IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, Florida, June 20-25, 2009

Linear Solution to Scale and Rotation Invariant Object Matching

Hao Jiang and Stella X. Yu
Computer Science Department
Boston College, Chestnut Hill, MA 02467, USA
{hjiang,syu}@cs.bc.edu

Abstract

Images of an object undergoing ego- or camera- motion often appear to be scaled, rotated, and deformed versions of each other. To detect and match such distorted patterns to a single sample view of the object requires solving a hard computational problem that has eluded most object matching methods. We propose a linear formulation that simultaneously finds feature point correspondences and global geometrical transformations in a constrained solution space. Further reducing the search space based on the lower convex hull property of the formulation, our method scales well with the number of candidate features. Our results on a variety of images and videos demonstrate that our method is accurate, efficient, and robust over local deformation, occlusion, clutter, and large geometrical transformations.

1. Introduction

Images of a bee flapping its wings and moving around a daisy appear to be related by global translation, rotation, scaling, and local deformation. Our goal is to detect and match the 2D pattern of the object with a template built from a single sample image (Fig. 1).

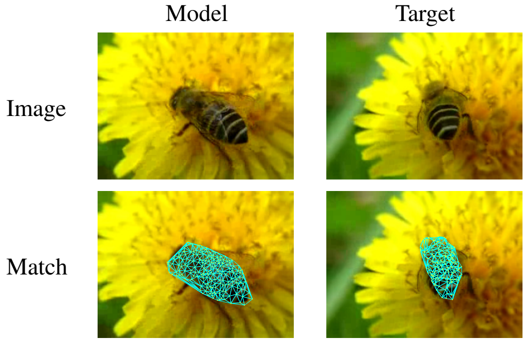


Figure 1. Our goal is to find the correspondence between the model image of a deformable object and the target image of the same object with unknown scaling, rotation, and local deformation.

The basic idea in pattern matching is that distinctive feature points should maintain both local appearances and relative spatial relationships. The spatial consistency has to be enforced if we need point-to-point correspondences rather than the mere pattern detection [1]. Geometrical transformations such as scaling and rotation introduce such a computational complexity that few methods have been able to deliver fast, accurate, and robust solutions.

Hough transform [2, 3] and RANSAC [4] have been widely used in shape matching. Hough transform often requires a careful selection of its parameters (e.g. bin size), and easily breaks down in the presence of clutter. RANSAC is more resistant to clutter: it generates random matching hypotheses for a small number of anchor points, and then evaluates the hypotheses with all the points. As it becomes increasingly slow with more outliers in the image, heuristics and general Hough transform have been used [4] to preprocess the local matches and prune the unpromising ones. However, these remedies help little when local feature matching becomes increasingly ambiguous. In addition, neither Hough transform nor RANSAC directly produces point-to-point feature correspondences.

Object matching has also been studied as a graph matching problem. For special cases where the graphs have no loops or the target candidates have linear orders, exact polynomial-time algorithms such as dynamic programming [5] and max flow [6] can be used. Graph matching in general is NP-hard and an exact solution is often too slow to be feasible for large scale problems.

Various approximation methods have been developed. Iterative Conditional Modes (ICM) [8] is a local optimization method that gets easily trapped in local optima. Back tracking is a graph search method with pruning heuristics [7]. Graph cut [9] and belief propagation (BP) [11] are recent global search methods that have been applied to a range of matching problems including stereo [10, 12], motion estimation [18], object pose estimation [14], tracking [13], and recognition [15]. They often have a linear to quadratic complexity with respect to the number of target candidates and become slow when searching over large ranges in target images. Large range searching is precisely what scale and rotation invariant matching demands.

Mathematical programming is yet another approach to object matching. Linear programming has been used in object matching without scale and rotation changes [20]. Integer quadratic programming and its linear relaxation [19] have been proposed for scale-invariant matching. Softassign [16] with its extension [17] is one of the few methods that handle large object deformations. It employs an iterative optimization routine that alternates between local point matching and global deformation matching.

We propose a linear method for scale and rotation invariant object matching. We explicitly model scaling and rotation, and approximate the resulting formulation by a convex program. Using the lower convex hull property, we can effectively solve the program on a small number of variables. Our method thus has a complexity rather independent of the number of target candidates, making it suitable for very large scale problems. The relaxed results can be further improved by successively shrinking trust regions.

Our extensive experimentation demonstrates that the proposed linear solution is accurate, fast, and robust. It works well with both scale and rotation invariant features such as SIFT [4] and non-invariant features such as shape context [21] and simple image patches.

2. Scale and Rotation Invariant Matching

Given two sets of points, each point associated with a feature vector, we would like to detect the model set in a target set that contains a globally translated, scaled, rotated, and locally deformed version of those model points among some irrelevant clutter points (Fig. 2).

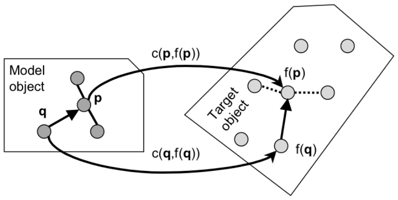


Figure 2. Our matching criterion minimizes feature matching cost and spatial matching cost. The feature matching cost is determined by the difference of the feature vectors associated with each model point $\mathbf{p}$ and its match $\mathbf{f}(\mathbf{p})$, and the spatial matching cost is determined by the difference between the vectors $(\mathbf{p}, \mathbf{q})$ and $(\mathbf{f}(\mathbf{p}), \mathbf{f}(\mathbf{q}))$, invariant to unknown global scaling and rotation.

2.1. Criterion

Our objective is to find every model point a corresponding target point so that they share similar local features and pairwise spatial connections. Formally, let $\mathcal{M}$ be the set of model points and $\mathcal{N}$ the set of all pairs of neighboring model points. Let $\mathbf{f}(\mathbf{p})$ be the target point matched to model point $\mathbf{p}$. Our objective function minimizes both feature and spatial matching costs:

$$\min_{\mathbf{f}} \left\{ \sum_{\mathbf{p}\in\mathcal{M}} c(\mathbf{p}, \mathbf{f}(\mathbf{p})) + \lambda \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} g(\mathbf{p}, \mathbf{q}, \mathbf{f}(\mathbf{p}), \mathbf{f}(\mathbf{q})) \right\}$$

Here, c is the feature matching cost which is small if the model point $\mathbf{p}$ and target point $\mathbf{f}(\mathbf{p})$ have small feature difference; g is the spatial matching cost which is small if the spatial connection $(\mathbf{p}, \mathbf{q})$ is similar to $(\mathbf{f}(\mathbf{p}), \mathbf{f}(\mathbf{q}))$ under some global geometrical transformation; and λ controls the relative weight of the two terms.

The objective function that admits scaling and rotation invariance in the spatial matching cost can be written as:

$$\begin{aligned} &\min_{\mathbf{f},s,R} \{ \sum_{\mathbf{p}\in\mathcal{M}} c(\mathbf{p}, \mathbf{f}(\mathbf{p})) + \\ &\lambda \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \|R \cdot (\mathbf{p} - \mathbf{q}) - s \cdot (\mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{q}))\| \} \end{aligned} \quad (1)$$

where s and R are unknown scaling factor and rotation matrix respectively. We thus have to find the point correspondence and estimate the scale and rotation simultaneously.

Note that the feature matching cost c is not a function of s and R. However, it is by no means restrictive to features that are scale and rotation invariant, e.g. SIFT [4]. For general features, we compute matching cost as follows. For model point $\mathbf{p}$ and target point $\mathbf{t}$, we compute the features for $\mathbf{t}$ at multiple scales s and angles θ, and use the minimal distance between the features of $\mathbf{t}$ and $\mathbf{p}$ as the matching cost $c(\mathbf{p}, \mathbf{t})$:

$$c(\mathbf{p}, \mathbf{t}) = \min_{s,\theta} \text{distance}(\text{feature}(\mathbf{p}), \text{feature}(\mathbf{t}; s, \theta)). \quad (2)$$

The nonlinear optimization problem in Eqn. (1) involves both discrete and continuous variables. For real applications, there are a large number of model points and target points. Exhaustive search is not an option. Our idea is to convert the problem into a small set of convex programs which can be efficiently solved. For the rest of the paper, we assume points are in 2D and their spatial matching cost is measured with L_1 norm, although our approach can be extended to higher dimensions as well as L_2 norm.

2.2. Matrix Formulation

We write Eqn. (1) in a succinct form using assignment matrix X. Let 1_n denote a column vector of n 1's, $'$ matrix transpose, tr the trace of a matrix, and $|\cdot|$ the summation of absolute values of all the elements in a matrix. Let n_m and n_t be the numbers of model and target points respectively.

$$\begin{aligned} &\min\ \varepsilon(X, s, R) = \text{tr}(C'X) + \lambda|EMR - sEXT| \quad (3) \\ &\text{subject to } X1_{n_t} = 1_{n_m}, X \in \{0,1\}^{n_m \times n_t} \\ &\qquad s > 0 \\ &\qquad R'R = I. \end{aligned}$$

There are 3 unknown variables:

$X = n_m \times n_t$ binary assignment matrix. Each row of X contains exactly one 1: $X(i,j) = 1$ indicates that the model point i is matched with target point j.

$s =$ global scaling factor s.

$R = 2 \times 2$ global coordinate rotation matrix. It is in fact the transpose of the R in Eqn. (1).

and 4 known matrices:

$M = n_m \times 2$ model point 2D coordinate matrix.

$T = n_t \times 2$ target point 2D coordinate matrix.

$C = n_m \times n_t$ feature matching cost matrix. $C(i,j)$ is the feature matching cost between model point i and target point j, i.e., $c(.)$ in Eqn. (1).

$E = n_e \times n_m$ edge-node incidence matrix for the model graph, where $n_e = |\mathcal{N}|$. Each row describes an edge with exactly two non-zero numbers: 1 and -1, and their signs can be switched. For example, $E(e,i) = 1$, $E(e,j) = -1$ indicate edge e connects nodes i and j in the model.

This optimization problem has a nonlinear objective function subject to linear and quadratic constraints and includes both continuous and integer variables. It is NP hard.

We make two additional comments regarding the formulation. **1)** We can take reflection into account by dropping $s > 0$: a negative s simply means that the spatial connection in the image could be a mirror reflection of that in the model (after rotation), scaled by factor $|s|$. Our method can be extended accordingly. **2)** It is essential to separate scale and rotation in the spatial matching cost. If we combine scale s and rotation R into one similarity transform $S = R/s$, i.e.,

$$\min \varepsilon(X,S) = \mathrm{tr}(C'X) + \lambda |EMS - EXT|$$

we would introduce a strong bias favoring small scales. The formulation is seemingly simpler but essentially wrong, resulting in matching a spurious small pattern in the image.

2.3. Linearization

Instead of directly solving the hard mixed integer nonlinear program in Eqn. (3), we convert it into linear program which can be efficiently solved. There are three obstacles in linearizing Eqn. (3): **1)** the L_1 norm in the spatial matching term, **2)** the nonlinearity introduced by the multiplication of the integer variable X and the continuous variable s, and **3)** the quadratic constraint on the rotation matrix R.

First, we introduce $n_e \times 2$ non-negative auxiliary matrices, Y and Z, to turn the L_1 norm optimization into a linear objective with linear constraints. It is well known that:

$$\min |x| \Leftrightarrow \begin{array}{ll} \min & y + z \\ \text{subject to} & y - z = x \\ & y \geq 0, z \geq 0 \end{array}$$

Applying to every element of $EMR - sEXT$, we have:

$$\begin{aligned} \min\ & \varepsilon(X,s,R,Y,Z) = \mathrm{tr}(C'X) + \lambda 1'_{n_e}(Y+Z)1_2 \\ \text{subject to}\ & Y - Z = EMR - sEXT \\ & Y \geq 0, Z \geq 0. \end{aligned} \tag{4}$$

Intuitively, for each element pair in Y and Z, since at most one of them is nonzero as ε is minimized, the sum of all the elements in Y and Z must be equal to $|EMR - sEXT|$.

Next, we introduce assignment variables at multiple scales $\{X_l\}$ such that $X = \sum_l X_l$, to transform the multiplication of s and X in Eqn. (4) into a linear function of X_l subject to linear constraints among s, X_l, and X. Illustrated in Fig. 3, we quantize s into n_s discrete values, $0 < s_1 < \ldots < s_{n_s}$. When $s = s_l$, $X = X_l$. Since only one scale is realized in a matching solution, $sX = \sum_{l=1}^{n_s} s_l X_l$ and hence $sEXT = \sum_{l=1}^{n_s} s_l EX_l T$. Recall that $X 1_{n_t} = 1_{n_m}$, we thus also have $\sum_{l=1}^{n_s} s_l X_l 1_{n_t} = s 1_{n_m}$, which enforces each model point to select the same scale in matching. As we relax the binary constraints on X_l to any value within $[0,1]$, s is no longer restricted to a discrete level s_l, but can be any real number within $[s_1, s_{n_s}]$.

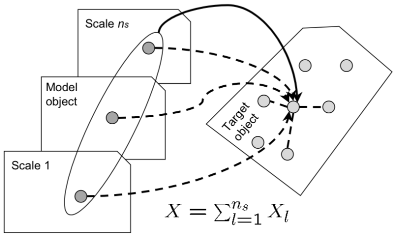


Figure 3. Scale linearization. As the assignment matrix X is represented as a combination at basis scales, the scaled assignment term sX in the spatial matching cost becomes linear.

Finally, we re-parameterize the rotation matrix R in terms of its elements u, v, and approximate the unit normal constraint $u^2 + v^2 = 1$, i.e., a circle in the uv plane, with four line segments (Fig. 4):

$$R'R = I \quad \approx \quad R = \begin{bmatrix} u & -v \\ v & u \end{bmatrix}$$
$$u \pm v = \pm 1, \quad |u| \leq 1, \ |v| \leq 1$$

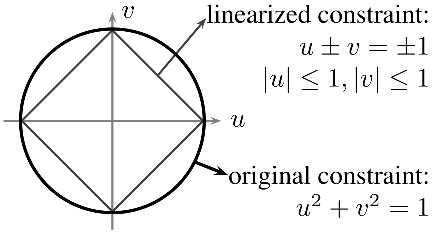


Figure 4. Rotation linearization. The orthonormal constraint on the rotation matrix R is approximated by four line segments.

Overcoming the three obstacles, we reach a complete linearization of the original optimization problem in Eqn. (3):

$$\text{LP:} \quad \min \varepsilon(X, s, u, v, Y, Z, X_1, \ldots, X_{n_s}) = \text{tr}(C'X) + \lambda 1'_{n_e}(Y+Z)1_2 \quad (5)$$

$$\text{subject to } Y - Z = EM \begin{bmatrix} u & -v \\ v & u \end{bmatrix} - \sum_{l=1}^{n_s} s_l E X_l T$$

$$Y, Z \geq 0, \quad u \pm v = \pm 1, \quad |u| \leq 1, \quad |v| \leq 1$$

$$X = \sum_{l=1}^{n_s} X_l, \quad X_l \geq 0, \forall l$$

$$\sum_{l=1}^{n_s} s_l X_l 1_{n_t} = s 1_{n_m}$$

$$X 1_{n_t} = 1_{n_m}, X \geq 0$$

We relax X into the continuous domain, the optimal target point coordinates are computed by $T^* = XT$.

2.4. Lower Convex Hull Speedup

The solution to the LP in Eqn. (5) can still be slow when the number of variables, proportional to $n_m \times n_t \times n_s$, is large. Fortunately, we can throw away many variables without changing the optimum of the linear program.

The key observation is that, for each model point $\mathbf{p}$, we only need to keep those target points $\mathbf{t}$ and their associated feature matching costs $c(\mathbf{p}, \mathbf{t})$ that correspond to the vertices of the lower convex hull of the point cloud $\{(\mathbf{t}, c(\mathbf{p}, \mathbf{t})) : \forall \mathbf{t}\}$.

Since the LP uses a linear combination of basis costs to approximate the original cost, at any fixed scale and rotation, ε is minimized on the lower convex hull of each feature's matching cost surface. The lower convex hull vertices are what matters. The effective number of variables becomes many times smaller and relatively independent of the number of target points.

The lower convex hull trick maintains the optimality of LP in the relaxed continuous domain, but the approximation to the original discrete solution could be rough, especially if the search range is large. We solve this problem with a successive trust region shrinkage scheme similar to [20]. Note that we only shrink the trust region for point locations, not for scale or rotation parameters. As it becomes smaller, the linear approximation becomes more accurate.

The average complexity of a linear program is roughly logarithmic to the number of variables and linear to the number of constraints [22]. In our formulation, the numbers of variables and constraints are largely decoupled from the number of target points with the lower convex hull trick. The average complexity of our LP is therefore nearly independent of the number of target points. Typically, for matching 100 model points and thousands of target points, each LP iteration takes less than 1 second on a 2.8GHz PC, and the trust region shrinkage runs 4-8 iterations.

2.5. Algorithm

Our scale and rotation invariant matching algorithm is:

1. Compute the feature matching cost matrix C between model features and target features.
2. Initialize trust region for each model point to be the entire target image.
3. Compute the lower convex hull vertices of matching costs for each model point within its trust region.
4. Solve 4 programs in Eqn. (5), one for each uv line.
5. Update the trust regions. If they are small enough, find the linear program that has the lowest matching cost ε and output the matching result; otherwise go to 3.

We simplify the algorithm in the implementation. Instead of solving 4 linear programs in each iteration, we only refine the one that yields the lowest cost in the first iteration. The simplified algorithm is still accurate and robust.

3. Experiments and Discussions

We choose the greedy matching scheme ICM for main comparison. BP and graph cut methods are not used in comparison because they are not easily extended to solve the proposed energy function efficiently.

3.1. A Working Example

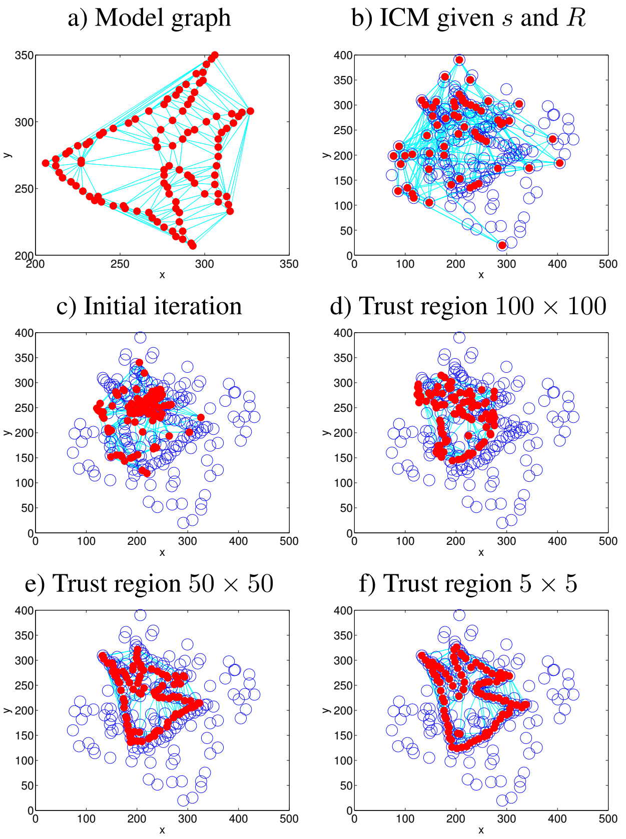
We first use the synthetic point pattern matching in Fig. 5 to illustrate our algorithm. The model graph is a Delaunay triangulation of 98 points of a fish shape. The target object is a locally deformed, globally scaled and rotated version of the template with 100 additional random noise points.

For each target point, we compute the shape context at 7 different scales ranging from 0.5 to 2 times of the template size, and at multiple angles by shifting the shape context along the angular axis. The feature matching cost is the minimal χ^2 distance between the model feature and all the scaled and rotated versions of the target feature (Eqn. (2)).

While ICM fails to find the right correspondences even with the correct scale and rotation (Fig. 5b), our algorithm gets roughly the right scale and orientation after the initial iteration (Fig. 5c). As we narrow down the trust region from 100 × 100 to 5 × 5 (Fig. 5d-f), the match is progressively refined in scale, rotation, and correspondence.

3.2. Benchmark with Synthetic Point Sets

For synthetic data, we give ICM the advantage of knowing the right scale s and rotation R, i.e., the same energy function is used, but s and R are fixed to the correct values. We also consider the simplest greedy method, which finds the best match for each model point separately.



a) Model graph

b) ICM given s and R

c) Initial iteration

d) Trust region 100×100

e) Trust region 50×50

f) Trust region 5×5

Figure 5. An example of the linear solution in matching a scaled and rotated deformable shape. a) A deformable fish shape template. b) Match found by ICM with known scale and rotation. c) Our initial match when the trust region is the entire image. d-f) Our results over iterations that shrink the trust regions. The algorithm converges to the nearly perfect match.

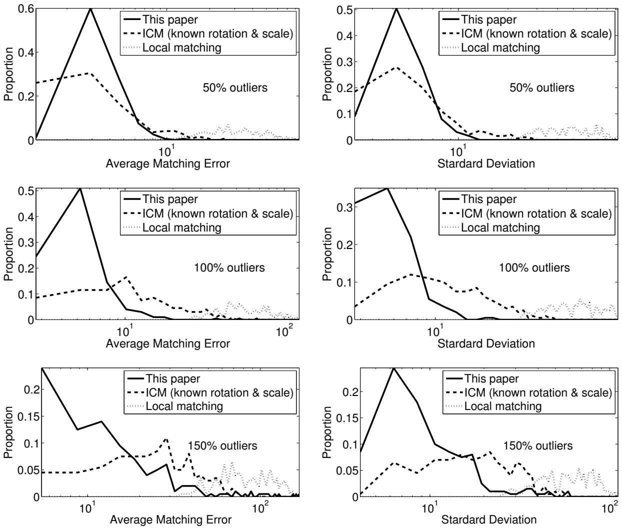


Figure 6. Our matching error for *fish* is smaller than ICM and local matching in terms of both the mean (left) and the standard deviation (right), for all three levels of clutter (one for each row).

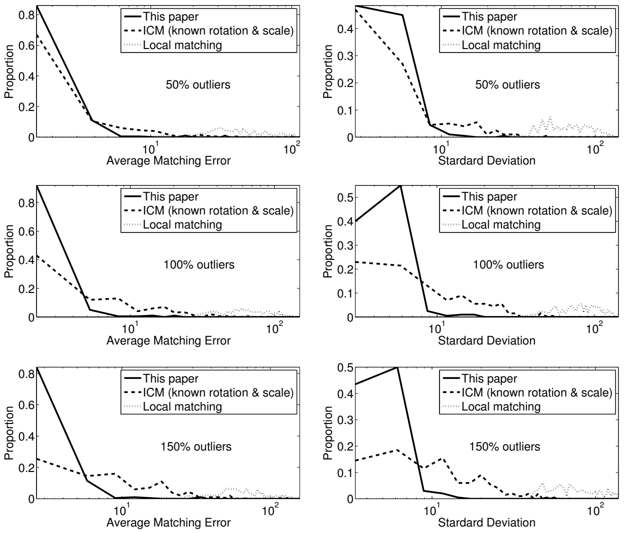


Figure 7. Our matching error for *random point could* is smaller than ICM or local matching in terms of both the mean (left) and the standard deviation (right), for all three levels of clutter.

Two synthetic point models are used to benchmark our method, ICM, and the simple local search. One is the fish in Fig. 5, the other is a random point cloud. The local deformation is smooth for the fish, and restricted to a disturbance of $[0, 10]$ for the random cloud. The scale varies within $[0.5, 2]$, and the angle of rotation varies within $[0°, 360°]$. We consider three clutter levels, where the number of random noise points is 50%, 100%, 150% of the number of model points respectively.

For each test, we compute the mean and standard deviation of the point matching errors. The matching performance is quantified by the distributions of the error means and standard deviations over all the trials. Good performance has the distributions concentrated on the low matching error range for both measures. Fig. 6 and Fig. 7 show the histograms computed over 200 random trials for each configuration. Our LP solution consistently gives much smaller error than ICM with the right scale and rotation, and the local matching results are always the worst. With increasing clutter, our method still has a high chance of finding the right scale, rotation, and point-to-point correspondences.

3.3. Real Videos

We test our method on two kinds of videos: our own videos (*book, magazine, bear*) demonstrating scaling, rotation, deformation, and occlusion, and YouTube videos (*butterfly, bee, fish*) of animals in their natural habitats.

Given a sample image for each video (Fig. 8 Row 1), we label the object region, and build a model graph with interest points and their neighboring connections through Delaunay triangulation. We use SIFT points for all the videos except the fish, for which small image patches on randomly

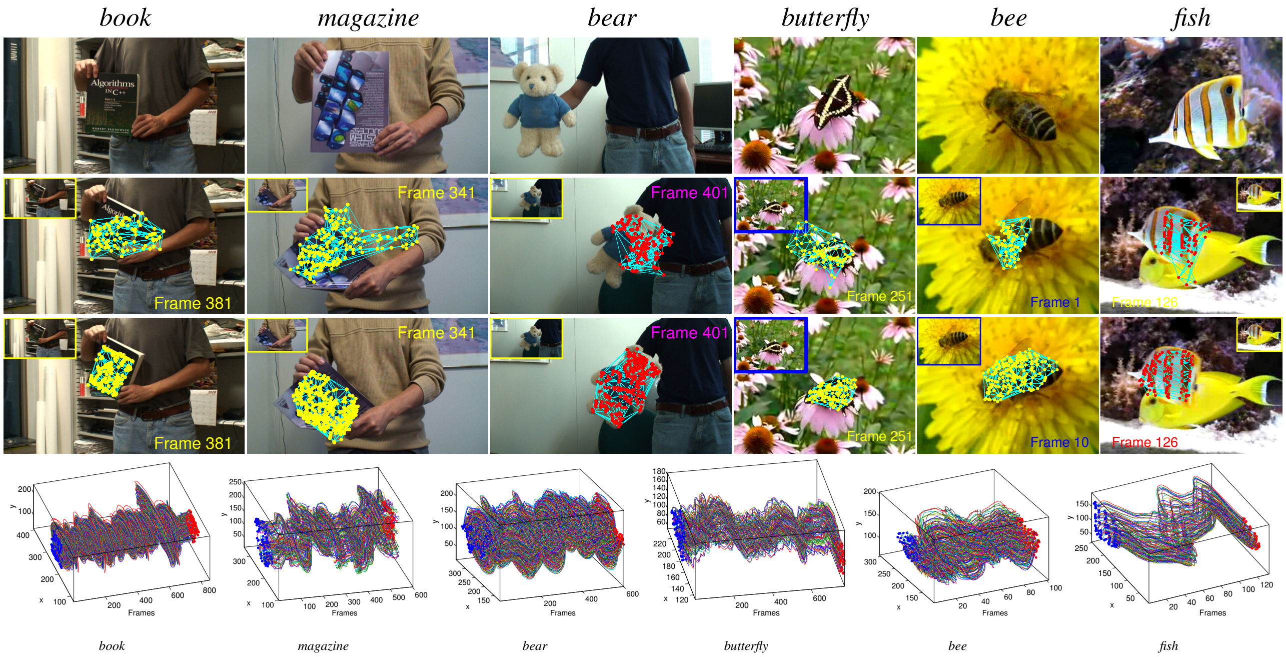


Figure 8. Our LP is accurate and robust in matching objects in challenging real videos where ICM fails. **Row 1**: images used to construct object templates from the Delaunay triangulation of detected interest points. **Row 2**: ICM results and **Row 3:** our LP results on the same frames. **Row 4**: point trajectories from the first (blue dots) to the last frame (red dots). Their shapes give telltale signs of scaling and rotation (*book* and *bear*), complex local warping (*magazine* and *butterfly*), and segments of smooth movement (*bee* and *fish*).

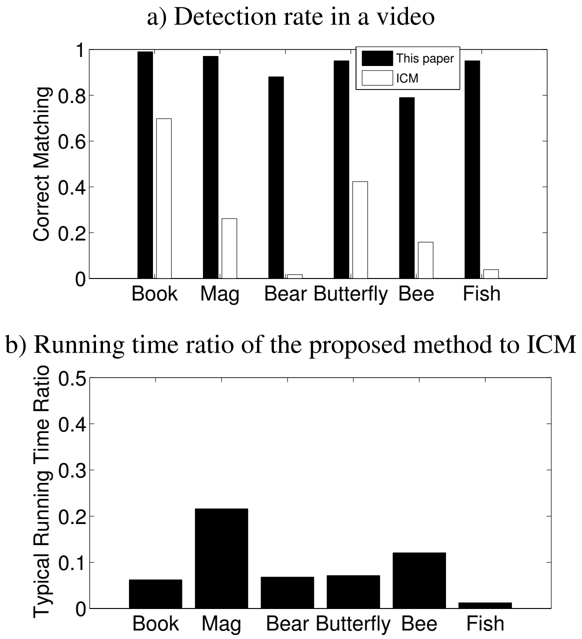


Figure 9. Our LP method (black bar) has a much higher detection rate than ICM (white bar) yet requiring a fractional running time.

video	*book*	*magazine*	*bear*	*butterfly*	*bee*	*fish*
#frames	856	601	601	771	101	131
#model	151	409	235	124	206	130
#target	2143	1724	1683	1405	1029	7316
time	1.6	11	2.2	1	2	0.9
accuracy	99%	97%	88%	95%	79%	95%

Figure 10. Performance statistics of our LP method on real videos. The five rows give the total number of frames, the number of model points, the average number of target points per frame, the typical running time measured by the number of seconds that one LP iteration takes on a 2.8GHz PC, and the accuracy measured by the detection rate over the entire video, as shown in Fig. 9.

selected edge points within the object region are used.

To apply ICM to these real videos, with no ground-truth for scale and rotation available, we have to employ exhaustive search for s (7 scales with step 0.25 from 0.5 to 2) and R (every 30°). Since ICM is rather efficient, its performance is an indicator of the complexity of the matching problem.

Our LP solution greatly outperforms ICM in terms of the quality of individual matches (Fig. 8 Rows 2-3), as well as the robustness (Fig. 9a) and efficiency (Fig. 9b) in matching the entire sequences.

Our single-frame-based matching algorithm requires no initialization and can track a deformable object undergoing large and complex motion over long video sequences (Fig. 8 Row 4). The shapes of these long tracks are characteristic of the object's deformation and movement patterns, which could be useful for activity recognition.

Fig. 11 shows sample matching results and Fig. 10 shows the overall performance statistics.

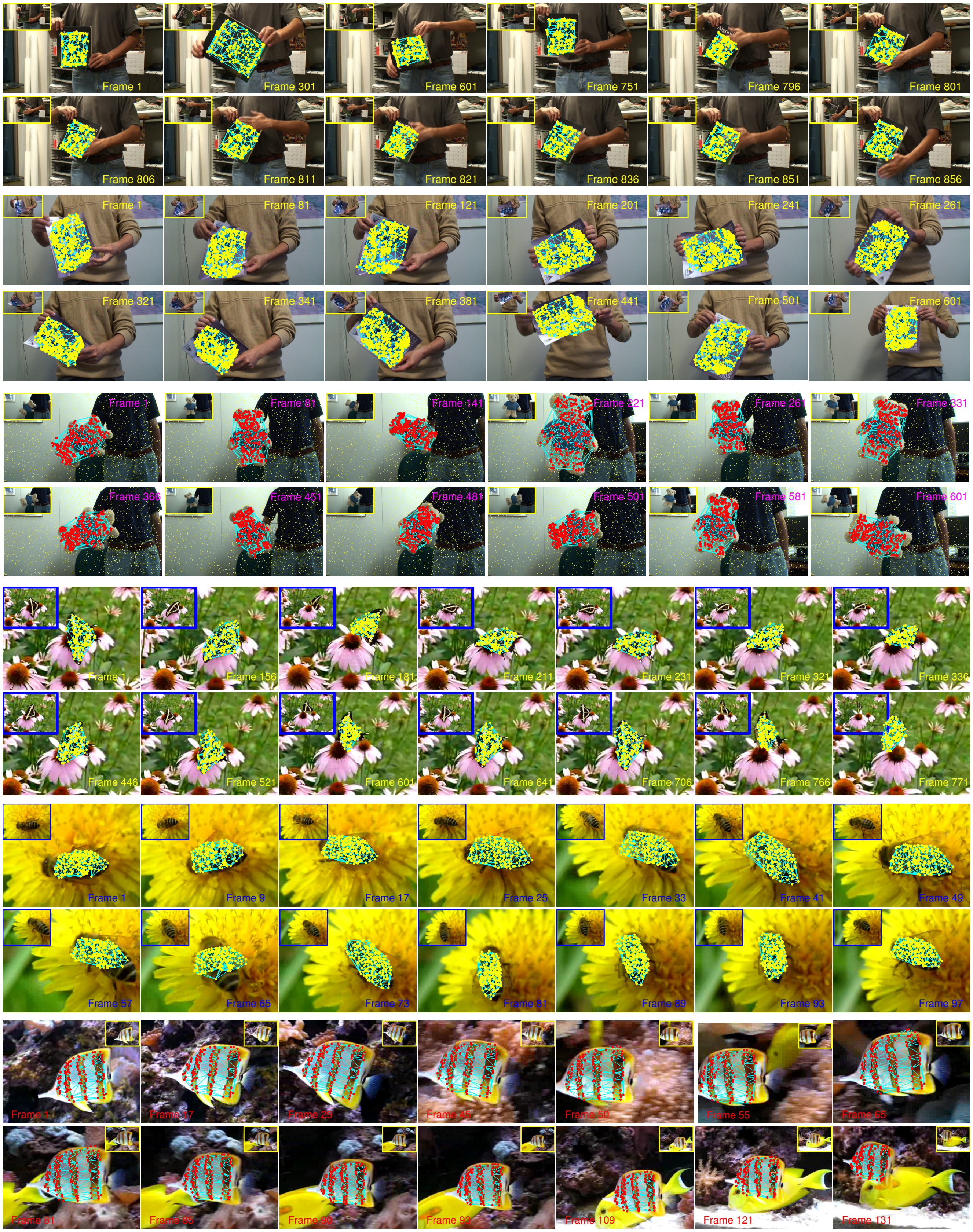


Figure 11. Sample matching results. These objects all have large scaling and rotation. In particular, *book* could be occluded, *magazine* has large warping, *bear* is textureless, *butterfly* flaps wings, *bee* is striated and circling in depth, and *fish* has weakly distinguishable features.

The first three video sequences all have a large range of scaling and rotation. In addition, we test matching performance on occlusion with a hand in front of the book, on complex deformation with significant warping of the magazine, on appearance with texturelessness of the furry toy bear, where local SIFT features (marked by yellow dots in Fig. 11) become highly ambiguous.

The last three are YouTube videos of animals moving naturally in their habitats. The image quality is low due to heavy compression. In addition, we test matching performance on large deformation with wing flapping of the butterfly, on appearance with indistinguishable texture features of the bee and the fish, and on large viewpoint change with the in-depth circling of the bee.

The results in Fig. 11 and Fig. 10 show that our LP solution is accurate and robust in face of all these challenges. The running time is largely dependent on the number of model points, and insensitive to the number of target points. The detection rate is largely dependent on the distinctiveness of features that allows the objective function to tell model points apart from each other: It is higher than 95% for the book, magazine, butterfly, and fish, lower for the textureless bear, and lowest for the striated bee.

Matching the tropical fish only appears simple. There are few distinctive SIFT features. We detect edges instead to locate target points, and use small image patches as their features. The feature matching cost is defined as the minimum color block Euclidean distance at different rotations, which is roughly scale invariant for these stripes. Our method actually works pretty well with such crude features considering the large changes in scale, rotation, and color. This example also demonstrates that our method is versatile and robust with various features and matching cost functions.

4. Summary

Scale and rotation invariant object matching is an NP hard problem that few existing methods can deal with effectively. We develop a linear solution with a computational complexity insensitive to the number of target points, making it suitable for large scale matching problems.

Our results on both synthetic and real data demonstrate the accuracy, robustness, and efficiency of our method. It can be directly used to track an object with large shape deformations and geometrical transformations, and has potential to be applied to object and activity recognition.

Acknowledgments

This research is supported by Boston College startup funding for HJ and by NSF CAREER IIS-0644204 and a Clare Boothe Luce Professorship to Yu.

References

[1] D. Nistér and H. Stewénius, “Scalable recognition with a vocabulary tree”, CVPR 2006. 1

[2] J. M. Gonzlez-Linares, N. Guil and E. L. Zapata, “An efficient 2D deformable objects detection and location algorithm”, Pattern Recognition, 36(11), 2003, pp.2543-2556. 1

[3] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval”, IEEE TPAMI, vol.19, no.5, 1997. 1

[4] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” IJCV, 60(2), 2004, pp. 91-110. 1, 2

[5] P.F. Felzenszwalb and D.P. Huttenlocher “Pictorial structures for object recognition”, IJCV, 61(1), 2005, pp.55-79. 1

[6] S. Roy and I.J. Cox, “A maximum-flow formulation of the n-Camera stereo correspondence problem”, ICCV 1998. 1

[7] W. Eric L. Grimson, “The combinatorics of object recognition in cluttered environment using constrained search”, A.I. Memo No. 1019, Feb. 1988. 1

[8] J. Besag, “On the statistical analysis of dirty pictures”, J. of the Royal Stat. Soc., Series B 48(3):259-302, 1986. 1

[9] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?” TPAMI, Feb. 2004. 1

[10] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph guts”, ICCV, 2001. 1

[11] Y. Weiss and W.T. Freeman “On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs”, IEEE Trans. Info. Theory, 47:2(723-35), 2001. 1

[12] J. Sun, N.N. Zheng and H.Y. Shum, “Stereo matching using belief propagation”, TPAMI 25(7), 2003, pp.787-800. 1

[13] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky, “Distributed occlusion reasoning for tacking with nonparametric belief propagation”, NIPS 2004. 1

[14] D. Ramanan and C. Sminchisescu, “Training deformable models for localization”, CVPR 2006. 1

[15] A. Quattoni, M. Collins and T. Darrell, “Conditional random fields for object recognition”, NIPS, 2004. 1

[16] H. Chui and A. Rangarajan, “A new algorithm for non-rigid point matching”, CVPR, 2000. 2

[17] Z. Tu, and A. Yuille, “Shape matching and recognition-using generative models and informative features”, ECCV 2004. 2

[18] P.F. Felzenszwalb and D.P. Huttenlocher. “Efficient belief propagation for early vision”, IJCV, 70(1), 2006. 1

[19] A.C. Berg, T.L. Berg and J. Malik “Shape matching and object recognition using low distortion correspondence”, CVPR 2005. 2

[20] H. Jiang, M.S. Drew and Z.N. Li, “Matching by linear programming and successive convexification”, TPAMI, 2007. 2, 4

[21] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts”, TPAMI 2002. 2

[22] V. Chvátal, Linear Programming, W.H. Freeman and Co., New York, 1983. 4