A Game-Theoretical Approach for Joint Matching of Multiple Feature throughout Unordered Images

Luca Cosmo, Andrea Albarelli, Filippo Bergamasco, Andrea Torsello
Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia, Venice Italy
http://www.dais.unive.it

Emanuele Rodolà, Daniel Cremers
Department of Computer Science
Technische Universität München, Garching, Germany
http://vision.in.tum.de

***Abstract*—Feature matching is a key step in most Computer Vision tasks involving several views of the same subject. In fact, it plays a crucial role for a successful reconstruction of 3D information of the corresponding material points. Typical approaches to construct stable feature tracks throughout a sequence of images operate via a two-step process: First, feature matches are extracted among all pairs of points of view; these matches are then given in input to a regularizer that provides a final, globally consistent solution. In this paper, we formulate this matching problem as a simultaneous optimization over the entire image collection, without requiring previously computed pairwise matches to be given as input. As our formulation operates directly in the space of feature across multiple images, the final matches are consistent by construction. Our matching problem has a natural interpretation as a non-cooperative game, which allows us to leverage tools and results from Game Theory. We performed a specially crafted set of experiments demonstrating that our approach compares favorably with the state of the art, while retaining a high computational efficiency.**

I. Introduction

Jointly matching multiple objects is an active topic of research in computer vision, graphics, and pattern recognition, and it naturally arises in several different contexts. The problem of matching multiple images simultaneously originates in the area of Structure from Motion, and can be traced back to early works in this field [1], [2]. However, in its more general formulation, different instances of this problem also arise in other fields of research. In 3D shape analysis, it translates to the problem of finding dense maps between deformable shapes in a collection [3], [4]; when the surfaces are partially overlapping and they are only allowed to undergo rigid transformations, it is known as the problem of multi-view registration [5], [6]; finally, the same concept appears in the graph-theoretical community under the more abstract guise of multiple graph matching [7], [8]. Common to all these approaches is the requirement that the correspondence between the different objects should be "consistent"; a natural criterion in this sense is *cycle-consistency* or *transitivity* [9]. Namely, such criterion requires that compositions of maps across cycles in the object collection should approximate the identity map.The dominant approach, so far, has been to compute point-to-point maps [10], [11] between pairs of objects in isolation; then, the set of pairwise solutions is fed into a global optimizer which adjusts the maps to improve their accuracy and to meet the consistency requirement. These approaches tend to work well when the input maps are accurate, however they are less effective if initial matches are sparse and in the presence of outliers both in the maps themselves (wrong matches) and in the collection (unrelated objects).

In this paper, we introduce a novel technique for matching multiple images simultaneously. Differently from existing approaches, we do not require pairwise maps to be given as input; instead, we operate directly over the space of consistent matches. As such, our method produces matches that are cycle-consistent by construction. Our formulation exhibts also two additional key advantages: the method is *robust*, as it is able to cope with clutter, partial similarity, repeated structures and with outlier images in the collection, moreover, the algorithm is demonstrably *efficient* and can generate valid solutions orders of magnitude faster than other high-accuracy approaches.

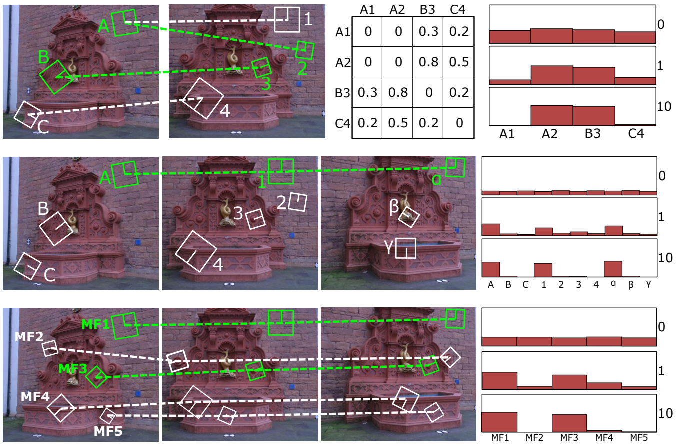
II. Game Theory for hypothesis validation

Evolutionary game theory [12] considers a scenario where pairs of individuals, each pre-programmed with a given strategy, are repeatedly drawn from a large population to play a game, and a selection process allows "fit" individuals (*i.e.*, those selecting strategies with high support) to thrive, while driving "unfit" ones to extinction. The general idea is to model each hypothesis as a strategy and let them be played one against the other according to a fixed payoff function until a stable population emerges. These notions of *hypothesis*, *payoff*, *population* are thus central to our approach.

Definition 1 (Hypothesis). *A fact, derived from observed data, that is assumed to be produced by the phenomenon to be characterized. We define $H = \{1, \cdots, n\}$ be the set of n available hypotheses derived from data.*

Definition 2 (Payoff). *A measure of the degree of reciprocal support between two hypotheses. The payoff is usually expressed as a function $\pi(i, j) : H \times H \to \mathbb{R}_{\geq 0}$, where i and j are hypotheses. Since payoffs are defined between all the pairs, an alternative notation is the payoff matrix $\Pi = (\pi_{ij})$, where $(\pi_{ij}) = \pi(i, j)$.*

Definition 3 (Population). *A probability distribution $\vec{x} = (x_1, \dots, x_n)^T$ over the strategies H. Any population vector is bound to lie within the n-dimensional standard simplex $\Delta^n = \{\vec{x} \in \mathbb{R}^n \ : \ x_i \geq 0 \text{ for all } i \in 1 \dots n, \ \sum_{i=1}^{n} x_i = 1\}$ The* support *of a population $\vec{x} \in \Delta^n$, denoted by $\sigma(\vec{x})$, is defined as the set of elements chosen with non-zero probability: $\sigma(\vec{x}) = \{i \in O \mid x_i > 0\}$.*



	A1	A2	B3	C4
A1	0	0	0.3	0.2
A2	0	0	0.8	0.5
B3	0.3	0.8	0	0.2
C4	0.2	0.5	0.2	0

Fig. 1. Examples of the three kind of game discussed in the paper (from top to bottom: pairwise, multi-way, track validation). Only a subset of features is shown in order to enhance readability, however shown payoffs and evolutionary process are computed correctly.

In order to find a set of mutually coherent hypotheses, we are interested in finding configurations of the population maximizing the average payoff. Since the total payoff obtained by hypothesis i within a given population $\vec{x}$ is $(\Pi\vec{x})_i = \sum_j \pi_{ij}x_j$, the (weighted) average payoff over all the considered hypotheses is exactly $\vec{x}^T\Pi\vec{x}$.

Unfortunately, it is not immediate to find the global maximum for $\vec{x}^T\Pi\vec{x}$, however local maxima, called Evolutionary Stable State (ESS), can be obtained by letting an initial population vector $\vec{x}$ evolve by means of a rather wide class of evolutionary dynamics called *Payoff-Monotonic Dynamics*. In our implementation we use replicator dynamics which are governed by the following equation:

$$x_i(t+1) = x_i(t)\frac{(\Pi\vec{x}(t))_i}{\vec{x}(t)^T\Pi\vec{x}(t)} \tag{1}$$

A. *Pairwise Game-theoretical Matching*

During the last few years, Game Theory has been adopted to perform hypothesis validation within a wide range of scenarios. This is the case, for instance, of the seminal paper by Albarelli *et al.* [13], where a game-theoretical framework is used for finding correspondences between segments and to perform point-pattern matching; Registration of rigid and deformable shapes have been also addressed in [14] and [15], and object-in-clutter recognition in [16]. Other relevant works are about feature selection [17] and image segmentation [18]. Most of these methods follow a common script:

- A set of initial hypotheses is selected from the solution space of the problem;
- A payoff function is defined between each pair of hypotheses in order to express the level of mutual validation;
- An hypotheses population, represented as a probability distribution, is evolved through some dynamics.

A highly relevant application, is the one presented in [10], called Game Theoretical Matcher (GTM). Here, the initial hypotheses are putative matches between features and the selection process operates according to a payoff that accounts for how well the affine transformation induced by one match can be applied to a competing hypothesis. This specific scenario is particularly interesting because it addresses the same problem of this paper, *i.e.*, to extract coherent feature matches between images in order to enable 3D recovery. Moreover, it also adopts a game theoretical framework, albeit it uses it in a very different way.

In the top row of Fig. 1 we show an example of a *pairwise* matching game performed according to [10]. The example is simplified, since it includes only a small set of initial hypotheses. A total of three SIFT features are extracted from the first image (labeled $A - C$) and a total of four from the second one (labeled $1 - 4$). Subsequently, a set of four initial hypotheses is created, corresponding to the four most likely matches according to the feature descriptor. A payoff is computed between each pair of hypotheses, producing matrix Π. Note that matches $A1$ and $A2$ have zero compatibility since they break the one-to-one relation between source and target images. Also, $A2$ and $B3$, although wrong, exhibit a higher compatibility between them than with respect to $C4$ since they agree on a similar feature transformation. In the rightmost part of the figure we show the evolution of an initial random population after one and ten iterations of the validation game. In this case, $A2$ and $B3$ are the only surviving strategies, and can be considered the final matches selected by this specific game theoretical matcher. Note that in this example the pairwise matching process fails to recover the correct matches $A1$ and $C4$. This is due to the limited mutual support that the individual matches can establish in the simple pairwise setting, which may give rise to visual ambiguities and is in fact more prone to the presence of structured noise in the images and to random outliers.

In the next section we will propose an approach to sidestep these limitations by leveraging the information contained in the *whole* collection of images.

III. Multi-feature matching games

Feature matching methods exploiting the game-theoretical framework (*e.g.* [10], [15], [16]) usually consider matches between two images as hypothesis and validate them to find the set of assignments that are the most suitable. As discussed in Sec. I, these pairwise solutions can in principle be recomposed so as to form coherent tracks. We propose exactly the opposite approach: Validating multiple correspondences of the same feature among several images by finding a mutually coherent set according to the feature descriptor. In this view, each game will produce just a *single* multi-feature match rather than a set of pairwise matches.

The motivating idea is that, if n images are available, searching for a set of landmarks exhibiting strong compatibility between each pair should result in a much stronger validation of the feature descriptors, which are required to be repeatable over all the $n(n-1)/2$ pairs. This, in turn, helps to avoid wrong matches resulting from random descriptor similarities that can easily happen if only two images are involved. Furthermore, the obtained tracks will be inherently multi-way, ruling out the need for an explicit merge of pairwise correspondences. Finally, when the single tracks are grouped together, enforcing their geometrical coherence throughout several images will benefit from the increased dimensionality.

These two steps (multi-feature selection and multi-feature validation) will be performed through two separate games, using different hypothesis sets and payoff functions, which we describe in the next two sections.

A. Multi-feature selection game

The goal of this game is to find features that agree on their descriptor throughout the whole image sequence (or at least images where the feature is visible). The result will be the extraction of a single track characterized (hopefully) by high reliability. Note that, differently from [10], no geometric information is used as we just rely on the descriptor vectors.

First, we select query candidates from all the feature points of all the images. We do this by estimating the density of the features in the feature-descriptor space and selecting low-density (*i.e.*, uncommon) descriptors under the assumption that these descriptors are more distinctive. The density estimation is performed non-parametrically through a k-nn density estimation: Let x be a point in the descriptor space, and $B_k(x)$ be the minimal ball centered at x containing k the descriptors of k features. Then the k-nn density at x is $d_k(x) = \frac{k}{|B_k(x)|}$, where $|A|$ is the volume of the set A. With the density at hand, we select the N least common features (lowest density) as query points and create a selection game for each of them. Given a query point, the selection game is as follows:

Hypotheses: for each image we extract a fixed proportion p of the features extracted from that image that are closest (in the descriptor space) to the query point.

Payoff: The payoff is defined as a Gaussian over the descriptor distances. This makes sense, since we are considering descriptors to be originated from the same phenomenon and their drift can be reasonably modeled as a non-biased random error with standard deviation σ_a. Note, however, that features from the same image are incompatible with one another, thus their payoff is set to 0 regardless of their descriptor:

$$\pi(i,j) = \begin{cases} \frac{1}{\sigma_a\sqrt{2\pi}} e^{-\frac{|D(f_i)-D(f_j)|^2}{2\sigma_a^2}} & \text{if } I(f_i) \neq I(f_j) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Parameter σ_a can be used to tune the expected drift, which is clearly dependent on the feature descriptor adopted, on its dimensionality and, finally, on the strictness that we want to enforce on the selection process. Smaller σ_a values will result in a more selective process and vice versa. Note that by setting payoff 0 between features coming from the same image we are enforcing a very important theoretical property of our method.

Theorem 1. *The support of a population evolved through a* Feature Selection Game *contains at most one feature from each image.*

Proof. If features i and j belong to the same image, then $\pi_{ij} = 0$. Assume that, after playing the game, $\vec{x}$ reaches an ESS and both $\vec{x}_i$ and $\vec{x}_j$ are not 0, thus we have $i, j \in \sigma(\vec{x})$. Let $\vec{y} = \delta(\vec{1}_i - \vec{1}_j) + \vec{x}$, where $0 < \delta \leq x_j$ and $\vec{1}_k$ is a vector with entry k equal to one and all other entries equal to zero. Note that $\vec{y}$ is a best reply to $\vec{x}$, in fact

$$\vec{y}^T\Pi\vec{x} = \delta(\vec{1}_i - \vec{1}_j)^T\Pi\vec{x} + \vec{x}^T\Pi\vec{x} = \vec{x}^T\Pi\vec{x}\,,$$

with $(\vec{1}_i - \vec{1}_j)^T\Pi\vec{x} = 0$ by Nash condition on $\vec{x}$. However,

$$\begin{aligned}(\vec{x}-\vec{y})^T\Pi\vec{y} &= -\delta(\vec{1}_i - \vec{1}_j)^T\Pi\left[\vec{x} + \delta(\vec{1}_i - \vec{1}_j)\right] \\ &= -\delta^2(\vec{1}_i - \vec{1}_j)^T\Pi(\vec{1}_i - \vec{1}_j) \\ &= -\delta^2(\pi_{ii} + \pi_{jj} - \pi_{ij} - \pi_{ji}) = \delta^2(\pi_{ij} + \pi_{ji}) \leq 0\,,\end{aligned}$$

which contradicts the evolutionary stability of $\vec{x}$. □

This theorem is key to the feasibility of the proposed approach since it allows to avoid to include in the final solution two or more hypotheses originating from the same image (which is indeed the case, for instance with highly repeated structure such as walls, facades, and many man-made objects).

In the central row of Fig. 1 we show a simplified example (in practice the method should be used with many more features and images) of the multi-way feature selection game using the same two images from the first row and by adding a third one with three additional features. The payoff matrix (not shown in the figure) has size 10, since we have a total of 10 features coming from the three images, and it exhibits three zero blocks on the diagonal, due to the incompatibilities among features from the same images. On the right part of the figure we show the evolution of a random initial population, plotting the starting distribution and the result after respectively 1 and 10 rounds. As expected, the population evolve to a final state with no more than one feature per image.

This configuration, which we call *multi-feature*, collects the most repeatable instances among all the features close to the query point. Note that the obtained multi-feature ($[A, 1, \alpha]$) includes a match ($A - 1$) which was not selected by the pairwise game (top row). This process can be repeated for each query point resulting in the computation of exactly N distinct multi-features.

In our experiments we used SIFT descriptors compared using the SIFT distance [19] and we set $k = 10$, $N = 2000$ and $p = 20\%$.

B. Multi-feature validation game

While the tracks extracted in the selection game are highly similar from a photometric point of view, their extraction does not enforce any form of mutual geometric consistency among them. We propose a validation scheme that selects the geometrically consistent multi-features by performing an additional game over them.

Hypotheses: The set of multi-features extracted with the selection games. Each multi-feature refers to a single material point, thus it must contain at most one feature from each image. More formally, multi-features are sets $\alpha = \{f_i, i \in 1..n | f_i, f_j \in \alpha \Rightarrow I(f_i) \neq I(f_j)\}$.

Payoff: In order to play this second game, we must define a payoff between multi-features. We differ from GTM [10] as we do not assume the transformations to be locally affine, neither we can perform epipolar validation, since we need to define a payoff between two tracks, and we would need at least 5 to produce a fundamental matrix. Rather, we need to define some property that can be preserved throughout subsequent shots of the same subject and that can be verified between two tracks. Given the 3D position of each tracked point, the distance between two of them would be a suitable measure. Unfortunately, we only know the projection on the image plane of the observed points. However, each feature f_i comes with an observed scale $S(f_i)$ and changes of the depth of the point with respect to a camera would result in inversely proportional changes in the observed scale through a constant k, which is a characteristic of the planar patch responsible for the observed feature. Under the assumption of moderate rotation between views, such constant is related to the size of the original object and can be used to express a point's 3D position: If k is known for the object that generated feature f_i, its position with respect to the reference frame of camera $I(f_i)$ is:

$$P(f_i, k) = \frac{k}{S(f_i)} \begin{bmatrix} U(f_i) & V(f_i) & 1 \end{bmatrix}^T , \tag{3}$$

where $U(f_i)$ and $V(f_i)$ are normalized coordinates of f_i on the image plane of the observing camera $I(f_i)$.

Let us consider features f^i_α and f^i_β extracted from the same image I_i observing two material objects (*i.e.*, belonging to two tracks) α and β. If we know the values of k for such objects, we can compute the estimated length of the segment connecting α and β as $L(f^i_\alpha, f^i_\beta, k_\alpha, k_\beta) = \|P(f^i_\alpha, k_\alpha) - P(f^j_\alpha, k_\beta)\|$. If tracks α and β have no outliers and perfectly accurate feature localization, then the variance σ^2_L of the distance L between the 3D points can be used as a measure of geometric inconsistency that is intrinsically multi-way. However, since k_a and k_b are not known, it is not possible to compute a value for σ^2_L, but we can compute a lower bound by minimizing its value over the unknown parameters. To this end, we have to account for an unrecoverable scale factor in the two patch sizes since we can always trade patch size for depth, thus we fix this scale setting $k^2_\alpha + k^2_\beta = 1$, obtaining the following multi-feature variance:

$$\sigma^2_{\alpha\beta} = \min_{k_\alpha k_\beta} \sigma^2_L \quad \text{s. t. } k^2_\alpha + k^2_\beta = 1. \tag{4}$$

With this we can define the payoff as follows:

$$\pi(\alpha, \beta) = \frac{1}{\sigma_b \sqrt{2\pi}} e^{-\frac{\sigma_{\alpha\beta}}{2\sigma_b^2}} . \tag{5}$$

In order to compute this variance, we define v_α and v_β as long vectors concatenating all the observed 3D points modulo the parameters k_α and k_β:

$$v_i = \left(\tfrac{U(f^1_i)}{S(f^1_i)}, \tfrac{V(f^1_i)}{S(f^1_i)}, \tfrac{1}{S(f^1_i)}, \dots, \tfrac{U(f^n_i)}{S(f^n_i)}, \tfrac{V(f^n_i)}{S(f^n_i)}, \tfrac{1}{S(f^n_i)} \right)^T \tag{6}$$

With these vectors at hand, we note that $||k_\alpha v_\alpha - k_\beta v_\beta||^2 = n(\sigma^2_L + \mu^2_L)$. In order to estimate k_α and k_β we substitute in the computation of μ_L the Euclidean distance between the points with their Manhattan distance, obtaining: $n\mu_L \approx s^T(k_\alpha v_\alpha - k_\beta v_\beta)$, where s is a vector satisfying $s_i = \mathrm{sign}(k_\alpha v_\alpha - k_\beta v_\beta)$. While this approximation is a bit rough, we only use it to estimate k_α and k_β, and not to compute the variance $\sigma_{\alpha\beta}$ directly:

$$n\sigma^2_l \approx (k_\alpha \, k_\beta) \, A \, (k_\alpha \, k_\beta)^T , \; A = \begin{pmatrix} v^T_\alpha E v_\alpha & -v^T_\alpha E v_\beta \\ -v^T_\beta E v_\alpha & v^T_\beta E v_\beta \end{pmatrix} \tag{7}$$

where $E = I - \frac{1}{n} s s^T$. Hence, we estimate k_α and k_β by initializing them to $\sqrt{2}/2$ and iteratively computing s with the current values and re-estimating $(k_\alpha, k_\beta)^T$ as the eigenvector associated with the smallest eigenvalue of A. With the estimated values of k_α and k_β at hand, we can compute the actual Euclidean distances between feature points (modulo global scale) and thus their variance $\sigma_{\alpha\beta}$. as:

$$\sigma_{\alpha\beta} = \frac{1}{n} \sum_{i=1}^{n} L(f^i_\alpha, f^i_\beta, k_\alpha, k_\beta)^2 - \left(\frac{1}{n} \sum_{i=1}^{n} L(f^i_\alpha, f^i_\beta, k_\alpha, k_\beta) \right)^2 . \tag{8}$$

A simplified example of such selection process, performed over tracks obtained with a multi-feature selection game, is shown in the bottom of Fig. 1. Here we have 5 different multi-features, labelled from MF1 to MF5. Features MF1 and MF3 are consistent with respect to the tracked physical point and exhibit a high compatibility level and, as a consequence, also a high payoff. The remaining tracks include at least one mismatched feature, resulting in a low mutual payoff which, in turn, drives them to extinction.

IV. Experimental evaluation

To evaluate the effectiveness of our approach in different practical settings, we performed a wide range of experiments over a selection of standard datasets. Specifically, we employed "dino" and "temple" ring from the Middlebury Multi-View Stereo dataset [21], "fountain-P11" from [22], "house" and "hotel" from the CMU motion database.

This selection has been performed so as to cover several of the nuisance factors commonly found in practical applications, such as bilateral symmetries in the depicted shapes (*e.g.*, "dino ring"), repeated structure, multiple object instances, wide baseline, and so forth. For each dataset, we computed SIFT keypoints [19] and then we used the associated descriptors for the multi-feature selection process described in Sec. III-A.

Before presenting a complete quantitative analysis of the approach, in Fig. 2 (left) we show an example of our method applied to a subset of the "fountain-P11" sequence. In order to better appreciate the complementary role of the two games we have been more loose in the multi-feature selection game, thus allowing more outliers. Indeed, only tracks which are actually consistent from a geometrical point of view survive at convergence of the final evolutionary process and are retained in the final solution. The obtained matches were used to densify the 3D reconstruction created by means of the method proposed in [20]. The resulting model is shown on the right.

Next, we are going to study the sensitivity of the method to its parameters. These include the standard deviations σ_a, σ_b of the two payoff functions (Sec. III-A and III-B), the minimum

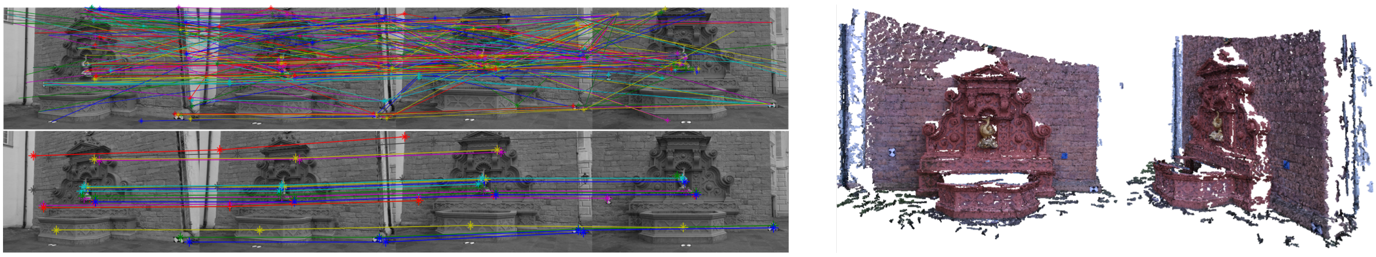


Fig. 2. Left: example on real data of multi-feature matches extracted by several iterations of the multi-feature selection game (first row), and the final solution after the multi-feature validation game (second row). Right: reconstruction obtained by densifying our correspondences using [20].

track length, and the desired number of tracks. In particular, the former two regulate the selective behavior of the inlier selection process and as such directly affect the quality of the extracted correspondence. We analyze this dependency by running our matching pipeline for various choices of σ_a, σ_b. In order to provide a quantitative assessment, we measure the quality of the extracted feature tracks by their average epipolar distance, namely the measure defined as:

$$\epsilon(\alpha) = \frac{1}{|\alpha|^2} \sum_{f_i, f_j \in \alpha} d(f_i, e(f_j)) \,, \tag{9}$$

where α is a given track, $d(f, \ell)$ is the Euclidean distance between point f and line ℓ, and $e(f_j)$ is the epipolar line associated to feature f_j. Notice that we take the average over *all* pairs of features in the track, hence considering the errors induced by both directions (f_i, f_j) and (f_j, f_i).

A track α is deemed *correct* if $\epsilon(\alpha) < 3$ pixels.

For these experiments we used the ”dino” and ”temple” datasets as they come with ground-truth transformations, from which we are able to compute the correct epipolar lines. In Fig. 3 we show the average results obtained for different values of σ_a and σ_b. We can derive a few interesting observations. First, the value of σ_b, which regulates the selectivity of the track validation process, has a limited influence on the final accuracy whenever the candidate tracks from the first game are already accurate enough (*i.e.*, when the multi-way payoff function (2) is made very selective). On the other hand, in order to get a larger collection of longer tracks one has to allow the track generator to be more permissive (larger σ_a); this, inevitably produces a higher outlier ratio and thus requires a more selective validation from the second game (small σ_b).

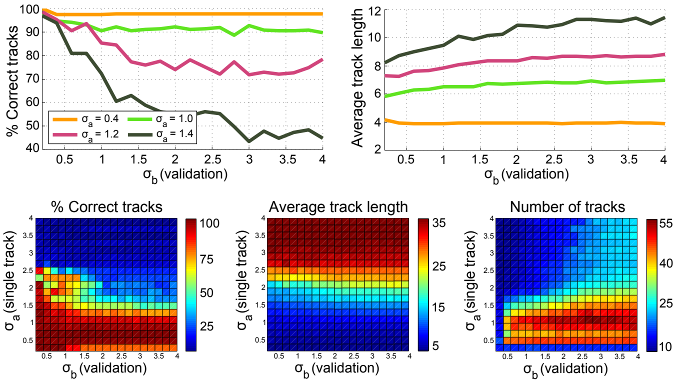


Fig. 3. Sensitivity analysis with respect to different choices of σ for the two payoff functions. The curve plots show a subset of the results reported in the second row, for fixed values of σ_a. A good trade-off between accuracy and selectivity can be obtained, for instance, by setting $\sigma_a = 1$ and $\sigma_b = 0.7$.

We also compared our method with some of the state-of-the art multi-way approaches. MatchLift [23] is a recent technique that works by taking as input noisy matches between pairs of images, and adjusts them so as to extract consistent tracks. It is currently the state-of-the-art within this family of approaches. VocMatch [24] is another recent technique that, similarly to our approach, operates *directly* on the space of consistent multi-features. To the best of our knowledge, this is the only other technique that tackles the problem from this multi-feature perspective.

For the comparison we use the CMU ”house” and ”hotel” data, for which hand-labeled ground truth matches are available for 30 keypoints across the whole sequences. These features are then given as input to each method in order to provide a fair evaluation. Based on the sensitivity analysis we performed in the previous section, for these experiments we set $\sigma_a = 1$ and $\sigma_b = 0.7$. We remark that, since these values were obtained on different datasets, they were not tuned to perform well in the comparisons. Table I reports the precision/recall attained by each method on three variants of the datasets. The first variant (*adj*) consists of 5 subsets of 15 adjacent frames each; in the second variant (*rnd*), the 15 images are selected randomly for each subset (*i.e.*, neighbor information is removed); finally, the third variant (*full*) consists of the full set of images (111 images for ”house”, 101 for ”hotel”).

MatchLift attains almost ideal results for the first variant, confirming previous reports on the same data [23]. However, its performance rapidly decreases as adjacency information is removed, and the method requires an unfeasibly long execution time for it to be applicable on the full sets. Conversely, VocMatch does not give any solution in the first two cases, due to its ”matchability” criterion according to which features

TABLE I
PRECISION/RECALL VALUES (%). BEST RESULTS IN BOLD.

	Ours	**MatchLift**	**VocMatch**
hotel (*adj*)	**99.8** / 87.6	98.9 / 94.7	-
house (*adj*)	99.8 / 83.7	**99.9** / 99.8	-
hotel (*rnd*)	90.9 / 63.4	**95.6** / 28.5	-
house (*rnd*)	93.7 / 62.1	**95.9** / 35.3	-
hotel (*full*)	**100.0** / 10.3	-	99.3 / 3.2
house (*full*)	**100.0** / 13.0	-	98.3 / 6.4

TABLE II
CORRECT TRACKS RATIO / NUMBER OF MATCHES. BEST IN BOLD.

	Ours(1)	**Ours(2)**	**MatchLift**	**VocMatch**
temple	**0.83** / 2.0k	0.25 / 4.4k	0.22 / 1.4k	-
dino	**0.92** / 3.9k	0.68 / 8.5k	0.49 / 4.9k	-
temple'	**0.98** / 26.5k	0.83 / 53k	-	**0.98** / 28.2k
dino'	**0.96** / 25.8k	0.79 / 51.7k	-	**0.96** / 5.6k

TABLE III
COMPARISON OF RUNTIMES (SEC) ON IMAGE SETS OF INCREASING SIZE.

	10	**20**	**30**	**40**	**100**
Ours	9.72	12.30	13.02	13.62	23.65
MatchLift	23.65	120.12	336.83	795.03	-
VocMatch	504.43	503.12	505.21	505.87	506.25

should be 1) unique in each image, and 2) rare across the collection. For this reason, the only extracted tracks arise from the full sets, where they are highly accurate but at the same time very sparse. We would like to clarify that VocMatch is indeed designed to work with very big datasets (thousands of images) [24], but was included in the comparison by virtue of its simultaneous nature. The results reported in Table I suggest two key properties of our method. First, it is able to extract most of the correct feature tracks in both ordered and unordered settings; in fact, our pipeline does *not* take into account any sequential information. Second, it is able to deal with medium-sized unordered datasets efficiently and accurately, where other approaches tend to fail. Notice how our technique recovers exact feature tracks (100$\%$ precision) on the full sets, with low recall due to its inlier selection behavior.

As a further experiment we provide a comparison between the three approaches when adopted as full-fledged pipelines, *i.e.*, we do not provide the same input features for each method. For this experiment we use two parametrizations of our method, namely: (1) $\sigma_a = 0.5, \sigma_b = 0.5$, and (2) $\sigma_a = 1, \sigma_b = 1.5$. We run the experiments on the "dino" and "temple" datasets. Due to the lack of ground-truth matches, performance is evaluated by the average epipolar distance defined in (9). In Table II we report, alongside the fraction of correct tracks, the number of matches (counted pairwise) extracted by each method.

Finally, in order to give an idea of the practical applicability of our method, in Table III we provide a comparison of runtimes obtained on collections of different sizes. Note that the reported times are representative of the pipelines taken as a whole, *i.e.*, from feature extraction to the generation of the final set of tracks. Our method is orders of magnitude faster than the state of the art on moderately sized collections.

V. CONCLUSIONS

In this paper we introduced a novel multi-feature matching approach for finding consistent feature correspondences in unordered image collections. Differently from existing literature, we cast the problem as a simultaneous optimization over the set of input images, without relying on pairwise feature matches to be given as input. The resulting correspondences are consistent by construction, and do not require a post-processing step in order to eliminate outliers. The problem is formulated by using standard game-theoretical tools, enabling strict guarantees on the structure of the solutions. We demonstrate the practical effectiveness of our method on standard datasets, where we outperform the state of the art in efficiency and matching accuracy.

REFERENCES

[1] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?"," in *Proc. ECCV*, 2002, pp. 414–431.

[2] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. J. Comput. Vision*, vol. 80, no. 2, pp. 189–210, Nov 2008.

[3] Q.-X. Huang and L. Guibas, "Consistent shape maps via semidefinite programming," *Computer Graphics Forum*, vol. 32, no. 5.

[4] L. Cosmo, E. Rodolà, A. Albarelli, F. Mémoli, and D. Cremers, "Consistent partial matching of shape collections via sparse modeling," *Computer Graphics Forum*, 2016.

[5] K. Pulli, "Multiview registration for large data sets," in *Proc. 3DIM*, 1999, pp. 160–168.

[6] A. Torsello, E. Rodolà, and A. Albarelli, "Multiview registration via graph diffusion of dual quaternions," in *Proc. CVPR*, 2011, pp. 2441–2448.

[7] D. Pachauri, R. Kondor, and V. Singh, "Solving the multi-way matching problem by permutation synchronization," in *Proc. NIPS*, 2013, pp. 1860–1868.

[8] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. M. Chu, "Graduated consistency-regularized optimization for multi-graph matching," in *Proc. ECCV*, vol. 8689, 2014, pp. 407–422.

[9] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proc. ICCV*, Dec 2013, pp. 3248–3255.

[10] A. Albarelli, E. Rodolà, and A. Torsello, "Imposing semi-local geometric constraints for accurate correspondences selection in structure from motion: A game-theoretic perspective," *Int. J. Comput. Vision*, vol. 97, no. 1, pp. 36–53, 2012.

[11] Z. Liu, P. Monasse, and R. Marlet, "Match selection and refinement for highly accurate two-view structure from motion," in *Proc. ECCV*, 2014, vol. 8690, pp. 818–833.

[12] J. Weibull, *Evolutionary Game Theory*. MIT Press, 1995.

[13] A. Albarelli, S. Rota Bulò, A. Torsello, and M. Pelillo, "Matching as a non-cooperative game," in *Proc. ICCV*, Sept 2009, pp. 1319–1326.

[14] A. Albarelli, E. Rodol, and A. Torsello, "Fast and accurate surface alignment through an isometry-enforcing game," *Pattern Recognition*, vol. 48, no. 7, pp. 2209–2226, 2015.

[15] E. Rodolà, A. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello, "A game-theoretic approach to deformable shape matching," in *Proc. CVPR*, June 2012, pp. 182–189.

[16] E. Rodolà, A. Albarelli, F. Bergamasco, and A. Torsello, "A scale independent selection process for 3d object recognition in cluttered scenes," *Int. J. Comput. Vision*, vol. 102, no. 1-3, pp. 129–145, 2013.

[17] A. Albarelli, E. Rodolà, and A. Torsello, "Loosely distinctive features for robust surface alignment," in *Computer Vision ECCV 2010*, 2010, vol. 6315, pp. 519–532.

[18] B. Ibragimov, B. Likar, F. Pernus, and T. Vrtovec, "A game-theoretic framework for landmark-based image segmentation," *Medical Imaging, IEEE Transactions on*, vol. 31, no. 9, pp. 1761–1776, Sept 2012.

[19] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, vol. 2, 1999, pp. 1150–1157.

[20] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[21] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. CVPR*, vol. 1, June 2006, pp. 519–528.

[22] C. Strecha, W. von Hansen, L. van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. CVPR*, June 2008, pp. 1–8.

[23] Y. Chen, L. Guibas, and Q.-X. Huang, "Near-optimal joint object matching via convex relaxation," in *Proc. ICML*, 2014, pp. 100–108.

[24] M. Havlena and K. Schindler, "Vocmatch: Efficient multiview correspondence for structure from motion," in *Proc. ECCV*, 2014, pp. 46–60.