# Object Matching with a Locally Affine-Invariant Constraint

Hongsheng Li† Edward Kim† Xiaolei Huang† Lei He§
† Department of Computer Science and Engineering, Lehigh University, PA 18015
§ National Library of Medicine, NIH, Bethesda, MD 20894
{h.li, edk208, xih206}@lehigh.edu, lei.he@nih.gov

## Abstract

*In this paper, we present a new object matching algorithm based on linear programming and a novel locally affine-invariant geometric constraint. Previous works have shown possible ways to solve the feature and object matching problem by linear programming techniques [9], [10]. To model and solve the matching problem in a linear formulation, all geometric constraints should be able to be exactly or approximately reformulated into a linear form. This is a major difficulty for this kind of matching algorithms. We propose a novel locally affine-invariant constraint which can be exactly linearized and requires a lot fewer auxiliary variables than the previous work [10] does. The key idea behind it is that each point can be exactly represented by an affine combination of its neighboring points, whose weights can be solved easily by least squares. The resulting overall objective function can then be solved efficiently by linear programming techniques. Our experimental results on both rigid and non-rigid object matching show the advantages of the proposed algorithm.*

## 1. Introduction

The problem of object matching in 2D images can be defined as matching a *model* graph representing an object to an instance of that object in a given *scene* image. It has extensive uses in object detection and tracking [9], shape matching [10], image classification [15], and image retrieval [17]. The nodes and edges of a model graph represent distinctive feature points and the neighborhood relationships between them, respectively (Fig. 1(a)). After feature points are detected in the scene image (Fig. 1(b)), point correspondences between model and scene feature points are established. Matched scene feature points should maintain consistency with the model graph in both local appearance and relative spatial relationships.

The object matching problem has been extensively studied as a graph matching problem [6], [8]. Compared with the RANSAC algorithms [4], the graph matching matching approaches usually can handle more complex deformations. Leordeanu and Hebert [11] proposed a spectral method working on a matrix where the diagonal elements

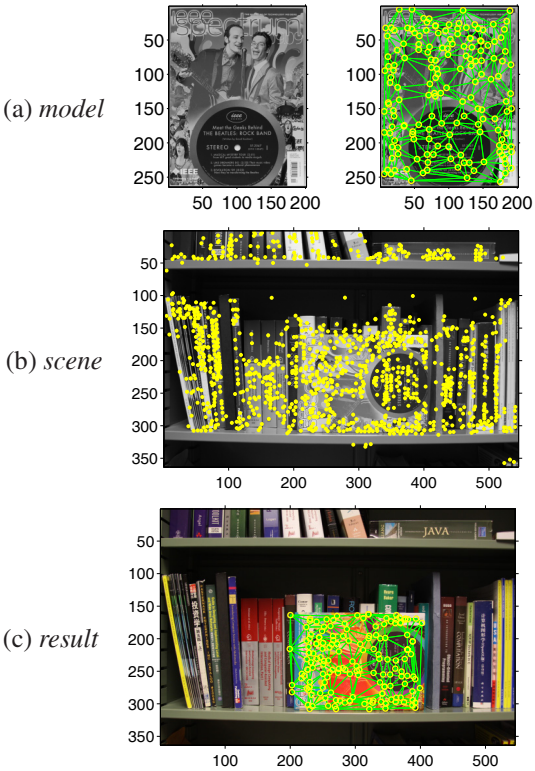


Figure 1. Matching a model graph to a scene image by our method. (a) A model graph representing a magazine, (b) a scene image and detected feature points in it, and (c) the final matching result by our proposed method. SIFT features [12] are used in this example.

represent one-to-one assignment costs, and other elements represent pairwise agreements between potential correspondences. The correspondences are then obtained by finding the principal eigenvector of this matrix. This method uses the distance between two points as the geometric constraint, which is only rotationally invariant. The same rotational invariant is also used in a point matching method [18]. Cour *et al*. [5] proposed a spectral relaxation method for the graph matching problem that incorporates one-to-one or one-to-many mapping constraints, and presented proper bistochastic normalization of the graph matching compatibility matrix to improve the overall matching performance. Duchenne *et al*. [7] used high-order (mostly 3) constraints

instead of unary or pairwise ones between nodes, which result in a tensor representing affinity between feature tuples. The resulting energy function can then be optimized using the power iteration method. Note that the definition of the graph in graph matching algorithms is different from that of our model graph; in our model graphs, edges are unweighted and are only used to specify the neighborhood for each node.

The matching problem has also been modeled as mathematical programming problems. Chui and Rangarajan [3] interpreted it as a mixed variable (binary and continuous) optimization problem. The correspondence problem is viewed as a linear assignment solved by softassign and deterministic annealing. Berg *et al*. [2] modeled the matching problem as a quadratic integer programming problem. It uses pairwise relationships between feature points and penalizes both rotation and scaling differences. Recently, linear programming has been used in object matching. Jiang *et al*. [9] proposed a linear solution to the feature matching problem. The main difficulty of this framework is to find geometric constraints which can be exactly or approximately linearized. In [9], the vectors defined by pairwise points are used as the geometric constraint for its objective function. It can only tolerate small local deformations and is not invariant to global transformations, such as similarity or affine transformations. To solve this problem, Jiang and Yu [10] explicitly modeled scaling and rotation, and approximated the resulting formulation by a convex program. The resulting solution is invariant to global rotation and scaling. Its extensive experimental results demonstrated the effectiveness and robustness of the pairwise geometric constraint in various object matching scenarios.

Along this line, we propose a new locally affine-invariant geometric constraint for the linear programming matching framework. For each point in the model graph, we represent it as an affine combination of its neighboring points. Such affine combinations can be easily and efficiently solved by least squares. As demonstrated in the next section, these representations are invariant to affine transformations. Moreover, since the coefficients of each affine combination are only calculated by using its corresponding point's neighboring points, this constraint is a local one.

Compared with the global rotation and scaling invariant constraint proposed in [10], our new geometric constraint has three major advantages over it: (1) our proposed geometric constraint is locally affine-invariant. Therefore, it can handle more complex and natural transformations of an object. For instance, objects undergoing articulated deformations in Section 3.3. (2) Unlike the explicit approximate linearization of the previous similarity invariant constraint, the exact linearization of our new constraint requires much fewer auxiliary variables. Therefore, it is asymptotically faster and is also easier to implement. And (3) for each point in the model graph, all of its neighboring points are used to calculate the affine combination coefficients. It is a higher order geometric constraint, which is more distinctive and can better exclude ambiguous matchings [7].

## 2. Methodology

### 2.1. Problem Formulation

Given a model graph $(V, E)$, where $V$ and $E$ represent the sets of nodes and edges of the model graph, the matching function $m(\cdot)$ matches every model feature point $\mathbf{p}_i = [x_i, y_i]^T \in V$ to a feature point $m(\mathbf{p}_i)$ in the scene image. $\mathcal{N}_{\mathbf{p}_i}$ stands for the set of ordered points in the neighborhood of $\mathbf{p}_i$. The order of points in each neighborhood is randomly set. The goal is to find the matching function $m(\cdot)$ that minimizes the overall objective function consisting of both feature and geometric matching costs:

$$\hat{m} = \arg\min_{m} \sum_{i=1}^{|V|} \{c(\mathbf{p}_i, m(\mathbf{p}_i)) + \lambda \cdot g(\mathbf{p}_i, \mathcal{N}_{\mathbf{p}_i}; m(\mathbf{p}_i), \mathcal{N}_{m(\mathbf{p}_i)})\}, \quad (1)$$

where $c(\mathbf{a}, \mathbf{b})$ is the feature matching cost between the feature points $\mathbf{a}$ and $\mathbf{b}$, $g(\cdot)$ is the geometric cost that measures the geometric dissimilarity between two sets of ordered points $\{\mathbf{p}_i, \mathcal{N}_{\mathbf{p}_i}\}$ and $\{m(\mathbf{p}_i), \mathcal{N}_{m(\mathbf{p}_i)}\}$, and $\lambda$ controls the relative weight between the feature and geometric costs.

Unlike the formulation proposed in [9], [10], where only pairwise geometric relationships are considered, our new formulation takes into consideration of higher order (at least order 3) geometric constraints, which are more distinctive and therefore can better exclude ambiguous matchings [7].

The neighborhood $\mathcal{N}_{\mathbf{p}_i}$ of $\mathbf{p}_i$ is specified by the edges connected to $\mathbf{p}_i$. It remains an open issue how to create model graphs better representing different objects for the object matching problem. In this paper, we focus on matching a given model graph to a scene image and used Delaunay Triangulation to create most model graphs.

Similar to [10], the choice of features is not restricted to similarity or affine invariant ones, *e.g*., SIFT [12]. For general non-transformation-invariant features, the matching cost between two feature points $\mathbf{a}$ and $\mathbf{b}$ can be defined by the minimal distance across all possible similarity or affine transformations $T$ with parameters $\Theta$,

$$c(\mathbf{a}, \mathbf{b}) = \min_{\Theta} distance(feature(\mathbf{a}), feature(T(\mathbf{b}; \Theta))). \quad (2)$$

### 2.2. A Locally Affine-Invariant Constraint

In this subsection, we present a novel locally affine-invariant geometric constraint for the geometric cost function $g(\cdot)$ in (1).

Our geometric constraint has two requirements on the structure of the model graph: 1) every node must have at least three neighbors, *i.e*., the degree of each node is at

least 3; and 2) every node's neighboring points must not be collinear, *i.e.*, they do not lie on a single straight line. Our goal is to create a way to characterize the geometric properties of the neighborhood of each node. To do so, we assume each $\mathbf{p}_i$ can be exactly represented by an affine combination of its neighboring points, *i.e.*,

$$\mathbf{p}_i = \sum_{\mathbf{p}_j \in \mathcal{N}_{\mathbf{p}_i}} W_{ij}\mathbf{p}_j, \tag{3}$$

where $W$ is a $|V| \times |V|$ weight matrix recording the affine combination coefficients for all points, and $W_i$ is the $i$th row of $W$ recording the affine combination coefficients for $\mathbf{p}_i$. Intuitively, $W_i$ reveals the local geometric layout around $\mathbf{p}_i$. There are two constraints on the weight matrix $W$: $W_{ij} = 0$ if $\mathbf{p}_j \notin \mathcal{N}_{\mathbf{p}_i}$ and each row must sum to one (equivalently, each point is represented by an affine combination of its neighbors). The first constraint reflects that this matrix only describes the local geometric properties of each point. The second makes the representation invariant to global translation.

It is easy to prove that a point can always be exactly represented by the affine combination of its neighbors if the above mentioned two requirements are satisfied. Assume $\mathbf{p}_i$ has only three neighbors $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$. The affine combination coefficients $W_i$ for $\mathbf{p}_i$ can be obtained by first solving the following linear equations:

$$\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \\ 1 & 1 & 1 \end{bmatrix} \tilde{W}_i^T = Q\tilde{W}_i^T = \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix}. \tag{4}$$

Because $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ are not collinear, the matrix $Q$ has full rank. $\tilde{W}_i^T = Q^{-1}[\mathbf{p}_i^T \ 1]^T$ is an exact solution of the affine combination coefficients for $\mathbf{p}_i$. We can then fill $W_i$ using $\tilde{W}_i$: $W_{ij} = \tilde{W}_{il}$ if $\mathbf{p}_j$ is the $l$th neighbor of $\mathbf{p}_i$, and $W_{ij} = 0$ if $\mathbf{p}_j \notin \mathcal{N}_{\mathbf{p}_i}$. If $\mathbf{p}_i$ has more than 3 neighbors, we can still obtain an exact affine combination by just using the first three neighbors. In practice, we use least squares to minimize the error of each point's affine combination. Since least squares guarantees to obtain a solution with minimal error under $L^2$ norm, and we just showed at least one solution with zero error exists, the solution by least squares is also an exact representation of that point. Although there might be an infinite number of affine representations for a point, any one of them can be used in our framework. We choose least squares because one of its desired properties is that it usually assigns nonzero weights to all neighbors, which means the local geometric properties of each point are described by all of its neighbors.

We calculate the reconstruction weights $\tilde{W}_i$ for each point $\mathbf{p}_i$ separately and transform them into the matrix form $W$ by the above mentioned way. The representation error for $\forall \mathbf{p}_i \in V$ is always zero no matter what type of norm is used, *i.e.*,

$$\left\| \mathbf{p}_i - \sum_j W_{ij}\mathbf{p}_j \right\|_{0,1,2,\cdots,F} = 0 \tag{5}$$

for $i = 1, \cdots, |V|$. For this particular application, we choose the $L^1$ norm for the representation error, since it can be exactly linearized (Section 2.5). Obviously, the error function (5) is affine invariant:

$$\begin{aligned} 0 &= \left\| \mathbf{p}_i - \sum_j W_{ij}\mathbf{p}_j \right\|_1 \\ &= \left\| A\mathbf{p}_i - \sum_j W_{ij} A\mathbf{p}_j \right\|_1 \\ &= \left\| (\mathbf{p}_i + t) - \sum_j W_{ij}(\mathbf{p}_j + t) \right\|_1, \end{aligned}$$

where $A$ and $t$ denote an arbitrary $2\times2$ affine transformation matrix and an arbitrary $2\times1$ translation vector, respectively.

Without any feature information, we seek a matching function $m(\cdot)$ which best preserves the geometric properties of the model graph specified by its weight matrix $W$:

$$\arg\min_m \sum_{i=1}^{|V|} \left\| m(\mathbf{p}_i) - \sum_j W_{ij} m(\mathbf{p}_j) \right\|_1. \tag{6}$$

On one hand, there are degenerate cases: matching all model points to one scene point also leads to a zero geometric cost because $\sum_j W_{ij} = 1$. Fortunately, in the object matching tasks, features have distinctive power. Those degenerate cases usually result in very large feature costs and thus are not likely to be the optima of the objection function (1). Even when the features used are not distinctive enough, we can further add constraints into our linear programming model to explicitly exclude those degenerate cases (Section 2.4). On the other hand, some parts of an object may be folded. If the features are invariant to such local deformations, matching several model points to one scene point also minimizes the error function (5) and should be considered as a correct matching (Section 3.4).

Compared with the approximately global rotation and scaling invariant constraint in [10], our new geometric constraint (1) is locally affine-invariant, (2) does not explicitly model any global transformation, and (3) can be exactly linearized with a lot fewer auxiliary variables.

### 2.3. Relation to Locally Linear Embedding [16]

Our affine invariant is inspired by the Locally Linear Embedding (LLE) and has a similar formulation, but our invariant is different from LLE in essence. Our invariant assumes each point can be represented by an "affine" combination of its neighboring points. Its reconstruction error by the affine combination is affine-invariant. In contrast, the LLE assumes a "convex" combination, and the resulting $W$ coefficients are similarity-invariant. However, LLE's reconstruction error for each point is not transformation-invariant, thus

its "convex" combination cannot be used in this matching framework.

### 2.4. Matrix Formulation

Following the representation in [10], we also present (1) in a succinct matrix form using an assignment matrix $X$ to help readers better understand the objective function and constraints.

Let $k_n$ denote a column vector of $n$ $k$'s where $k$ is a constant, $^T$ matrix transpose, $tr$ the trace of a matrix, $I_n$ an $n \times n$ identity matrix, and $|\cdot|$ the summation of the absolute values of all the elements in a matrix. Let $n_m$ and $n_s$ be the numbers of model and scene feature points, respectively. After we calculate the weight matrix $W$ of the model graph, the solution to the matching problem can be solved by

$$\begin{aligned} \min f(X) &= tr(C^T X) + \lambda |(I_{n_m} - W)XS| \qquad (7) \\ \text{subject to} \quad & X 1_{n_s} = 1_{n_m}, \\ & X \in \{0,1\}^{n_m \times n_s}, \\ & X^T 1_{n_m} \leq u_{n_s} \text{ (optional)}, \end{aligned}$$

The variable $X$ is a $n_m \times n_s$ binary assignment matrix that represents the matching function $m(\cdot)$. Each row of $X$ contains exactly one 1, meaning every point in the model graph must be matched to exactly one point in the scene image. $X(i,j) = 1$ denotes matching the $i$th model feature point to the $j$th scene feature point. If one model point's corresponding scene point is occluded or not detected, minimization of (7) would prefer matching it to another scene point which well approximates that model point's local geometric properties.

There are three known matrices in (7):

- $S$ is the $n_s \times 2$ coordinate matrix. It records the coordinates of $n_s$ 2D scene points.
- $C$ is the $n_m \times n_s$ feature matching cost matrix. $C(i,j)$ is the feature matching cost between the $i$th model point and the $j$th scene point.
- $W$ is the $n_m \times n_m$ coefficient matrix for point representations. The $i$th row of $W$ records the affine combination coefficients for representing the $i$th model point by its neighboring points.

There are three constraints:

- $X 1_{n_s} = 1_{n_m}$ denotes all model points should be matched into the scene.
- $X \in \{0,1\}^{n_m \times n_s}$ denotes the matching between a model and a scene feature point is either "yes" or "no".
- $X^T 1_{n_m} \leq u_{n_s}$ allows matching at most $u (u < n_m)$ model points to one scene point and thus avoids the degenerate cases we mentioned in Section 2.2. However, in practice, this constraint is usually not necessary since matching all model points to one scene point usually leads to a very large feature matching cost. It should be used only when features are not distinctive enough because it adds $n_s$ more constraints to the optimization model.

The objective function $f(X)$ consists of a feature and a geometric cost term. The feature cost term $tr(C^T X)$ is the matrix form of the first term in (1). The geometric cost term $|(I_{n_m} - W)XS|$ is the same as the cost definition in (6).

### 2.5. Exact Linearization and Relaxation

The problem (7) has a nonlinear objective function with integer constraints. It is NP-complete and cannot be efficiently solved. However, because $\lambda > 0$, the second term of (7) can be exactly linearized in the following way:

$$\begin{aligned} \min \textstyle\sum_{i=1}^N |x_i| \Leftrightarrow \quad \min \quad & \textstyle\sum_{i=1}^N x_i^+ \\ \text{subject to} \quad & x_i \leq x_i^+, x_i \geq -x_i^+ \\ & x_i^+ \geq 0, \\ & \text{for all } i = 1, \cdots, N, \end{aligned}$$

where $x_i^+$ is the $i$th auxiliary variable representing the upper bound of $|x_i|$.

We further relax the binary constraints, $X \in \{0,1\}^{n_m \times n_s}$, to continuous domains $[0,1]^{n_m \times n_s}$ to convert the original problem (7) into a linear programming (LP) form:

$$\begin{aligned} \min f(X) &= tr(C^T X) + \lambda 1_{n_m}^T X^+ 1_2 \qquad (8) \\ \text{subject to} \quad & X 1_{n_s} = 1_{n_m}, \; X \geq 0, \\ & (I_{n_m} - W)XS \leq X^+, \\ & (I_{n_m} - W)XS \geq -X^+, \\ & X^+ \geq 0, \\ & X^T 1_{n_m} \leq u_{n_s} \text{ (optional)}, \end{aligned}$$

where $X^+$ is a $n_m \times 2$ auxiliary variable matrix.

### 2.6. Numerical Scheme

Without any simplification trick, the number of variables in our LP model (8) is proportional to $n_m \times n_s$. In contrast, the number of variables of the LP model in [10] is proportional to $n_m \times n_s \times$ *the number of scaling discretizations*. Moreover, at the first step of the LP method in [10], it needs to solve 4 such LP problems because it models rotation as 4 different linear constraints. Therefore, our algorithm is asymptotically faster than that in [10].

We utilize the lower convex hull with successive trust region shrinkage method proposed in [9] to solve our LP problem (8). This method makes the complexity of the LP nearly independent of the number of scene points. The difference is that we only use its *consistent rounding* process in the last 2-3 iterations. Before that, we directly use LP's results as *anchors* for the shrinkage of trust regions.

LP with tens of thousands of variables and thousands of constraints can be solved within seconds on a standard PC using state-of-the-art solvers, such as CPLEX and

Gurobi. In our experiments, we use MATLAB with a non-commercial solver, *lp_solve* [1], which employs the simplex methods. Typically, for matching 100 model points and thousands of scene points, each LP iteration takes less than 1 second on an Intel E6850 3.0GHz CPU, and the trust region shrinkage runs for 4-8 iterations. Note that the running time can be further shortened because the simplex methods *lp_solve* uses are less efficient than the interior point methods when solving medium to large size LPs, and MATLAB is less efficient than C/C++ on iteration operations.

## 3. Experiments

In our experiments, we only use gray-scale images to create more challenging cases since features on gray-scale images are less distinctive than those on color images. SIFT [12] or MSER [13] detectors are used as feature point detectors. The SIFT descriptor [12] is used as the feature for every detected point, and feature dissimilarity is calculated by the $L^2$ distance between two feature vectors. For all experiments, we set $\lambda = 1$.

### 3.1. Objects Undergoing Similarity Transformations

We first modeled an *IEEE Spectrum* magazine (Fig. 1(a)) and matched it to its transformed instances in scene images with cluttered background (Fig. 2). For the model graph, nodes were selected as SIFT points with scales between 2 and 10, and edges were obtained by computing the Delaunay Triangulation of the nodes. Although there were many outlier feature points in the scene images, and some model points' corresponding scene points were not detected, our method still was able to match the magazine to the scene images efficiently and robustly (Fig. 2(1)). We also did experiments using feature information only to show the necessity of the geometric constraint (Fig. 2(2)).

### 3.2. Objects Undergoing Affine Transformations

Our geometric invariant is able to handle objects' natural affine transformations. One such example is the transformation caused by viewpoint change when viewing objects with planar surfaces. We used two sets of viewpoint change images (Fig. 3 and Fig. 4) from [14] to evaluate the performance of our matching method under approximately affine transformations. Each of the two sets, *graf* and *wall* set, contains 6 images of a planar wall. We created a model graph using the first image in each set and matched it to other images with different viewpoints in the same set.

For the nodes in the model graph, we used feature points detected by MSER in the central area of the first image. The two parameters of MSER, minimal region size and minimal margin, were set to 30 and 15, respectively. We further excluded duplicate nodes and nodes with too small scales. Edges of the graph were then obtained by Delaunay Triangulation (Fig. 3). For detecting feature points in the scene images, we used the default parameter settings of MSER.

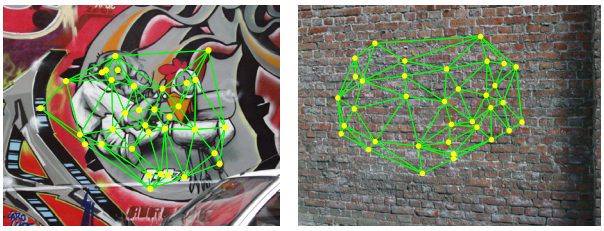

Figure 3. Model graphs created based on feature points detected by MSER in the first image of the *graf* set (left) and in the first image of the *wall* set (right) [14]. See Section 3.2 for details about how the graphs are built.

Note that the MSER does not only detect feature locations but also calculate three other shape parameters. The calculation of the SIFT descriptor relies on all those 5 parameters.

In the *graf* set, images of a painting on the wall are taken from different viewpoints. In the *wall* set, images of a brick wall are taken. The brick wall has a relatively uniform texture appearance which makes the MSER detector less accurate and the SIFT features less distinctive. Matching in the *wall* set is therefore a much more challenging problem. Our method matched the model graph of *graf* to all *graf* images, and matched the model graph of *wall* to the first 4 *wall* images. Some correct correspondences in the 5th *wall* image are recovered, but the overall matching is not satisfactory. In contrast, the LP based method in [10] only satisfactorily matched the 2nd image in each set where viewpoint change angles are small (Fig. 4.(2a)) but failed on all other images with larger viewpoint change angles (Fig. 4.(2b-e)) because it only models objects' rotation and scaling.

### 3.3. Objects Undergoing Articulated Deformations

Our local geometric constraint only tries to maintain each point's local geometric properties and thus can match objects undergoing articulated deformations. In Fig. 5, we show an experiment of matching a toy worm with distinctive features (Fig. 5(a)) to its bended instances in scene images. To obtain the model graph, we manually removed some edges after calculating the Delaunay Triangulation of feature points to avoid building strong connections between different moving parts. Results in Fig. 5(c) and 5(d) demonstrate the advantages of our local geometric constraint over the global constraint proposed in [10]. The LP method in [10] would fail on these cases because it models the global rotation for the entire model graph and can only tolerate small rotation disagreement between different parts.

### 3.4. Real Videos

We did experiments on real videos, some taken by ourselves (the *Computer* and *Spectrum* magazine videos) and one obtained from the YouTube (the *honeybee* video). Similar to the matching experiment in Section 3.1, we used SIFT points in the selected object regions as the nodes of model graphs and built their neighboring connections through Delaunay triangulation (Fig. 6(a)). We applied our method to

Figure 2. Matching the *Spectrum* magazine (Fig. 1(a)) in scene images with cluttered background. (From left to right) the magazine is not rotated, rotated by 90 degrees, rotated by 180 degrees, and occluded by a hand. Unmatched scene feature points are marked in light blue.

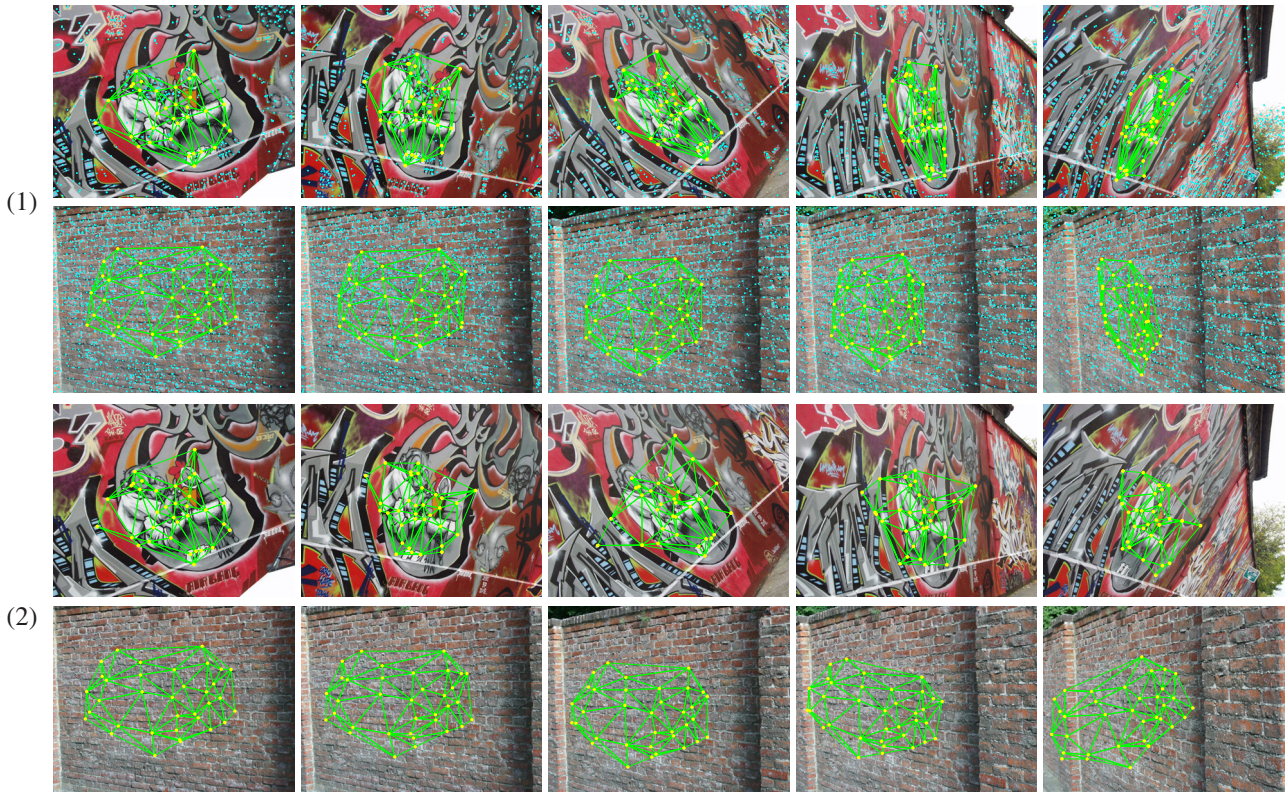


Figure 4. Matching the model graphs in Fig. 3 to the *graf* and the *wall* sets using (1) the proposed method and (2) the LP method in [10]. Unmatched scene feature points are marked in light blue.

every single frame of those videos and did not utilize any temporal information. The algorithm does not need initialization and can track an object undergoing large and complex deformations. We compared our method with the LP based method in [10] using those videos.

The *Computer* magazine video consists of mostly similarity transformations, with some occlusions and local deformations (Fig. 7(1))[1,2]. For this video, our method has similar matching accuracy as the LP method in [10] (Fig. 6(1)) but has an asymptotically faster running speed.

The *Spectrum* magazine video consists of mostly affine transformations and non-rigid deformations[3] (Fig. 7(2)). On this video, our method outperforms the LP method in [10] because our geometric constraint is affine-invariant, and its local property enables it to handle larger non-rigid deformations. One such example is shown in Fig. 6(2) where the magazine is wrapped inwards. The global geometric constraint of [10] prefers scaling the model graph globally. Our local constraint tries to maintain each point's local geometric properties so it can better handle such non-rigid deformations. Note that the binaries of [10] we obtained from its authors have a fixed parameter setting. We speculate that its performance may improve if its parameter setting is changed to give the geometric cost a smaller weight.

The *honeybee* video looks simple, but it has fewer distinctive feature points than the previous 2 videos which makes matching the the honeybee[4] a more challenging task

[1]Results: www.youtube.com/watch?v=QZpYP0DTENA.
[2]Feature points: www.youtube.com/watch?v=WarQ31l8HSk.
[3]Results: www.youtube.com/watch?v=yvd4Ma6YWVY.
[4]Results: www.youtube.com/watch?v=OovjtkPAHjk.


Figure 5. Matching a toy worm undergoing articulated deformations. (a) The original image of the toy worm, (b) the model graph, (c-d) two examples of matching the toy worm model to its instances that have undergone articulated deformations.

(Fig. 7(3)). Our method outperforms the LP method in [10] when a large portion of corresponding feature points are missing in the scene images. Fig. 6(3) shows such an example where only a fraction of the feature points on the honeybee's tail part are correctly detected. The global geometric constraint of the LP method in [10] favors all matched scene points maintaining a similar geometric structure as the model graph. It matches part of the tail correctly but wrongly matches other parts to the background (Fig. 6(3c)). In contrast, our geometric constraint only tries to keep local geometric structures and thus can match disappeared feature points to shrunken neighborhoods. The result by our method is shown in Fig. 6(3d) where the tail part is correctly matched.

## 4. Conclusions and Discussions

In this paper, we presented a novel locally affine-invariant constraint for the LP-based object matching framework. This constraint depends on exactly representing each point by an affine combination of its neighboring points. Such representations were proved to be exact and can be easily solved by least squares. Our proposed constraint showed several advantages over those in previous works. Experiments on various matching cases for rigid and non-rigid objects demonstrated the effectiveness and efficiency of our proposed algorithm.

However, how to create the model graph for a specific object and how to set a proper weight between feature and geometric costs remain important but open issues in our algorithm. Handling occlusions of the model graph remains a challenging problem for the graph matching [11], [7] and the linear programming based matching [9], [10] frameworks. Although in [9], an occlusion handling method is proposed to match all occluded model points to a "null" point, it has two limitations which make it difficult to use in practice: (1) the "null" point's matching cost needs to be smaller than the cost of a wrong match and larger than that of a correct match; and (2) the occluded model points cannot be completely removed from the geometric constraint term of the objective function. We would like to explore possible ways to solve these problems in the future.

**Acknowledgments.** We would like to thank Dr. Hao Jiang (Boston College) for providing us the code of [10] for comparison and for giving us precious suggestions on efficient trust-region-shrinkage algorithms.

## References

[1] lp_solve: sourceforge.net/projects/lpsolve.

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. *Proc. CVPR*, pages 26–33, 2005.

[3] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89:114–141, 2003.

[4] O. Chum and J. Matas. Optimal randomized ransac. *IEEE Trans. PAMI*, 30(8):1472–1482, 2008.

[5] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. *Proc. NIPS*, pages 313–320, 2006.

[6] A. D. J. Cross and E. R. Hancock. Graph matching with a dual-step em algorithm. *IEEE Trans. PAMI*, 20(11):1236–1253, 1998.

[7] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *Proc. CVPR*, 2009.

[8] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. PAMI*, 18:377–388, 1996.

[9] H. Jiang, M. S. Drew, and Z. Li. Matching by linear programming and successive convexification. *IEEE Trans. PAMI*, 29:959–975, 2007.

[10] H. Jiang and S. X. Yu. Linear solution to scale and rotation invariant object matching. *Proc. CVPR*, 2009.

[11] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *Proc. ICCV*, pages 1482–1489, 2005.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Comp. Vis.*, 60:91–110, 2004.

[13] J. Matas, O. Chum, M. Urba, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Proc. BMVC*, pages 384–396, 2002.

[14] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int'l J. Comp. Vis.*, 60:63–86, 2004.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *Proc. CVPR*, pages 2161–2168, 2006.

[16] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[17] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. PAMI*, 19:530–535, 1997.

[18] Y. Zheng and D. Doermann. Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE Trans. PAMI*, 28:643–649, 2006.


Figure 6. Example matching results by our method and the LP method in [10] on videos. (a) Model graphs, (b) scene frames, (c) matching results by the LP method in [10], and (d) matching results by the proposed method.


Figure 7. Sample matching results by our method from (1) the *Computer* magazine sequence, (2) the *Spectrum* magazine sequence, and (3) the *honeybee* sequence. Unmatched scene feature points are marked in blue.