CODE: Coherence Based Decision Boundaries for Feature Correspondence

Wen-Yan Lin, Fan Wang, Ming-Ming Cheng, Sai-Kit Yeung, Philip H.S. Torr, *Senior Member, IEEE*, Minh N. Do, *Fellow, IEEE*, and Jiangbo Lu, *Senior Member, IEEE*

Abstract—A key challenge in feature correspondence is the difficulty in differentiating true and false matches at a local descriptor level. This forces adoption of strict similarity thresholds that discard many true matches. However, if analyzed at a global level, false matches are usually randomly scattered while true matches tend to be coherent (clustered around a few dominant motions), thus creating a coherence based separability constraint. This paper proposes a non-linear regression technique that can discover such a coherence based separability constraint from highly noisy matches and embed it into a correspondence likelihood model. Once computed, the model can filter the entire set of nearest neighbor matches (which typically contains over 90 percent false matches) for true matches. We integrate our technique into a full feature correspondence system which reliably generates large numbers of good quality correspondences over wide baselines where previous techniques provide few or no matches.

Index Terms—Feature matching, wide-baseline matching, visual correspondence, RANSAC

1 Introduction

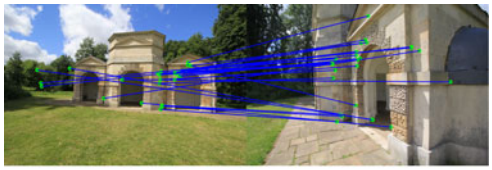
Correspondence between image pairs involve finding the projections of the same scene points in both images. By linking multiple images together, correspondence is a critical first input for many vision systems [1], [2], [3]. As applications vary wildly, correspondence algorithms must accommodate a wide range of baselines and scenarios, e.g., Internet images, noisy infrared images, high resolution images, low-resolution video frames, etc. In addition, the desired stability of downstream systems requires correspondence algorithms to find as many matches as possible while keeping false matches to a minimum. The simultaneous need for large numbers of matches, robustness and flexibility places a heavy strain on correspondence algorithms and has motivated intensive research along this direction.

To date, feature matching [4], [5], [6] is the correspondence solution of choice for many computer vision systems. While lacking the correspondence density offered by optical flow alternatives [7], [8], [9], [10], feature matchers provide an attractive blend of wide baselines, fast speed and fine localization. The goal of feature matching is to correspond sparsely scattered, distinctive key-points. Distinctiveness is enhanced by consolidating local patch information into transformation invariant descriptors and matching decisions are based on descriptor comparison. While generally effective, feature correspondence typically discards many true matches to suppress the number of false matches [11], [12], [13]. This can cause a paucity of matches which negatively impacts downstream algorithms. The problem is especially severe at the extremes of feature correspondence's working range shown in Fig. 1A). Lowering match acceptance thresholds provides many more (sometimes by a few orders of magnitude [11]) true matches. However, false matches increase more rapidly in number, leading to the mess shown in Fig. 1B).

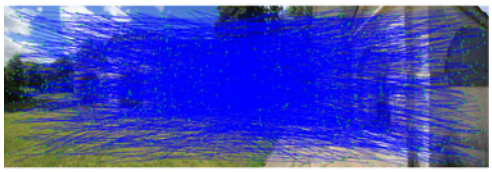
Despite appearances, matches in Fig. 1B) may actually be separable into true and false matches. This is because true matches tend to be coherent, with neighboring pixels sharing similar motions, while wrong matches are usually randomly scattered. This leads us to propose a novel approach termed *COherence based DEcision boundaries* or *CODE*, which computes a coherence based partition of the potential correspondence space into true and false regions. At first glance, this approach appears self-contradictory, since coherence estimation and thus the partition is intrinsically coupled with the unknown (or highly imperfectly estimated) correspondence. However, we observe that a coarse coherence estimate is sufficient as we can eventually rely on the feature's spatial localization for fine matching. This permits quasi-decoupling of the coherence and correspondence steps. In *CODE*, we develop a robust, non-linear regression formulation by treating feature matches as noisy data points. This is used to model a coherence cost (or the likelihood) for every possible motion. As the regression model only needs to be coarsely estimated, it can be effectively approximated from very noisy point correspondences obtained from low acceptance

- *W.-Y. Lin, F. Wang, and J. Lu are with the Advanced Digital Sciences Center, Singapore 138632. E-mail: {daniel.lin, wang.fan, jiangbo.lu}@adsc.com.sg.*
- *M.-M. Cheng is with CCCE, Nankai University, Nankai, Qu 300071, China. E-mail: cmm@nankai.edu.cn.*
- *S.-K. Yeung is with Singapore University of Technology and Design, Singapore 487372. E-mail: saikit@sutd.edu.sg.*
- *P.H.S. Torr is with Oxford University, Oxford OX1 3PA, United Kingdom. E-mail: philip.torr@eng.ox.ac.uk.*
- *M.N. Do is with the University of Illinois at Urbana-Champaign, Urbana, IL 61801. E-mail: minhdo@illinois.edu.*

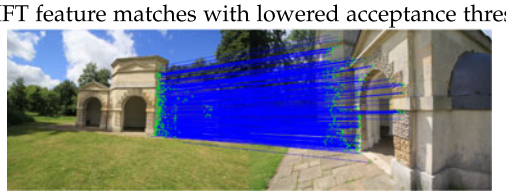
Manuscript received 16 Sept. 2015; revised 15 June 2016; accepted 20 Dec. 2016. Date of publication 15 Jan. 2017; date of current version 12 Dec. 2017. Recommended for acceptance by L. Zelnik-Manor. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2017.2652468



A) Traditional A-SIFT feature matches



B) A-SIFT feature matches with lowered acceptance thresholds



C) A-SIFT feature correspondence with *CODE* regression

Fig. 1. Feature matching for an example image pair. A) A-SIFT [6], a highly regarded wide-baseline feature matcher. B) Relaxed acceptance thresholds result in many more true matches hidden in a pile of false matches. C) Our fusion of A-SIFT with the proposed technique termed *CODE*. *CODE* regression can model the motion adequately even under such noisy matches in B). Note that only the right side of the main arch structure is visible in the second image.

thresholds. Once estimated, the model forms a coherence based decision boundary for verifying correspondence hypotheses. The result is efficient, coherence enforcement for feature correspondences.

The proposed approach has some distinctive theoretical and practical advantages. *1) Global motion modeling:* By applying our optimization on the bilateral domain which contains both spatial and motion coordinates $[x, y, u, v]$, a smooth curve can model spatially discontinuous motion (conceptually similar to edge preserving bilateral filters [14]). This dispenses with the need to handle discontinuities by weakening the smoothing function or demanding explicit discontinuity detection. Instead, a global as-smooth-as-possible function accommodates a wide variety of motions, making modeling both flexible and robust. *2) Computational tractability:* We show that an as-smooth-as-possible regression function can be formulated as a convex cost with a guaranteed global minimum. This avoids initialization problems and facilitates mathematically elegant solutions. *3) Soft modeling:* Given our input feature matches' noisy nature, modeling motion as a hard one-to-one mapping assignment between images is unlikely to be accurate. The proposed likelihood model expresses this ambiguity, and it adds a layer of robustness. As a result, mistakenly high scores for an incorrect motion need not affect the true motion's score, allowing graceful degradation on difficult scenes. *4) Efficiency:* Naive correspondence likelihood modeling is potentially very expensive. If the likelihood of every possible motion for every pixel is stored as a matrix, even a small 640×480 image, with potential horizontal and vertical motion ranges of $[-640, 640]$ and $[-480, 480]$, would require an enormous matrix of size $640 \times 480 \times (2 \times 640) \times (2 \times 480)$. By expressing likelihood with a regression model, we can estimate every matrix entry with a few hundred variables. This makes correspondence likelihood modeling computationally tractable.

In practice, *CODE* consists of a number of smooth likelihood decision boundaries for validating feature correspondences. These can be rapidly computed from noisy correspondences hypotheses (50 percent wrong matches are acceptable at this stage). Once computed, the decision boundaries can filter the entire nearest-neighbor matching set (which often has over 90 percent outliers) for inliers, with the one-to-one correspondence forming a natural complement to the likelihood based *CODE* boundaries. This process retains many more correspondences than traditional descriptor based thresholding methods while eliminating nearly all wrong matches. Our regression can be computed in a few seconds and since verification is cheap, the proposed technique scales efficiently to large numbers of features. In practice, *CODE* filtering time is $O(N)$ where N is the predefined number of feature matches used in the regression.

To make a complete feature correspondence package, we fuse *CODE* with our re-implementation of GPU A-SIFT [15] and Muja and Lowe's [16] fast approximate nearest neighbor matching. As *CODE*'s overhead is low, our full system is faster than the original A-SIFT implementation [6], while procuring many more matches. An example is shown in Fig. 1C).

This work extends our previous conference papers [11], [17]. In this paper, we provide a thorough exposition of the proposed technique, and introduce an approximation that improve its scalability to large numbers of matches. In addition, we perform a number of new and expanded experiments to validate the proposed technique against state-of-the-art methods. To summarize the main contributions:

- We propose *CODE*, a principled approach expressing the motion smoothness constraints as a matching likelihood estimate of every possible motion of every image pixel. *CODE* can be efficiently computed from highly noisy feature matches and forms an effective mechanics of discerning the difference between true and false matches.
- We integrate *CODE* with A-SIFT to create a feature matching system which yields high quality matches at wide baselines where previous techniques provide few or no matches. Despite the aggressive matching, it avoids matching images of different scenes/objects. We share our optimized implementation, in the form of executable files, to the research community at: http://www.kind-of-works.com/home/code.
- We demonstrate our high numbers of reliable correspondences bring compelling improvements when integrated into the existing structure-from-motion (SfM) systems.

1.1 Related Work

This paper owes a significant debt to pioneering research on affine invariant, wide-baseline features [6], [18], [19]. However, the original works could not fully exploit their features' invariance due to difficulty of identifying false matches. This fact forced these methods to adopt very strict similarity criteria which discarded many true match hypotheses. By modeling the true underlying motion from very noisy matches, our *CODE* technique can reliably differentiate true and false matches. This results in a flood of good matches shown in Fig. 1.

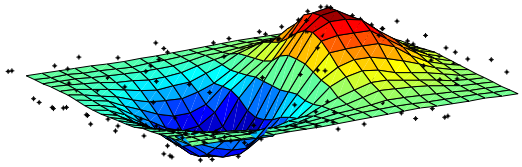


Fig. 2. Regression can be understood as finding a continuous surface that explains scattered data points (denoted by "+").

Similar to optical flow [7], [8], [9], [10], [20], we use motion smoothness to facilitate correspondence matching. However, we retain feature correspondence's design philosophy, trading matching density and fineness in favor for wide-baselines, higher speeds and robustness. As such our technique may ignore subtle motion details which might be retained by optical flow algorithms. If desired, density and fine motions may be recovered through a subsequent recomputation, but this is beyond the current paper's scope.

Our formulation builds on the motion coherence framework [21], [22]. Unlike the smoothness prior enforced in most optical flow algorithms which directly penalizes motion differences, motion coherence seeks the smoothest continuous motion field consistent with observed data. The global data integration has repeatedly demonstrated impressive levels of stability [22], [23], [24]. However, the original formulation does not accommodate motion discontinuities and is vulnerable to local minima. By adapting the motion coherence into a motion discontinuity-preserving, global regression technique, we avoid these problems while maintaining its robustness.

Edge-preserving bilateral filters [14] were a major inspiration for this work. However, unlike bilateral filtering which is a local operation, our regression on the bilateral domain computes a global model. This allows us to connect information from across the image for robustness to outliers. In addition, while bilateral filters return a single value at each pixel, *CODE* will return a continuous function encoding a distribution of likelihoods for all possible motion values a pixel can take.

We also draw inspiration from previous attempts to fuse smoothness constraints with a sparse feature correspondence. These include graph matching [13], [25], [26], [27], [28], bounded distortion (BD) [29] and mesh-based correspondence reasoning [30], with a number of approaches [29], [30] explicitly focused on removing false matches. However, the graph and distortion based techniques [29], [30] are vulnerable to local minima. Further, their computation cost increases with the number of features, making the techniques less scalable to high resolution imagery. While scalability and non-convexity are problems with most correspondence formulations, assuming pre-computed, albeit noisy correspondence creates a sub-problem which we show is amenable to convex regression modeling, thus avoiding both these issues.

In practical terms, a family of RANSAC techniques [31], [32], [33], including the recent branch-and-bound formulation [34], share our goal of removing false matches. However, the basic RANSAC formulation is designed for small (e.g., 10-20 variables) linearizable models, making them application specific. Our general smoothness based constraints cannot be enforced with RANSAC, thus motivating our investigation of regression techniques. In practice, it is best to use our method to boost the number of true matches first, then followed by RANSAC to estimate application specific parameters when appropriate.

Also worth mentioning are region growing correspondence algorithms [35], [36] which provide quasi-dense correspondence while handling large baselines and occlusions. These techniques provide many more correspondences than ours. However our solution retains the feature based techniques' innate advantage in speed, correspondence accuracy and handling large scale changes.

Finally we emphasize that different from the objectives of SIFT flow [37] or object matching [38], we do not desire correspondence between different objects. Given two images of physically different objects, our desired result is no correspondence.

2 Our Approach

This section outlines our general formulation for coherence based data regression. We begin by designing a multi-function regression in Section 2.1 based on motion coherence [21], [22]. Section 2.2 explains the bottlenecks of this approach and proposes an accelerated approximation. We provide a succinct summary of the regression process in Section 2.3. Based on the theoretical foundation laid in this section, Section 3 applies the regression techniques to the correspondence problem.

2.1 Generalized Coherence for Data Regression

The problem is formulated as fitting a smooth function to *observed data* points. The fitting function is $f : \mathbf{p} \mapsto q$, where $\mathbf{p} \in \mathbb{R}^D$ and q are its domain and co-domain. We assume $f(\mathbf{p})$ is a linear combination of K smooth functions $\{f_k(\mathbf{p})\}_{k=1,2,\ldots,K}$, such that

$$q = f(\mathbf{p}) = \sum_{k=1}^{K} a_k(\mathbf{p}) f_k(\mathbf{p}), \tag{1}$$

where $a_k(\mathbf{p})$ are known weighting functions over the D-dimensional domain of $\mathbf{p}$, which provide formulation flexibility we exploit later.[1]

The *observed data* consists of N noisy scalar values $\{\hat{q}_j\}$ at corresponding locations $\{\mathbf{p}_j\}$, which are assumed to be noisy observations of $f(\mathbf{p})$, such that

$$\hat{q}_j = f(\mathbf{p}_j) + n_j = \sum_{k=1}^{K} a_k(\mathbf{p}_j) f_k(\mathbf{p}_j) + n_j, \tag{2}$$

with n_j representing noise. With respect to Fig. 2, $\mathbf{p}$ is a spatial point on the two-dimensional X-Y domain. The regression function $f(\mathbf{p})$ is represented by a continuous valued surface, while $\hat{q}_j$ is the observed value for a spatial point $\mathbf{p}_j$.

Each individual $f_k(.)$ function is composed of two terms

$$f_k(\mathbf{p}) = H_k + \phi_k(\mathbf{p}). \tag{3}$$

H_k is an (optional) unknown scalar offset and $\phi_k(\mathbf{p})$ is a smooth function evaluated with a motion coherence smoothness penalty [21], [22]

1. The formulation is unchanged if q is a vector. However for computational speed, q is a scalar throughout this paper.

$$\Psi_k = \int_{\mathbb{R}^D} \frac{|\overline{\phi}_k(\omega)|^2}{\overline{g}(\omega)}\, d\omega, \tag{4}$$

where $\overline{\phi}_k(.)$ denotes the Fourier transform of a function $\phi_k(.)$, while $\overline{g}(\omega)$ is the Fourier transform of a Gaussian function with a spatial standard deviation γ. Hence, Eqn. (4) achieves smoothness by penalizing high frequency terms.

Our goal is to find the smoothest possible set of $f_k(.)$ functions consistent with the given data points $\{\mathbf{p}_j, \hat{q}_j\}$. This is expressed as the following energy minimization

$$\begin{aligned} E &= \sum_{j=1}^{N} C\Big(\hat{q}_j - f(\mathbf{p}_j)\Big) + \lambda \sum_{k=1}^{K} \Psi_k \\ &= \sum_{j=1}^{N} C\left(\hat{q}_j - \sum_{k=1}^{K} a_k(\mathbf{p}_j) f_k(\mathbf{p}_j)\right) + \lambda \sum_{k=1}^{K} \Psi_k. \end{aligned} \tag{5}$$

Here, $C(.)$ represents the Huber function as used in [17]:

$$C(z) = Huber(z) = \begin{cases} z^2 & \text{if } \|z\| \le \epsilon \\ 2\epsilon\|z\|_1 - \epsilon^2 & \text{if } \|z\| > \epsilon, \end{cases} \tag{6}$$

and λ is the weight given to the smoothness constraint Ψ_k.

Directly minimizing E with respect to functions $f_k(.)$ appears intractable as $\{f_k(.)\}$ are continuous functions. However, we can reduce the problem to an optimization over a finite number of variables as shown below.

Note the Fourier transform relation, $\phi_k(\mathbf{p}) = \int_{\mathbb{R}^D} \overline{\phi}_k(\omega) e^{2\pi\imath\langle \mathbf{p},\omega\rangle} d\omega$. We know that at the minimum of the energy E of Eqn. (5), its derivative is zero. Hence,

$$\begin{aligned} &\frac{\delta E}{\delta \overline{\phi}_k(\mathbf{z})} = 0, \forall \mathbf{z} \in \mathbb{R}^D, k \in \{1, 2, \ldots, K\} \\ &\Rightarrow \sum_{j=1}^{N} \mathbf{w}_k(j) \int_{\mathbb{R}^D} \frac{\delta \overline{\phi}_k(\omega)}{\delta \overline{\phi}_k(\mathbf{z})} e^{2\pi\imath <\mathbf{p}_j,\omega>} d\omega \\ &\quad + \lambda \int_{\mathbb{R}^D} \frac{\delta}{\delta \overline{\phi}_k(\mathbf{z})} \frac{|\overline{\phi}_k(\omega)|^2}{\overline{g}(\omega)} d\omega = 0 \\ &\Rightarrow \sum_{j=1}^{N} \mathbf{w}_k(j) e^{2\pi\imath <\mathbf{p}_j,\mathbf{z}>} + 2\lambda \frac{\overline{\phi}_k(-\mathbf{z})}{\overline{g}(\mathbf{z})} = 0, \end{aligned} \tag{7}$$

where $\mathbf{w}_k$ is a $N \times 1$ vector that serves as a place-holder for more complicated terms.

Rearranging the terms gives

$$\overline{\phi}_k(\mathbf{z}) = \overline{g}(-\mathbf{z}) \sum_{j=1}^{N} \mathbf{w}_k(j) e^{-2\pi\imath <\mathbf{p}_j,\mathbf{z}>}, \tag{8}$$

Taking the inverse Fourier transform of Eqn. (8), we can write our continuous functions $\{\phi_k(\mathbf{p})\}$ in terms of a finite set of N-dimensional vectors $\{\mathbf{w}_k\}$,

$$\phi_k(\mathbf{p}) = \sum_{j=1}^{N} \mathbf{w}_k(j) g(\mathbf{p}, \mathbf{p}_j) = \sum_{j=1}^{N} \mathbf{w}_k(j) e^{-\frac{\|\mathbf{p}-\mathbf{p}_j\|}{\gamma^2}}, \quad \forall k \in \{1, 2, \ldots, K\}, \tag{9}$$

where $g(\mathbf{p}, \mathbf{p}_j)$ is a Gaussian radial basis function and $\{\mathbf{w}_k(j)\}$ are unknown variables. Thus Eqn. (9) states that the smoothest possible functions $\{\phi_k(.)\}$ which minimize the energy in Eqn. (5) must lie in the space spanned by N radial basis functions $\{g(\mathbf{p}, \mathbf{p}_j)\}$.[2]

For the smoothness constraint, substituting Eqn. (8) into Eqn. (4) allows the continuous regularization function Ψ_k to be expressed in terms of $\mathbf{w}_k$

$$\Psi_k = \mathbf{w}_k^T G \mathbf{w}_k, \qquad k \in \{1, 2, \ldots, K\}, \tag{10}$$

where G is a symmetric matrix with its elements given as

$$G(i, j) = g(\mathbf{p}_i, \mathbf{p}_j) = e^{-\|\mathbf{p}_i - \mathbf{p}_j\|^2/\gamma^2}. \tag{11}$$

Substituting Eqns. (9) and (10) into Eqn. (5) yields

$$\begin{aligned} &\arg\min_{\{f_k(\mathbf{p})\}} \sum_{j=1}^{N} C\Big(\hat{q}_j - f(\mathbf{p}_j)\Big) + \lambda \sum_{k=1}^{K} \Psi_k \\ &= \arg\min_{\{f_k(\mathbf{p})\}} \sum_{j=1}^{N} C\left(\hat{q}_j - \sum_{k=1}^{K} a_k(\mathbf{p}_j) f_k(\mathbf{p}_j)\right) + \lambda \sum_{k=1}^{K} \Psi_k \\ &= \arg\min_{\{\mathbf{w}_k, H_k\}} \sum_{j=1}^{N} C\left(\hat{q}_j - \sum_{k=1}^{K} a_k(\mathbf{p}_j)\left(H_k + \sum_{i=1}^{N} \mathbf{w}_k(i) g(\mathbf{p}_j, \mathbf{p}_i)\right)\right) \\ &\quad + \lambda \sum_{k=1}^{K} \mathbf{w}_k^T G \mathbf{w}_k, \end{aligned} \tag{12}$$

where the energy is dependent only on a finite number of variables, i.e., $\{\mathbf{w}_k\}$ and H_k. Since G is a Gram matrix [22], this makes the coherence term Ψ_k in Eqn. (10) convex. As the Huber loss function $C(.)$ is also convex and the sum of convex functions is convex, the overall energy minimization problem in Eqn. (12) is convex. Therefore, a gradient descent minimization of Eqn. (12) will lead to a guaranteed global minimum.

Based on the estimated variables $\{\mathbf{w}_k\}$ and H_k, the global regression function $f(\mathbf{p})$ is constructed from $\{f_k(\mathbf{p})\}$

$$\begin{aligned} f(\mathbf{p}) &= \sum_{k=1}^{K} a_k(\mathbf{p}) f_k(\mathbf{p}) = \sum_{k=1}^{K} a_k(\mathbf{p})(H_k + \phi_k(\mathbf{p})) \\ &= \sum_{k=1}^{K} a_k(\mathbf{p}) \left(H_k + \sum_{i=1}^{N} \mathbf{w}_k(i) g(\mathbf{p}, \mathbf{p}_i)\right). \end{aligned} \tag{13}$$

This formulation mirrors that developed in our previous conference paper [17]. For applications of this global regression approach to data fitting and image warping, we encourage interested readers to refer to [17].

2.2 Accelerated Coherence Based Global Regression

Examining the formulation presented in Section 2.1, one can find that the length of $\mathbf{w}_k$ vectors and hence the number of variables in Eqns. (8) and (12) increase linearly with N, the number of observed data points under consideration. This creates a computational burden when N is large. Motivated to address this challenge, we propose an approximation that decouples this linear computational dependency, with its full derivation given in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2017.2652468.

2. This may partially explain the success of radial basis functions in learning networks [39].

TABLE 1
Summary of Parameters and Variables in Our Multi-Function Regression in Eqns. (16), (17), and (18)

Estimated	$\{H_k\}$	optional bias variable in $\widetilde{f}_k(\mathbf{p})$
	$\{\widetilde{\mathbf{w}}_k\}$	function variables in $\widetilde{f}_k(\mathbf{p})$
User-defined	λ	weight of smoothness terms
	γ	std. deviation of smoothing Gaussian
	ϵ	threshold in the Huber function $C(.)$
	K	number of smooth functions for $\widetilde{f}(\mathbf{p})$
	$\{a_k(\mathbf{p})\}$	user defined weighting function
	M	length of $\widetilde{\mathbf{w}}_k$
	$\{\widetilde{\mathbf{p}}_j\}$	representative subsampled points
	N	number of data points
Others	$\widetilde{f}(\mathbf{p})$	regression function in Eqn. (16)
	cost	given in Eqn. (18)

Consider $f_k(\mathbf{p})$ in Eqn. (13), the kth smooth function to be estimated. If many data points in the input set $\{\mathbf{p}_i\}$ are adjacent, a good approximation of $f_k(\mathbf{p})$ can be given as follows,

$$f_k(\mathbf{p}) \approx \widetilde{f}_k(\mathbf{p}) = H_k + \sum_{j=1}^{M} \widetilde{\mathbf{w}}_k(j) g(\mathbf{p}, \tilde{\mathbf{p}}_j), \tag{14}$$

where $\widetilde{\mathbf{w}}_k$ is an M-dimensional vector ($M \ll N$), and $\{\tilde{\mathbf{p}}_j\}$ are M representative cluster centroids distributed across the space occupied by the original points $\{\mathbf{p}_j\}$. These centroids are usually obtained by K-means clustering [40] on $\{\mathbf{p}_j\}$. If most of the $\mathbf{p}_j$ points have near duplicates in the set $\{\tilde{\mathbf{p}}_j\}$, the function $\widetilde{f}_k(\mathbf{p})$ will be able to closely approximate all possible values of the original $f_k(\mathbf{p})$ function in Eqn. (13).

Interestingly, the approximated regression functions $\{\widetilde{f}_k(\mathbf{p})\}$ can be easily integrated into the global energy function (12), which becomes

$$\arg \min_{\{\widetilde{\mathbf{w}}_k, H_k\}} \sum_{j=1}^{N} C\left(\hat{q}_j - \sum_{k=1}^{K} a_k(\mathbf{p}_j)\widetilde{f}_k(\mathbf{p}_j)\right) + \lambda \sum_{k=1}^{K} \widetilde{\mathbf{w}}_k^T \widetilde{G} \widetilde{\mathbf{w}}_k, \tag{15}$$

where $\widetilde{G}$ is an $M \times M$ matrix, with $\widetilde{G}(i,j) = g(\tilde{\mathbf{p}}_i, \tilde{\mathbf{p}}_j)$ and the overall cost remains convex. Note that in this approximation, the values of λ and M are not coupled, i.e., the same λ can be properly used for different settings of M.

This clustering-based approximation is actually useful in accelerating many computer vision tasks, where closely located pixels create many redundant variables. Apart from this paper, we believe our approximation can also aid the original motion coherence [22] and its derivations [23], [24].

2.3 Summary of the Coherence Based Regression

To provide a succinct summary of the regression process, we give a recap of it here. Given N noisy scalar values $\{\hat{q}_j\}$ at corresponding locations $\{\mathbf{p}_j\}$, where $\mathbf{p}_j$ is a D-dimensional vector, we can explain the *observed data* $\{\mathbf{p}_j, \hat{q}_j\}$ with a continuous function $\widetilde{f}(\mathbf{p})$. The function $\widetilde{f}(\mathbf{p})$ is composed of a linear combination of K smooth functions, which returns a value for each query point $\{\mathbf{p}\}$

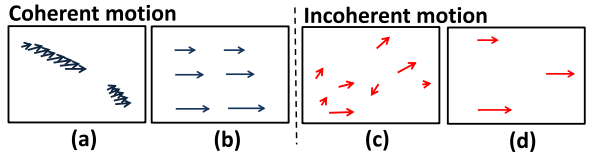


Fig. 3. Coherence based separation of true and false matches. Motions are considered coherent if (a) many local points make similar motions or (b) there is broad spatial support for the motion. This is enforced via the likelihood function in Eqn. (21). In contrast, feature matches in (c) and (d) do not give coherent motions, as the matches are not consistent in (c), while there are insufficient smoothly moving points to justify a long-range motion coherence model in (d).

$$q = \widetilde{f}(\mathbf{p}) = \sum_{k=1}^{K} a_k(\mathbf{p}) \widetilde{f}_k(\mathbf{p}), \tag{16}$$

where $\{a_k(\mathbf{p})\}$ are known weighting functions over the D-dimensional domain $\mathbf{p}$. The function $\widetilde{f}_k(\mathbf{p})$ is in turn parametrized by $\{\widetilde{\mathbf{w}}_k, H_k\}$ variables

$$\begin{aligned} \widetilde{f}_k(\mathbf{p}) &= H_k + \sum_{j=1}^{M} \widetilde{\mathbf{w}}_k(j) g(\mathbf{p}, \tilde{\mathbf{p}}_j) \\ &= H_k + \sum_{j=1}^{M} \widetilde{\mathbf{w}}_k(j) e^{-\|\mathbf{p}-\tilde{\mathbf{p}}_j\|^2/\gamma^2}, \end{aligned} \tag{17}$$

which can be estimated by a gradient descent minimization of the convex cost

$$\begin{aligned} &\arg \min_{\{\widetilde{\mathbf{w}}_k, H_k\}} \sum_{j=1}^{N} C\left(\hat{q}_j - \widetilde{f}(\mathbf{p}_j)\right) + \lambda \sum_{k=1}^{K} \widetilde{\mathbf{w}}_k^T \widetilde{G} \widetilde{\mathbf{w}}_k \\ &= \arg \min_{\{\widetilde{\mathbf{w}}_k, H_k\}} \sum_{j=1}^{N} C\left(\hat{q}_j - \sum_{k=1}^{K} a_k(\mathbf{p}_j)\widetilde{f}_k(\mathbf{p}_j)\right) + \lambda \sum_{k=1}^{K} \widetilde{\mathbf{w}}_k^T \widetilde{G} \widetilde{\mathbf{w}}_k. \end{aligned} \tag{18}$$

A Huber loss function $C(.)$ is adopted to evaluate the data fitting quality. A summary of the estimated variables and user-defined parameters are given in Table 1.

3 Coherence Based Decision Boundaries

With the general formulation for coherence based data regression presented in Section 2, we now apply the regression techniques to the correspondence problem. The key idea is to estimate a coarse coherence based model from a sparse set of very noisy feature matches. This model quantifies the coherent matching evidence for every possible motion, providing a coherence based decision boundaries for verifying correspondence hypotheses. More specifically, we achieve this through two regression functions which act as a chain of cascaded filters for eliminating wrong feature matches. We describe individual modules below and elaborate on design decisions and properties in Section 3.1.

Matching Likelihood Boundaries. What constitutes coherent motion? We consider a local motion is coherent if it satisfies either of these criteria: a) a concentrated cluster of local features making the motion; and b) many features over a large spatial extent making the motion. A visual illustration is shown in Fig. 3.

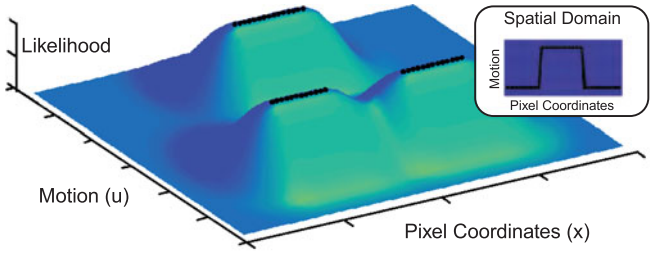


Fig. 4. Inset: A set of motion hypotheses with discontinuities over the one-dimensional X axis. Main figure: The same data (black dots) on the bilateral domain u and X gives rise to a motion likelihood model built from the proposed regression technique. Observe that estimating a regression function over the bilateral domain involves estimating a value for every possible motion at every spatial location. Thus, a cross section along the specified X-position yields a likelihood function over the admissible motion range that a query X-position can take from. Note also that a smooth likelihood function computed for a motion-augmented bilateral domain can explain motion data with discontinuities.

To enforce this coherence, we formulate a regression based likelihood function in which each given feature match is a noisy *observed data* of the form

$$\Big\{\mathbf{p}_j = [\mathbf{x}_j, \mathbf{m}_j, \mathbf{x}_j + \mathbf{m}_j, \mathbf{o}_j], \hat{q}_j = 1\Big\}, \tag{19}$$

where each match indexed by j hypothesizes a "1" value at location $\mathbf{p}_j$. Here $\mathbf{x}_j = \begin{bmatrix} x_j, y_j \end{bmatrix}$ and $\mathbf{m}_j = \begin{bmatrix} u_j, v_j \end{bmatrix}$ are two-dimensional vectors representing image coordinates and motion vectors respectively, while $\mathbf{o}_j$ is a 4×1 vector representing the relative affine orientation of the matched feature descriptor. From Eqn. (17), we choose a single function $K = 1$ without the H_k bias variable, and set $a_k(\mathbf{p}) = 1, \forall \mathbf{p}$. This leads to a regression function

$$\widetilde{f}(\mathbf{p}) = \sum_{j=1}^{M} \widetilde{\mathbf{w}}(j) g(\mathbf{p}, \tilde{\mathbf{p}}_j). \tag{20}$$

Substituting the regression function (20) into the cost (18), we attain

$$\arg\min_{\tilde{\mathbf{w}}} \sum_{j=1}^{N} C(1 - \tilde{f}(\mathbf{p}_j)) + \lambda \tilde{\mathbf{w}}^T \tilde{G} \tilde{\mathbf{w}}, \tag{21}$$

whose minimization estimates the parameters of $\widetilde{f}(\mathbf{p})$.

Observe that the smoothness cost $\tilde{\mathbf{w}}^T \tilde{G} \tilde{\mathbf{w}}$ in Eqns. (4) and (10) not only penalizes un-smooth motions but penalizes all motions by encouraging $\tilde{\mathbf{w}}$ to be zero. Thus, the lack of a bias term in $\widetilde{f}(\mathbf{p})$ given in Eqn. (20) means the likelihood function stays at zero, unless forced upward by the data. The likelihood function rises towards one if there exist a cluster of adjacent matches whose collective pull justifies an "un-smooth" peak. Alternatively, if a motion has significant supports across a wide area but with no strong local pull, the likelihood function can also rise to one, since a smooth surface over a large extent can be fitted for these matches by incurring a low smoothness penalty. In contrast, randomly scattered, incorrect matches do not exert sufficient collective pull and are ignored by the regression function. A simplified matching likelihood function is illustrated in Fig. 4.

From noisy feature correspondences, we use Eqn. (21) to fit a robust likelihood function given by $\widetilde{f}(\mathbf{p})$. The acceptance condition for a feature matching hypothesis $\mathbf{p}_i$ is

$$accept(\mathbf{p}_i) = \text{true} \quad \text{if } \widetilde{f}(\mathbf{p}_i) > \epsilon_{likelihood}. \tag{22}$$

This mechanics can take very noisy data and remove most gross matching errors. However, it lacks fine spatial awareness and some erroneous matches will remain.

Bilaterally Varying Affine Motion Boundaries. The matching likelihood function can be considered as a blind cluster discovery mechanics. However, we know that correct motions not only cluster but tend to approximate a piecewise smoothly varying affine model [23]. We enforce this knowledge through a bilaterally varying affine regression function, which computes a motion likelihood by checking a hypothetical motion's consistency against its estimated model.

We focus on the X motion direction first. Observed data take the form

$$observed\ data = \{\mathbf{p}_j, \hat{q}_j = x_j + u_j\}, \tag{23}$$

where x_j, u_j are obtained from the given feature correspondence. The definition of $\mathbf{p}_j$ remains unchanged from the likelihood function formulation in Eqn. (19).

By setting $K = 3$ in Eqn. (16) and setting $a_1(\mathbf{p}) = x, a_2(\mathbf{p}) = y, a_3(\mathbf{p}) = 1$, we have

$$\tilde{f}_x(\mathbf{p}) = \tilde{f}_1(\mathbf{p})x + \tilde{f}_2(\mathbf{p})y + \tilde{f}_3(\mathbf{p}), \tag{24}$$

where each $\widetilde{f}_k(\mathbf{p}) = H_k + \sum_{j=1}^{M} \widetilde{\mathbf{w}}_k(j) g(\mathbf{p}, \tilde{\mathbf{p}}_j)$ represents an affine motion parameter for the location $\mathbf{p}$. To estimate $\widetilde{f}_k(\mathbf{p})$, from Eqn. (17), we optimize the following cost function

$$\arg\min_{\{\tilde{\mathbf{w}}_k, \tilde{H}_k\}} \sum_{i=1}^{N} C(x_i + u_i - \tilde{f}_x(\mathbf{p}_i)) + \lambda \sum_{k=1}^{3} \tilde{\mathbf{w}}_k \tilde{G} \tilde{\mathbf{w}}_k. \tag{25}$$

Similarly for the Y direction, we have a bilaterally varying affine model as follows,

$$\tilde{f}_y(\mathbf{p}) = \tilde{f}_4(\mathbf{p})x + \tilde{f}_5(\mathbf{p})y + \tilde{f}_6(\mathbf{p}), \tag{26}$$

which can be computed from the cost

$$\arg\min_{\{\tilde{\mathbf{w}}_k, \tilde{H}_k\}} \sum_{i=1}^{N} C(y_i + v_i - \tilde{f}_y(\mathbf{p}_i)) + \lambda \sum_{k=4}^{6} \tilde{\mathbf{w}}_k \tilde{G} \tilde{\mathbf{w}}_k. \tag{27}$$

While not a direct likelihood function, the bilaterally varying affine model can be applied to distinguish correct and wrong feature matches through a thresholding step

$$\begin{aligned} &accept(\mathbf{p}_i) = \text{true} \\ &\text{if } \sqrt{(\tilde{f}_x(\mathbf{p}_i) - x_i - u_i)^2 + (\tilde{f}_y(\mathbf{p}_i) - y_i - v_i)^2} < \epsilon_{spatial}. \end{aligned} \tag{28}$$

In practice, we find this stage gives more refined estimates but lacks the robustness of the preceding likelihood module. Thus, we cascade the affine motion boundary estimation after the likelihood boundary, which has removed most of the grossly incorrect matches.

3.1 Important Design Considerations and Properties

Having outlined our basic formulation, we will now delve into the properties and design considerations of our coherence based regression functions.

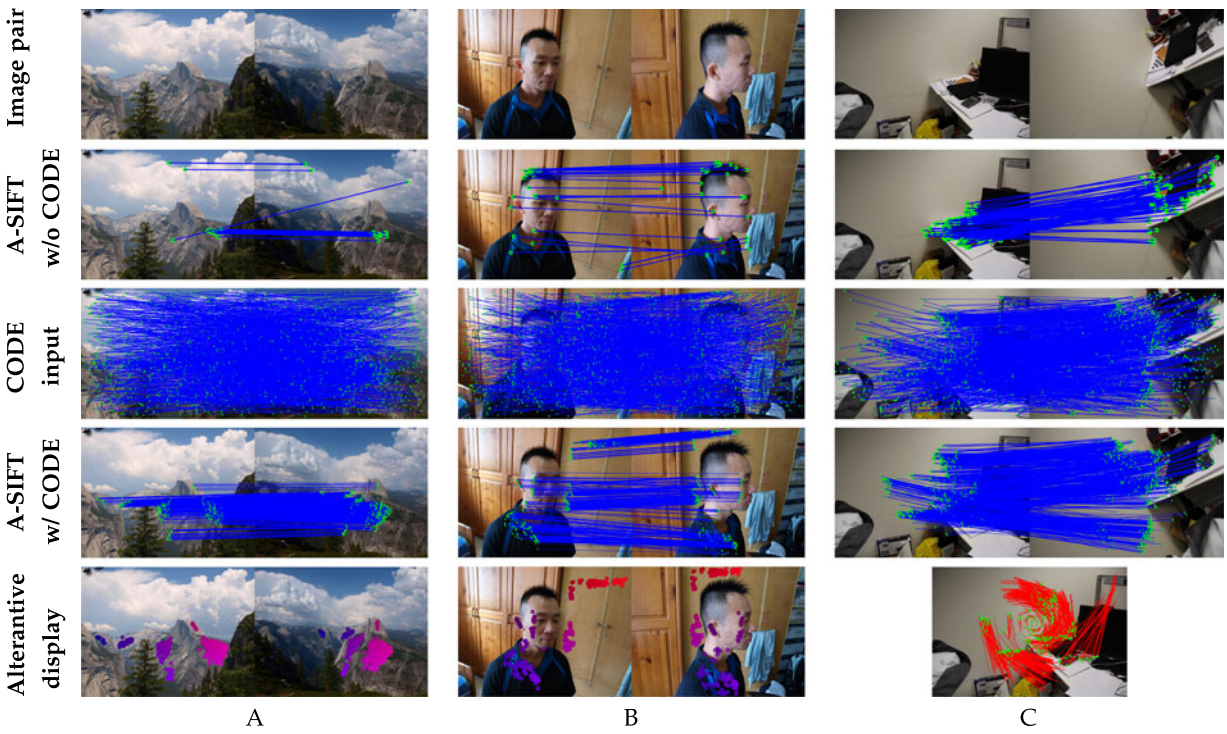


Fig. 5. *CODE* applied to A-SIFT features [6]. A and B demonstrate *CODE*'s ability to handle large motion discontinuities and occlusions, while C demonstrates its ability to handle large rotations. In alternative display, we color code feature matches according to their spatial locations on the left image. The matching transfers these color codes to the right image. To better illustrate the large rotation, C uses a vector display.

Global Smoothness. It is enforced in all the functions. Unlike a local smoothness penalization which directly penalizes (motion) differences between neighbors, global smoothness enforced in our formulation seeks the globally smoothest function. By integrating information from across the image, global functions provide a high level of robustness, allowing the regression functions to be computed from very noisy input as shown in Fig. 5.

Domain Choice. The choice of the domain for $\mathbf{p}$ in the as-smooth-as-possible regression formulation can be arbitrary. We are motivated to choose the domain shown in Eqn. (19) out of three concerns:

Discontinuity Preservation: Typically, global smoothness interferes with discontinuity preservation. Consider a function $f_k(\mathbf{p})$ where $\mathbf{p}$ represents pixel coordinates, while the range of $f_k(\mathbf{p})$ represents motion. The smoothness (or infinite differentiability) implies a continuity constraint

$$\lim_{\Delta\mathbf{p}\to 0} f_k(\mathbf{p}+\Delta\mathbf{p}) - f_k(\mathbf{p}) = 0, \quad \forall \mathbf{p} = [x,y]^T \in \mathbb{R}^2, \tag{29}$$

which forces the function value in the neighborhood of $\mathbf{p}$ to be similar. This means smooth functions must incur large errors at discontinuous motion boundaries.

We tackle this problem by using the bilateral domain spanning both the spatial and motion dimensions, i.e., $\mathbf{p} = [x,y,u,v]^T$, for conceptual understanding, we use a simpler form than that in (19). This redefines neighbors such that spatially neighboring points with different velocities are no longer adjacent. Therefore, we can assign very different function values to points with adjacent spatial coordinates, while retaining the constraint that $f_k(\mathbf{p})$ must be smooth. If the motion difference $(\Delta u, \Delta v)$ between two points in the bilateral domain $\mathbf{p}$ tends to infinity, the point separation also tends to infinity, significantly reducing their influence on each other. This separation holds irrespective of the spatial coordinates, and allows motion discontinuity to be accommodated by a smooth function shown in Fig. 4. Examples of this discontinuity handling are shown in Figs. 5A and 5B).

Affine Smoothness: It has been noted in [23] that because of the local generalization property of affines, motions are smoother when over-parameterized in the affine domain. Thus, we concatenate the affine parameters $\mathbf{o}$ to $\mathbf{p}$ in (19). This allows us to better handle ciruclar motions such as in Fig. 5C).

Symmetry: Finally, for the sake of mathematical symmetry between left and right images, we concatenate the arguably redundant $\mathbf{x}_j + \mathbf{m}_j$ term to $\mathbf{p}$ in (19).

Note that we find many different domain choices give good results and the domain's chosen in this paper serve as guidelines rather than canonical selections.

Multiple Motion Hypotheses. Computing regressions on the bilateral domain means estimating a function value for every $[x,y,u,v]^T$. This forces the function to consider every possible value that each pixel can take. Thus, unlike bilateral filters which estimate a single value for each pixel, *CODE* regression functions encode the likelihood for every possible motion each pixel can take. This soft-modeling adds a significant layer of robustness, as mistakenly giving a high score to an incorrect motion need not affect the score of the true motion. The computed regression functions and the resulting decision boundaries of *CODE* are subsequently used to verify feature correspondence hypotheses, with the hard, sub-pixel accurate feature matches forming a natural complement to the soft likelihood models.

Convex Cost. The cost functions for estimating likelihood and bilaterally varying affine boundaries are convex. This

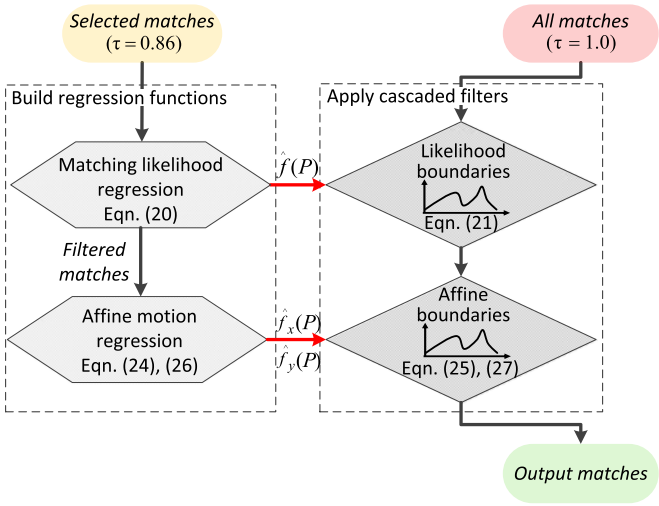


Fig. 6. Workflow of the proposed *CODE* technique. It consists of two major steps: building regression functions and applying cascaded filters, which take noisy feature matches obtained with different threshold values τ as the input. See the text for details.

avoids the need for initialization and allows our algorithm to handle challenging motions such as the large rotation shown in Fig. 5C).

3.2 Implementation

Our algorithm begins with feature matching. View-invariant features are computed through a GPU A-SIFT [15] with nearest-neighbor matching computed by FLANN [16]. Typically, feature matchers choose to accept matches based on a ratio test with a threshold setting $\tau = 0.6$. i.e., a match is accepted if its nearest-neighbor difference is at least 0.6 times smaller than its second nearest neighbor. However, this can lead to few matches as shown in Fig. 1A). In this paper, we take *selected matches* obtained with a weaker threshold setting $\tau = 0.86$ as the input. While matching results appear like a mess in Fig. 1B), the weaker threshold provides many true matches, from which information can be mined.

The *selected matches* are used to compute likelihood boundaries through the regression step in Eqn. (21). The current set is filtered by the likelihood boundaries and acceptable matches are used to compute bilateral affine motion boundaries in Eqns. (25) and (27). Once the regression functions and the resulting decision boundaries of *CODE* are computed, *all matches* obtained by setting $\tau = 1.0$ are passed through the cascaded filters. The finally accepted feature matches are *output matches*. The minimizer used is the Ceres solver [41] and the process is illustrated in Fig. 6.

In theory, better results can be obtained through iterations, where the *output matches* form a new set of *selected matches*. In practice, we find the gain is small and hence restrict the study to a single iteration in this paper.

Parameter Settings. To accommodate images of different sizes, the Hartley normalization is applied to the *selected matches* such that they have zero mean and average distance from the center of $\sqrt{2}$. The user-defined parameters in Table 1 for the likelihood boundaries are $\lambda = 1, \gamma = 1, \epsilon = 0.1, K = 1, M = 100, N = 30{,}000$, with the set $\{\widetilde{\mathbf{p}}_j\}$ being centroids obtained from K-means clustering of the selected data. Matches are accepted with an $\epsilon_{likelihood} = 0.6$ in Eqn. (22). The user-defined parameters for the bilateral affine boundaries are $\lambda = 1, \gamma = 1, \epsilon = 0.1, K = 3, M = 100, N = 1{,}000$. Matches are accepted with an $\epsilon_{spatial} = 0.01$ in Eqn. (28). The design details of the regression function $f(\mathbf{p})$ and weighting functions $\{a_k(\mathbf{p})\}$ are provided in Eqns. (20), (24) and (26).

4 Experimental Results

To evaluate the effectiveness of the proposed *CODE* technique, we performed a series of experiments on different feature matching tasks and applications. Experimental results are reported in four sections. Section 4.1 shows how the proposed regression technique *CODE* significantly improves the A-SIFT feature matcher [6]'s performance. Section 4.2 applies an A-SIFT with *CODE* system (*A-SIFT w CODE*) to the Structure from Motion problem. Section 4.3 compares our *A-SIFT w CODE* to other correspondence algorithms. Finally Section 4.4 discusses relative computational time of various correspondence approaches.

4.1 A-SIFT and *CODE*

CODE's central purpose is true-false feature match differentiation to retain ambiguous matches, which are often discarded in existing methods due to a conservative thresholding step. To evaluate effectiveness on real scenes, we computed A-SIFT, with and without *CODE* on twenty image pairs shown in Appendix D in the supplementary material. These images are grouped into sets A and B. Set A has moderate viewpoint changes but some image pairs with large illumination changes, while set B has very large viewpoint changes. All images are resized to a height of 600 pixels (to accommodate the original A-SIFT's limitation in image resolution).

We show feature matching results in Fig. 7. *Av. Precision* shows *A-SIFT w CODE* is more accurate than the original A-SIFT. A match is considered incorrect if its distance exceeds seven pixels (on an image normalized to 640×480)[3] from the fundamental matrix, and we used the implementation of RANSAC algorithms from Peter Kovesi [42]. As reflected from a comparison on *Av. Match Num*, *CODE* achieves higher precision while at the same time providing a few orders of magnitude more matches than the A-SIFT matcher. This illustrates the sheer number of potential correspondences that are discarded due to the inability to differentiate between true and false matches. The improvements are more marked at wide-baselines as can be seen by comparing sets A and B.

Apart from quantitative improvements, Fig. 7 also shows the proposed *CODE* approach produces qualitatively better results, yielding quasi-dense feature correspondences compared to the scattered matches of A-SIFT. *Area %* is an alternative measure of feature match numbers. Raw matching numbers are not so informative in themselves since an algorithm can interpolate and extrapolate to obtain more correspondences. To simulate this possibility, each match is dilated with a radius of 12 pixels (on its 640×480 rescaled image) to define a matching area. The total matching area covered by the A-SIFT is then measured against that of *A-SIFT w CODE*,

3. We consistently set a threshold value of 7 as a compromise between the thresholds of 12 and 5 pixels used in [36] and [29], respectively.

	Algo	Av. Precision	Av. Match Num	Area %	Av. Time/ s	Max time/ s	Total Failures	**F-number**
Set A	A-SIFT w/o *CODE*	0.9471	396.84	37.8%	-	-	4/13	0.799
	A-SIFT w CODE	0.9673	7.3447e+03	100%	2.58	5.29	2/13	0.9027
Set B	A-SIFT w/o *CODE*	0.8073	43.714	20.9%	-	-	3/7	0.6692
	A-SIFT w CODE	0.9143	1.4807e+03	100%	1.96	2.76	0/7	0.9552

Fig. 7. Comparison between the original A-SIFT matcher [6] and *A-SIFT w CODE*, applying the proposed *CODE* regression to A-SIFT. For this test, we constructed a challenging dataset of static scene images. It includes representative images from a variety of prior papers [11], [13], [24], [29]. Set A has limited out-of-plane rotation, while Set B features very large viewpoint changes. *CODE* recovers many true feature matches that A-SIFT discards while maintaining high precision. This gain is especially noticeable for wide-baseline image pairs contained in Set B.

and the resulting area ratio is reported as *Area %*. Even on this measure, *CODE* improves A-SIFT quite significantly and allows to cover an appreciably larger area. Note that this measure under-weights more difficult scenes which have small areas of overlap. Therefore, we also report *Total Failures* which gives the fraction of the tested scene pairs with no matches returned from an algorithm. In the absence of ground-truth, an approximate *Recall* is given as (1-*Total Failures*). Finally, *F-Number* is calculated as a summary statistic

$$F\text{-}number = \frac{2 \times Recall \times Precision}{Recall + Precision}. \quad (30)$$

Av. Time and *Max. Time* of 1.96 and 2.76 seconds reflect the computational overhead of integrating *CODE*. The computational overhead is marginal relative to the runtime of the original A-SIFT matcher (about 20 seconds) and our re-implementation (about 10 seconds). Runtimes are measured on a laptop with Intel i7CPU (2.4 GHz), 8 GB RAM and NVIDIA GPU GeForce GTX 660 M.

4.2 Structure from Motion

In the last section, we performed a direct comparison between the A-SIFT matcher and our integration of *CODE* in feature matching. The experiments justified *CODE* significantly improves A-SIFT. In this section, we choose to evaluate the two different feature matchers by integrating them into a complete end-to-end Structure from Motion system, where feature matching is a fundamental early processing module.

Structure from Motion seeks to infer 3D models from 2D images. It typically comprises two main stages: a) camera position computation where camera poses are inferred from feature matches such as A-SIFT, and b) a multi-view stereo reconstruction step, where images with known relative positions are fused into a full 3D model. An interesting fact is a test dataset, designed more for one of the stages, can be difficult for the other. For example, images from the multi-view stereo database [43] permit high quality dense multi-view reconstruction, if camera positions are assumed known; but the same set of images are less amenable to camera position recovery, due to difficulty in computing reliable feature correspondences. Hence, most SfM systems fail to reconstruct significant 3D scene sections. We attempted to reconstruct these scenes by integrating *A-SIFT w CODE* in the well-regarded Visual SfM system [3], [44], [45], [46], [47]. While reconstruction results are not always perfect, we can recover a complete 3D structure for all multi-view scenes. An example is shown in Fig. 8.

This experiment serves two purposes. First, it demonstrates that *CODE*'s high quality feature matching results can translate into meaningful improvements on other computer vision tasks. Second, feature correspondence is indeed critical to the final 3D model's quality. Our experiment provides an indirect visualization of feature correspondence quality on challenging multi-view stereo datasets like [43]. Here each of the six image sequences has twenty-four images, leading to a total of 1,656 pairwise matches. Since the 3D reconstruction task here is computationally much heavier than all the other tests, we ran this comparative experiment on a desktop PC with an NVIDIA GeForce GTX980-Ti GPU. The complete 3D reconstruction results for A-SIFT with and without *CODE* are shown in Appendix C in the supplementary material.

4.3 Comparison to Other Correspondence Algorithms

Thus far evaluations have focused on *CODE*'s ability to facilitate A-SIFT feature correspondence. We now turn our

A set of multi-view images [43]

Agisoft [48]: A commercial 3D reconstruction software

Visual SfM [3], [44], [45], [46], [47]

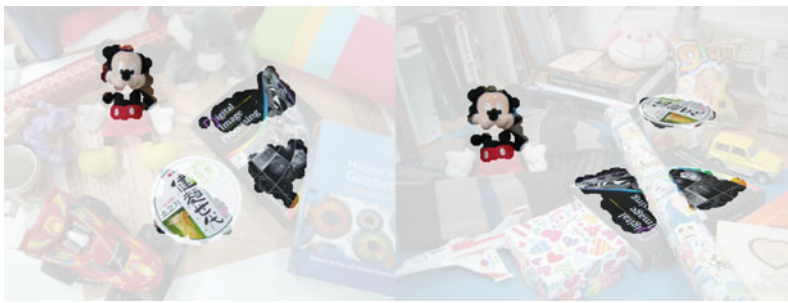
Visual SfM using feature matches returned by *A-SIFT w CODE*

Fig. 8. Structure-from-motion seeks to fuse 2D images into 3D models. *CODE* recovers large numbers of reliable feature matches, making this task easier and more robust.

attention to comparing our full *A-SIFT w CODE* to other matching techniques. In this section, we drop the cumbersome name *A-SIFT w CODE* and call our technique *CODE*. The most related algorithms are those which seek to remove or refine incorrect feature matches. Two recent examples are bounded distortion by Yaron et al. [29], and *Mode Seeking* by Chao et al. [13], [26]. At the time of writing, they represent a fair representation of the state-of-the-art. For experimental completeness, we also evaluate quasi-dense correspondence *NRDC* by HaCohen et al. [36] and agglomerative clustering *ACC* by Cho et al. [38]. As *NRDC* seeks dense correspondence, computation time on large images is prohibitively long. Because of this, unlike other algorithms, *NRDC* is run on down-sampled images of 640×480 resolution. As stated in [36], *NRDC* requires different settings for evaluation on the co-recognition dataset [49]. However, this impacts its general performance. Thus, we evaluate *NRDC* using its default parameters, and also *NRDC(t)* tuned for the co-recognition dataset. Next, we conduct evaluations for these competing algorithms on three datasets, and we present the evaluation protocol and experimental results for each of these datasets one by one.

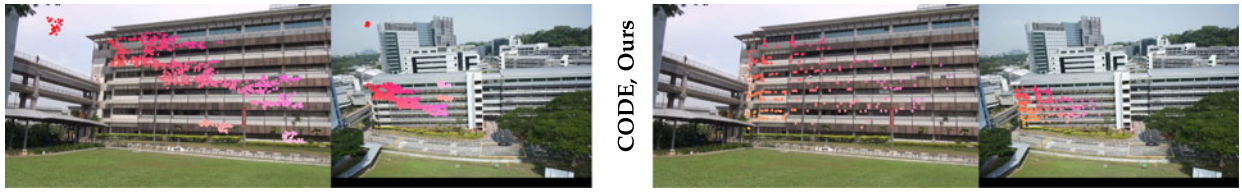
1) *Co-recognition dataset* [49] consists of six image pairs where significant sections of the scene is re-shuffled. It tests an algorithm's ability to handle large independent motions while providing large numbers of correspondence. Ground-truth segmentation boundaries are provided. A match is considered correct if it lies within a seven-pixel radius of the ground-truth segment boundary. The fraction of true matches is tabulated as *Av. Precision* in Fig. 9. We compute a correspondence area by dilating each match with a 12-pixel boundary [36]. The percentage of the total area overlapping with ground-truth segmentation is the *Precision (area)*, a stricter measure than *Av. Precision*. *Recall (area)* is the percentage of ground-truth segmentation detected after dilation. *F-number* from Eqn. (30)

CODE, Ours



Algo	Av. Precision	Av. Match Num	Av. Time /s	Max time/ s	Total Failures	Recall (area)	Precision (area)	**F-number (area)**
BD [29]	0.81	31.833	10.17	18.01	0/6	0.0781	0.786	0.1421
Mode Seeking [13]	0.96	915.16	14.09	20.10	0/6	0.4445	0.877	0.5901
NRDC(t) [36]	0.95	3.565e+04	27.30*	30.75*	0/6	0.5096	0.776	0.6155
NRDC [36]	0.80	2.653e+04	25.12*	35.28*	0/6	0.3291	0.716	0.4510
ACC [38]	0.99	68	3.00	3.62	0/6	0.1901	0.934	0.3159
CODE, Ours	0.99	6.668e+03	2.94	3.43	0/6	0.5900	0.939	0.7249

Fig. 9. Evaluation on the co-recognition dataset [49]. We use a Recall (area) metric which favors dense correspondence over feature matchers. Despite this, *CODE* is surprisingly competitive at the co-segmentation task. This is a testament to *CODE*'s number of correspondences and ability to handle multiple motion models. Example results are visualized in the top panel.



	Algo	Av. Precision	Av. Match Num	Area %	Av. Time/ s	Max time/ s	Total Failures	**F-number**
Set A	BD [29]	0.84	616.76	41.96%	102.85	577.6	6/13	0.657
	Mode Seeking [13]	0.86	606.07	40.64%	40.39	270.23	3/13	0.815
	NRDC(t) [36]	0.95	4.4946e+04	105.8%	24.9*	29.6*	5/13	0.747
	NRDC [36]	0.95	4.8556e+04	112.5%	24.7*	31.4*	4/13	0.803
	ACC [38]	0.96	65	18.8%	3.68	6.07	5/13	0.751
	CODE, Ours	0.96	4.41e+03	100%	2.97	6.27	1/13	0.940
	Algo	Av. Precision	Av. Match Num	Area %	Av. Time/ s	Max time/ s	Total Failures	**F-number**
Set B	BD [29]	0.61	14.1	13.5%	27.51	64.88	2/7	0.657
	Mode Seeking [13]	0.84	416.4	22.07%	8.14	9.98	3/7	0.680
	NRDC(t) [36]	0.94	1.843e+03	14.29%	25.69*	31.17*	5/7	0.439
	NRDC [36]	0.96	7.985e+03	48.03%	21.93*	26.59*	2/7	0.822
	ACC [38]	0.99	9.28	6.90%	1.87	2.60	5/7	0.444
	CODE, Ours	0.92	4.41e+03	100%	1.73	2.05	0/7	0.960

Fig. 10. Evaluating *CODE* against other feature matchers. A challenging image dataset and a set of metrics as described in Section 4.1 are used for this evaluation. The dataset contains full-resolution, representative images primarily from a variety of papers [13], [24], [26], [29]. Set A has limited out-of-plane rotation while Set B has very large viewpoint changes. *CODE* with constant parameters excels in a wide variety of scenarios. This is shown by low *Total Failures* and high *Precision*.

provides a performance summary. When comparing computational time, we exclude the runtime of feature detection and nearest-neighbor matching (our implementation is faster), and focus only on that of a core, correspondence evaluation/filtering algorithm. As *NRDC* does not use features, we provide its full computational time, and acknowledge this difference by indicating its runtime with a superscript '*'. A more detailed discussion on timing is provided in Section 4.4. We report all the results based on the above metrics in Fig. 9.

The area based statistics used for co-segmentation evaluation favors *NRDC*, a quasi-dense matching technique. However, our sparse feature matching method *CODE* remains surprisingly competitive, with its *Recall (area)* performance comparable to *NRDC(t)*. In addition, it retains the computational speed and precision traditionally associated with feature matchers.

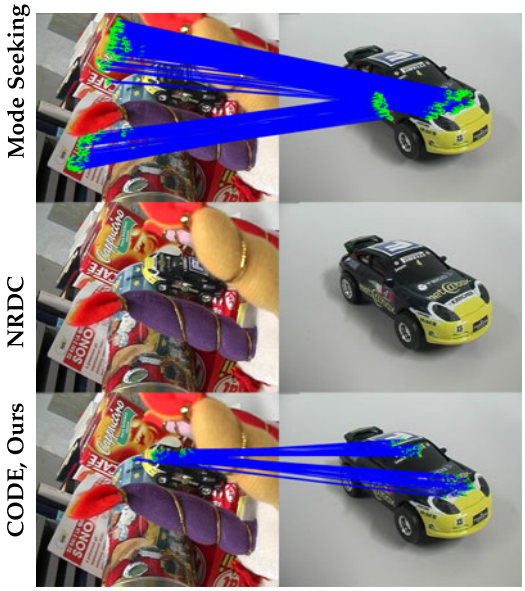
2) *Our dataset* consists of 20 images provided with the code of [13], [26], [29] with some additions. The images chosen have wide variations in scale, illumination, viewpoint change and image resolution. This is also the dataset used in Section 4.1, except that images are no longer down-sampled. Image pairs are divided in two sets. Set A has no large out-of-plane rotational motion and is in the working range of all matching algorithms. Images in Set B have large out-of-plane rotational motions and cannot be meaningfully handled by [29] which utilizes SIFT features. Results are tabulated in Fig. 10. For elaboration on individual statistics, please refer to Section 4.1. *CODE* works quite well across a wide variety of scenes as shown by the low number of *Total Failures*. Its matches are also of good quality as reflected by the over 90 percent precision.

3) *ETHZ-toys* [50], [51] consist of nine toys in different positions making a total of 40 object images. There are another 23 scene images where toys are hidden. In many scenes, the toy is, for instance, partially occluded, at a scale significantly different from that in its object image, distorted due to physical bending of the object, or having a glass encapsulation. We compute correspondence between every scene-toy image pair. If a scene contains a toy and a correspondence algorithm finds matches, it is considered a correct retrieval. If a scene does not contain a toy but an algorithm finds correspondences, it is considered an incorrect retrieval. Recall and precision statistics are computed for these trials and tabulated in Fig. 11. As processing 920 image pairs with all algorithms is prohibitively costly, we only compare to the top performers from the previous tests over the image sets A and B. The results in Fig. 11 show that *CODE* is general enough to accommodate large scale changes, non-rigid deformations and significant partial occlusion. It does so while avoiding correspondence on image pairs of different scenes. Over 920 image pairs, we had 0 incidence of incorrect retrieval and correctly retrieved 95:4 percent of the objects. This compares favorably to the next best algorithm, *NRDC*, with a precision and recall of 75 percent and 48:8 percent respectively.

To summarize the experiments, *CODE* demonstrates superior standings on a variety of challenging test cases, as a very competitive feature matching technique. In terms of precision, number of matches and computational time, it is consistently better than alternatives like *BD* [29] and *Mode Seeking* [13], [26]. Its match numbers is exceeded by quasi-dense matching *NRDC*. However, *CODE* can obtain reliable feature correspondences on many scenes where *NRDC* has no matches as seen in *Total Failures* and *Recall* from Figs. 10 and 11, respectively.

4.4 Computational Time

It is true that computational efficiency is not the most critical issue when the primary concern is on estimating quality



Algo	Precision	Recall	**F-number**
Mode Seeking [13]	0.2333	0.9767	0.3767
NRDC [36]	0.7500	0.4884	0.5915
CODE, Ours	1	0.9535	0.9762

Fig. 11. The *ETHZ-toys* dataset [50], [51]. We measure the object-retrieval correctness. Mode Seeking [13] tends to find correspondence on image pairs even when there exists no common objects. This produces high recall but low precision. NRDC [36] represents a better balance. In contrast, our algorithm is aggressive in match retrieval but also strict in enforcing correctness, leading to a high F-number.

feature matches for an image pair. However, applications like Structure from Motion and image stitching typically require all-pair feature correspondence, where the complexity scales quadratically with the dataset size. As a result, an innocuous 30-second runtime scales to 50 minutes for all-pair matching on a small 10-image set. In this respect, feature matchers have a significant advantage over dense matchers as the heavy cost absorbed in the feature computation is only incurred once per image. This leaves a relatively low recurring time shown in Table 2. Observe that our theoretical time on a 10-image dataset is 12.4 minutes, while *NRDC*'s is 45 minutes. In addition, our timings reported in Table 2 include many high resolution images, while *NRDC* used down-sampled 640×480 images. Such a clear runtime advantage enables a practical algorithm to leverage the enhanced accuracy offered by high resolution imagery to benefit challenging tasks such as Structure from Motion.

Next, we compare our computational time against other match verification or refinement alternatives like *BD* [29] and *Mode Seeking* [13], [26]. From Figs. 9 and 10, it can be seen that *CODE*'s average time is typically 5 to 6 times lower than alternatives. However, *CODE*'s maximum time can be one to two orders of magnitude lower, suggesting a scalability advantage. The picture is clearer in Fig. 12, which shows *CODE*'s computational cost scales well to handling large numbers of feature matches. This is due to *CODE*'s generalization ability. As proposed in Section 2.2, the computationally expensive regression step can be computed on a subset of M matches, where M is pre-defined. The subsequent match verification step is fast and scales efficiently with the increasing number of input matches N ($N \gg M$). This makes the overall cost $O(M)$.[4]

TABLE 2
Breakdown of Average Timing of Set A in Fig. 10

Time in seconds	Feature computation	Match	*CODE*	Recurring time	Total time
A-SIFT w CODE	8.62	4.37	2.96	7.33	15.95
NRDC [36]	-	26.98	-	26.98	26.98

We compare our feature matcher with NRDC, a dense matcher. Note that while our algorithm is only about 2 times faster, its recurring time is 3 times faster than NRDC, leading to significant efficiency gains on image sets of moderate or large size. See the text for a detailed discussion.

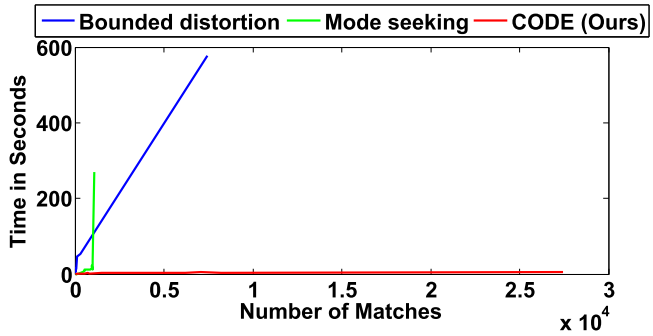


Fig. 12. *CODE* scales efficiently to high resolution images with many thousands of matches. Runtimes over different numbers of input feature matches N are plotted: red for *CODE*, blue for *BD* [29] and green for *Mode Seeking* [13].

5 Conclusion and Future Work

We present a feature correspondence algorithm in which a smoothness based regression is utilized to identify correct matches based on their shared coherence. By reducing the strain on feature uniqueness, it provides many more matches over wider-baselines than previous solutions. These results are especially remarkable for a novel formulation and suggest a promising research direction. Our current experiments are limited to sparse feature matching with an emphasis on speed and reliability. As such our solution ignores fine motion details and small independent motions. More visual illustrations are shown in Appendix B. However, by reliably modeling general motion, we believe our technique lays a solid foundation for subsequent works to enhance the density and fineness of motion estimates. Note that we have recently customized the algorithm for pose estimation in [52].

Acknowledgments

We thank Siying Liu and Nguyen Tan-Dat's for their contributions. In addition, Philip Torr was supported by the EPSRC, ERC grant ERC-2012-AdG 321162-HELIOS, EPRSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1. This study is supported by the Human-Centered Cyber-physical Systems Programme at Advanced Digital Sciences Center, DSO grant for Highly Accurate Camera Localization by Effective Image Matching, EPSRC and NSFC (NO. 61572264) and CAST young talents plan. Jiangbo Lu is the corresponding author.

4. Code is available at http://www.kind-of-works.com/

REFERENCES

[1] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, 2007.

[2] Microsoft, *Microsoft Image Composite Editor*, Microsoft, Redmond, WA, USA, 2011.

[3] C. Wu, "VisualSfM: A visual structure from motion system," 2011. [Online]. Available: http://ccwu.me/vsfm/

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[6] J. M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.

[7] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[9] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[10] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1385–1392.

[11] W.-Y. Lin, M. Cheng, J. Lu, H. Yang, M. Do, and P. Torr, "Bilateral functions for global motion modeling," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 341–356.

[12] H. Yang, W.-Y. Lin, and J. Lu, "DAISY filter flow: A generalized discrete approach to dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3406–3413.

[13] C. Wang, L. Wang, and L. Liu, "Progressive mode-seeking on graphs for sparse feature matching," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 788–802.

[14] C. Tomasi and R. Manduch, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.

[15] V. Codreanu, et al., "GPU-ASIFT: A fast fully affine-invariant feature extraction algorithm," in *Proc. Int. Conf. High Performance Comput. Simul.*, Jul. 2013, pp. 474–481.

[16] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

[17] W.-Y. Lin, M.-M. Cheng, S. Zheng, J. Lu, and N. Crook, "Robust non-parametric data fitting for correspondence modeling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2376–2383.

[18] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. British Mach. Vis. Conf.*, 2002, pp. 36.1–36.10.

[19] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 128–142.

[20] D. Min and K. Sohn, "Edge-preserving simultaneous joint motion-disparity estimation," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, pp. 74–77.

[21] A. L. Yuille and N. M. Grzywacz, "The motion coherence theory," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 1988, pp. 344–353.

[22] A. Myronenko, X. Song, and M. Á. Carreira-Perpiñán, "Non-rigid point set registration: Coherent point drift," in *Proc. Advances Neural Inf. Process. Syst.*, 2006, pp. 1009–1016.

[23] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong, "Smoothly varying affine stitching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 345–352.

[24] W.-Y. Lin, L. Liu, Y. Matsushita, K.-L. Low, and S. Liu, "Aligning images in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1–8.

[25] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 596–609.

[26] C. Wang, L. Wang, and L. Liu, "Density maximization for improving graph matching with its applications," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2110–2123, Jul. 2015.

[27] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 313–320.

[28] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 492–505.

[29] Y. Lipman, S. Yagev, R. Poranne, D. W. Jacobs, and R. Basri, "Feature matching with bounded distortion," *ACM Trans. Graph.*, vol. 33, no. 3, May 2014, Art. no. 26.

[30] D. Pizarro and A. Bartoli, "Feature-based deformable surface detection with self-occlusion," *Int. J. Comput. Vis.*, vol. 97, no. 1, pp. 54–70, Mar. 2012.

[31] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[32] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 500–513.

[33] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understanding*, vol. 78, pp. 138–156, 2000.

[34] J.-C. Bazin, H. Li, I. Kweon, C. Demonceaux, P. Vasseur, and K. Ikeuchi, "A branch and bound approach to correspondence and grouping problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1565–1576, Jul. 2013.

[35] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418–433, Mar. 2005.

[36] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," in *Proc. ACM SIGGRAPH*, 2011, pp. 70:1–70:9.

[37] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "SIFT flow: Dense correspondence across different scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 28–42.

[38] M. Cho, J. Lee, and K. M. Lee, "Feature correspondence and deformable object matching via agglomerative correspondence clustering," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 1280–1287.

[39] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, no. 3, pp. 321–355, 1988.

[40] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Statist. Soc.*, vol. 28, no. 1, pp. 100–108, 1979.

[41] S. Agarwal, K. Mierle, and Others, "Ceres solver," (2010). [Online]. Available: http://ceres-solver.org

[42] Peter Kovesi: Matlab and octave functions for computer vision and image processing. (1991). [Online]. Available: http://www.peterkovesi.com/matlabfns/, Accessed on: Jul. 20, 2015.

[43] Yasutaka Furukawa and Jean Ponce: 3d photography dataset. (2010). [Online]. Available: http://www-cvr.ai.uiuc.edu/ponce_grp/data/mview/, Accessed on: Jul. 20, 2015.

[44] C. Wu, "Towards linear-time incremental structure from motion," in *Proc. Int. Conf. 3D Vis.*, 2013, pp. 127–134.

[45] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3057–3064.

[46] M. Jancosek and T. Pajdla, "Multi-view reconstruction preserving weakly-supported surfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3121–3128.

[47] M. Waechter, N. Moehrle, and M. Goesele, "Let there be color! Large-scale texturing of 3D reconstructions," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 836–850.

[48] Agisoft: 3D reconstruction system. [Online]. Available: http://www.agisoft.com, Accessed on: Jul. 20, 2015.

[49] M. Cho, Y. M. Shin, and K. M. Lee, "Co-recognition of image pairs by data-driven Monte Carlo image exploration," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 144–157.

[50] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Simultaneous object recognition and segmentation by image exploration," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 145–169.

[51] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *Int. J. Comput. Vis.*, vol. 67, no. 2, pp. 159–188, 2006.

[52] W.-Y. Lin, S. Liu, N. Jiang, M. N. Do, P. Tan, and J. Lu, "RepMatch: Robust feature matching and pose for reconstructing modern cities," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 562–579.

Wen-Yan Lin received the PhD degree from National University of Singapore, in 2012, supervised by Prof. Loong-Fah Cheong and Dr. Dong Guo. He subsequently worked for the Institute of Infocomm Research Singapore and Prof. Philip Torr. He is currently a research scientist in the Advanced Digital Sciences Center Singapore.

Fan Wang received the BEng degree in computer science and technology from Lanzhou University, Lanzhou, China, in 2012 and the MS degree in electrical engineering from National University of Singapore, Singapore, in 2013. He is with the Advanced Digital Sciences Center, Singapore, as a software engineer. His research interests include 3D computer vision and 3D reconstruction.

Ming-Ming Cheng received the PhD degree from Tsinghua University, in 2012, supervised by Prof. Shi-Min Hu. Then, he did two years research fellow, with Prof. Philip Torr in Oxford. He is currently an associate professor with Nankai University. His research interests include computer graphics, computer vision, and image processing. He has received the Google PhD fellowship award, the IBM PhD fellowship award, etc.

Sai-Kit Yeung received the BEng (First Class Hons.) degree in computer engineering and the MPhil degree in bioengineering from HKUST, in 2003 and 2005, respectively, and the PhD degree in electronic and computer engineering from Hong Kong University of Science and Technology (HKUST), in 2009. He is an assistant professor with Singapore University of Technology and Design (SUTD), where, he leads the Vision, Graphics and Computational Design (VGD) Group. He is also a visiting assistant professor with Stanford University and MIT. Before joining SUTD, he had been a postdoctoral scholar in the Department of Mathematics, University of California, Los Angeles (UCLA). He was also a visiting student in the Image Processing Research Group, UCLA, in 2008 and in the Image Sciences Institute, University Medical Center Utrecht, The Netherlands, in 2007.

Philip H.S. Torr received the PhD degree from Oxford University. After working for another three years with Oxford, he worked for six years as a research scientist for Microsoft Research, first in Redmond, then in Cambridge, founding the vision side of the Machine Learning and Perception Group. He is now a professor with Oxford University. He has won awards from several top vision conferences, including ICCV, CVPR, ECCV, NIPS, and BMVC. He is a senior member of the IEEE, Royal Society Wolfson Research Merit Award holder, and program co-chair of ICCV 2013.

Minh N. Do received the BEng degree from the University of Canberra, Australia, in 1997, and the DrSci degree from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2001. Since 2002, he has been a faulty with the University of Illinois at Urbana-Champaign (UIUC) and holds joint appointments with the Coordinated Science Laboratory, the Beckman Institute for Advanced Science and Technology, and the Department of Bioengineering. His research interests include image and multidimensional signal processing, wavelets and multiscale geometric analysis, computational imaging, augmented reality, and visual information representation. He received the Silver Medal from the 32nd International Mathematical Olympiad in 1991, a University Medal from the University of Canberra in 1997, a CAREER Award from the National Science Foundation in 2003, and a Young Author Best Paper Award from IEEE in 2008. He was named a Beckman fellow with the Center for Advanced Study, UIUC, in 2006, and received of a Xerox Award for Faculty Research from UIUC, in 2007. He was a member of the IEEE Signal Processing Theory and Methods Technical Committee, Image, Video, and Multidimensional Signal Processing Technical Committee, and an associate editor of the *IEEE Transactions on Image Processing*. He is a fellow of the IEEE.

Jiangbo Lu received the BS and MS degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively, and the PhD degree in electrical engineering from Katholieke Universiteit Leuven, Leuven, Belgium, in 2009. From April 2003 to August 2004, he was with VIA-S3 Graphics, Shanghai, China, as a Graphics Processing Unit (GPU) Architecture Design Engineer. In 2002 and 2005, he conducted visiting research with Microsoft Research Asia, Beijing, China. Since October 2004, he has been with the Multimedia Group, Interuniversity Microelectronics Center, Leuven, Belgium, as the PhD researcher. Since September 2009, he has been working in the Advanced Digital Sciences Center, Singapore, which is a joint research center between the University of Illinois at Urbana-Champaign, Urbana, and the Agency for Science, Technology and Research (A*STAR), Singapore, where, he is leading a few research projects. His research interests include computer vision, visual computing, image and video processing, interactive multimedia applications and systems, and efficient algorithms for various architectures. He served as an associate editor of the *IEEE Transactions on Circuits and Systems for Video Technology* in 2012-2016. He received the 2012 TCSVT Best Associate Editor Award. He is a senior member of the IEEE.