ..., S_m\}) = m-1+D(S_1 \cup S_2)+\sum_{j=3}^{m} D(S_j) <$$

$$E(\{S_1, S_2, ..., S_m\}) = m + D(S_1) + D(S_2) + \sum_{j=3}^{m} D(S_j)$$

For 3D traffic signs reconstruction, D can be the residuals of a reconstruction of a sign from the 2D detections normalized by an estimate of the uncertainty on the image orientations (cf Section 3.2).

The problem posed above is very generic and can be applied to a broad variety of problems:

- Points clustering: the objects are points in $\mathbb{R}^n$ and the dissimilarity is the variance of a subset of these input points.
- Unsupervised classification: same as above with a vector of features.
- Shape detection (such as RANSAC, Hough): the objects are points in $\mathbb{R}^n$ and the dissimilarity is the sum of residuals of a least square estimate of a subset of point by the shape to detect.
- Reconstruction of similar 3D shapes: objects are 2D detections of the similar 3D shapes and the dissimilarity is a normalized sum of residuals of the 3D reconstruction of an object from a subset of detections (cf Sections 3.1 and 3.2).

We will now propose a methodology to solve this combinatorial clustering problem, that is to find a minimum (or at least a good approximation) of (1) over all possible partitions of the set (O) of input objects to cluster. To our knowledge, the numerous clustering methods that have been developed in the fields of computer vision and machine learning mainly deal with bivariate similarity measures (the dissimilarity measure is only defined for a pair of object), such that the dissimilarity of a set is sum of the pairwise dissimilarities of its constituents. For our specific problem, such approaches are insufficient as a very high similarity can be coincidental for detection corresponding to different 3D objects. This is why we investigated means to solve this more general clustering problem.

2.2 Compatibility graph

The number of partitions of a set of n elements is the Bell number $|\mathcal{P}(\mathcal{O})| = B_n$ that increases (very roughly) as n^n which is even faster than exponential. Minimizing (1) over all possible partitions of the set (O) thus cannot be done by brute force in reasonable time for more than 20 objects (for which $B_{20} \approx 5.10^{13}$) while we were regularly confronted to problems of size exceeding 100 in the context of road sign reconstruction. In this paper, we propose heuristics to make this problem tractable for large number of objects.

The first simplification of this problem is to list pairs of incompatible objects and forbid putting them in the same set. In other terms, we propose to find a criterion ensuring that two objects o_1 and o_2 are not in the same part of the optimal solution. We found that the criterion $D(\{o_1, o_2\}) > 2$ gives such an insurance, because in this case:

$$E(\{o_1, o_2, o_3, ..., o_m\}) = 1 + D(\{o_1, o_2, o_3, ..., o_m\}) \geq$$

$$1+D(\{o_1, o_2\})+D(\{o_3, ..., o_m\}) > 3+D(\{o_3, ..., o_m\}) =$$

$$E(\{o_1\}, \{o_2\}, \{o_3, ..., o_m\})$$

This result has a simple geometric interpretation: the minimization ensures that the clusters have a maximum radius of 1 (in normalized dissimilarity), so two objects further than 2 cannot belong to the same cluster.

If more is known on D, a finer incompatibility criterion should be investigated (cf Section 3.2 in the case of 3D traffic signs reconstruction). This incompatibility relationship allows us to build an object compatibility graph $\mathcal{G}_C$ where nodes are objects and an edge exists between two nodes if the corresponding objects are compatible (cf Figure 2, last page). Acceptable partitions will then be partitions of this graph into cliques (subsets of the graph where each node is connected to each other, or equivalently sets of objects all compatible one with another). This approximation decreases drastically the number of possible partitions, especially if the compatibility criterion is well chosen. It also naturally splits the problem into smaller problems given by the connected components of $\mathcal{G}_C$ that can be processed separately (in different threads and on different machines).

Finding the optimal partitions of a connected component $\mathcal{C}$ of a graph in cliques can be done in the following way:

1. List all the cliques C_k of $\mathcal{C}$ and compute their energies $E(C_k)$.
2. Build a clique compatibility graph $\mathcal{C}_C$ where the nodes are the enumerated cliques C_k, and there is an edge iff cliques are non intersecting (intersecting would mean the same object belongs to two clusters).
3. Find the maximal clique of minimum energy in $\mathcal{C}$. A maximal clique of $\mathcal{C}$ corresponds to a partition of $\mathcal{C}$ in non overlapping cliques. The standard being maximum weighted clique, we will simply take the opposites of the individual clique energies.

Note that we handle two different types of cliques:

1. Cliques of $\mathcal{C}$ that are sets of objects compatible one with another (potential clusters).
2. Maximal cliques of $\mathcal{C}_C$ that define partitions of $\mathcal{C}$ in cliques (of the previous type).

2.3 Clique enumeration

We propose a simple and memory efficient approach to build the set of all the cliques of a connected component $\mathcal{C}$, illustrated in Figure 3. It relies on creating a tree with the empty set as root, each tree node t_k is labelled by a graph node n_i and corresponds to the clique C_k containing n_i and all the labels of the ancestors of t_k (tree nodes above t_k):

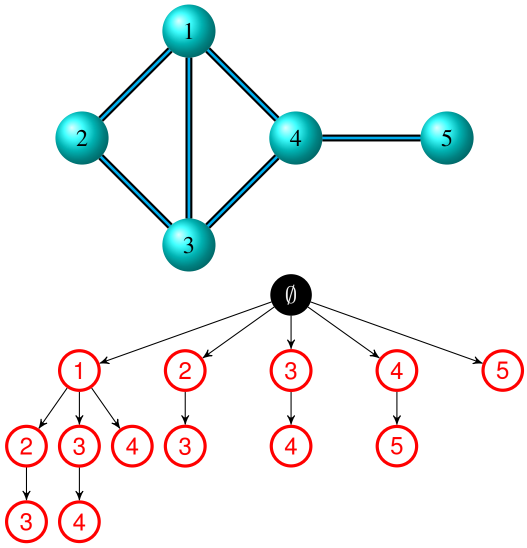

Figure 3: A simple graph (top) and its clique tree (bottom). Each red node t_k corresponds to the clique of the graph defined by listing the nodes n_i from $\emptyset$ to t_k

1. Create the tree root t_0 labelled with $\emptyset = C_0$ and add one child to this root labelled with n_i for each node n_i of $\mathcal{G}$.
2. For each tree node t_k, labelled with n_i list all the nodes n_j $(j > i)$ of $\mathcal{C}$ that are connected to all nodes in C_k and add a child to t_k labelled with n_j for each of them. The criterion $(j > i)$ ensures that each clique is enumerated only once (with nodes in increasing order).
3. Iterate step 2 recursively on the new tree nodes until no new node is found.

The energy $E(C_k)$ can be computed efficiently and stored on each tree node during this construction, especially if the update of the dissimilarity measure $D(C_k)$ is a simple update of the dissimilarity of a sub-clique of C_k (given by the father node of t_k). For large, highly connected graphs, this construction may exceed the memory size of the machine. In our experiments, this happens for connected components with more than 200 nodes and 3000 edges. We propose a solution to that problem in Section 2.5.4.

2.4 Maximum weighted clique

The maximum weighted clique problem is rather classical and efficient implementations exist to solve it, such as Cliquer (Niskanen and Östergård, 2003). However, the number of enumerated cliques grows rapidly with the number of nodes and edges of $\mathcal{C}$, which makes the maximal clique problem too long to run. We can however give a certain time budget to the algorithm and ask it to give the best solution found after this amount of time if the optimal solution cannot be found. In this case, we will compare the solution with a greedy search (iteratively add the best clique not overlapping previously added cliques) and take the best. In extreme cases (more than 10^5 enumerated cliques) even the graph construction may exceed memory size so we directly choose the greedy solution.

2.5 Improvements

For clarity, we have only given the core of the method in the previous sections. In our implementation, we have added several useful improvements to enhance the quality of the result in the cases where optimality cannot be guaranteed and reduce the computing time and memory footprint of the algorithm.

2.5.1 Singletons removal By definition, singletons have an energy of 1 because their disparity is null. As singletons are cliques, they should be added to the clique compatibility graph $\mathcal{C}_C$, which increases its size by n. However, a simple modification of the energy allows to bring their energy to 0 making singletons indifferent (they can be added or not to the solution without modification to the energy): $E(S) = 1 + D(S) + |S|$ Because $\sum |S|$ over a partition is exactly the number n of objects, this only corresponds to adding the constant n to the partition energy which will not change its minimum. This way, if an object does not appear in the maximum weighted clique, it simply means that it is a singleton in the corresponding partition.

2.5.2 Local optimality The incompatibility criterion being quite weak, the compatibility graph $\mathcal{G}_C$ might have large cliques that require to be cut in several clusters to optimize the energy. If a clique listed by the method of Section 2.3 has a partition that has a better energy, then we are sure that it won't belong to the optimal solution, so we should not add it to the clique graph $\mathcal{C}_C$. This way we can reduce the size and complexity of $\mathcal{C}_C$ without loss of generality. We call "locally optimal" a clique for which no partition has a better energy, and we add to $\mathcal{C}_C$ only such locally optimal cliques.

Computing local optimality can be done by computing an "optimal energy" which is the energy of the best partition of a clique. A clique is then simply defined locally optimal if its optimal energy is equal to its energy. This can be done by increasing clique size:

- A pair $\{o_1, o_2\}$ is locally optimal iff

  $$E(\{o_1, o_2\}) = 1 + D(\{o_1, o_2\}) < E(\{o_1\}, \{o_2\}) = 2$$

  $$\Leftrightarrow D(\{o_1, o_2\}) < 1$$

  Define its optimal energy as

  $$E_{opt}(\{o_1, o_2\}) = min(E(\{o_1, o_2\}), E(\{o_1\}, \{o_2\}) = 0)$$

- For a n-clique C, compute the minimum energy:

  $$E_{bipart}(C) = min_{C_1 \cup C_2 = C, C_1 \cap C_2 = \emptyset} E_{opt}(C_1) + E_{opt}(C_2)$$

  over all bipartitions of C (there are $2^{n-1} - 1$). The optimal energies of the two parts C_1 and C_2 have already been computed as their sizes are $< n$ and this computation is done by increasing clique size. The optimal energy is then simply $E_{opt}(C) = min(E(C), E_{bipart}(C)$. C is locally optimal if the min is $E(C)$.

This local optimality has two benefits:

1. It reduces the complexity of the maximum weighted clique computation

2. It ensures that the found solution (even if not optimal) only contains clusters that should not be split. In particular, it ensures that the ambiguity problem mentioned in Figure 1 will be solved even if a greedy algorithm is used, because the clique (1,2,3,5,6) will not be locally optimal (its partition (1,2,3)(5,6) is better) so it will be removed from the clique graph beforehand.

2.5.3 Useless edges removal The cliques compatibility graph has a number of edges that increases very rapidly. Fortunately, a simple criterion allows us to remove some useless edges: if merging two clusters gives a better energy, then the corresponding edge is useless as we know it won't be part of the optimal solution. If as a result, $\mathcal{C}_C$ is not connected any more, then each connected component can be processed separately which reduces both computing time and memory footprint.

2.5.4 Problem splitting In practice, the method described above is only tractable for connected components containing up to 150-200 objects. If the compatibility criterion is sufficiently good, connected components might have such reasonable sizes even with a much larger number of input objects. However, if the problem is too complex and the compatibility criterion not discriminative enough, connected components larger than this size may appear. In this case, too many cliques are listed and the cliques graph $\mathcal{C}_C$ becomes so large that it cannot even be stored in memory (Cliquer stores graphs in a matrix so a graph with n nodes takes roughly a memory space of n^2).

The solution that we propose is to find the optimal solution on subsets of $\mathcal{C}_C$ then merge these solutions. We rely on the strong assumption that the optimal solution of a subset of the problem is a subset of the optimal solution of the full problem, which is only true for some dissimilarity measures, and an approximation necessary to make the problem tractable in other cases. We split the input set $\{o_i\}_{i\in\mathbb{N}^n}$ into two sets of objects with even and odd indices. The two optimal partitions P_1 and P_2 of these subsets are computed with the method described above. To merge the results we build a solution merge graph $\mathcal{G}_M$ where the nodes are pairs $\{p_i^1, p_j^2\}, p_i^1 \in P_1, p_j^2 \in P_2$ for which $E(p_1 \cup p_2\}) < E(p1, p2)$. This means that a node represents the fact that merging the two parts reduces the energy. Two such merges $\{p_i^1, p_j^2\}$ and $\{p_{i'}^1, p_{j'}^2\}$ will be called incompatible iff $i = i'$ or $j = j'$, that is if they share two parts, because in that case applying the two merges will merge together two parts from an optimal solution, which is against our assumption.

Once again, we find the optimal merge by finding the minimum weighted clique of the merge graph $\mathcal{G}_M$, where the weights are the energy reductions $E(p_1 \cup p_2\}) - E(p1, p2) < 0$. This ensures to find the optimal compatible merges between parts of the two solutions.

The splitting/solution merging process described in this section can be applied recursively if the subsets are still too large.

2.5.5 Combinatorial Gradient descent Because we need to do some approximations if the problem is too large (problem splitting, not waiting for the end of maximal clique computation), we are in general not guaranteed to find a global minimum. This is why, in such cases, we can try to improve locally the solution by trying to change the class of individual objects, and validating the change if the energy decreases. This can be done in a greedy manner until no cluster change decreases the energy. These operations do not change the number of clusters but only their composition.

3. RESULTS

The combinatorial clustering defined above is very general and requires only a normalized measure of dissimilarity on any subset of objects, and optionally a finer criterion to discard pairs of objects. We first applied it to (synthetic) particle reconstruction in order to validate it (Section 3.1) then to our initial problem of traffic sign reconstruction (Section 3.2).

3.1 Particle reconstruction

We call particle reconstruction the problem of reconstructing individual (and indistinguishable) 3D points (particles) from their 2D projections in images. More precisely, we generate N_{3D} such particles P_i in the unit cube ($N_{3D} = 10$ in our experiments), and place N_{cam} virtual cameras around the cube in which we project the particles. Each 2D projection of a particle P_i in an image j corresponds to a 3D line L_i^j, so somehow our clustering problem corresponds to finding the optimal way to intersect these lines. For a given subset S of lines, the optimal intersection I^* is obtained by minimizing:

$$E_S(I) = \sum_i \sum_j d^2(L_i^j, I)$$

over all points $I \in \mathbb{R}^3$, where d is the Euclidian distance. The minimum I^* is easy to compute using a closed form formula, and we define the dissimilarity of subset S as $D(S) = E_S(I^*)/E_{ref}$. To take into account the uncertainties that arise from both the (intrinsic and extrinsic) calibration and detection in the real world, we introduce a Gaussian noise of variance σ on the three coordinates of the camera centres, implying roughly a $d_{av}^2 = 2.7\sigma$ average particle to line squared distance. For a correct cluster of size n, we can expect residuals of $E_S(I^*) \approx nd_{av}^2$, thus a good choice is $E_{ref} = 4\sigma$ such that objects adding more than $1.5d_{av}^2$ to the disparity of a cluster will be rejected, while others will be favoured. Experimental results are presented in three tables that give:

- $Prec$ (%): Precision of the reconstruction=number of good reconstructions/number of reconstructions, a reconstruction being considered good if it is at less than d_{av} from the input particle.
- Rec (%): Recall of the reconstruction=number of particles correctly reconstructed/N_{3D}, same criterion for correct reconstruction.
- $Dupl$: Number of duplicates (multiple reconstructions for one particle=over-segmentation). We do not count under-segmentation (one reconstruction for multiple particles) as it will result in a decrease in Rec.
- Acc (cm): Accuracy=average distance between a particle and its corresponding reconstruction (in cm). Each line of the tables corresponds to one experiment for which

the 10 particles were randomly generated. The difficulty of the problem depending on the minimum distance between two particles D_{min} (in cm), we systematically indicate it. Another alternative would have been to average the results on numerous experiments, but that would not be much more informative as each experiment might have a very different difficulty depending on the exact configuration of particle.

N_{cam}	D_{min}	$Prec$	Rec	$Dupl$	Acc
2	11	71	70	0	7.2
3	11	87	80	0	4.3
4	15	90	90	0	4.4
5	19	100	90	0	3.6
7	13	100	100	0	3.6
9	13	100	100	0	4.4
12	22	100	100	0	3.2
15	12	100	100	3	2.6

Table 1: Results on synthetic dataset for increasing N_{cam} with $N_{3D} = 10$, $\sigma = 4cm$ and $E_{ref} = 4\sigma$.

Table 1 shows that with only two cameras, the problem is too ambiguous to be well solved. Additional cameras help disambiguating (5 are sufficient in this setting). The accuracy of the reconstruction is also increased with more cameras, which was one of the motivations of this work (making large clusters improves accuracy). However, increasing the number of cameras also increases drastically the potential false detections (sets of 3D lines coincidentally all close to a point in space where no particle is present). Our algorithm still proves quite robust to that (precision stays high), even if the combinatorial difficulty increases rapidly with the number of cameras (problem required splitting above 7 cameras). This combinatorial explosion is probably the reason for the 3 splits (2 reconstructions for a single particle) occuring with 15 cameras. Finally, we observe as expected that the accuracy increases with the number of cameras used for constant uncertainty on the camera positions.

Note that the problem that we are trying to solve here is very hard as the average line to particle distance is 6.6cm for $\sigma = 4cm$ and the minimum particle to particle distance between 10 and 20cm, so for a line passing between two particles, the attribution may be ambiguous even for a perfect algorithm.

σ	D_{min}	$Prec$	Rec	$Dupl$	Acc
1	13	100	100	0	0.9
2	13	100	100	0	1.8
4	17	100	90	2	4.2
8	26	78	80	2	9.5

Table 2: Results on synthetic dataset for increasing σ (in cm), $N_{cam} = 5$ and $E_{ref} = 4\sigma$

Table 2 shows that when the uncertainty on camera orientation increases, the problem becomes more ambiguous. The problem is perfectly solved if uncertainty is sufficiently low, then fails for higher uncertainties. For this test with 5 cameras, the limit occurs around $\sigma = 4cm$ which is roughly a theoretical limit as stated above. Increasing the number of points has the same impact as increasing σ as the point density increases which is equivalent to scaling down.

Table 3 investigates the choice of the dissimilarity normalization factor E_{ref}. Choosing E_{ref} below 2σ generates many small compact clusters that do not necessarily correspond to particles (they might be coincidental with so many lines). Conversely, large values will under-segment (one reconstruction for multiple particles), which lowers both precision and recall. Moreover, high E_{ref} values make the problem more difficult as the consistency graph becomes much denser, so it has much more cliques, which decreases the quality of the result for the maximum clique computation that has a constant time budget, and increases computation time for the local optimality.

E_{ref}	D_{min}	$Prec$	Rec	$Dupl$	Acc
σ	16	9.5	20	0	5.1
2σ	20	69	90	0	4.3
4σ	16	90	100	0	4.5
6σ	19	89	90	1	5.0

Table 3: Results on synthetic dataset for increasing D_{ref} with $\sigma = 4cm$, $N_{cam} = 5$

For all the experiments, we used a time budget for maximal clique computation of 30s which proved sufficient in most cases. The computation time thus mainly depends on the local optimality computation. For 100 nodes, this computation takes from a minute to 10 for very dense graphs. For lager sizes, we recursively split and merge the component, so the computing time will be roughly linear, plus the time budget allocated to each merge (which are also maximum clique problems).

3.2 Application to 3D traffic sign reconstruction

To apply our methodology, we simply average the dissimilarity from previous section over the sign corners (a specific measure should be used for circular signs). Note that the reconstruction used is constrained by the shape of the 3D sign (square, equilateral triangle) in the same manner as (Soheilian et al., 2013). For the normalization factor E_{ref}, we estimate that our geroreferencing system drifts of a few centimeters during the acquisition of images of a given sign. Given the results of previous section, we assumed that $E_{ref} = (20cm)^2$ was a good choice.

For the discarding criterion, we used the same constraints as (Timofte et al., 2009) and (Soheilian et al., 2013):

- Geometric constraints:
  - Epipolar geometry: both detected sign centers should lie on the same epipolar line of the camera pair.
  - Size: the size of resulting 3D traffic sign should lie in some range (specified by a traffic sign reference document).
  - Visibility: the resulting 3D traffic sign should face both camera centers.
- Similarity constraint: both detected signs should have the same visual characteristics.

We evaluated the reconstruction on a large dataset of 4800 full HD images acquired on a 1 km long path which resulted in 1057 2D detections, from which 62 signs were reconstructed. Only one sign from the ground truth was omitted

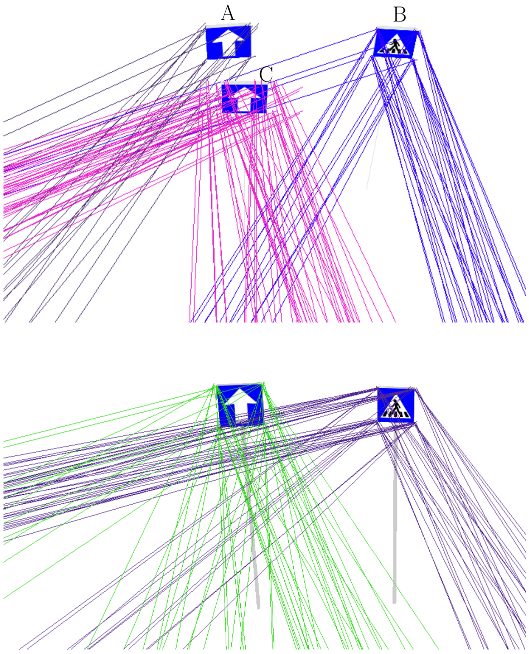

Figure 4: An ambiguous clustering problem. Top: a greedy heuristic generates a "ghost" 3D signs at C when reconstructing the real signs A and B. Bottom: our combinatorial clustering successfully removed C, reattributing all 2D detections to the correct 3D sign, also improving accuracy.

(1 False Negative) and two reconstructions had no counterpart in the ground truth (2 False Positives, against 9 with a greedy heuristic). Seven signs were duplicated. Most of the "ghost" signs from the greedy heuristic of (Soheilian et al., 2013) were eliminated as illustrated in Figure 4, validating its practical interest. Processing time (for the combinatorial clustering) was around 1 hour.

4. CONCLUSIONS

This paper proposed a general framework for combinatorial clustering that can be applied in a wide range of contexts, in particular to disambiguate the reconstruction of similar 3D objects such as traffic signs. Even if our formulation of the problem makes it hard to solve in complex cases, we proposed several improvements that makes it tractable on most cases that we encountered. Moreover our interlaced strategy for splitting very large problems shows good performance even if we cannot give guarantees that the resulting partition is optimal. Our experiments have shown a significant improve compared to our previous greedy heuristic, with a correct handling of most ambiguities. Thus we consider that this work has overcome a significant difficulty inherent to the problem of reconstruction of similar 3D object, which may be applicable to other contexts. The major limit, inherent to the problem, is the ratio between the precision of the detection and the minimum distance between objects. When it becomes too high, the problem becomes too ambiguous for a correct solution to be found.

In the future, we plan on validating more finely the approximations necessary to make the problem tractable, but also on refining the criteria used to create the consistency graph. For instance, we would like to make use of a 3D city model in order to predict occlusions more finely in order to have a simpler consistency graph. Finally, the complexity of the problem could be reduced if we track the signs for each camera, in which case we would cluster not the signs but the series of signs from tracking.

ACKNOWLEDGEMENTS

This study has been performed as part of the Cap Digital Business Cluster TerraMobilita Project.

REFERENCES

Cornelis, N., Leibe, B., Cornelis, K. and Gool, L., 2008. 3D Urban Scene Modeling Integrating Recognition and Reconstruction. International Journal of Computer Vision 78(2-3), pp. 121–141.

Fang, C.-y., Member, A., Chen, S.-w., Member, S. and Fuh, C.-s., 2003. Road-sign detection and tracking. IEEE Transactions on Vehicular Technology 52(5), pp. 1329–1341.

Früh, C. and Zakhor, A., 2004. An automated method for large-scale, ground-based city model acquisition. International Journal of Computer Vision 60, pp. 5–24.

Fu, M. and Huang, Y., 2010. A survey of traffic sign recognition. In: International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), IEEE, pp. 119–124.

Lafuente-Arroyo, S., Maldonado-Bascon, S., Gil-Jimenez, P., Acevedo-Rodriguez, J. and Lopez-Sastre, R. J., 2007. A Tracking System for Automated Inventory of Road Signs. In: IEEE Intelligent Vehicles Symposium, Ieee, pp. 166–171.

Leonardis, A., Gupta, A. and Bajcsy, R., 1995. Segmentation of range images as the search for geometric parametric models. International Journal of Computer Vision 14(3), pp. 253–277.

Li, H. and Nashashibi, F., 2010. Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data. In: Intelligent Transportation Systems, Madeira Island, Portugal, pp. 1657–1662.

Meuter, M., Kummert, A. and Muller-Schneiders, S., 2008. 3D Traffic Sign Tracking Using a Particle Filter. In: 2008 11th International IEEE Conference on Intelligent Transportation Systems, pp. 168–173.

Niskanen, S. and Östergård, P. R. J., 2003. Cliquer user's guide, version 1.0. Communications Laboratory, Helsinki University of Technology, Espoo, Finland.

Soheilian, B., Paparoditis, N. and Vallet, B., 2013. Detection and 3D reconstruction of traffic signs from multiple view color images. ISPRS Journal of Photogrammetry and Remote Sensing 77, pp. 1–20.

Timofte, R., Zimmermann, K. and Van Gool, L., 2009. Multi-view traffic sign detection, recognition, and 3D localisation. In: 2009 Workshop on Applications of Computer Vision (WACV), IEEE, Snowbird, UT, pp. 1–8.

Wang, K. C., Hou, Z. and Gong, W., 2010. Automated road sign inventory system based on stereo vision and tracking. Computer-Aided Civil and Infrastructure Engineering 25, pp. 468–477.

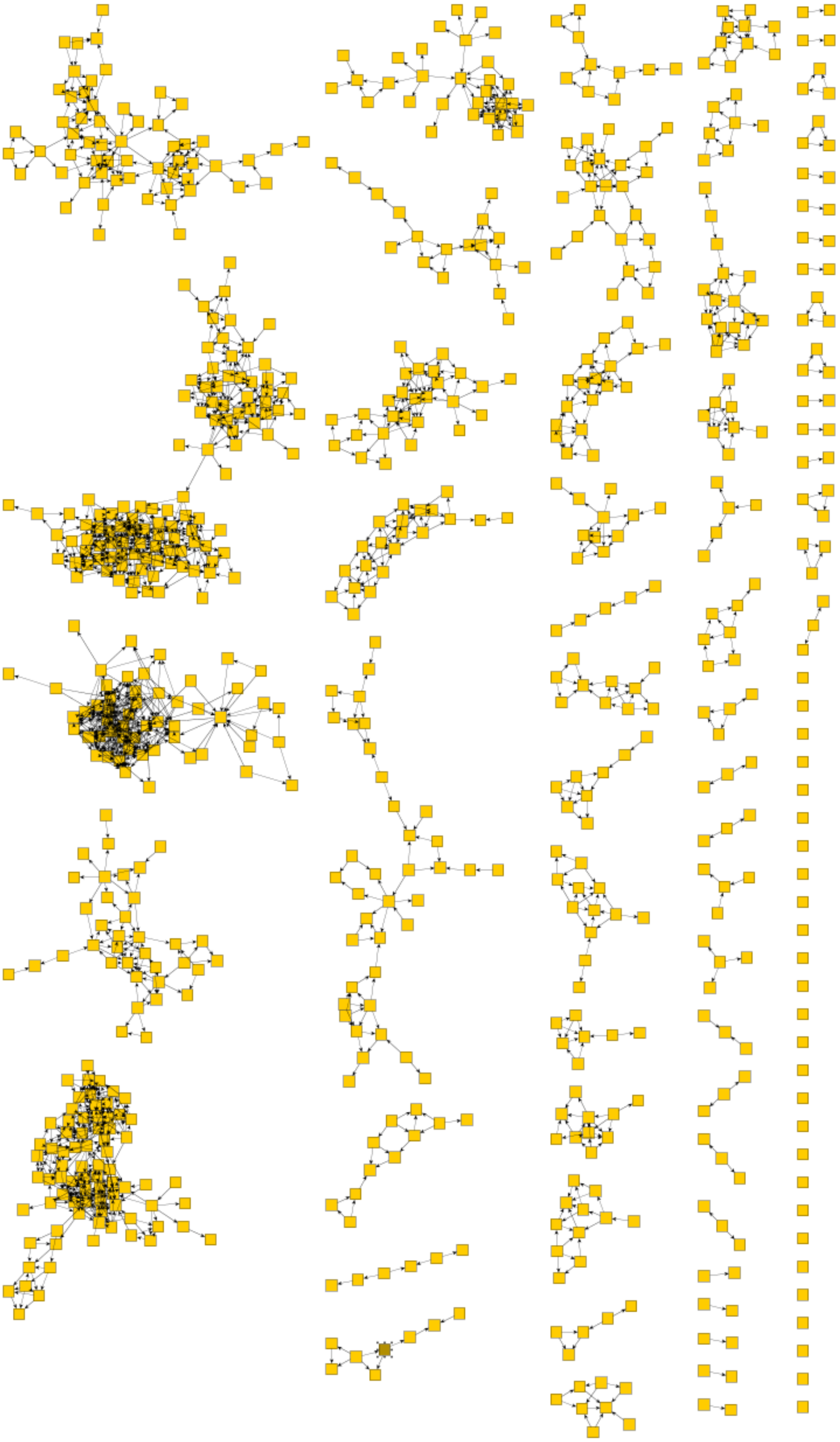

Figure 2: Consistency graph for around 1000 triangular 2D traffic signs detections. Nodes are 2D detections of 3D signs, and an edge exists if the detections corresponding ton the connected nodes are compatible (cf Section 2.2). Large components indicate very ambiguous clustering situations. Our method aims at partitioning the connected components in a minimum number of cliques of minimum dissimilarity.