Int J Comput Vis
DOI 10.1007/s11263-013-0657-5

An Interactive Approach to Solving Correspondence Problems

Stefanie Jegelka · Ashish Kapoor · Eric Horvitz

Received: 24 February 2013 / Accepted: 29 August 2013

Abstract Finding correspondences among objects in different images is a critical problem in computer vision. Even good correspondence procedures can fail, however, when faced with deformations, occlusions, and differences in lighting and zoom levels across images. We present a methodology for augmenting correspondence matching algorithms with a means for triaging the focus of attention and effort in assisting the automated matching. For guiding the mix of human and automated initiatives, we introduce a measure of the expected value of resolving correspondence uncertainties. We explore the value of the approach with experiments on benchmark data.

Keywords Human interaction · Active learning · Value of information · Matching · Correspondence problems

1 Introduction

Identifying correspondences among similar or identical objects appearing in different images is a ubiquitous problem in computer vision, and promising advances have been made with algorithms for identifying such correspondences. Nevertheless, the success of these methods is variable and can be sensitive to multiple factors, including differences in image resolution, lighting conditions and zoom level across images, occlusions that block views, and rigid or non-rigid deformations of objects. In hard cases, correspondence algorithms may return partial results where some subset of matches is identified with confidence. We describe a methodology for refining such partial matching results. Our methods selectively seek human or machine effort to resolve key uncertainties in correspondences.

We specifically pursue answers to the following questions: (1) *What kind of* additional information can be used to improve the mapping while being obtainable with reasonable effort, (2) how can such information be obtained *efficiently* in terms of computational effort and other costs, and finally (3) how can such additional information be *integrated* with ease so as to refine the correspondences?

We analyze the information gained with verifying correct and incorrect matches in a partial solution to a correspondence problem. Such verification resolves uncertainty about selected correspondences and, importantly, also introduces new structural and topological constraints in an interactive manner that guide forthcoming human efforts at resolving uncertainties about other correspondences. Beyond focusing the attention and effort of people, our methods can be used to triage the application of computationally intensive subroutines.

We focus on the use of methods that alternate between recruiting human assistance to verify the most informative matches and propagating their implications to compute an updated solution. Engaging people to assist introduces additional considerations of usability where we wish the tasks to be simple enough to be completed successfully by people. For example, we limit the verification of correspondences to pairwise checks.

Core contributions of this paper include (1) a decision-theoretic criterion for a cost-efficient, active selection of cor-

Work done during an internship of S.J. at Microsoft Research.

S. Jegelka (✉)
UC Berkeley, Berkeley, CA, USA
e-mail: stefje@eecs.berkeley.edu

A. Kapoor · E. Horvitz
Microsoft Research Redmond, Redmond, WA, USA
e-mail: akapoor@microsoft.com

E. Horvitz
e-mail: horvitz@microsoft.com

Published online: 22 September 2013

respondence refinement tasks, (2) a general model for incorporating human input in correspondence problems, and (3) crowdsourcing experiments whose results demonstrate how human input improves results in the combinatorial matching problem.

1.1 Preliminaries

We are given two point sets $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_m\}$ between which we aim to establish pairwise correspondences. Each point x is characterized by one or more features $\boldsymbol{\psi}(x)$, e.g. location or appearance. In addition, we might construct neighborhood graphs $\mathcal{G}_\mathcal{X}$, $\mathcal{G}_\mathcal{Y}$. We aim to find a mapping $f : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ that shows correspondences between $\mathcal{X}$ and $\mathcal{Y}$. If a point x_i is mapped to $\perp$, then it has no correspondent in $\mathcal{Y}$, e.g. in case of occlusion.

We can demonstrate the gain of interaction in the simplest, linear assignment model. The approach integrates in a straightforward manner into quadratic or more sophisticated models as well, where it can be viewed as creating more informative features. We define pairwise costs $c(x, y)$ for matching point $x \in \mathcal{X}$ to $y \in \mathcal{Y}$. Initially, we set the costs $\widetilde{C}$ to the matrix of distances $c_{ij} = d(\boldsymbol{\psi}(x_i), \boldsymbol{\psi}(y_j))$. A feasible matching is injective, i.e., $f(x_i) \neq f(x_j)$ whenever $x_i \neq x_j$ and $f(x_i) \neq \perp$. To account for unmatched points, we introduce m auxiliary points $\mathcal{X}^\perp = \{x_1^\perp, \ldots, x_m^\perp\}$ and n points $\mathcal{Y}^\perp = \{y_1^\perp, \ldots, y_n^\perp\}$. Now, a feasible matching is a bijective function between elements of $\mathcal{X} \cup \mathcal{X}^\perp$ and $\mathcal{Y} \cup \mathcal{Y}^\perp$. We denote the set of all feasible matchings by $\mathcal{M}$, and we aim to find the matching that minimizes the costs $c(f)$:

$$\min_{f \in \mathcal{M}} \sum_{x \in (\mathcal{X} \cup \mathcal{X}^\perp)} c(x, f(x)). \tag{1}$$

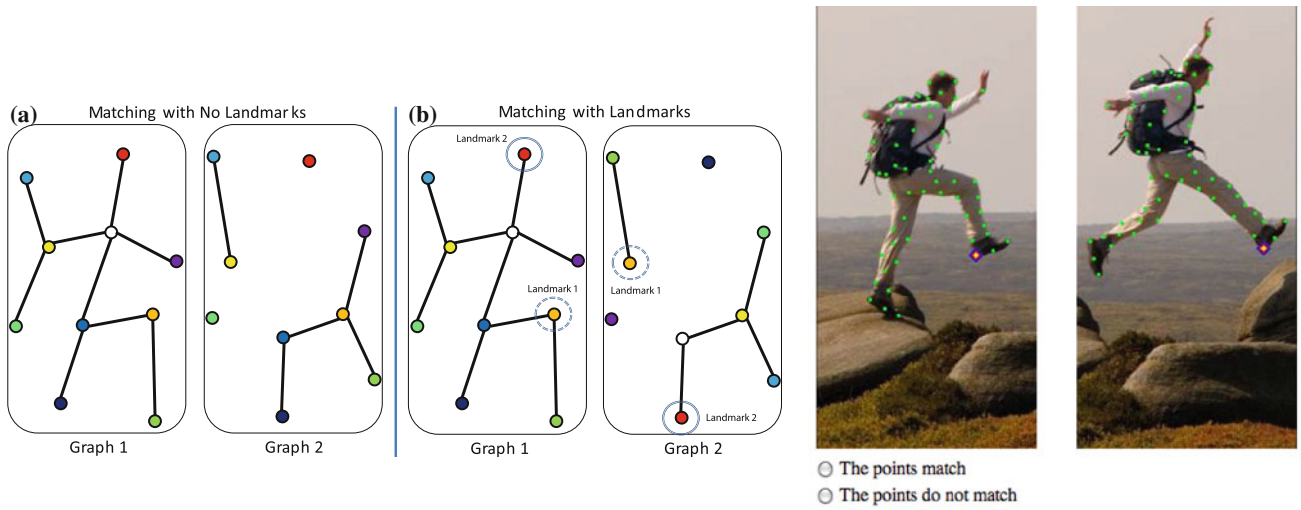
The cost of matching any auxiliary point is defined by a threshold θ: $c(x_i^\perp, y) = c(x, y_j^\perp) = \theta$. As θ is lowered, increasing numbers of points remain unmatched. For ease of notation, in the sequel we will implicitly include $\mathcal{X}^\perp$ in $\mathcal{X}$ and $\mathcal{Y}^\perp$ in $\mathcal{Y}$. The optimization problem (1) can be solved by the Hungarian algorithm or Munkres' method (Munkres 1957).

Not all features may be equally suited for a direct comparison $d(\boldsymbol{\psi}(x), \boldsymbol{\psi}(y))$ across data sets. Coordinates, for example, can fail for rotations or non-rigid objects. In such cases, it may be more appropriate to use *relative* features, capturing as attributes of points their relation to *reference* points within the data set, and to compare such relations. In this paper, we will use such relational features. These features introduce parts of the quadratic assignment problem into our simple model, but, as opposed to quadratic assignment problems, the resulting optimization problem will still be solvable exactly.

1.2 Related Work

Point correspondence problems are employed in a multitude of applications in computer vision. Mapping points across images is important in object or (3D) shape matching, 3D reconstruction, motion segmentation, and image morphing. These problems differ in terms of assumptions on the nature of the transformations, the objects under consideration, and in assumptions on the given information. Among the simplest are transformations of rigid bodies, where geometry can be exploited (Goodrich and Mitchell 1999; McAuley et al. 2008), while correspondences among non-rigid objects, and between non-identical objects, can pose significant challenges. Algorithms applied to more general correspondence problems largely combine the compatibility of points by features with the local geometric compatibility of matches. Such models can be formulated as graphical models (McAuley et al. 2008; Torresani et al. 2008; Starck and Hilton 2007) or as selecting nodes in an association graph (Lordeanu and Hebert 2005; Cho et al. 2010; Cour et al. 2006), and have been extended to higher-order criteria (Duchenne et al. 2009; Zass and Shashua 2008; Lee et al. 2011). Other methods consider the Laplacian constructed from a neighborhood graph (Umeyama 1988; Escolano et al. 2011; Mateus et al. 2008), and some models are learned from full training examples (Torresani et al. 2008; Caetano et al. 2009). Closest to the idea of using reference points are approaches based on seed points (Sharma et al. 2011), coarse-to-fine strategies (Starck and Hilton 2007), and guessed points that help orient the remaining points in a rigid body (McAuley and Caetano 2012). None of these models, however, explicitly seek and incorporate updates from user interactions. Our focus on actively gaining information is orthogonal to ongoing work on enhancing matching methods as described above. While we use simple low-order models for exposition and experiments, we note that the proposed method is compatible with higher-order models, and easily extends to the procedures described in this section.

Other related work includes multiple efforts to use human input for improving computer vision (Vijayanarasimhan 2011; von Ahn and Dabbish 2004). Many of these approaches pursue *active learning* to guide human annotation effort for curating training data. Criteria such as uncertainty (Kapoor et al. 2009), disagreement among a committee of classifiers (Freund et al. 1997), the structure of the version space (Tong and Koller 2000), or expected informativeness (MacKay 1992; Lawrence et al. 2002) have been proposed for choosing unlabeled points for tagging data for supervised machine classification. Active learning has also been used for image annotation (Joshi et al. 2009) and object detection (Vijayanarasimhan and Kapoor 2010). These and other related studies focus inherently on classification and on the goal of minimizing misclassification rates. Recent



(a) Landmarks **(b)** Sample query

Fig. 1 **(a)** Example of how landmarks help identify correct correspondences. Matching in the absence of landmarks can lead to a suboptimal solution (*left*, matched pairs indicated with *same color*) out of sets of ambiguous solutions. Ambiguity can be removed by providing two landmarks (*large circles*), which results in the correct solution (*right*). **(b)** Sample query to the user on confirming a match

approaches explore the decision-theoretic notion of *value of information* (VOI) (Howard 1967; Heckerman et al. 1992), where the expected value of information under uncertainty is computed to balance the cost of making a mistake with the costs of acquiring labels from human experts. The use of VOI as a criterion for selective supervision has been explored in the realm of supervised learning (Kapoor et al. 2007), sensor placement (Krause et al. 2008), and more recently in the context of visual recognition and detection (Vijayanarasimhan and Kapoor 2010). In related work on human computation and crowdsourcing, a Monte Carlo procedure for computing value of information for long sequences of human inputs (Kamar and Horvitz 2013) is used to fuse machine vision and human perception in a citizen science task for astronomy (Kamar et al. 2012). Interaction different from the verifications employed in this work has been used for 3D reconstruction (Kowdle et al. 2011; Debevec et al. 1996). Maji and Shakhnarovich (2012) propose a framework that lets users decide on locations of landmarks, but without active querying. Also relevant to the current study are earlier efforts referred to as *active matching* that aim at reducing the search space in matching problems (Chli and Davison 2008; Handa et al. 2010). Our work differs from those methods in that we address how to seek additional information from people, with the challenge of posing tasks that are feasible for humans to solve.

2 Improving Matchings Via Interaction

An algorithm that perfectly solves all types of correspondence problems has been an elusive goal, but many existing methods can achieve partially correct matchings. For refining an initial imperfect matching, we first examine how additional information can help to achieve better results. Then we discuss how we can obtain this information efficiently. First, we observe that any settled correspondence propagates information to the remaining candidate matches, as good mappings are coupled by the structural bijection constraint. Second, we use the concept of *landmarks* that provide orientation for the matching task.

Definition 1 A *landmark* is a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ that is a correct match and that is used as a reference for creating features that are comparable across $\mathcal{X}$ and $\mathcal{Y}$.

Figure 1a illustrates this intuitively. Graph 2 is a simple perturbation of Graph 1, derived by removing a single node followed by a 180 degree rotation. Matching without any landmarks (left) is a weakly constrained problem. Depending on the choice of algorithm, we can obtain numerous solutions (e.g., Fig. 1a left). However, a few landmarks (Fig. 1a right) make the problem significantly easier, as the added constraints rule out ambiguities. Landmarks will play a central role in the approach that we next describe in detail.

2.1 Using Landmarks

Knowledge of landmarks can help to solve a correspondence task by inducing constraints that impose topological and geometric invariants. We propose to use the *relation* of points to the collection $\mathcal{L}$ of landmarks (x^ℓ, y^ℓ) as additional information for a better matching. To distinguish landmark points from regular points, we index them by superscript ℓ.

Given a collection $\mathcal{L}$, we compute additional feature vectors $\boldsymbol{\phi}(x), \boldsymbol{\phi}(y) \in \mathbb{R}^{|\mathcal{L}|}$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ that are comparable *across* point sets. In particular, for a given distance d and landmarks $\ell = (x^\ell, y^\ell) \in \mathcal{L}$, we introduce the features

$$\phi_\ell(x) = d(x, x^\ell) \quad \text{and} \quad \phi_\ell(y) = d(y, y^\ell). \tag{2}$$

These *landmark features* describe the relationship of each point to the set of landmarks. Such descriptors easily extend to multiple landmarks, for example for describing angles. For a candidate match (x, y) the features $\phi_\ell(x)$ and $\phi_\ell(y)$ provide new compatibility information, as they allow for comparing the relation of x to x^ℓ with that of y to y^ℓ. This information is similar to compatibilities of pairs of matches used in quadratic methods, but, contrary to those approaches, landmark features do not affect the hardness of the optimization problem (1). We integrate the landmark features into an additional cost matrix $D_\mathcal{L}$ of distances between feature vectors, e.g., with ℓ_2 distances, $d_{ij} = \|\boldsymbol{\phi}(x_i), \boldsymbol{\phi}(x_j)\| = \sqrt{\sum_\ell (\phi_\ell(x) - \phi_\ell(y))}$, or ℓ_1 distances $d_{ij} = \sum_\ell |\phi_\ell(x) - \phi_\ell(y)|$.

We linearly combine $D_\mathcal{L}$ and the matrix C_{init} of initial costs (e.g., distances between $\boldsymbol{\psi}(x)$, $\boldsymbol{\psi}(y)$) to a joint cost $C = (1-\alpha)\bar{C}_{\text{init}} + \alpha \bar{D}_\mathcal{L}$, where the bar denotes normalized matrices, $\bar{C} = (\max_{i,j} c_{ij})^{-1} C$. A mixing coefficient α balances initial and newly introduced information, and can be adjusted as $\mathcal{L}$ grows. We have found a concave increase in α to be suitable.

2.1.1 Distance Functions

We propose two variants of distance functions to compute the features ϕ_ℓ in Eq. (2): Euclidean and commute distances. For Euclidean distances, each point x must have a descriptor $\boldsymbol{\psi}(x)$ which includes its location, SIFT, or other local features. Then we have $d(x, x^\ell) = \|\boldsymbol{\psi}(x) - \boldsymbol{\psi}(x^\ell)\|$ (and analogously for $d(y, y^\ell)$).

The commute distance arises from a graph representation, and applies for example if only neighborhood relations within $\mathcal{X}$ and $\mathcal{Y}$ are known or decisive. Given a neighborhood graph $\mathcal{G}_\mathcal{X}$ on $\mathcal{X}$, the *commute distance* between x and x^ℓ is the square root of the expected time a random walk on $\mathcal{G}_\mathcal{X}$ would take to wander from x to x^ℓ and back. This can be computed as a distance between features derived from the eigenvectors of a graph Laplacian (Lovász 1993). In practice, we found it often more robust to truncate commute distances to a maximum threshold.

2.1.2 Updates

We propose a simple design where, given the current matching f, the algorithm selects a proposed matched pair $(x_i, f(x_i)) \in \mathcal{X} \times \mathcal{Y}$ and asks the user for verification: "is $(x, f(x))$ a match?" (illustrated in Fig. 1b). If the engaged human confirms, the introduced landmark is used to update the cost matrix with a new feature. If the human judges the match as incorrect, the system adds a large constant γ to the entry of C that refers to $(x, f(x))$. This constant is chosen to depict a high enough cost to prevent that those candidates are ever matched in future refinements. If the acquired input on the match confirms that it is correct, a new landmark ℓ is introduced. The matrix $D_\mathcal{L}$ can be updated efficiently: Let $\Phi_\ell(\mathcal{X}, \mathcal{Y})$ be the matrix whose (i, j)th entry is the ℓ_2 distance $\|\phi_\ell(x_i) - \phi_\ell(x_j)\|$ between added features. Then $D_{\mathcal{L}\cup\{\ell\}} = \sqrt{D_\mathcal{L}^2 + \Phi_\ell(\mathcal{X}, \mathcal{Y})^2}$, where the square root and square are element-wise.

2.2 Seeking Good Landmarks

Starting with an initial matching based on point features, the algorithm continues to incorporate additional (higher-order) information at every query to refine the solution. With this flexibility, we aim to be query-*efficient* and achieve the best possible match with as few queries as possible. To select maximally informative queries, we select the pair $(x, f(x))$ that maximizes the *expected gain*. This gain is computed as the sum of the gains for the outcomes where people confirm versus disconfirm a proposed match, weighted by the probabilities of each outcome:

$$p(\text{match})\text{gain}(\text{match}) + p(\neg\text{match})\text{gain}(\neg\text{match}). \tag{3}$$

Two quantities needed for this computation are (1) the confidence that the query will be assessed as a match, and (2) the gain associated with either answer.

2.2.1 Estimates of Gain

The gain represents the amount of additional information about correspondences that can be obtained via learning whether a candidate pair is a match. We define two different criteria for estimating the gain associated with landmark candidates. Each moves beyond the local element-wise cost function defined earlier. The first criterion involves the propagation of information from the assessed landmarks across the set of points. The second factor considers the structure of the combinatorial optimization problem and relates to margin maximization and version spaces.

2.2.2 Covering

The first criterion aims at "covering" the set of points with landmarks, ensuring that each point has at least one landmark sufficiently close by. We formulate this criterion as a covering problem. The common algorithm to cover a space with as few landmarks as possible in polynomial time is greedy

(Chvatal 1979): sequentially, always add the landmark that covers the maximum number of additional points. The value of information essentially implements this rule in a probabilistic setting.

Formally, we say a point x is *covered* by x^ℓ if it falls in a ball $\mathcal{B}_r(x^\ell) = \{x | d(x, x^\ell) < r\}$ around the landmark x^ℓ. Conversely, a landmark ℓ covers all points in the balls around its components x^ℓ, y^ℓ. The gain of landmark ℓ is

$$\rho(\ell) = |\mathcal{B}_r(x^\ell)| + |\mathcal{B}_r(y^\ell)| \tag{4}$$

$$= |\{x | d(x, x^\ell) < r\}| + |\{y | d(y, y^\ell) < r\}|. \tag{5}$$

If we have already selected a set of landmarks $\mathcal{L}$, then we only count the *marginal* gain with respect to those landmarks. Let $\mathcal{B}_{r,\mathcal{X}}(\mathcal{L}) = \bigcup_{\ell \in \mathcal{L}} \mathcal{B}_r(x^\ell)$ be the union of the points in $\mathcal{X}$ covered by any landmark, and analogously $\mathcal{B}_{r,\mathcal{Y}}(\mathcal{L})$. Then the marginal gain of a new pair ℓ given $\mathcal{L}$ is

$$\rho(\ell \mid \mathcal{L}) = |\mathcal{B}_r(x^\ell) \cup \mathcal{B}_{r,\mathcal{X}}(\mathcal{L})| - |\mathcal{B}_{r,\mathcal{X}}(\mathcal{L})| + |\mathcal{B}_r(y^\ell) \cup \mathcal{B}_{r,\mathcal{Y}}(\mathcal{L})| - |\mathcal{B}_{r,\mathcal{Y}}(\mathcal{L})|. \tag{6}$$

When judging gain by covered area, the radius r of the balls determines the density of landmarks. We initialize r by a large value (the average distance to the $\sqrt{n}/3$th nearest neighbor point) to spread the first few landmarks widely. When nearly all points are covered, there is no more difference in the gain of any additional landmark. In this case, if there is budget for more landmarks, we reduce r by one third, so that subsequent landmarks fill the gaps among existing landmarks and ensure a closer landmark for each point.

A finer measure of covering allows each point to be covered by k landmarks. Let $\mathrm{cov}(x)$ be the number of landmarks whose balls cover x. A refined gain is

$$\rho_k(\ell \mid \mathcal{L}) = \sum_{x \in \mathcal{B}_r(x^\ell)} [k - \mathrm{cov}(x)]_+ + \sum_{y \in \mathcal{B}_r(y^\ell)} [k - \mathrm{cov}(y)]_+,$$

where $[a]_+ = \max\{a, 0\}$.

The gain of a non-match (negative user feedback) is a *no-match* constraint, and scored as a constant ν for all pairs. Using (3), we query for assessments about the pair (x, y) that maximizes the score

$$\hat{p}(f(x) = y)\rho((x, y)|\mathcal{L}) + (1 - \hat{p}(f(x) = y))\nu. \tag{7}$$

2.2.3 Stability

Some active learning methods are aimed at minimizing the version space—the set of likely hypotheses that are consistent with the current observations (Dasgupta 2004). This goal of stability is addressed by selecting a query whose answer leaves a version space with little mass, meaning that only few likely solutions remain. If we view our cost as a potential, then this rule means that we select a landmark ℓ whose features ϕ_ℓ rule out many good candidate solutions and thereby reduce ambiguity.

As computing the mass of the version space is expensive, we estimate it by the gap between the best and the second-best solution. Maximizing this gap is also the aim of methods that maximize a margin. A wide gap indicates little ambiguity. Thus, we seek landmarks whose addition in expectation maximizes this gap. We compute the gap both for the case where the query pair is indeed a match, and for the case of negative feedback. Negative feedback can be beneficial if it helps rule out one of two nearly good solutions and thereby widens the gap between the two best allowed solutions.

The second-best solution of a matching can be computed via shortest paths in a bipartite graph (Chegireddy and Hamacher 1987): given the optimal matching f, we construct a complete bipartite graph $\mathcal{G} = (\mathcal{X}, \mathcal{Y}, \mathcal{E})$. For every pair (x, y) that is currently not matched, i.e., $f(x) \neq y$, there is a directed edge (x, y) with weight $c(x, y)$. In addition, for each pair $(x, f(x))$, there is an edge $(f(x), x)$ in the other direction, with weight $-c(x, f(x))$. For each match $(x', f(x'))$, the shortest path from x' to $f(x')$ in $\mathcal{G}$ forms a cycle together with the edge $(f(x'), x')$. The cost of this cycle $\mathcal{C}(x') \subset \mathcal{E}$, i.e., the sum of its edge weights, is the difference between the cost of the optimum solution f and the optimum solution that does not map x' to $f(x')$:

$$\min_{g \in \mathcal{M}, g(x') \neq f(x')} c(g) - c(f) = \sum_{e \in \mathcal{C}(x')} w(e) \tag{8}$$

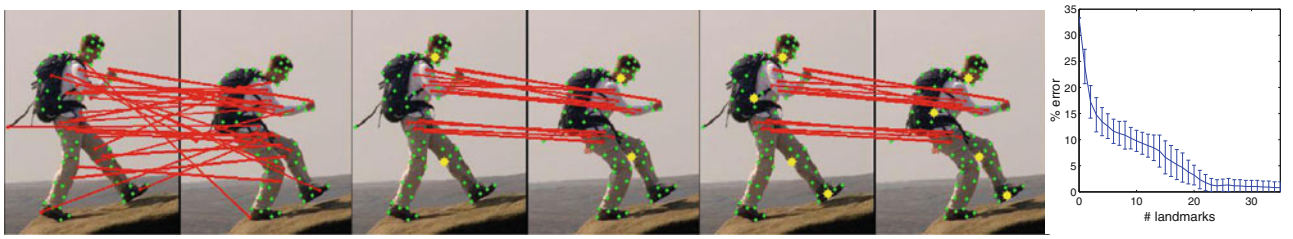
Thanks to the optimality of f, the graph $\mathcal{G}$ does not have any negative cycles (Chegireddy and Hamacher 1987), and therefore the shortest cycle $\mathcal{C}(x')$ is easy to compute. The length of the shortest cycle is the desired gap. The same method applies to find gaps for the best solutions that exclude any given tentative landmark pair; this is the gain if the feedback is negative. We substitute these gains into Equation (3).

2.2.4 Probability of a Match

The confidence in the match $(x, f(x))$ can be estimated by comparing the fit of $f(x)$ to that of other possible matches $y \in \mathcal{Y}$. The more good candidates, the lower the confidence. We estimate confidences conservatively as

$$\hat{p}(f(x_k) = y_i) = \min \left\{ \frac{\exp(-c(x_k, y_i))}{\sum_j \exp(-c(x_k, y_j))}, \frac{\exp(-c(x_k, y_i))}{\sum_j \exp(-c(x_j, y_i))} \right\}. \tag{9}$$

This quantity estimates at the same time how easy a human may find it to verify the candidate match, and gives preference to more identifiable candidates.



(a) no landmark: err = 44% **(b)** 2 landmarks: err = 20% **(c)** 4 landmarks: err = 18% **(d)** random landmarks

Fig. 2 Effect of adding landmark features. As few as 1–2 landmarks eliminate global mismatches. Mismatches are highlighted with *red arcs*, *green points* are matched correctly, *yellow marks* indicate landmarks; **(d)** average error over 20 image pairs if random landmarks are added (Color figure online)

2.2.5 Thresholds

A further potential gain of feedback is to adapt the threshold θ that determines when we "trust" a match, and when we leave a point unmatched. This threshold is equivalent to the penalty that we assign to unmatched points. We update the threshold multiplicatively down to a given lower bound. When a match has small distance (resulting in higher confidence) and the feedback indicates that there is no match, we reduce the current θ multiplicatively. Otherwise, when observing a match whose distance is above the threshold, we adapt θ by multiplying by a factor larger than one.

3 Experiments

We now report on experiments for evaluating the proposed approaches. The experiments suggest that improvements can be achieved by adding landmark features in a selective manner. We compare the proposed methodology to the baselines of (1) not adding any new features, and (2) selecting queries uniformly at random from the pairs $(x, f(x))$. We always query matched pairs, keeping in mind that this is still more informative than querying arbitrary pairs from $\mathcal{X} \times \mathcal{Y}$. We use mostly Euclidean distances for computing ϕ_ℓ, and ℓ_1 or ℓ_2 distances between the vectors of landmark features.

3.1 Usefulness of Landmarks in General

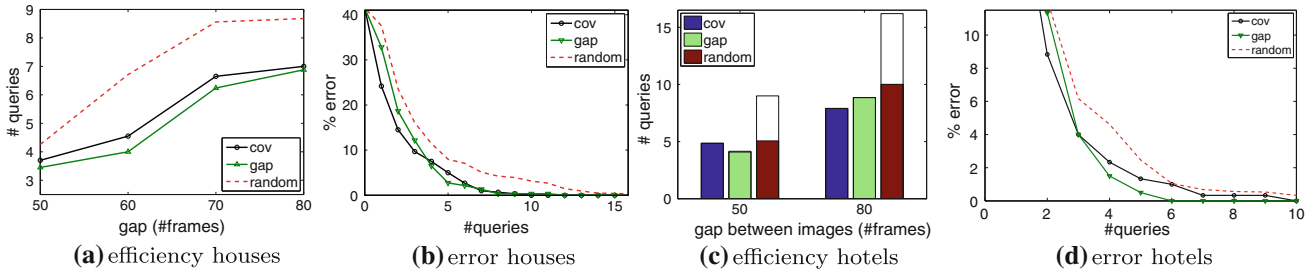
First, we establish whether landmarks can improve the quality of matchings. Figure 2 shows a sample matching computed from initial SIFT features only, and subsequent improvements when landmark information is added. Here, we introduced landmarks (correct matches) in a random manner, with α increasing from 0.65 to 0.95 over 15 landmarks. Few landmarks suffice to rule out mismatches where $f(x)$ is very far from the true match $y(x)$, such as matches between a point on the foot of Person 1 to the neck of Person 2. This effect is similar to the effect of structural constraints illustrated in (Torresani et al. 2008, Fig. 3). Figure 2d shows that over many randomly drawn landmarks, the added information improves the results on average. The variance suggests that the actual gain depends on the set of landmarks, and how well the landmarks complement one another.

3.2 Selective Querying

Knowing that landmarks can be beneficial, we test the effectiveness and efficiency of employing selective *queries*. We compare the two selection strategies from Sect. 2.2.1 with a baseline of randomly selected queries. The strategies *cov* and *gap* vary in how they estimate gain: 'cov' employs the covering criterion $\rho(\ell \mid \mathcal{L})$, and 'gap' uses the stability criterion measuring the gap between the best and second-best solution. To analyze the random sequence, we run ten independent repetitions for each image pair, compute the error and efficiency for each, and then average. The 'cov' and 'gap' methods resolve ties between equally scoring potential landmarks randomly. Therefore, we repeat those methods five times and average.

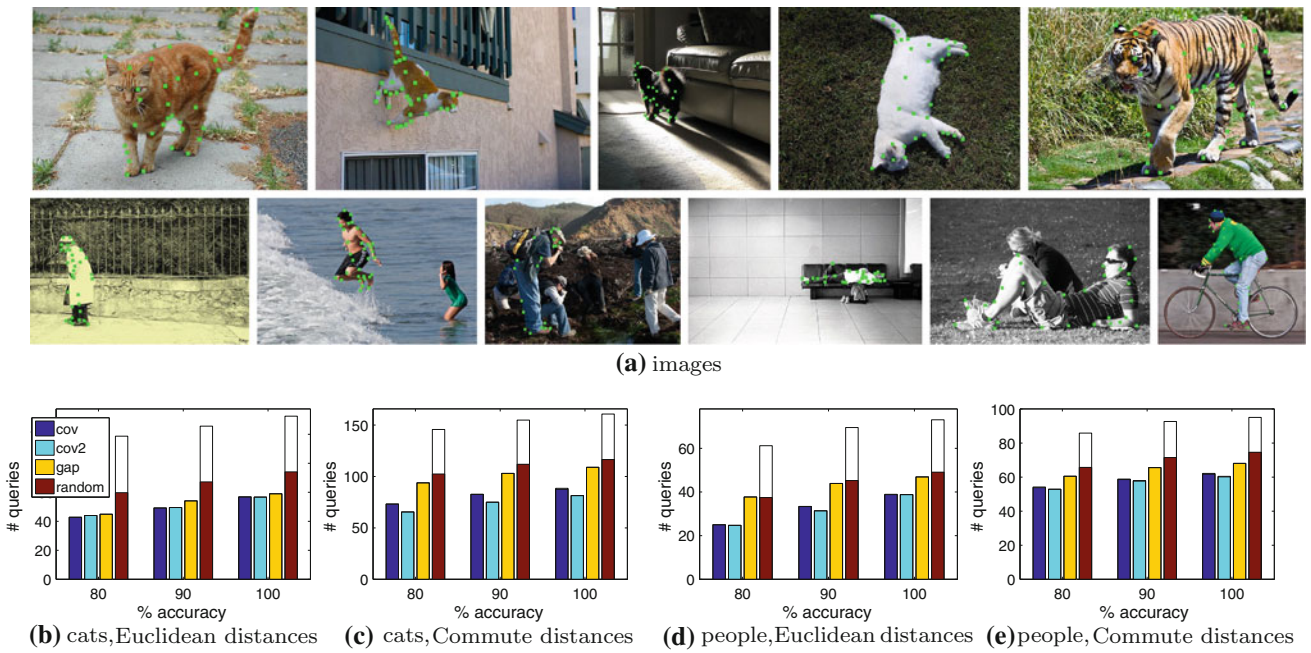
3.2.1 House/Hotel Sequence Dataset

We begin with the CMU house sequence data, with the 30 labeled points per image and shape context features used in (Caetano et al. 2009). The sequence consists of 111 frames of a rotated toy house. We compute C as Euclidean distances between the shape context features, and use Euclidean distances to landmarks. We match 20 pairs of images with a fixed distance of frames. The error is computed as the fraction of mismatched points, $\text{err}(f) = |\{x | f(x) \neq y(x)\}|/|\mathcal{X}|$, where $y(x)$ is the true correspondent of x. Figure 3a shows the average number of queries needed for a completely correct matching for gaps varying between 50 and 80 frames. The random sequence needs more than 50 % more queries than the other methods, and more than twice as many in the worst case. Furthermore, Fig. 3b indicates that the average



(a) efficiency houses **(b)** error houses **(c)** efficiency hotels **(d)** error hotels

Fig. 3 Efficiency (average number of queries needed for zero error) and average error for the CMU Houses and Hotels data sets. The *white bar* illustrates the worst case (over random repetitions), averaged over all image pairs



(a) images

(b) cats, Euclidean distances **(c)** cats, Commute distances **(d)** people, Euclidean distances **(e)** people, Commute distances

Fig. 4 Average number of queries needed to attain 80, 90 or 100 % accuracy. Averages are over all possible pairs of the depicted images. The results refer to the case that no initial landmark pairs are given, or that ten correct random landmark pairs are given

error achievable with a fixed budget of queries is lower for decision-theoretic selections.

The results on the related CMU hotel sequence are similar. Figure 3c indicates that the variance for the number of queries needed is very high when querying randomly. The 'gap' and 'cov' criteria lead to progress more consistently.

In general, pursuing correspondences for the rigid house sequence is a relatively easy task: For geometric transformations of rigid bodies, many sets of landmarks are equally good. Thus, the advantage of the decision-theoretic scores stems from preferably querying pairs that we are more confident about. Those are more likely to lead to a new landmark (correct pair). Given that about 70 % of the initial matches are correct, a random query is likely to yield an additional hit. For a more objective evaluation, we test the guided interaction on more challenging matching instances that exhibit more variation.

3.2.2 Non-Rigid Objects with Variations

When the matched objects are not identical, such as the forks in Fig. 5, then simple features such as SIFT features may be very uninformative: they lead to very high initial matching errors. In such cases, human interaction can be beneficial.

We obtained such harder instances by labeling photos of humans, cats and objects from Flickr and simulate query sequences as before. We again use Euclidean distances to landmarks as they appear to be more robust, and begin with a cost matrix C computed from SIFT features at 37–87 points. By themselves, these features provide very little guidance for correspondences and match about 5 % of the points correctly. As a result, random queries are not very likely to query a correct pair and thereby identify new landmarks. Therefore, such a poor initial solution serves as a difficult test for querying methods.

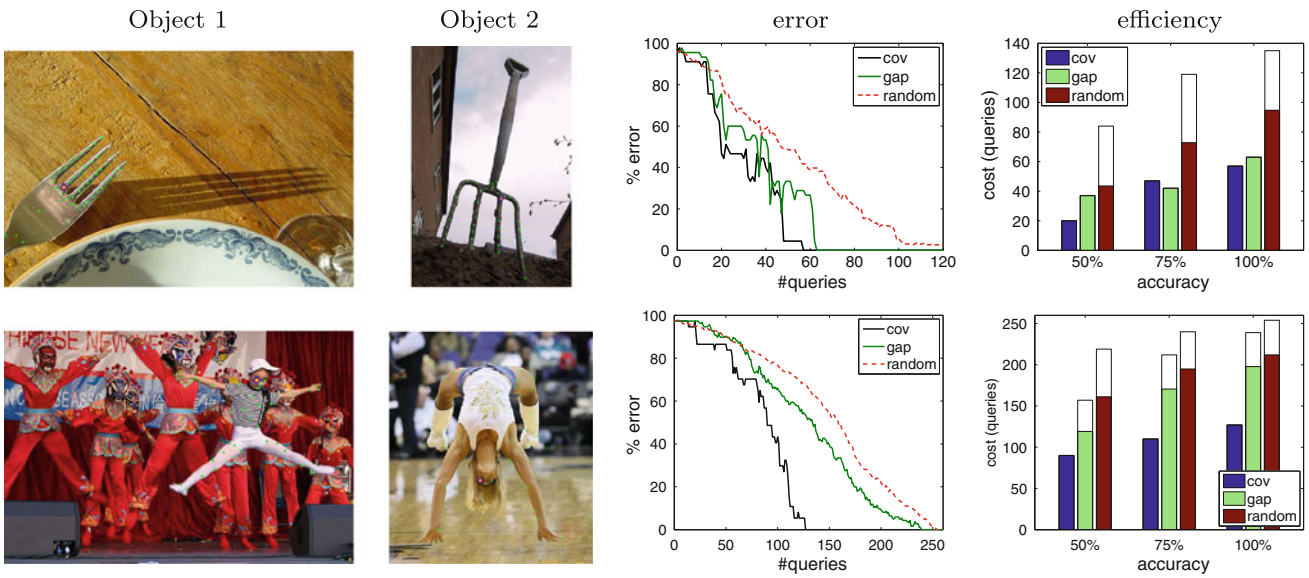


Fig. 5 Sample results for '*cov*' (coverage), '*gap*' (stability) and '*random*' (randomly selected queries) methods. *Colored bars* depict averages, *white bars* worst encountered cases (Color figure online)

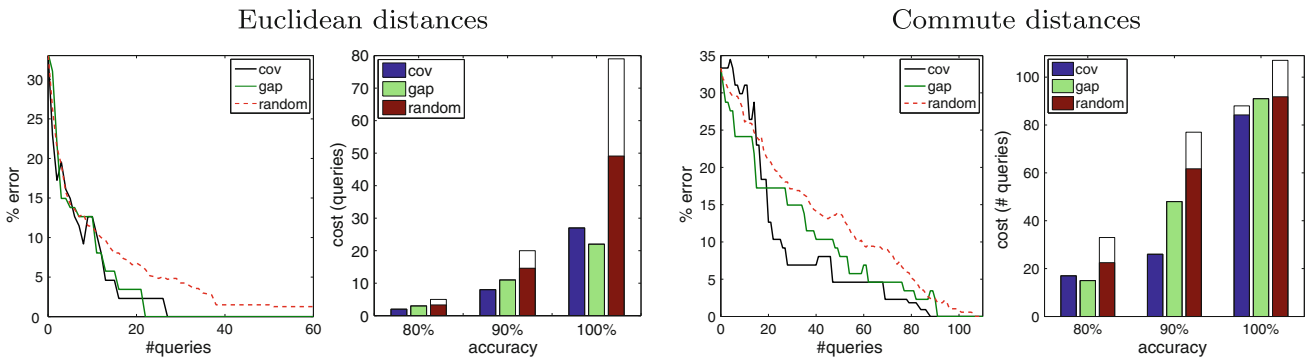


Fig. 6 Sample results for Euclidean and commute distances to landmarks. *Colored bars* depict averages, *white bars* worst encountered cases. The images are from the same sequence as those in Fig. 2 (Color figure online)

Figure 4 shows the average number of queries required to attain a certain target accuracy, across all possible pairs of five cats (10 pairs) and six humans (15 pairs). Here, we also compare to a variant of the covering criterion where each point can be covered by $k = 2$ landmarks, as described in Sect. 2.2.1. This criterion is in some cases more efficient than 'cov'. Since the two covering criteria still often behave similarly, we restrict ourselves to $k = 1$ in the other experiments. Figures 5 and 6 display sample results for single pairs of images of humans and forks. For those, we also show the error as a function of the number of queries. Both statistics, error and efficiency, indicate that (1) engaging humans with queries about matches helps to reduce error, and (2) selecting queries by expected value of information reduces the number of queries needed for a given accuracy, and this decreases human effort. As an example, for the forks in Fig. 5, the decision-theoretic query selection procedures require half as many queries as the random scheme.

In Fig. 6, where the SIFT features carry more information and the initial match is more accurate, the stability criterion becomes very useful and is the most efficient method. Note that the active *querying* methods (that do not know correct matches and may therefore fail to add a landmark in some steps) achieve full accuracy faster than a method that randomly adds *known* landmarks (one in *each* step), as shown in Fig. 2d. This suggests that both the selective gathering of positive and negative feedback and the location of landmarks matter.

3.2.3 Difficulties

A closer inspection of the results shows that matching becomes more difficult when the SIFT features are less informative and there are very few correct initial matches, and when there are symmetries in the objects. In those cases, more initial queries are needed before the error decreases rapidly.

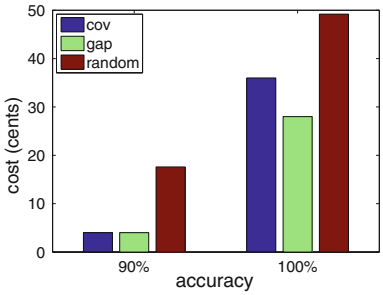


Fig. 7 Efficiency with human labelers for the images in Fig. 1b; the *y* axis shows the cost when each query costs 5 cents

While symmetries can be resolved with a few landmarks, landmark-based matching can become more difficult when there are strong deformations in the objects that place points together in one image but not the other. This is the case for the second pair in Fig. 5, where in particular the 'gap' method needs more queries. Once a few landmarks are established, the active querying methods still become effective.

3.2.4 Human Expertise

As a complement to the simulations, we explored feedback provided by human labelers via Mechanical Turk and recruited users for evaluating matches. Figure 1b shows the user interface for an example query. All possible points are shown in green, and the query points are marked by multicolored diamonds. We labeled a subset of points on the objects, and then added unlabeled points. The algorithm could query any pair, and the error was computed on the 24 hand-labeled points. Figure 7 shows the average number of queries needed to achieve a certain accuracy. When paying 0.05 dollars for a query, a query selection by stability can save 26.5 cents on average for a completely correct labeling, and 17 cents for a labeling with 90 % accuracy, for which random queries cost more than four times more than the selective ones.

4 Conclusion

We have explored methods for harnessing the perceptual abilities of people to help to refine partial correspondences between images that are identified via automated procedures. We employ a measure of the value of information to selectively direct human attention on correspondence problems. We proposed two different objectives for computation of value of information. In the first formulation, we seek to maximize coverage. The other formulation seeks to find stability via reducing the gap between the best and second-best solutions. We found that the covering criterion tends to be more robust when very few correct matches have been found. The stability criterion tends to become increasingly effective as more knowledge is gathered. Both criteria substantially outperform the random selection of query points and sometimes exceed the strategy where confirmed landmarks are added randomly at each step. The methods and results demonstrate the value of developing interactive approaches to challenging matching problems. More generally, the interactive approach we have taken to solving correspondence problems highlights the promise of endowing computational systems with the ability to engage and collaborate with people so as to ideally leverage the complementary skills of people and machines.

References

Caetano, T., McAuley, J., Cheng, L., Le, Q., & Smola, A. (2009). Learning graph matching. In *IEEE Trans. on Pattern Analysis and Machine Intelligence* (pp. 2349–2374).

Chegireddy, C. R., & Hamacher, H. W. (1987). Algorithms for finding *k*-best perfect matchings. *Discrete Applied Mathematics*, *18*, 155–165.

Chli, M., & Davison, A. J. (2008). Active matching. In *European Conference on Computer Vision (ECCV) 2008, Part I. Lecture Note in Computer Science* (Vol. 5302, pp. 72–85). Heidelberg: Springer.

Cho, Y., Lee, J., & Lee, K. M. (2010). Reweighted random walks for graph matching. In *European Conference on Computer Vision (ECCV)*.

Chvatal, V. (1979). A greedy heuristic for the set covering problem. *Math of Operations Research*, *4*(3), 233–235.

Cour, T., Srinivasan, P., & Shi, J. (2006). Balanced graph matching. In *Advances in Neural Information Processing Systems (NIPS)*.

Dasgupta, S. (2004). Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems (NIPS)*.

Debevec, P., Taylor, C., & Malik, J. (1996). Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Computer Graphics SIGGRAPH 1996 Proceedings*.

Duchenne, O., Bach, F., Kweon, I., & Ponce, J. (2009). A tensor-based algorithm for high-order graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Escolano, F., Hancock, E., & Lozano, M. (2011). Graph matching through entropic manifold alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*(2—-3), 133–168.

Goodrich, M., & Mitchell, J. (1999). Approximate geometric pattern matching under rigid motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(4), 371–379.

Handa, A., Chli, M., Strasdat, H., & Davison, A. J. (2010). Scalable active matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heckerman, D., Horvitz, E., & Nathwani, B. N. (1992). Toward normative expert systems: Part i the pathfinder project. *Methods of Information in Medicine*, *31*, 90–105.

Howard, R. (1967). Value of information lotteries. *IEEE Transaction on Systems, Science and, Cybernetics*, *SSC–3*(1), 54–60.

Joshi, A. J., & Porikli, F. N. P. (2009). Multi-class active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kamar, E. , & Horvitz, E. (2013). A Monte-Carlo approach to computing value of information: Procedure and experiments. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kamar, E., Hacker, S., & Horvitz, E. (2012). Combining human and machine intelligence in large-scale crowdsourcing. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kapoor, A., Horvitz, E., & Basu, S. (2007). Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence*.

Kapoor, A., Grauman, K., Urtasun, R., & Darrell, T. (2009). Gaussian processes for object categorization. *International Journal of Computer Vision*, *88*(2), 169–188.

Kowdle, A., Chang, Y., Gallagher, A., & Chen, T. (2011). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, *9*, 235–284.

Lawrence, N., Seeger, M., Herbrich, R. (2002). Fast sparse Gaussian process method: Informative vector machines. In *Advances in Neural Information Processing Systems (NIPS)* (Vol. 15). Cambridge: MIT Press

Lee, J., Cho, M., & Lee, K. (2011). Hyper-graph matching via reweighted random walks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lordeanu, M., & Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *International Conference on Computer Vision (ICCV)*.

Lovász, L. (1993). Random walks on graphs: a survey. *Combinatorics: Paul Erdös is Eighty*, *2*, 1–46.

MacKay, D. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*(4), 589.

Maji, S., Shakhnarovich, G. (2012). Part annotations via pairwise correspondence. In *4th Workshop on Human Computation, AAAI*.

Mateus, D., Horaud, R., Knossow, D., Cuzzolin, F., & Boyer, E. (2008). Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

McAuley, J., & Caetano, T. (2012). Fast matching of large point sets under occlusion. *Pattern recognition*, *45*, 563–569.

McAuley, J., Caetano, T., & Barbosa, M. S. (2008). Graph rigidity, cyclic belief propagation and point pattern matching. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *30*(11), 2047–2054.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, *5*(1), 32–38.

Sharma, A., Horaud, R. P., Cech, J., & Boyer, E. (2011). Topologically-robust 3d shape matching based on diffusion geometry and seed growing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Starck, J., & Hilton, A. (2007). Correspondence labelling for wide-timeframe free-form surface matching. In *International Conference on Computer Vision (ICCV)*.

Tong, S., & Koller, D. (2000). Support vector machine active learning with applications to text classification. In *International Conference on Machine Learning (ICML)*.

Torresani, L., & Kolmogorov, V. (2008). *Rother, C*. Feature correspondence via graph matching: Models and global optimization. In *European Conference on Computer Vision (ECCV)*.

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *10*(5), 695–703.

Vijayanarasimhan, S. (2011). *Active visual category learning*. PhD Thesis, UT Austin.

Vijayanarasimhan, S., & Kapoor, A. (2010). Visual recognition and detection under bounded computational resources. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *CHI: SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM.

Zass, R., & Shashua, A. (2008). Probabilistic graph and hypergraph matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.