GUIDED SEARCH CONSENSUS: LARGE SCALE POINT CLOUD REGISTRATION BY CONVEX OPTIMIZATION

Manuel Marques and João Costeira

Instituto Superior Tecnico - Av. Rovisco Pais, 1049-001 Lisbon, Portugal

ABSTRACT

In this paper we propose a point matching algorithm that computes correspondences between images and/or 3D objects in affine camera settings. We formulate the point correspondence problem by minimizing an error function over the set of all binary decisions. This function has two components: a geometric error (akin to retro-projection error) and an image dissimilarity component. Originally a combinatorial problem we obtain its exact solution through a convex relaxation. Hinging on a recent theorem, this solution is derived by carefully designing the minimizing function. The large scale optimization problem is handled by an extremely fast algorithm based on Nesterov's projected gradient method. Experimental results show both computational efficiency and high robustness to large deviations and we demonstrate that it can cope with very hard real situations such as scenes with repetitive patterns. The methodology can be applied seamlessly to both 2D or/and 3D cameras.

***Index Terms*—** Image matching, Point correspondence, Structure-from-Motion, Convex optimization

1. INTRODUCTION

This article proposes an optimal matching framework to solve the point correspondence problem for affine uncalibrated 2D and/or 3D cameras. The solution is framed in the convex optimization domain thanks to a recent mathematical result[**?**]. The framework applies seamlessly to a wide variety of fundamental problems in computer vision, such as pose estimation (2D-3D), image to image matching (2D-2D) or range-data registration (3D-3D). We demonstrate its capabilities through a set of complex experiments that few other approaches can handle. The methodology, nicknamed GuiSaC, takes into account global geometric criteria, as well as photometric or local image descriptors. Furthermore, we derive a computationally efficient algorithm, specially for large scale datasets. Figure 1 illustrates one challenging example for most image matching algorithms. The foreground of fig 1 shows in detail the kind of patterns that makeup the large artistic tile panel on the background. These panels are pieces of decoration in palaces and historic buildings. In the context of image retrieval and recognition applications, the goal here is localizing the core pattern in the large panel. In figure 2 we show the template with a superimposed green mesh. The nodes of this mesh are the location of a set of SIFT features, that must be matched to the panel on the right. The two small panels on the right show the results of our method (red mesh) against a local-feature approach (purple mesh). The hardship of the task is illustrated by the completely wrong solution found by the robust method based on local information, despite its low cost (high local similarity of the matched points). On the

Fig. 1. Typical application: Searching for repetitive patterns on a tile panel.

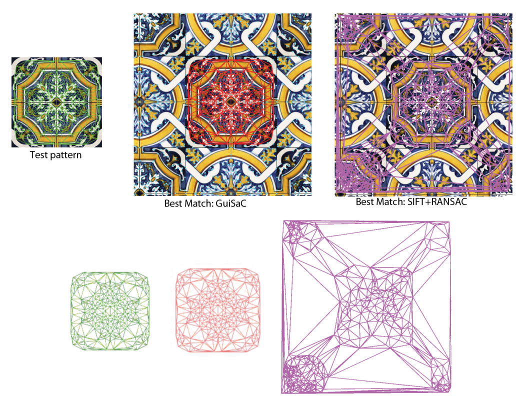

Fig. 2. Left:Template pattern with selected features (green mesh). Middle: Matched points obtained by GuiSaC. Right: Matched points from SIFT features and RANSAC. Below the meshes without background for better visualization

This work was partially funded by FCT Project PRINTART (PTDC/EEA- CRO/098822/2008).

contrary, GuiSaC found the correct pattern without any recurrence to combinatorics, sampling or decision making. This example highlights the fundamental role that geometry plays in image/pattern matching. It seems evident that it is impossible to solve this problem relying solely on local properties, neglecting their relative location in the image.

1.1. Previous Work

Successful approaches in point matching tend to be clustered around two main classes of algorithms: a)those that rely on very robust image descriptors followed by a simple local matching stage -e.g. Nearest Neighbor- and b) algorithms that provide global solutions to a class of combinatorial problems. Examples of the first class are headed by David Lowe's SIFT features[1], $Shapecontext$ by Belongie et.al[2] or [3], where feature descriptors and geometric constraints (like proximity) are combined with search strategies that lead to "mixed" schemes. Geometric information (rigidity, homography) are used to validate the matching, inherently suboptimal. A profound improvement was produced when Kolmogorov-Zabih introduced the graph-cut technique to solving quadratic binary problems[4, 5]. The main inconvenience here is the restricted number of admissible functions that can be efficiently optimized: quadratic functions with negative values on the off-diagonal elements of the Hessian matrix. Recently, [6] uses concave programming, such as [7], to find the registration (rotation and translation) between two 2D point clouds. Here, authors replace the intrinsic non-linear observation model by a linearized version. Because the linearization depends on each rotation angle, they have to solve a sequence of optimization problems to covering the search space. Along the same line, [8] also linearizes the model (rotation matrix), formulating a ℓ_1-norm program, solved by linear programming. The orthogonal constraints defining a rotation matrix are relaxed to linear constraints which relax the unit circle to the inscribed square. Besides the disadvantage of the approximations, it is tailored for 2D problems.

1.2. Contributions

This article contributes with a general framework that optimally computes correspondences between point clouds of any dimension that are related by affine maps. Previous works [9, 10] enables the matching between two point clouds with same number of points. In this paper, we introduce a methodology to deal with outliers.

Correspondences are the result of a search for the permutation matrix that minimizes a global function,

$$\begin{array}{rl} \mathbf{P}^* = \arg\min_{\mathbf{P}} & f_{geom}(\mathbf{P}) + f_{similar}(\mathbf{P}) \\ \text{s.t.} & \mathbf{P} \in \mathcal{P} \end{array} . \quad (1)$$

The first term of this function take into account geometric properties of the point cloud and de second term local image/feature similarities. Despite the convexity of the cost function ($f_{similar}$ will be a linear function and f_{geom} is a quadratic function with positive definite Hessian), problem (1) is non-convex because the domain is integer. However the theorem proved in [11], demonstrates that the search space can be made convex by relaxing the set of permutation matrices $\mathbf{P} \in \mathcal{P}$ to its convex-hull, $\mathbf{P} \in \mathbf{D}_s$, the set of doubly-stochastic matrices.

The convex function was designed in such a way that its suitable for Nesterov's [12] or Barzilai-Bowein [13] projected gradient algorithms, which provably converge, sometimes with superlinear convergence rate.

We also contribute with a procedure, named GuISaC, which exploits convergence properties to speed up the minimization process.

2. MODELING THE OBSERVATIONS

The image projection of a 3D point-cloud (3D shape) is modeled by the bilinear model (see [14, 15]):

$$\lambda_i \begin{bmatrix} u_i \end{bmatrix} = \mathbf{Q}\mathbf{X}_i \quad (2)$$

$$\begin{bmatrix} \lambda_1 \dots \lambda_N \\ \lambda_1 \dots \lambda_N \\ \lambda_1 \dots \lambda_N \end{bmatrix} \odot \begin{bmatrix} u_1 \dots u_N \\ v_1 \dots v_N \\ 1 \dots 1 \end{bmatrix} = \Lambda \odot \mathbf{W} = \mathbf{Q}\mathbf{X} \quad (3)$$

where $\mathbf{Q}$ is the projective camera matrix, $\mathbf{X}$ the homogeneous 3D point coordinates and λ_i the projective depth. Considering the object length is small when compared to the distance to the camera (small variance for λ_i), the above model simplifies to the bilinear (affine) observation model,$\mathbf{W} = \lambda^{-1}\mathbf{Q}\mathbf{X} = \mathbf{M}\mathbf{S}$where matrix $\mathbf{M}$ and $\mathbf{S}$ are the 2×4 camera/motion and $4 \times N$ shape matrices respectively and $\mathbf{W}$ the $2 \times N$ matrix of the image point coordinates. Consider now the pose estimation problem where 3D shape is known but the affine projection $\mathbf{M}$ is unknown.The columns of $\mathbf{W}$(image coordinates) when captured in real situations, are not in correspondence to columns of $\mathbf{S}$(object coordinates). Instead of the ideal $\mathbf{W}$, the real observation matrix $\overline{\mathbf{W}}$ has its columns permuted by an unknown permutation modeled by

$$\mathbf{W} = \overline{\mathbf{W}}\mathbf{P} \Leftrightarrow \mathbf{M}\mathbf{S} = \overline{\mathbf{W}}\mathbf{P}, \quad (4)$$

where $\mathbf{P}$ belongs to $\mathcal{P}$, the set of permutation matrices. Elements of this set are defined by $\sum_i p_{ij} = 1, \sum_j p_{ij} = 1, p_{ij} \in \{0, 1\}$. If $p_{ij} = 1$ point i in the image is assgined to point j in the model and only one assignment per point. Algebraically speaking, equation (4) states that rows of $\overline{\mathbf{W}}\mathbf{P}$ must lie on the row subspace $range(\mathbf{S})$. Thus, in a noiseless case, the following equality holds:

$$\overline{\mathbf{W}}\mathbf{P}\mathbf{S}^{\perp} = 0, \mathbf{P} \in \mathcal{P} \quad (5)$$

where $\mathbf{S}^{\perp}$ is a basis for the orthogonal space to $range(\mathbf{S})$. In a real situation, the image data contains a possibly large set of outliers which implies that our data should have a block structure to allow outlier elimination.

$$\underbrace{\begin{bmatrix} \overline{\mathbf{W}}_{inl} \overline{\mathbf{W}}_{outl} \end{bmatrix}}_{\overline{\mathbf{W}}} \underbrace{\begin{bmatrix} \mathbf{P}_1 & A_1 \\ A_2 & \mathbf{P}_2 \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} \mathbf{S}_1^{\perp} \\ 0 \end{bmatrix}}_{\mathbf{S}^{\perp}} = 0, \mathbf{P} \in \mathcal{P} \quad (6)$$

If matrix $\mathbf{P}_1$ is a permutation, it constraints $A_1 = A_2 = 0$ and $\mathbf{P}_2$ should be also a permutation. The conditions for unique solution in the relaxed domain still hold, except if one situation occurs: The coordinate of the correct image points are generated by a convex combination of outliers.In this case, the solution could be incorrect if a inlier is located in the convex hull of a set of outliers and both the inliers and outliers have very similar image properties.

3. SOLVING FOR THE OPTIMAL CORRESPONDENCE

Our goal is to sort image coordinates of $\overline{\mathbf{W}}$ and eliminate outliers such that its rows lye on the range of shape space $range(\mathbf{S})$ and have high image similarity. These principles take concrete form in the following optimization problem:
Problem 1

$$\begin{aligned} \mathbf{P}^* = \; & argmin \; \left|\left|\overline{\mathbf{W}}\mathbf{P}\mathbf{S}^{\perp}\right|\right|_F^2 + \alpha \sum_{ij} C_{ij} p_{ij} + \\ & + \tfrac{c}{2} \left|\left|\mathbf{P}\mathbf{1} - \mathbf{1}\right|\right|_F^2 + \tfrac{c}{2} \left|\left|\mathbf{P}^T\mathbf{1} - \mathbf{1}\right|\right|_F^2 \\ s.t. \quad & p_{ij} \geq 0 \end{aligned} \tag{7}$$

where $\mathbf{1}$ is a column of 1's. Referring to equation 1, f_{geom} and $f_{similar}$ are represented by the first and second term of Problem 1. The third and forth terms are penalties that, in conjunction with constraint $p_{ij} \geq 0$, make $\mathbf{P}$ a doubly-stochastic matrix. The value C_{ij} is the appearence/color dissimilarity between features i and j.

Note that we transformed a hard problem on the set of permutation matrices to a minimization over the set of positive matrices. This design (a simple constraint set) enables very simple and efficient algorithm implementation.

3.1. Large Scale Optimization Algorithm for Point Correspondence

The above solution to the correspondence problem increases in size quite fast: the number of variables grows with N^2. Consequently, Newton-like algorithms are prohibitive in terms of storage (the Hessian requires N^4 entries) and number of operations (large matrix inversion). The large number of constraints and the existence of a inequality, naturally points to Nesterov's *Optimal Gradient*, the provably best in terms of convergence rate and complexity. For the cost function $f(P)$ in Problem 1, the algorithm is given by algorithm 3.1

Algorithm 3.1: Nesterov's *Optimal Gradient*
Searching for $\mathbf{P}^* = argmin\{f(P)\}, \;\; s.t. \;\; p_{ij} \geq 0$

1 - Initialize $P_0 \in \mathbb{R}^{N \times N}$ and set $P_0 = y_0$
2 - Repeat for $k = 1, 2, \ldots$
3 - $\quad P_k = proj(y_{k-1} - \frac{1}{L_k} \nabla f\left(y_{k-1}\right))_+$
4 - $\quad y_k = P_k + \frac{k-1}{k+2}\left(\mathbf{P}_k - \mathbf{P}_{k-1}\right)$

The $proj()$ operator is quite simple since projecting a matrix on the set of positive matrices amounts to $P = proj(X)$: $p_{ij} = max(0, x_{ij})$. The performance of Nesterov's method hinges on two key issues: Computing constant L_k (Lipschitz constant) and projecting $\mathbf{P}_k$ on the constraint set. In our case, constant L is the largest singular value of the Hessian matrix of $f(x)$ and $proj(x)_+$ is the projection of x in $\mathbb{R}_0^+$. Due to the careful design of f the $N^2 \times N^2$ Hessian matrix H is constant and L can be easily computed.

Speeding Up the Search: In the case of object pose, we can compute the camera model (matrix $\mathbf{M}$) if 4 hypothetical matches are known. A simple model checking procedure evaluates how good is the fit. Unlike RANSAC procedure, which selects hypothetical matches randomly, we have a convex problem and that convergence is smooth towards an integer minimizer. Thus, we can monitor the set of (at least) 4 points that converges fastest (p_{ij} is approaches 1), such as the guided search strategy (Algorithm 3.1) shows.

Algorithm 3.1: Guided Search Consensus (GuiSaC)
Searching for $\mathbf{P}^* = argmin\{f_{geom}(P)\}, s.t. p_{ij} \geq 0$

1 - Initialize $\mathbf{P}$, $k = 0$
2 - While $ErrorCriteria(\mathbf{P}_k) > Thresh$
3 - $\quad$ Compute $\mathbf{P}_{k+1}$ using Nesterov's iteration (Alg 3.1)
4 - $\quad$ If at least 4 "reliable" matches exist
$\quad\quad$ Compute pose M and project model into image $\mathbf{MS}$
$\quad\quad$ Using Nearest Neighbours get new match $\mathbf{P}_{NN}$
$\quad\quad$ If $f_{geom}(P_{NN}) < f_{geom}(P_{k+1})$
$\quad\quad\quad$ then $P_{k+1} = P_{NN}$
$\quad\quad$ end
$\quad$ end
5 - $\quad$ k=k+1
end

This procedure fits naturally any optimization algorithm.

4. EXPERIMENTAL RESULTS

In this section we will test the methodology and quantify performance indicators. We will indicate "nominal" performance and synthetic experiments demonstrate empirical evidence of the gains. Experiments with real data were intended to be very challenging to any correspondence algorithm and we used a (small) known dataset for some reference with other works.

4.1. Complexity and convergence

The worst case scenario of Nesterov's algorithm is the following: **Memory requirements**: If N points must be matched, the largest data structures are the estimate of $\mathbf{P}_k$ and the gradient which are both square matrices with N^2 elements. **Convergence**: If E is the maximum allowed error to the minimum of $f_{geom}(\mathbf{P}^2)$, Nesterov's algorithm achieves this error in $k = \mathcal{O}(1/\sqrt{E})$ iterations. **Complexity**: The number of operations per iteration amounts to a $N \times N$ matrix multiplication. We will show GuiSaC has an exponential improvement over Nesterov's. Experiments with synthetic data were made

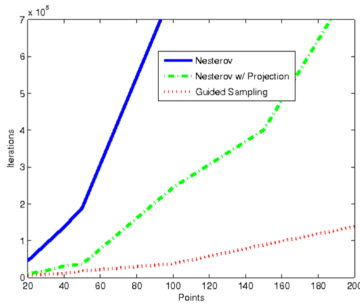

Fig. 3. Number of iteration with problem dimensionality. GuiSac exhibits a relatively constant convergence speed regardless of size of data.

to provide a quantitative evaluation on the rate of convergence and robustness to noise. Convergence speed was tested by running 200 random experiments for a varying number points. To convey a correct perception of the gains introduced by the Spacer Step (model check in Algorithm 3.1) we plot the performance indicators of other variations of the methodology, namely

- Nesterov: Pure gradient algorithm.
- Nesterov w/ Projection: Similar to Algorithm 3.1 with a a new initialization instead of the Spacer Step. Every other m iterations, we compute the closest permutation matrix to the current estimate using the Hungarian method. If the global cost is lower we reinitialize the current estimate with this projection.

Figure 4.1 confirms GuiSac's superior resilience to data size and the much faster convergence rate introduced by the camera estimation and Nearest Neighbour match. Nevertheless, the first remark should be about the extremely fast initial convergence of Nesterov's Gradient algorithm. Comparing the two "truncation" methods mentioned before, we verify that the proposed algorithm is computationally cheaper than Nesterov's with projection. Since the algorithm tends to obtain correct correspondences for some points quite early, the computed affine transform with a such subset of points finds all correspondences much faster.

4.2. Real experiments

In order to benchmark our approach, we use pairs of images from Youtube videos, with animals in their natural habitats (butterfly and fish) as in [8] (see fig. 4). Top images in 4 show the model data matrix $\mathbf{W}$. Points are detect by a feature extractor. The red mesh which connects all selected points is used for visualization purposes only. In the bottom row, the green mesh shows the result of the point assignments found by the algorithm. Outliers are indicated by small red circles. Remark that, in both datasets, there are more than 60% of outliers. The RGB values are used to measure the similarity between points (the c_{ij} are the absolute difference of pixel color).

Fig. 4. 2D shapes with some local deformations. Top: Red mesh denotes the object model. Bottom: Image data $\overline{\mathbf{W}}$ with outliers (red circles). The green mesh is the estimated position of the object points

Despite some local deformations of their shape, correspondences were obtained with the global geometric constraint alone without the need for inclusion of extra knowledge. In a close look, we can observe that few points are off their correct location by a few pixels. This is due to the inaccurate feature extraction mechanism and also to the fact that we used objects with homogeneous areas, therefore with high color similarity in neighbouring points. However even with these harsh conditions the global geometric shape of the object is found and the matching never produced gross mistakes.

5. CONCLUSIONS AND FUTURE WORK

In this article we presented a point matching algorithm that finds correspondences by minimizing an image error function. Combining, in the same optimization framework, intrinsic geometric constraints with local descriptors, our methodology finds the correspondence between two point clouds with a large amount of outlier data. Furthermore the cost functions are targeted to the state-of-the-art projected gradient algorithms that can handle large sized data. Also, due to a special behaviour of the search procedure, an intermediate step is introduced that speed up the algorithm dramatically. The proposed method is validated and its more relevant feature are highlighted by real experiment. We expect to extend this methodology to cope with large number of missing points.

6. REFERENCES

[1] D. Lowe, "Object recognition from local scale-invariant features," *IEEE-International Conference on Computer Vision*, 1999.

[2] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, Apr 2002.

[3] A.C. Berg, T.L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 26–33 vol. 1, June 2005.

[4] V. Kolmogorov and Zabih R., "What energy functions can be minimizedvia graph cuts?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[5] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *European Conf. on Computer Vision*, 2008, pp. II: 596–609.

[6] Wei Lian and Lei Zhang, "Robust point matching revisited: A concave optimization approach," *European Conference on Computer Vision*, Oct. 2012.

[7] J. Maciel and J.P. Costeira, "A global solution to sparse correspondence problems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 187–199, Feb. 2003.

[8] Hao Jiang, Stella X. Yu, and David R. Martin, "Linear scale and rotation invariant matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1339–1355, 2011.

[9] M. Marques, M. Stošić, and J. Costeira, "Subspace matching: Unique solution to point matching with geometric constraints," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 1288–1294.

[10] M. Marques and J. Costeira, "Lamp: Linear approach for matching points," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 2113–2116.

[11] M. Stošić, M. Marques, and J. Costeira, "Convex solution of a permutation problem," *Linear Algebra and its Applications*, vol. 434, no. 1, pp. 361–369, 2011.

[12] Y. Nesterov, *Introductory Lectures on Convex Optimization:A Basic Course*, Kluwer Academic Publishers, 2003.

[13] J. Barzilai and J. Borwein, "Two point step size gradient methods," *IMA Journal of Numerical Analysis,*, vol. 8, June 1988.

[14] C. Tomasi and K. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137–154, 1992.

[15] P. Sturm and W. Triggs, "A factorization based algorithm for multi-image projective structure and motion," 1996.