IEEE Conference on Computer Vision and Pattern Recognition 2011 (CVPR'11)

Scale and Rotation Invariant Matching Using Linearly Augmented Trees

Hao Jiang
Boston College
hjiang@cs.bc.edu

Tai-Peng Tian and Stan Sclaroff
Boston University
tian@cs.bu.edu, sclaroff@cs.bu.edu

Abstract

We propose a novel linearly augmented tree method for efficient scale and rotation invariant object matching. The proposed method enforces pairwise matching consistency defined on trees, and high-order constraints on all the sites of a template. The pairwise constraints admit arbitrary metrics while the high-order constraints use L1 norms and therefore can be linearized. Such a linearly augmented tree formulation introduces hyperedges and loops into the basic tree structure, but different from a general loopy graph, its special structure allows us to relax and decompose the optimization into a sequence of tree matching problems efficiently solvable by dynamic programming. The proposed method also works on continuous scale and rotation parameters; we can match with a scale up to any large number with the same efficiency. Our experiments on ground truth data and a variety of real images and videos show that the proposed method is efficient, accurate and reliable.

1. Introduction

Matching objects in cluttered images is a challenging task because the target object may appear rotated, scaled and locally deformed. To handle shape variation, object matching is naturally formulated as a graph matching problem, in which the object is divided into parts represented by graph nodes and the part coupling is represented by graph edges. Matching is to assign the target candidates to graph nodes so that the assignment has low cost and it is consistent with the constraints defined on the graph edges.

We propose a novel formulation to tackle scale and rotation invariant object matching. In our model, the object parts follow basic tree relations and we also introduce global constraints that couple all the tree nodes. These global constraints can be linearized and we call this class of constraints *linearly augmented tree* (LAT) constraints.

A large class of matching techniques are based on discrete energy minimization. If the energy function is submodular then it can be efficiently minimized using maxflow algorithms [5, 3]. Alternatively, if the underlying graph is a tree then dynamic programming is used [2]. For general loopy graphs, popular approximation techniques include loopy Belief Propagation [6], convergent Tree Reweighted Message Passing [12], integer quadratic programming [8], primal-dual techniques [11] and dual decomposition [18, 17]. Unfortunately, these discrete energy minimization techniques do not generalize easily to handle a mixture of discrete and continuous variables in the LAT constraints, *e.g.*, the scale and rotation parameters are continuous. A typical workaround is to quantize these continuous variables. Our optimization algorithm avoids such an ad hoc quantization step by using a decomposition method that splits the relaxed problem into a master and slave optimization. The master problem optimizes over the set of continuous variables, and the slave problem performs efficient combinatorial optimization over the discrete variables in order to generate proposals for the master problem. Furthermore, these cited works only model low-order constraints (typically up to two) and in contrast, we use LAT to model high-order constraints that couple all the model points in the matching.

Other matching techniques do not use an explicit graph template. For example, the Hough Transform [4] is a robust and efficient voting method but it requires careful quantization of the parameter space. Techniques such as Softassign [7], spectral graph methods [14] and RANSAC [16] do not need to quantize the global transformation parameters, but their matching performance deteriorates rapidly when clutter increases and features weaken. In our experiments, we show that matching using LAT constraints is reliable even when the scene is highly cluttered and the features are not distinctive.

Two closely related works are [13, 19]. These methods have different drawbacks, namely, restricting the pairwise cost to the L1 norm [13], quantizing the scale parameter [13] and the tendency to match small structures when the features are weak [19]. The lower convex hull method these methods rely on is unsuitable for very weak feature matching. Our new formulation removes these drawbacks.

The contributions of this paper are twofold:

New Formulation for Rotation and Scale Invariance
The proposed linear augmented tree model (LAT) allows arbitrary metrics for the pairwise costs on trees

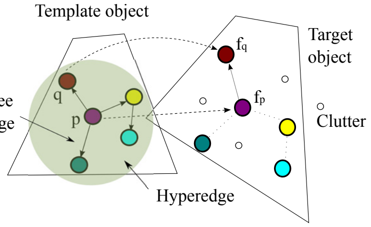


Figure 1. Matching a template to a target object using a linearly augmented tree (LAT) model. Our method allows arbitrary pairwise constraints defined on the basic tree edges and linear high-order constraints that couple all the model nodes.

and it also allows powerful high-order constraints that couple all the nodes. It works on continuous scale and rotation parameters. It also allows virtually unbounded scaling so that users do not have to guess the scale range.

Efficient Matching Algorithm Our algorithm efficiently solves the matching with LAT constraints by relaxing the problem and decomposing it into a sequence of efficient dynamic programming problems. Furthermore, the relaxed problem can be solved optimally.

2. Scale and Rotation Invariant Matching

We formulate scale and rotation invariant matching using linearly augmented tree constraints (Fig. 1). Given a set of template points $\mathcal{I}$ and target candidate points $\mathcal{J}$, the matching problem is formulated to search for three items, namely the mapping $f : \mathcal{I} \rightarrow \mathcal{J}$, rotation θ_0 and scale s_0 to minimize the objective function

$$\mathbf{c}(f, \theta_0, s_0) = \mathbf{c}_u(f) + \mathbf{c}_t(f, \theta_0, s_0) + \mathbf{c}_g(f, \theta_0, s_0). \quad (1)$$

The unary cost term

$$\mathbf{c}_u(f) = \sum_{i \in \mathcal{I}} c(i, f_i) \quad (2)$$

is the sum of the matching cost $c(i, f_i)$ between model point i and its target point f_i. The scale and rotation term,

$$\begin{aligned}\mathbf{c}_t(f, \theta_0, s_0) = \mu & \sum_{(p,q) \in \mathcal{N}} d(\theta((p,q),(f_p,f_q)), \theta_0) \\ & + \gamma \sum_{(p,q) \in \mathcal{N}} |s((p,q),(f_p,f_q)) - s_0|,\end{aligned} \quad (3)$$

encourages pairs of model points to have similar rotation angle θ_0 and scale factor s_0 in the matching. $\mathcal{N}$, the set of neighboring model points, corresponds to the edges of a tree. $\theta((p,q),(f_p,f_q))$ is the rotation angle from vector $\overrightarrow{pq}$ to $\overrightarrow{f_p f_q}$. $s((p,q),(f_p,f_q))$ is the scaling factor between the two vectors. Fig.1 illustrates the matching of a pair of model points. In Eq.(3), $d(.)$ computes the difference of two angles; coefficients μ and γ control the weight between terms. The global cost term

$$\mathbf{c}_g(f, \theta_0, s_0) = g(s_0, \theta_0, h(1), \ldots, h(n), h(f_1), \ldots, h(f_n))$$

introduces global constraint across all the model points ($|\mathcal{I}| = n$), and $h(.)$ is a function that maps model points and target points to some quantities, *e.g.*, the coordinates. We require that $g(.)$ contain only the L1 norm and linear operations on quantities of the model and target points. As shown later, the scale-rotation term $\mathbf{c}_t$ and the global term $\mathbf{c}_g$ can be linearized and form hyperedges on the basic tree nodes. The formulation therefore follows a LAT model.

Even though the basic structure of a LAT model is a tree, the linear high-order constraints make the optimization difficult to solve. Naive discretization is infeasible if the scale upper bound is unknown; quantizing rotation angles and scales would result in too many discrete cases. We propose to encode the problem as a mixed integer linear program and show how to exploit its special LAT structure to design an efficient algorithm.

2.1. Linearization

We describe how to encode the minimization of $\mathbf{c}(f, \theta_0, s_0)$ (Eq. (1)) as a mixed integer linear program. Assume that there are n model points and m target points. Let $[\![\pi]\!] = 1$ if the predicate π holds and 0 otherwise. We introduce an $n \times m$ matrix X and $m \times m$ matrix $Y_{p,q}$ whose elements

$$x_{i,j} = [\![f_i = j]\!] \quad \text{and} \quad y^{p,q}_{i,j} = [\![f_p = i \wedge f_q = j]\!].$$

The matrix X indicates the matching of model points to target points, and the matrix $Y_{p,q}$ indicates the matching of a model point pair $(p,q) \in \mathcal{N}$ to target point pairs. We enforce X to be an assignment matrix with the unity constraint $X 1_m = 1_n$, where 1_m is an all one element vector of length m. The X matrix and Y matrices are related by

$$X^T e_p = Y_{p,q} 1_m, \quad \text{and} \quad X^T e_q = Y^T_{p,q} 1_m,$$

where the n-vector $e_p = [0, 0, \ldots, 1, 0, \ldots, 0]^T$ has a single unity element at p.

The unary cost term defined in Eq.(2) can be represented as $\mathrm{tr}(C^T X)$ where $C = [c(i,j)]$ is the matching cost matrix whose element $c(i,j)$ is defined in Eq.(2).

The rotation term: For a model point pair $(p,q) \in \mathcal{N}$, we assume p matches target point i and q matches j. Let the rotation angle from vector $\overrightarrow{pq}$ to $\overrightarrow{ij}$ be $\theta^{p,q}_{i,j}$ and the $m \times m$ rotation angle matrix $\Theta_{p,q} = [\theta^{p,q}_{i,j}]$. If the target vector $\overrightarrow{ij}$ degenerates to a single point then $\theta^{p,q}_{i,j}$ is assigned a random number in $[0, 2\pi]$. The rotation for the model point pair (p,q) can be represented as $\mathrm{tr}(Y^T_{p,q} \Theta_{p,q})$. We require that all the model point pairs share similar rotation in the matching so that the object spatial structure is maintained. To this end, we may minimize $\sum_{(p,q) \in \mathcal{N}} |\mathrm{tr}(Y^T_{p,q} \Theta_{p,q}) - \theta_0|$, where θ_0 is the overall (unknown) rotation angle, but this method does not work at the rotation boundaries. To avoid this difficulty, we split the rotation term into cos and sin terms:

$$\sum_{(p,q) \in \mathcal{N}} \{ |\mathrm{tr}(Y^T_{p,q} \cos(\Theta_{p,q})) - u_0| + |\mathrm{tr}(Y^T_{p,q} \sin(\Theta_{p,q})) - v_0| \}$$

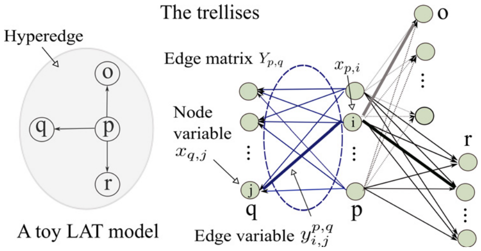


Figure 2. LAT model and trellises. Thick lines indicate the paths.

where u_0 and v_0 correspond to the cos and sin of the unknown rotation angle θ_0; $\cos(.)$ and $\sin(.)$ apply to each element of matrix $\Theta_{p,q}$. The absolute value terms are converted into linear functions by using a standard auxiliary variable trick [10]. The complete linearization is shown in Eq.(4).

The scaling term: The spatial consistency constraint further enforces that the line segments between neighboring model points should scale uniformly. Similar to the rotation matrix $\Theta_{p,q}$, we define an $m \times m$ scaling matrix $S_{p,q}$ for each pair $(p,q) \in \mathcal{N}$. The scaling for the model point pair (p,q) is therefore $\text{tr}(Y^T_{p,q} S_{p,q})$. To enforce the scaling consistency, we minimize

$$\sum_{(p,q)\in\mathcal{N}} |\text{tr}(Y^T_{p,q} S_{p,q}) - s_0|,$$

where s_0 is the global scaling factor. We can linearize this term with auxiliary variable tricks similar to the rotation term.

The global constraint: Term $\mathbf{c}_g$ in Eq.(1) is composed of L1 norms and linear functions of the quantities attached to model and target points. $\mathbf{c}_g$ may have θ_0 and s_0 as parameters. It can be linearized: each $|v|$ term in $\mathbf{c}_g$ turns into the summation of two non-negative auxiliary variables in the objective and their difference is set to equal v in the constraints. As shown in Eq.(4), $\mathbf{c}_g$ is transformed to g_o in the objective and g_c in the constraints.

We now obtain a mixed integer linear formulation of the nonlinear optimization in Eq.(1).

$$\max\{-\text{tr}(C^T X) - \textstyle\sum_{(p,q)\in\mathcal{N}} [\mu(u^+_{p,q} + u^-_{p,q} \\ + v^+_{p,q} + v^-_{p,q}) + \gamma(s^+_{p,q} + s^-_{p,q})] - g_o(w)\} \qquad (4)$$

Subject to:

$$\boxed{\begin{array}{c} \text{tr}(Y^T_{p,q}\cos(\Theta_{p,q})) - u_0 - u^+_{p,q} + u^-_{p,q} = 0, \\ \text{tr}(Y^T_{p,q}\sin(\Theta_{p,q})) - v_0 - v^+_{p,q} + v^-_{p,q} = 0, \\ \text{tr}(Y^T_{p,q} S_{p,q}) - s_0 - s^+_{p,q} + s^-_{p,q} = 0, \\ g_c(X, w) = 0, \end{array}}$$

$$X^T e_p = Y_{p,q} 1_m, \; X^T e_q = Y^T_{p,q} 1_m, \quad X 1_m = 1_n,$$
$$0 \le u^+_{p,q}, u^-_{p,q}, v^+_{p,q}, v^-_{p,q}, s^+_{p,q}, s^-_{p,q}, w \le M,$$
$$a \le u_0 \le b, c \le v_0 \le d, \epsilon \le s_0 \le L, \; u_0 \pm v_0 = \pm 1$$

where X and Y are binary matrices. g_o is a linear function induced by global constraints $\mathbf{c}_g$. By merging its X terms to the first cost term in the objective, we denote g_o as a function of non-negative auxiliary variables w. g_c corresponds to the L1 norm terms in $\mathbf{c}_g$.

In Eq.(4), the original minimization is changed to maximization of the negative. An extra constraint $|u_0| + |v_0| = 1$ is included to approximate the orthonormal constraint $u_0^2 + v_0^2 = 1$. The bounds $[a,b]$ and $[c,d]$ are determined by the quadrant of the approximation line. For instance, if $u_0 + v_0 = 1$, we have $a = 0, b = 1$ and $c = 0, d = 1$. We find the optimum among four quadrants. This constraint is optional; when included it is able to improve the quality of the relaxation. We also include an upper bound M for the auxiliary variables; M is a large number to avoid the unbounded solution. The scale is upper bounded by L and lower bounded by a small number ϵ. In this paper, $M = L = 1000$, and $\epsilon = 0.001$.

It helps to visualize the mixed integer linear program using coupled trellises as illustrated in Fig.2. By expanding the augmented tree nodes, we obtain a set of coupled trellises. Each trellis node corresponds to an X variable, and the edges between the candidate nodes of two neighboring model points correspond to a Y matrix. The optimization can thus be treated as searching for the optimal "paths" starting from a tree root node candidate and ending at a candidate of each tree leaf node. If the paths pass a node, the corresponding X variable is 1 and otherwise 0. If the paths pass an edge, the corresponding Y variable is 1 and otherwise 0. The cost of the feasible paths is the summation of the node cost, the scale-rotation cost and the global cost induced by $\mathbf{c}_g$. Due to the global constraints, searching for the optimal paths in the trellises is a hard problem.

The optimization in Eq.(4) can be relaxed into linear programs and solved via the Simplex Method. However, when the target point number approaches thousands or millions, solving the large scale optimization becomes infeasible. Fortunately, with the LAT constraints, it can be decomposed into a sequence of efficient dynamic programming problems.

2.2. Decomposition into Dynamic Programming

It is the scale, rotation and global constraints, the boxed constraints in Eq.(4), that complicate the optimization. Without them, we can simply select the best match for each model point to optimize the objective. Another observation is that, without the "complex" constraints, the problem turns into an optimization on a tree. The complex constraints introduce links (hyperedges) among all the tree nodes. If we find feasible solutions on the tree, we may use their linear combinations to satisfy the complex constraints and to optimize the objective. The original large scale problem can therefore be decomposed.

We use Dantzig-Wolfe decomposition [20] to break large linear programs into small ones. However, a naive decomposition slows down the optimization and increases the memory usage. We use the special LAT structure and convert our problem into a sequence of efficient dynamic programming on trellises.

We rewrite Eq.(4) in a compact format:

$$\max\left\{c^T x : Ax = r, Bx = e\right\}. \tag{5}$$

Abusing the notation, we use the vector x to indicate the variables in Eq.(4), *i.e.*, x includes X, Y, u_0, v_0, s_0 and the auxiliary variables. We use vector c to denote the objective coefficients in Eq.(4). The complex constraints (boxed) are denoted as $Ax = r$ and other constraints as $Bx = e$.

Initialization: Removing the complex constraints $Ax = r$, we obtain a linear program LP_s. We select the lowest cost target point for each model point to maximize LP_s. The auxiliary variables $u^+_{p,q}, u^-_{p,q}, v^+_{p,q}, v^-_{p,q}, s^+_{p,q}, s^-_{p,q}$ and w are bounded in LP_s. Since their coefficients are negative in the objective function, they all should take their lower bounds. We determine the values of s_0, u_0 and v_0 using the same method.

From the initial solution for LP_s, we can always reset $u^+_{p,q}, u^-_{p,q}, v^+_{p,q}, v^-_{p,q}, s^+_{p,q}, s^-_{p,q}$ and w to obtain x_1 and x_2 so that $\alpha_i^T x_1 = r_i + 1$ and $\alpha_i^T x_2 = r_i - 1$, where α_i is the coefficient vector corresponding to the ith row of A and r_i is the ith element of r. $0.5x_1 + 0.5x_2$ is feasible to both simple and complex constraints. We use x_1 and x_2 to serve as the first two proposals.

Updating tree trellis for new proposals: The goal is to find new proposals satisfying simple constraints $Bx = e$ so that we can combine these proposals to optimize the objective and satisfy the complex constraints $Ax = r$. Assume that we already have $k - 1$ proposals and we introduce the kth proposal x_k so that

$$[F_k]\text{:} \max_{\lambda_1,\ldots,\lambda_k \geq 0} \left\{ \sum_{j=1}^{k} \lambda_j c^T x_j : \sum_{j=1}^{k} \lambda_j \alpha_i^T x_j = r_i, \sum_{j=1}^{k} \lambda_j = 1 \right\}, \tag{6}$$

where λ is the weight vector for the proposals. For the previous $k - 1$ known proposals, we price out the constraints of $F_{(k-1)}$ with shadow prices, which equal the optimal dual variable values. We denote the ith constraint's shadow price as d_i and the unit sum constraint's shadow price as δ. Based on the Simplex Method, by introducing the new proposal, the maximal gain of the objective is $\lambda_k(c^T x_k - \sum_i \alpha_i^T d_i x_k - \delta)$, recalling that shadow price is the change of the objective per unit change of the right hand side (the constant part) of a constraint. To improve the objective, the gain has to be greater than 0. Using Dantzig-Wolfe decomposition [20], instead of randomly searching for a new proposal, we choose the x_k that maximizes the gain: stripping away the λ_k and δ, we maximize $\hat{c}(x) = (c^T - \sum_i \alpha_i^T d_i)x$, *i.e.*,

$$x_k = \arg\max_x \left\{\hat{c}(x) : Bx = e\right\}, \tag{7}$$

where the set of constraints includes the tree constraints and other bound constraints.

Using the LAT structure, we solve Eq.(7) by dynamic programming. Since the constraint matrix B excluding the columns and rows for variables other than X and Y is totally unimodular, and X, Y and other variables are separable, we always have integer solutions for X and Y, and the optimization is equivalent to finding the longest paths on trellises expanded from the tree defined by $\mathcal{N}$. We first update the edge weight based on the Y variable coefficients in $\hat{c}(x)$ (all X variables have been substituted by Y variables) and then we use dynamic programming to implicitly enumerate all the feasible paths. The auxiliary variables and s_0, u_0, v_0 in Eq.(7) take their lower bounds or upper bounds depending on the signs of their coefficients in $\hat{c}(x)$.

Termination condition and looping: We check the optimal objective $\hat{c}(x^*)$ of the dynamic programming and proceed as follows

$$\hat{c}(x^*) \begin{cases} > \delta & \text{add } x^* \text{ as a proposal} \\ \leq \delta & \text{terminate} \end{cases} \tag{8}$$

Therefore, if the gain $\hat{c}(x)$ is greater than δ, we introduce a new proposal, update the trellises and solve a new dynamic program; otherwise, the iteration terminates. The iterative process is finite and terminates with the optimum solution for the relaxed problem [20]. The optimal solution is a linear combination of the proposals.

Obtaining integral solution: The optimal solution for the relaxed solution is fractional and we convert it into an integral solution by solving a mixed-integer program. We solve the mixed-integer program that keeps only the non-zero value target points. We observe that in practice, there are very few non-zero assignment variables in the relaxed solution; therefore, the complexity of this stage is negligible. This scheme ensures that if the optimal target point for each model point is non-zero in the relaxation, then the global optimum is achieved. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Scale and rotation invariant matching on linearly augmented tree (LAT) (Eq.4)

```
Initialize Get feasible solutions $x_1$ and $x_2$ and set $k = 2$.
repeat
    Solve $F_k$ (Eq.6) and $k := k + 1$
    Update trellis weight (Eq.7) and use dynamic programming to solve for $x_k$.
until convergence (Eq.8).
Obtain integral solution.
```

Toy Example: We match a 3-point template in red to the blue target points (Fig.5). The template's basic graph is a tree with $\mathcal{N} = \{(1, 2), (1, 3)\}$. Model point 1 matches target point 1 with cost 10, and the other points with cost 9. Model points 2 and 3 match every target point with cost 10.

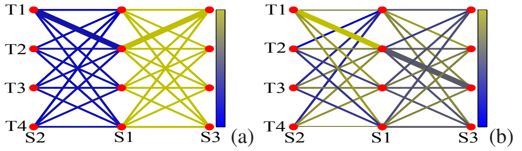
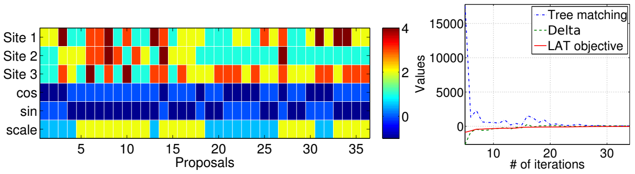


Figure 3. The trellises for tree matching in the first (a) and last (b) stage of iteration. S1, S2 and S3 denote the 3 model points and T1, T2, T3 and T4 denote the 4 target candidates. The warm color indicates high value and cool color indicates low value. The tree optimization finds the matching such that the total edge value is the highest. The thick edges indicate the optimal matching.



(a) (b)

Figure 4. (a): Tree proposals. (b): Convergence of the example.

In this example, we ignore the global term $\mathbf{c}_g$. However, we still include the hyperedge term that involves scale and rotation consistency; it is non-trivial to solve. We construct the proposed model and solve the optimization using Algorithm 1. The solution process involves a sequence of trellis updates. In the following, we show one of the 4 linear programs that achieve the global optimum. Initially, the trellises are as shown in Fig.3(a). The color of the edges illustrates their weight. The assignment on trees can be efficiently computed by dynamic programming: the result is $1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 1$. s_0, u_0, v_0 and auxiliary variables take their lower or upper bounds based on the signs of their objective coefficients. We update the trellises using the proposed scheme so that a new tree solution linearly combined with previous proposals improves the objective. The trellises evolve and at the last stage they are as shown in Fig.3(b). The tree solution is $1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 3$, which is the optimum. Fig.4(a) shows the assignment and rotation-scale parameters in different proposals. Fig.5 shows how the floating-point assignments for model points 1, 2, 3 and the values for the objective, $\hat{c}(x)$ and δ change in the iteration. Fig.4(b) shows the convergence process: the dynamic programming solution approaches δ and the gap scaled by λ equals the improvement of the objective. The proposed method achieves the integral solution; it is the global optimum.

Complexity: The complexity of the proposed method depends on the tree structure matching and the linear programming for fusing the proposals. A standard dynamic programming solution for tree matching is $O(nm^2)$ where n is the number of model points and m is the number of target points. If we can embed the target points on grids, the complexity can be reduced to $O(nm)$. The complexity of F_k is independent of the n and m. It is mostly determined by the number of complex constraints l and the number of proposals k. With the Simplex Method, the average complexity is roughly $l \log(k)$ [10]. Fig.6 illustrates the complexity of the proposed method based on the statistics of a large number of synthetic problems. The proposed method is much more efficient than a direct simplex solution. By embedding target points on grids, the method is able to solve problems with millions of target points.

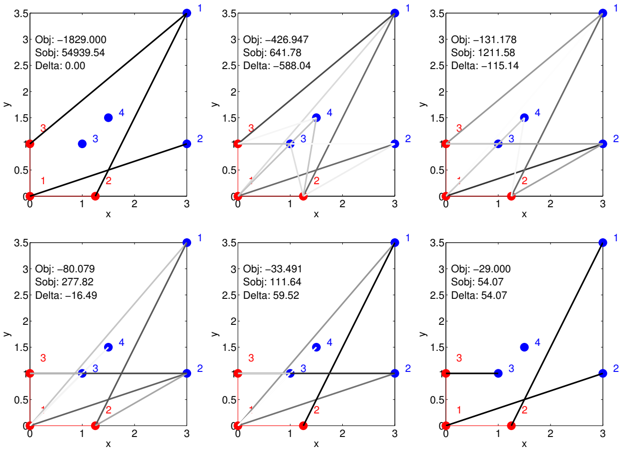
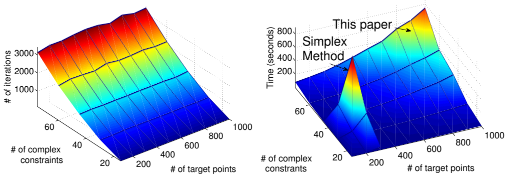


Figure 5. Matching a 3-point template. The proposed algorithm achieves optimum in 34 iterations, in which 6 samples are shown. The gray levels of lines indicate the assignment strength.



(a) (b)

Figure 6. (a): The number of iterations is determined by the number of complex constraints but is independent of the number of target points. (b): The proposed method is more efficient than the Simplex Method.

As a further remark, our approach is different from the dual decomposition [18]. Instead of decomposing the objective to obtain a set of easy problems we decompose the constraints and optimize on a tree.

3. Benchmarking Using Ground Truth Data

We evaluate the performance of the proposed method on synthetic point datasets, which have been widely used in testing matching performance. There are two kinds of test patterns: one is the fish and Chinese character in [7, 9], the second test pattern is random dots. In the experiment, we randomly select 10 model points from the template image to form a template graph. The matching is a challenging task even for clean target images since other points act as clutter points and there are 10 times more clutter points than model points. The target patterns of the first class are smoothly deformed from their templates, while for the second class, the target points are randomly perturbed in 0-20 pixels to simulate deformation. Clutter points are also included in the

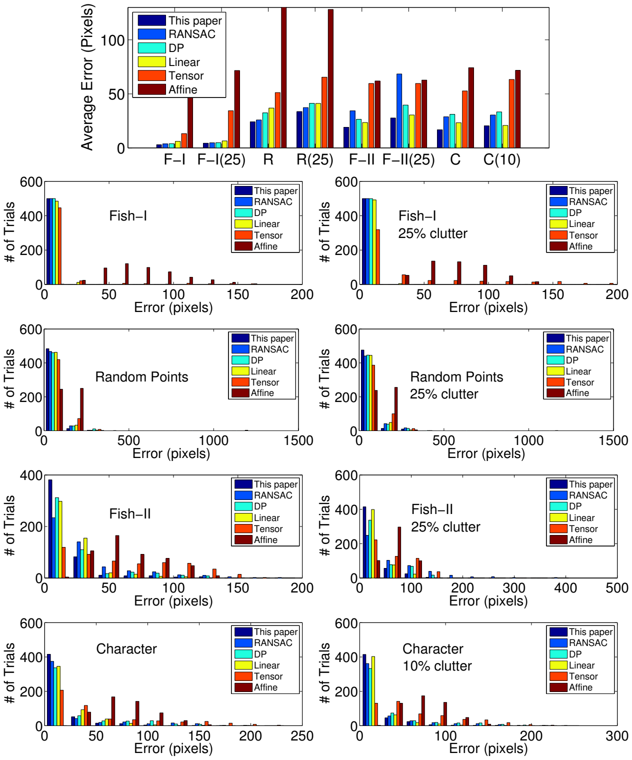
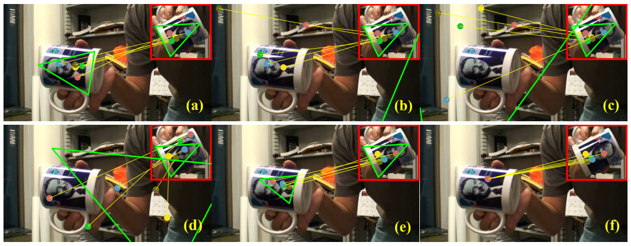


Figure 7. Benchmarking using ground truth data. Row one shows the average matching errors; other rows show the error histograms in different test cases. Good performance is indicated by high values in the lower error ranges and a short tail in the high error ranges.

target patterns. We randomly rotate and scale the patterns to form the final target images. For each pair of template and target candidate points, the matching cost is the lowest chi-square distance between their shape contexts [9] over a set of different scales and rotations.

We compare our method with a tensor method [14], RANSAC [16], linear matching [13] and local affine invariant matching [19]. These matching methods represent the state of the art. A dynamic programming (DP) approach is also compared: by quantizing the scale and rotation angle, each discrete case contains only unary and pairwise constraints and can be solved by dynamic programming. This DP approach is in fact a variant of the Hough Transform. The quantization intervals for the scale and rotation are 0.1 and 5 degrees respectively. The DP method uses the same set of parameters as the proposed method in the objective. In this experiment, the proposed method sets $\mathbf{c}_g$ to 0.

We randomly generate 500 matching problems for each test case and we use the error histograms and average matching errors to quantify the performance of each method. The error histograms record the frequency of matching errors in different ranges. As shown in Fig.7, the proposed method has the lowest average matching errors in all the tests. The error histograms indicate that it also yields matching results in lower error ranges than all the competing methods. Interestingly, our method outperforms the discretized “exhaustive” search method (DP) under the same parameter setting: search in continuous domain helps.



This paper	DP	RANSAC	Tensor	Linear	Affine
90%	63%	48%	5%	88%	8%

Figure 8. Matching 429-frame cup sequence. The sample result shows how the proposed method (a) improves the result over DP (b), RANSAC (c), tensor (d), linear [13] (e) and local affine [19] (f) method. The table summarizes the detection rates for the video.

4. Evaluations on Real Images and Videos

We evaluate the proposed method on a variety of videos on different features including SIFT [1], image patches, and unreliable regions. With a randomly selected template in each experiment, we use the proposed method to match the target object in cluttered videos. We also compare with the five competing methods in the synthetic data experiments.

Matching SIFT: We first match SIFT features and test whether the proposed method still has an advantage over other competing methods. In this experiment, the proposed method uses a global affine constraint, *i.e.*, the target of the root model point is constrained to be close to a point that is the linear combination of all the other target points. The coefficients are determined from the layout of the model points. We select the top five model points with the lowest best and second-best matching candidate cost ratio [1] to form the model graph. To simulate hard situations, for each model point, we corrupt the best matching candidate cost and set it equal to the second best matching cost. Fig.8 shows the comparison results for the 429-frame cup sequence. Due to the complexity of the tensor method [14] we have to use a higher threshold to reduce the number of SIFT features. We use visual inspection to quantify the detection rate: if all the model points match correctly, we have a correct detection. In this experiment, the proposed method achieves a 90% detection rate, which is the highest. It also has a complexity similar to the efficient linear [13] and affine [19] methods.

Matching image patches: We test the reliability of the algorithm when using non-distinctive features. We use edge pixels and image patches for matching. The target candidate points include all the edge pixels in the target image. The local matching cost is the lowest cost of image patch matching at different rotations; because the image patch is small it is roughly scale invariant. We ran the algorithm on a 200-frame sequence of a person running (Fig. 9). With such

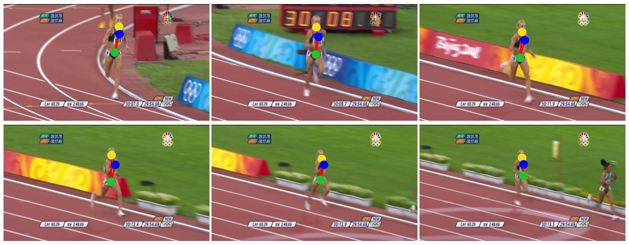
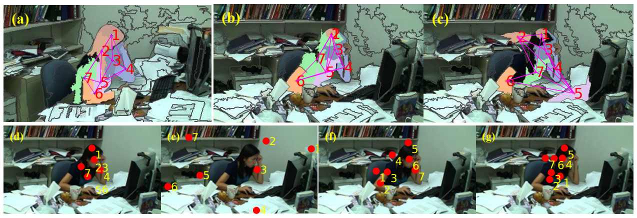


Figure 9. Matching using edge pixels on a 200-frame sequence of a person running. The detection rate is 97%. Average running time is 0.7s per frame.



This paper	DP	RANSAC	Tensor	Linear	Affine
91%	73%	43%	11%	74%	14%

Figure 10. Matching using superpixels on the 264-frame `girl` sequence. The `girl` sequence has strong background clutter and unstable superpixels. **(a)** is the template. The sample result of **(b)** the proposed method is superior to **(c)** DP, **(d)** RANSAC, **(e)** tensor [14], **(f)** linear [13] and **(g)** local affine [19] method. The table summarizes the detection rates in the whole sequence.

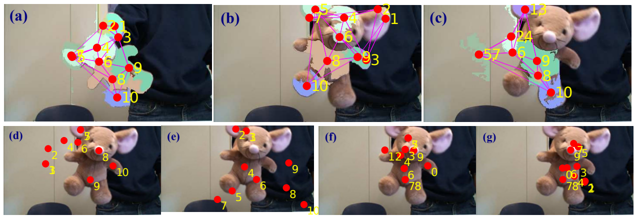
rough features, the proposed method still reliably matches the target with a 97% detection rate. It is also efficient, the typical running time of the optimization is $0.7s$ per frame.

Matching unreliable regions: We demonstrate the ability of LAT method on using cheap features to match unreliable regions. This setting enables fast object matching. However, drastic region variation, non-distinctive features and strong clutter also pose a great challenge. Previous techniques [21, 22] rely on strong features that are expensive to compute or hierarchical region matching that has high complexity.

We over-segment images into superpixels using [15]. The model points and target points are superpixel weight centers in the template and target images. We use two weak features: the average chromaticity of each superpixel and a shape feature defined as the ratio between the two eigenvalues of the xy coordinate covariance matrix.

We use a linear global constraint $\mathbf{c}_g$ to enforce the total area consistency. Even though superpixels may change size randomly, the overall object size equals the template size scaled by a factor. It is defined as $\mathbf{c}_g = |\text{tr}(R^T X) - s_0^2 t_a|$ where R is the target area matrix and t_a is the template area. $\mathbf{c}_g$ can be linearized by letting $g_c = \text{tr}(R^T X) - s_0^2 t_a - w^+ + w^-$ and $g_o = \phi(w^+ + w^-)$ in Eq.(4), where ϕ is a constant coefficient. s_0^2 can be further changed to s_0 and we squared the scaling matrices S in Eq.(4).

We compare our method with other five competing methods on the challenging `girl` (Fig. 10) and `mouse` (Fig. 11) sequences. Results for the `girl` and `mouse` sequences are shown in Fig. 10 and Fig. 11 respectively.



This paper	DP	RANSAC	Tensor	Linear	Affine
90%	86%	20%	16%	34%	1%

Figure 11. Matching using superpixels on the 551-frame `mouse` sequence. The superpixels are unstable and change drastically from frame to frame due to the subtle color difference on the object and shading changes when the object rotates. The color of the mouse is also similar to the superpixels on the wall. **(a)** is the template. In the target frame, **(b)** the proposed method succeeds, but competing methods **(c)** DP, **(d)** RANSAC, **(e)** tensor [14], **(f)** linear matching [13], and **(g)** local affine matching [19] fail. The table summarizes the detection rates in the video.

	Mouse	Girl	Dance-I	Gym	Dance-II	Skate
Num. Frames	551	264	713	386	792	472
Rate	90%	91%	98%	90%	94%	91%
Avg. Time (s)	0.78	0.42	0.03	0.02	0.07	0.05

Figure 13. The average running time for optimization in one frame is measured on a 2.8GHZ machine.

We also apply the proposed method on four other challenging video sequences downloaded from YouTube (Fig. 12). The detection rates and average running time of the proposed method when applied to the six different videos are listed in Fig.13. The detection rate is determined by visual inspection. Due to unreliable segmentation, we check the global detection result, *i.e.*, we transform the model points using an affine transformation that is based on the region center correspondence and examine whether the matching is correct. The proposed method robustly matches the target in these sequences with a detection rate from 90% to 98%. It is also efficient: optimization in a frame takes less than a second for a target image with hundreds of superpixels. The proposed method achieves significantly better results than all the competing methods. It has similar complexity to the linear, affine and RANSAC methods and many times faster than the discretized parameter DP and tensor methods.

5. Conclusion

We propose a novel formulation for scale and rotation invariant matching using Linearly Augmented Tree (LAT) constraints. Due to the LAT's special structure, we can solve the relaxed matching problem efficiently with a sequence of dynamic programming. Our experimental results on ground truth data and real images demonstrate that the proposed method is more reliable than previous methods. The experiments confirm that our method maintains high performance even on very weak features such as unreliable regions. We believe our method is generic and can be

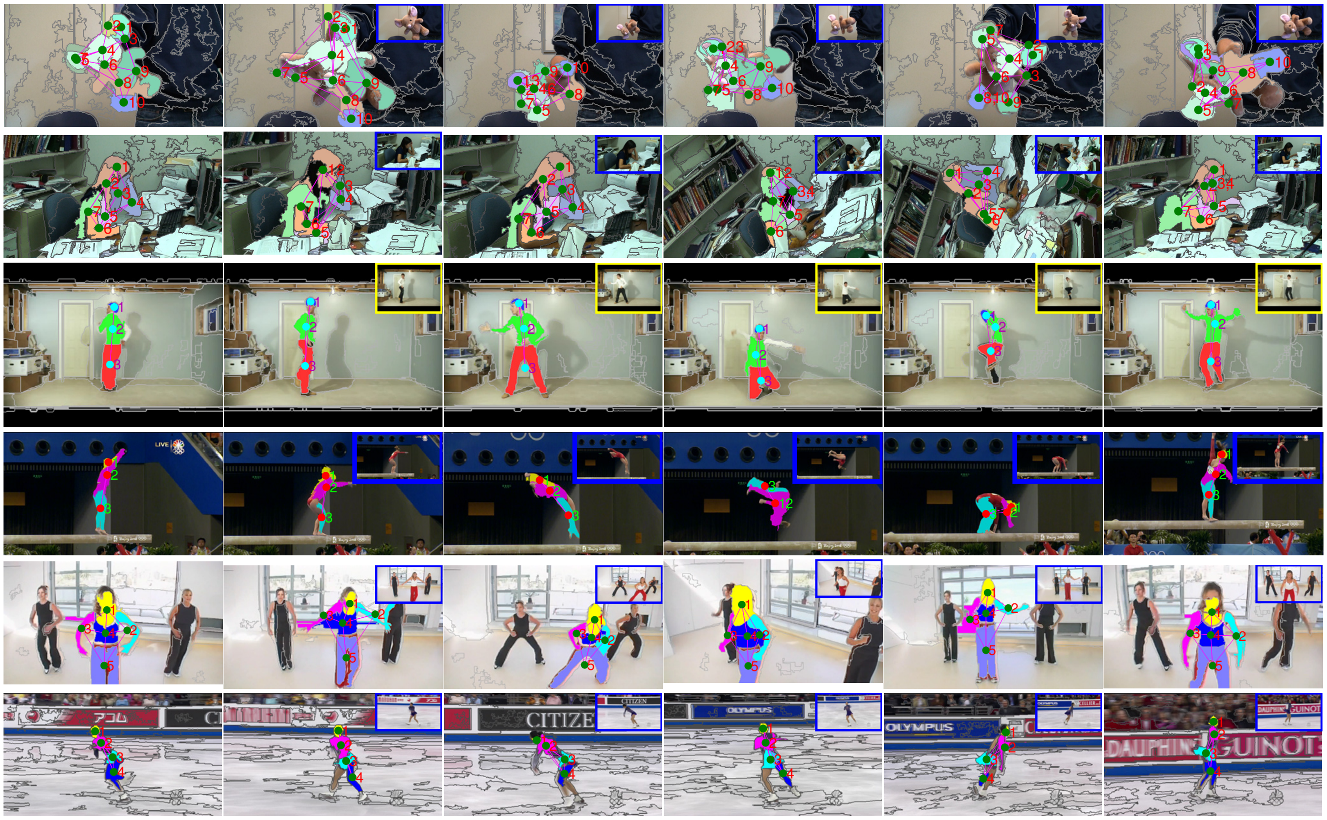


Figure 12. Results of the proposed method on real world data. The first column shows the randomly selected templates. **Row 1:** `mouse` (551 frames) has drastic superpixel changes and foreground is similar to the background. **Row 2:** `girl` (264 frames) has strong clutter. **Row 3-6:** `dance-I` (713 frames), `gym` (386 frames), `dance-II` (792 frames), and `skate` (472 frames) have complex articulated movement, large deformation and self-occlusion. `Dance-II` also includes a few human subjects with similar shapes and colors that form challenging structured clutter.

adapted to solve problems in other domains including pose estimation and object tracking.

Acknowledgment: This work was supported in part through US National Science Foundation grants 0713168 and 1018641.

References

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

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

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

[4] R.O. Duda, P.E. Hart, "Use of the hough transform to detect lines and curves in pictures", Comm. of the ACM, vol.15, no.1, pp.11-15.

[5] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" TPAMI, 26(2), 2004.

[6] 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.

[7] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching", CVPR 2000.

[8] A.C. Berg, T.L. Berg and J. Malik, "Shape matching and object recognition using low distortion correspondence", CVPR 2005.

[9] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts", TPAMI, vol.24, no.4, 2002.

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

[11] N. Komodakis and G. Tziritas, "Approximate labeling via graph-cuts based on linear programming". TPAMI, 29(8), 2007.

[12] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization", TPAMI, 28(10), 2006.

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

[14] O. Duchenne, F. Bach, I. Kweon and J. Ponce, "Tensor-based algorithm for high-order graph matching", CVPR 2009.

[15] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation", IJCV, vol 59, no. 2, 2004.

[16] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, vol.24, no.5, pp.381-395.

[17] L. Torresani, V. Kolmogorov and C. Rother, "Feature correspondence via graph matching: models and global optimization export", ECCV 2008.

[18] N. Komodakis, N. Paragios and G. Tziritas, "MRF energy minimization and beyond via dual decomposition", TPAMI, vol.33, no.3, pp.531-552, 2011.

[19] H. Li, E. Kim, X. Huang and L. He, "Object matching with a locally affine-invariant constraints", CVPR 2010.

[20] G.B. Dantzig and P. Wolfe, "Decomposition principle for linear programs", Operations Research, vol.8, no.1, 1960.

[21] V. Hedau, H. Arora, N. Ahuja, "Matching images under unstable segmentations", CVPR 2008.

[22] S. Todorovic and M.C. Nechyba, "Dynamic trees for unsupervised segmentation and matching of image regions", TPAMI, 27(11), 2005.