Linear Programming Matching and Appearance-Adaptive Object Tracking

Hao Jiang, Mark S. Drew, and Ze-Nian Li

School of Computing Science, Simon Fraser University,
Burnaby BC, Canada V5A 1S6

Abstract. In this paper, we present a novel successive relaxation linear programming scheme for solving the important class of consistent labeling problems for which an L_1 metric is involved. The unique feature of the proposed scheme is that we use a much smaller set of basis labels to represent the label space. In a coarse to fine manner, the approximation improves during iteration. The proposed scheme behaves very differently from other methods in which the label space is kept constant in the solution process, and is well suited for very large label set matching problems. Based on the proposed matching scheme, we develop a robust multi-template tracking method. We also increase the efficiency of the template searching by a Markov model. The proposed tracking method uses a small number of graph templates and is able to deal with cases in which objects change appearance drastically due to change of aspect or object deformation.

1 Introduction

Matching is one of the most important tasks in computer vision, and is key for stereo, motion estimation, 3D object reconstruction, tracking, and object recognition. Matching can be mathematically formulated as a consistent labeling problem, in which we need to assign labels to sites such that a predefined energy function is minimized. For consistent matching problems, labels are usually defined in a metric space and therefore the distances of labels can be measured. Although simple in concept, consistent labeling is NP-hard in general. For some special cases, for instance, when sites have linear or tree order, dynamic programming can be applied to solve labeling problem in polynomial time. Another special case is when labels for each site have linear order, and the metric defined in the label space is convex. In this case, polynomial-time max-flow schemes [1] can be applied. Other searching schemes, e.g. branch and bound schemes [2], whose worst and average complexities are exponential, have also been applied to medium sized matching problems. For the general case in image matching, approximation algorithms are preferred. Relaxation labeling (RL) [3] was one of the earliest methods for solving labeling problems, and has had a great deal of influence on later matching schemes. RL uses local search, and therefore relies on a good initialization. ICM — Iterative Conditional Modes [4] — another widely applied method for solving labeling problems, is greedy and has been found to be easily trapped in a local minimum. In recent years, graph cut (GC) [5] and belief propagation (BP) [6][7][8] have become standard methods for solving consistent labeling problems. Graph cut based methods have been successfully applied in matching problems such as stereo [9] and

A. Rangarajan et al. (Eds.): EMMCVPR 2005, LNCS 3757, pp. 203–219, 2005.

motion [10]. Loopy belief propagation has also been widely applied in stereo [11] and object matching [12]. GC and BP are more robust than traditional labeling schemes and are also found to be faster than stochastic annealing based methods. But GC and BP are still very complex for large scale problems that involve a large number of labels. Besides GC and BP, many other schemes have been presented, such as spectrum graph theory based methods [13]. Although intensively studied, the large scale matching problem is still an unsolved problem. In this paper, we present methods based on linear programming to solve the class of consistent labeling problems with L_1 regularity terms. For this class of problems, an efficient LP method can be formulated, which is found to be faster and more robust in solving large label set problems than the standard methods such as BP and GC.

The work most related to the proposed scheme is the mathematical programming schemes, which have received much interest in formulating and solving labeling problems. The early RL schemes belong to this class. One of the major challenges of a labeling algorithm is to overcome the problem of local minima in the optimization process. Different schemes have been proposed. Deterministic annealing schemes [14][15] have been successfully applied for matching point sets and graphs. Quadratic programming schemes [16] and most recently semidefinite programming schemes [17] have been proposed for matching problems. Up to now, these schemes could only be applied to small scale problems. On the contrary, because of its efficiency, Linear Programming has been applied in many vision problems, such as estimating motion of rigid scenes [18]. A linear programming formulation [19] has been presented for uniform labeling problems and for approximating general problems by tree metrics. Another general LP scheme, studied in [20], is quite similar to the linear relaxation labeling formulation [3]. This LP formulation is found to be only applicable to small problems because of the large number of constraints and variables involved.

We present a linear programming relaxation scheme for L_1-regularity consistent labeling problems, and we study an efficient successive relaxation scheme to solve the optimization problem. Different from other methods, the proposed scheme uses a much smaller number of basis labels to represent the matching space. This is one key component of the method to speed up the algorithm. In our scheme, basis labels correspond to the coordinates of the 3D lower convex hull of the matching cost function for each site. We propose a successive relaxation scheme to increase the accuracy of the approximation iteratively. During the iteration, we shrink the trust region for each site and locate the new trust region based on the solution of the previous relaxation solution, but reconvexify the matching cost based on the *original* cost function. This process continues until the trust region becomes small. Since the convexification process eliminates many false local minima in the earlier stages of the solution process, the proposed scheme is able to find a good approximated solution quickly. Iteratively, the successive relaxation process refines the labeling result.

Based on the proposed matching scheme, we propose a robust template tracking scheme. In object tracking, handling drastic shape and appearance changes of objects, due to severe viewpoint and aspect changes and object shape deformation, is a difficult problem. Two classes of methods have been used to try to solve the problem. The first class uses features resistant to object aspect and deformation, such as the color histogram. The mean-shift based method [21] belongs to this class. Methods relying on invariant features cannot handle large appearance changes and usually do not produce an accurate correspondence in tracking. The second class of method is most correlated with the method studied in this paper, which is appearance-based and requires a set of key templates representing an object's appearance. Black and Jepson [22] propose a PCA based approach to select and represent templates. Researchers also study methods to learn the templates on-line [23]. In this paper, we present a scheme which does not rely on a complex training process. The system requires a very small number of exemplars. Graph templates are then generated and used in tracking. Apart from tracking objects of fixed appearance, the proposed scheme is also able to track an object that changes appearance dramatically by selecting the best template in matching. To further increase efficiency, the templates are organized into a digraph, so that one template can only be replaced by its scaled or rotated version or its neighbors with the same scale and rotation settings. The tracking process is then equivalent to finding a node transition sequence in the given digraph. Experiments show very promising results for tracking objects in cluttered backgrounds.

2 Matching by Linear Programming

The matching problem can be stated generally as the following consistent labeling problem,

$$\min_f \sum_{s \in S} c(s, f_s) + \sum_{\{p,q\} \in \mathcal{N}} \lambda_{p,q} d(f_p, f_q)$$

where $c(s, f_s)$ is the cost of assigning label f_s to site s; $d(f_p, f_q)$ is the distance between the labels assigned to neighboring sites p and q; S is a finite set of sites; $\mathcal{N}$ is the set of non-ordered neighboring site pairs; $\lambda_{p,q}$ are smoothing coefficients. For computer vision problems, labels can be discrete or continuous. When the labels are discrete, we denote a problem as a discrete labeling problem, and otherwise as a continuous labeling problem. For a discrete labeling problem, we can interpolate the cost $c(s, f_s)$ for each site piecewise-linearly such that $c(s, f_s)$ becomes a surface, and allow f_s to take values in the convex hull supported by the discrete labels: we thus obtain the *continuous extension* of a discrete problem. Continuous labeling problems such as motion estimation can be well approximated by such a continuous extension of a discrete system. In the following discussions, without loss of generality we assume both the set $\mathcal{S}$ and the label set $\mathcal{L}_s$ to be discrete. In this paper, we focus on that subset of consistent labeling problems such that $d(f_p, f_q) = ||f_p - f_q||$; $|| \cdot ||$ is the L_1 norm and f are vectors defined in the L_1 metric space. When f degenerate into scalars, a maximum-flow scheme can be used to solve the problem. For problems with label dimensionality greater than 1, the problem becomes much more complex. We rewrite the formulation as follows, with boldface symbols for site $\mathbf{s}$ and label $\mathbf{f}$ to emphasize that they are vectors:

$$\min_{\mathbf{f}} \sum_{\mathbf{s} \in S} c(\mathbf{s}, \mathbf{f_s}) + \sum_{\{\mathbf{p},\mathbf{q}\} \in N} \lambda_{\mathbf{p},\mathbf{q}} ||\mathbf{f_p} - \mathbf{f_q}||$$

In the following, we assume the $\mathbf{f}$ vectors are 2D. The methods proposed below can be easily extended to cases where the labels have higher dimensionality. To simplify notation, $c(\mathbf{s}, \mathbf{f_s})$ is also used to represent the continuous extension matching cost surfaces.

2.1 Approximation by Linear Programming

The above energy optimization problem is nonlinear and usually non-convex, which makes it difficult to solve in this original form without a good initialization process. We now show how to approximate the problem by a linear programming formulation via linear approximation and variable relaxation, as outlined in [24, 25] by Jiang et al. To linearize the first term, the following scheme is applied. A basis $\mathcal{B}_\mathbf{s}$ is selected for the labels for each site $\mathbf{s}$. Then the label $\mathbf{f_s}$ can be represented as a linear combination of the label basis as $\mathbf{f_s} = \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \mathbf{j}$, where $\xi_{\mathbf{s},\mathbf{j}}$ are real-valued weighting coefficients. The labeling cost of $\mathbf{f_s}$ can then be approximated by the linear combination of the cost of the basis labeling costs $c(\mathbf{s}, \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \mathbf{j}) \simeq \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot c(\mathbf{s}, \mathbf{j})$. We also further set constraints $\xi_{\mathbf{s},\mathbf{j}} \geq 0$ and $\sum_{\mathbf{j}\in\mathcal{B}_s} \xi_{\mathbf{s},\mathbf{j}} = 1$ for each site $\mathbf{s}$, so as to constrain the space spanned by the basis to the convex hull of the basis labels. Clearly, if $\xi_{\mathbf{s},\mathbf{j}}$ are constrained to be 1 or 0, and the basis contains all the labels, i.e. $\mathcal{B}_\mathbf{s} = \mathcal{L}_\mathbf{s}$, the above representation becomes exact. Note that $\mathbf{f_s}$ are *not* constrained to the basis labels, but can be any convex combination. To linearize the regularity terms in the nonlinear formulation we can represent a free variable by the difference of two nonnegative auxiliary variables and introduce the summation of the auxiliary variables into the objective function. If the problem is properly formulated, when the linear programming problem is optimized the summation will approach the absolute value of the free variable.

Based on this linearization process, a linear programming approximation of the problem can be stated as

$$\min \sum_{\mathbf{s}\in S} \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} c(\mathbf{s}, \mathbf{j}) \cdot \xi_{\mathbf{s},\mathbf{j}} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2} (f^+_{\mathbf{p},\mathbf{q},m} + f^-_{\mathbf{p},\mathbf{q},m})$$

$$s.t. \qquad \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} = 1, \forall \mathbf{s} \in S$$

$$\sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \phi_m(\mathbf{j}) = f_{\mathbf{s},m}, \ \forall \mathbf{s} \in S, \ m = 1, 2$$

$$f_{\mathbf{p},m} - f_{\mathbf{q},m} = f^+_{\mathbf{p},\mathbf{q},m} - f^-_{\mathbf{p},\mathbf{q},m}, \forall \{\mathbf{p}, \mathbf{q}\} \in \mathcal{N}, \ m = 1, 2$$

$$\xi_{\mathbf{s},\mathbf{j}}, \ f^+_{\mathbf{p},\mathbf{q},m}, f^-_{\mathbf{p},\mathbf{q},m} \geq 0$$

where $\mathbf{f_s} = (f_{\mathbf{s},1}, f_{\mathbf{s},2})$ and function ϕ_m returns the mth component of its argument. It is not difficult to show that the linear programming formulation is equivalent to the general nonlinear formulation if the linearization assumptions hold. In general situations, the linear programming formulation is an approximation of the original nonlinear optimization problem.

Property 1: *If* $\mathcal{B}_\mathbf{s} = \mathcal{L}_\mathbf{s}$, and *the cost function of its continuous extension* $c(\mathbf{s}, \mathbf{j})$ *is convex,* $\forall \mathbf{s} \in S$ *, the LP exactly solves the continuous extension of the discrete labeling problem.* $\mathcal{L}_\mathbf{s}$ is the label set of $\mathbf{s}$.

Proof: We just need to show that when LP is optimized, the configuration $\{\mathbf{f}_\mathbf{s}^* = \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}}^* \cdot \mathbf{j}\}$ also solves the continuous extension of the nonlinear problem. Since $c(\mathbf{s},\mathbf{j})$ is convex, $\sum_{\mathbf{j}\in\mathcal{L}_\mathbf{s}} c(\mathbf{s},\mathbf{j})\xi_{\mathbf{s},\mathbf{j}}^* \geq c(\mathbf{s},\mathbf{f}_\mathbf{s}^*)$. And, when the LP is minimized we have $\sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2}(f_{\mathbf{p},\mathbf{q},m}^+ + f_{\mathbf{p},\mathbf{q},m}^-) = \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f}_\mathbf{p}^* - \mathbf{f}_\mathbf{q}^*||$. Therefore

$$\min \sum_{\mathbf{s}\in S,\mathbf{j}\in\mathcal{L}_s} c(\mathbf{s},\mathbf{j})\xi_{\mathbf{s},\mathbf{j}} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2}(f_{\mathbf{p},\mathbf{q},m}^+ + f_{\mathbf{p},\mathbf{q},m}^-)$$
$$\geq \sum_{\mathbf{s}\in S} c(\mathbf{s},f_\mathbf{s}^*) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f}_\mathbf{p}^* - \mathbf{f}_\mathbf{q}^*||$$

According to the definition of continuous extension, $\mathbf{f}_\mathbf{s}^*$ are feasible solutions of the continuous extension of the non-linear problem. Therefore the optimum of the linear programming problem is not less than the optimum of the continuous extension of the nonlinear problem. On the other hand, it is easy to construct a feasible solution of LP that achieves the minimum of the continuous extension of the nonlinear problem. The property follows.

In practice, the cost function $c(\mathbf{s},\mathbf{j})$ is usually highly non-convex for each site $\mathbf{s}$. In this situation, the proposed linear programming model approximates the original non-convex problem by a convex programming problem.

Property 2: *For general cost function $c(\mathbf{s},\mathbf{j})$, and if $\mathcal{B}_\mathbf{s} = \mathcal{L}_\mathbf{s}$, $\forall \mathbf{s} \in S$, the linear programming formulation solves the continuous extension of the reformulated discrete labeling problem, with $c(\mathbf{s},\mathbf{j})$ replaced by its lower convex hull for each site $\mathbf{s}$.*

The proof is similar to that of Property 1, by replacing $c(\mathbf{s},\mathbf{j})$ in the non-linear function with its lower convex hull. An example for lower convex hull and the coordinates of the lower convex hull vertices are illustrated in Fig. 1. Fig. 2(a) shows the convexification effect introduced by LP relaxation.

Property 3: *For general cost function $c(\mathbf{s},\mathbf{j})$, the most compact basis set $\mathcal{B}_\mathbf{s}$ contains the vertex coordinates of the lower convex hull of $c(\mathbf{s},\mathbf{j})$, $\forall \mathbf{s} \in S$.*

By Property 3, there is no need to include all the labeling assignment costs in the optimization: we only need to include those corresponding to the basis labels. This is one of the key steps to speed up the algorithm.

Property 4: *If the convex hull* of *the cost function* $c(\mathbf{s},\mathbf{j})$ is strictly convex*, nonzero weighting basis labels must be "adjacent".*

Proof: Here "adjacent" means the convex hull of the nonzero weighting basis labels cannot contain other basis labels. Assume this does not hold for a site $\mathbf{s}$, and the nonzero weighting basis labels are $\mathbf{j}_k$, $k = 1..K$. Then, there is a basis label $\mathbf{j}_r$ located inside the convex hull of $\mathbf{j}_k$, $k = 1..K$. Thus, $\exists \alpha_k$ such that $\mathbf{j}_r = \sum_{k=1}^{K} \alpha_k \mathbf{j}_k$ and $\sum_{k=1}^{K} \alpha_k{=}1$, $\alpha_k \geq 0$. According to *Karush-Kuhn-Tucker Condition* (*KKTC*), there exists $\lambda_1, \lambda_2, \lambda_3$ and $\mu_\mathbf{j}$ such that

$c(\mathbf{s},\mathbf{j}) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}) + \lambda_3\phi_2(\mathbf{j}) - \mu_\mathbf{j} = 0$ and $\xi_{\mathbf{s},\mathbf{j}}\mu_\mathbf{j} = 0, \mu_\mathbf{j} \geq 0, \forall \mathbf{j} \in \mathcal{B}_\mathbf{s}$

Therefore we have,

$c(\mathbf{s},\mathbf{j}_k) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}_k) + \lambda_3\phi_2(\mathbf{j}_k) = 0, k = 1..K$

$c(\mathbf{s},\mathbf{j}_r) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}_r) + \lambda_3\phi_2(\mathbf{j}_r) \geq 0$

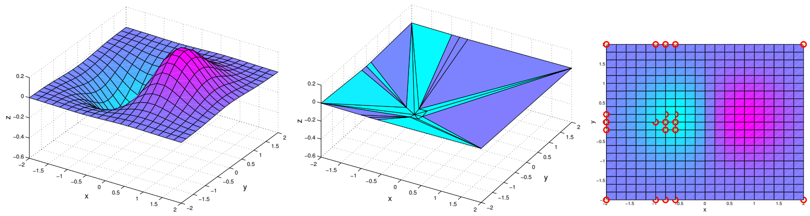


Fig. 1. Lower convex hull. Left: The surface; Middle: Lower convex hull facets; Right: Coordinates of the lower convex hull vertices.

On the other hand,

$$\begin{aligned}
&c(\mathbf{s}, \mathbf{j}_r) + \lambda_1 + \lambda_2 \phi_1(\mathbf{j}_r) + \lambda_3 \phi_2(\mathbf{j}_r) \\
&= c(\mathbf{s}, \textstyle\sum_{k=1}^{K} \alpha_k \mathbf{j}_k) + \lambda_1 + \lambda_2 \phi_1(\sum_{k=1}^{K} \alpha_k \mathbf{j}_k) + \lambda_3 \phi_2(\sum_{k=1}^{K} \alpha_k \mathbf{j}_k) \\
&< \textstyle\sum_{k=1}^{K} \alpha_k c(\mathbf{s}, \mathbf{j}_k) + \lambda_1 + \lambda_2 \sum_{k=1}^{K} \alpha_k \phi_1(\mathbf{j}_k) + \lambda_3 \sum_{k=1}^{K} \alpha_k \phi_2(\mathbf{j}_k) = 0
\end{aligned}$$

which contradicts the *KKTC*. The property follows.

It is not difficult to show that any basic feasible solution of the linear program has at most 3 basic variables from ξ of each site. Therefore, when using the simplex method, there will be at most 3 nonzero-weight basis labels for each site. After convexification, the original non-convex optimization problem turns into a convex problem and an efficient linear programming method can be used to yield a global optimal solution for the approximation problem. Note that, although this is a convex problem, standard local optimization schemes are found to work poorly because of quantization noise and large flat areas in the convexified objective function.

Approximating the matching cost by its lower convex hull is intuitively attractive since in the ideal case, the true matching will have the lowest matching cost and thus the optimization becomes exact in this case. In real applications, several target points may have equal matching cost and, even worse, some incorrect matching may have lower cost. In this case, because of the convexification process, in a one-step relaxation, the resulting fractional (continuous) labeling could be far from the true solution, as shown

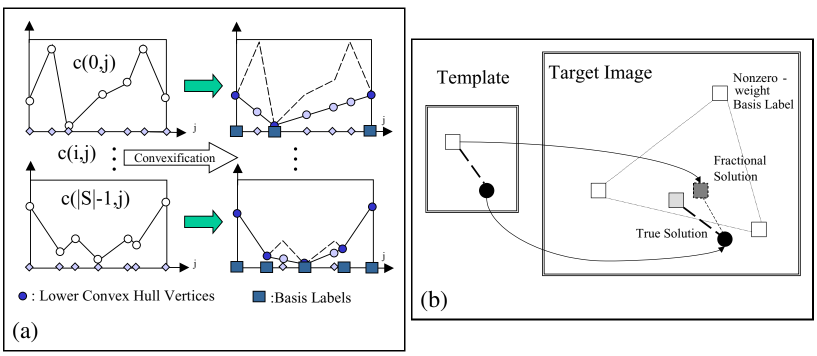


Fig. 2. (a): The convexification process introduced by LP relaxation. (b): An example when a single LP relaxation produces a fractional labeling.

in the Fig 2(b). In this simple example, there are 2 sites in the source image and we construct a simple 2-node graph template. There are 5 target points in the target image. The labels are displacement vectors. We assume that a white square will match a white square with zero cost. And the circles will match with zero cost. Matching between different shape points have large matching cost. The light gray square is in fact the true target for the white one in the source image, but the match cost is a very small positive number because of noisy measurement. By solving the LP relaxation problem, we get a fractional solution, as illustrated in Fig 2(b), that has zero cost for the LP's objective function but is not the true solution. Adjusting the smoothing parameter will not help because we already achieve the minimal (zero) cost. A traditional rounding scheme will try to round ξ into 0 and 1. Unfortunately, the rounding will drive the solution even farther from the true solution, in which the square template node will match one of the white squares in the target image. Intuitively, we can shrink the searching region for each site based on the current LP solution, and do a further search by solving a new LP problem in the smaller trust region. Clearly, if the trust region shrinks slowly, we will find the true optimal solution. In the following section, we expand this idea and propose a successive convexification scheme to improve the approximation iteratively.

3 Successive Relaxation Method

Here we propose a successive convexification linear programming method to solve the non-linear optimization problem, in which we construct linear programming problems recursively based on the previous searching result and gradually shrink the matching trust region systematically.

Assume $\mathcal{B}_{\mathbf{s}}^n$ is the basis label set for site $\mathbf{s}$ at stage n linear programming. The trust region $\mathcal{U}_{\mathbf{s}}^n$ of site $\mathbf{s}$ is determined by the previous relaxation solution $\mathbf{f}_{\mathbf{s}}^{n-1} = (f_{\mathbf{s},1}^{n-1}, f_{\mathbf{s},2}^{n-1})$ and a trust region diameter d_n. We define $\mathcal{Q}_{\mathbf{s}}^n = \mathcal{L}_{\mathbf{s}} \cap \mathcal{U}_{\mathbf{s}}^n$. $\mathcal{B}_{\mathbf{s}}^n$ is specified by $\mathcal{B}_{\mathbf{s}}^n = \{$the vertex coordinates of the lower convex hull of $\{c(\mathbf{s},\mathbf{j}), \forall \mathbf{j} \in \mathcal{Q}_{\mathbf{s}}^n\}\}$, where $c(\mathbf{s},\mathbf{j})$ is the cost of assigning label $\mathbf{j}$ to site $\mathbf{s}$.

Algorithm 1. *Successive Convexification Linear Programming*

1. *Set* $n = 0$*; Set initial diameter* $= d_0$;
2. *FOREACH*$(\mathbf{s} \in S)$
3. $\{$ *Calculate the cost function* $\{c(\mathbf{s},\mathbf{j}), \forall \mathbf{j} \in \mathcal{Q}_{\mathbf{s}}^0\}$*;*
4. *Convexify* $\{c(\mathbf{s},\mathbf{j})\}$ *and find basis* $\mathcal{B}_{\mathbf{s}}^0$*;* $\}$
5. *Construct and solve* $\mathcal{LP}_0$;
6. *WHILE* $(d_n \geq 1)$
7. $\{$ *n ←n+1;*
8. $d_n = d_{n-1} - \delta_n$;
9. *FOREACH*$(\mathbf{s} \in S)$
10. $\{$ *IF* $(\mathcal{Q}_{\mathbf{s}}^n$ *is empty*$)$ $\mathcal{Q}_{\mathbf{s}}^n = \mathcal{Q}_{\mathbf{s}}^{n-1}; \mathcal{U}_{\mathbf{s}}^n = \mathcal{U}_{\mathbf{s}}^{n-1}$;
11. *ELSE update* $\mathcal{U}_{\mathbf{s}}^n$, $\mathcal{Q}_{\mathbf{s}}^n$*;*
12. *Reconvexify* $\{c(\mathbf{s},\mathbf{j})\}$ *and relocate basis* $\mathcal{B}_{\mathbf{s}}^n$*;* $\}$
13. *Construct and solve* $\mathcal{LP}_n$; $\}$
14. *Output* $\mathbf{f}_{\mathbf{s}}^*, \forall \mathbf{s} \in S$*;*

It is not difficult to verify that the necessary condition for successive LP converging to the global minimum is that $\mathcal{LP}_n \leq \mathcal{E}^*$, where $\mathcal{E}^*$ is the global minimum of the non-linear problem. Since the global minimum of the function is unknown, we estimate an upper bound $\mathcal{E}^+$of $\mathcal{E}^*$ in the iterative process. The configuration of targets that achieves the upper bound $\mathcal{E}^+$ is composed of *anchors* — an anchor is defined as the control point of the trust region for one site in the next iteration. A simple scheme is to select anchors as the solution of the previous LP, $\mathbf{r_s} = \mathbf{f_s}^{(n-1)}$. Unfortunately, in the worse case, this simple scheme has solutions whose objective function is arbitrarily far from the optimum. In fact, the fractional solution could be far away from the discrete label site. To solve this problem, we use a deterministic rounding process: we check the discrete labels and select the anchor that minimizes the non-linear objective function, given the configuration of fractional matching labels defined by the solution of the current stage. This step is similar to a single iteration of an ICM algorithm. In this step, we project a fractional solution into the discrete space. We call this new rounding selection scheme a *consistent rounding* process. Let $\mathbf{r_s}$ be the anchor; $\mathbf{m_s}$ be the global optimal solution; and $\mathbf{f_s}$ be the fractional labeling solution of LP.

Proposition 1: The energy with consistent rounding is bounded above by $3\varepsilon_{opt} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}(||\mathbf{m_p}-\mathbf{f_p}|| + ||\mathbf{m_q}-\mathbf{f_q}||)$, where ε_{opt} is the optimal energy.

Except for $\mathcal{LP}_1$, we further require that new anchors have energy not greater than the previous estimation: the anchors are updated only if new ones have smaller energy. The anchors are kept inside the new trust region for each site. The objective function for $\mathcal{LP}_n$ must be less than or equal to $\mathcal{E}^+$. This iterative procedure guarantees that the objective function of the proposed multi-step scheme is at least as good as a single relaxation scheme. In the following example, we use a simple scalar labeling problem to illustrate the solution procedure.

Example 1 (A scalar labeling problem): Assume there are two sites $\{1, 2\}$ and for each site the label set is $\{1..7\}$. The objective function is $\min_{\{f_1,f_2\}} c_{1,f_1} + c_{2,f_2} + \lambda|f_1 - f_2|$. In this example we assume that $\{c_{1,j}\} = \{2, 6, 1.7, 4, 5, 2, 2\}$; $\{c_{2,j}\} = \{5, 1, 3, 4, 1, 2, 5\}$, and $\lambda = 0.5$.

Based on the proposed scheme, the problem is solved by the sequential LPs: $\mathcal{LP}_0$, $\mathcal{LP}_1$ and $\mathcal{LP}_2$. In $\mathcal{LP}_0$ the trust regions for sites 1 and 2 both start as the whole label space $[1, 7]$. Constructing $\mathcal{LP}_0$ based on the proposed scheme corresponds to solving an approximated problem in which c for site 1 and 2 are replaced by their lower convex hulls respectively (see Fig. 3). Step $\mathcal{LP}_0$ uses convex hull basis labels $\{1, 3, 7\}$ for site 1 and $\{1, 2, 5, 6, 7\}$ for site 2. $\mathcal{LP}_0$ finds a solution with nonzero weights $\xi_{1,3} = 1$, $f_1 = 3$; and $\xi_{2,2} = 2/3$, $\xi_{2,5} = 1/3$, and resulting continuous label LP solution $f_2 = (2/3 * 2 + 1/3 * 5) = 3$. Based on the proposed rules for anchor selection, we fix site 1 at label 3 and search for the best anchor label for site 2 in [1, 7] using the nonlinear objective function. This label is 2, which is selected as the anchor for site 2. Similarly, the anchor for site 1 is 3. At this stage $\mathcal{E}^+ = c(1, 3) + c(2, 2) + 0.5 * |3 - 2| = 3.2$. Further, the trust region for $\mathcal{LP}_1$ is shrunk to $[2, 6] \times [2, 6]$ by reducing the previous trust region diameter by a factor of 2. The solution of $\mathcal{LP}_1$ is $f_1 = 3$ and $f_2 = 3$. The anchor site is 3 for site 1 and 2 for site 2, with $\mathcal{E}^+ = 3.2$. Based on $\mathcal{LP}_1$, $\mathcal{LP}_2$ has new trust region $[3, 5] \times [2, 4]$ and its solution is $f_1 = 3$ and $f_2 = 2$. Since 3 and 2 are the anchors

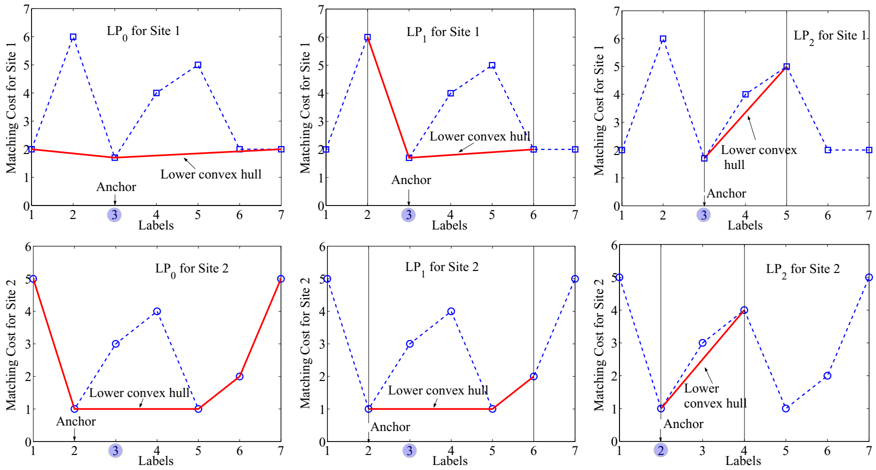


Fig. 3. Successive convexification LP in 1D. Labels in circles are LP fractional solutions.

for site 1 and 2 respectively and in the next iteration the diameter shrinks to unity, the iteration terminates. It is not difficult to verify that the configuration $f_1 = 3$, $f_2 = 2$ achieves the global minimum. Fig. 3 illustrates the proposed successive convexification process for this example.

Interestingly, for the above example, ICM or even the graph cut scheme only finds a local minimum, if initial values are not correctly set. For ICM, if f_2 is set to 5 and the updating is from f_1, the iteration will fall into a local minimum corresponding to $f_1 = 6$ and $f_2 = 5$. The GC scheme based on α-expansion will have the same problem if the initial values of f_1 and f_2 are set to 6 and 5 respectively.

A revised simplex method is used to solve the LP problem. Therefore, an estimate of the average complexity of successive convexification linear programming is $O(|S| \cdot |Q|^{1/2} \cdot (\log |Q| + \log |S|))$, where Q is the label set. Experiments also confirm that the average complexity of the proposed optimization scheme increases more slowly with the size of label set than previous methods such as the graph cut scheme, whose average complexity is linear with respect to $|Q|$.

4 Object Tracking with Multiple Templates

Based on the above matching framework, we present a robust multiple template tracking method that can be used to track objects with changing appearance. One assumption we use in the tracking process is that the template's scale and rotation remain continuous even if the template changes. This assumption is valid for appearance changes for most real-world objects, from simple rigid objects to complex articulated ones. We use a set of templates to represent possible object appearance in the tracking process. These templates can be further formulated as a digraph to represent the possible transitions from one appearance model to another. Models that can be reached in one step from the current model include the model itself and its neighbors. Other parameters in the tracking include the scale and rotation changes of the template. Based on this formulation, tracking becomes the process of locating the object with the best templates constrained by the model transition graph.

The deformable template defined includes feature nodes and neighbor relations. In this paper, the features are image blocks in the template images and target images centered on the edges. We use a very low edge detection threshold so as not to lose weak features. Usually we can downsample the feature points in the template and target images to reduce the complexity. To make the scheme resistant to changing illumination, we use chromaticity color space, in which the three color channels are normalized by their arithmetic mean. The L_1 norm is also used in calculating the cost of matching an image block with a target block. We also use non-square blocks at the boundary of the template, since values outside of the boundary are not defined. Therefore, each feature node also contains a feature mask in calculating the matching cost. We use baseline Delaunay triangulation to obtain the neighbor relations of the feature nodes. To simplify the matching problem, we decompose the geometrical transformation of the template into two cascaded transformations: a global transformation $\mathcal{G}$ and a local deformation $\mathcal{D}$. The global transformation is shared by all the sites in the template while the local deformation can be different for all sites. And, we assume that matching cost c is only influenced by global transformation but not local deformation. Intuitively, c is a function of the source pixel (site), the target pixel (label), and the global transformation (such as scaling and rotation). The energy minimization problem becomes:

$$\min_{\mathcal{G},\mathcal{D}} \mathcal{E} : \sum_{\mathbf{s}\in S} c_{\mathcal{G}}(\mathbf{s}, \mathcal{D}\circ\mathcal{G}(\mathbf{s})) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} ||\mathcal{D}\circ\mathcal{G}(\mathbf{p}) - \mathcal{D}\circ\mathcal{G}(\mathbf{q}) - \mathcal{G}(\mathbf{p}) + \mathcal{G}(\mathbf{q})||$$

In the tracking process, the global transformation $\mathcal{G}$ is specified as the previous rotation and scale estimation and is updated after each matching process. With $\mathcal{G}$ fixed, the problem is reduced to the consistent labeling problem discussed in the last section and we can apply the proposed LP scheme to solve for $\mathcal{D}$ by the successive convexification LP scheme.

After finding the matches of the feature points in the template with corresponding points in the target image based on the proposed method, we need to further decide how similar these two constellations of matched points are and whether the matching result corresponds to the same event as in the exemplar. We use the following quantities to measure the difference between the template and the matching object. The first measure is P, defined as the average of pairwise length changes from the template to the target. To compensate for the global deformation, a global affine transform $\mathcal{A}$ is first estimated based on the matching and then applied to the template points before calculating P. P is further normalized with respect to the average edge length of the template. The second measure is the average warped template matching cost M, which is defined as the average absolute difference of the target image and the warped reference image in the region of interest. The warping is based on a cubic spline. The total matching cost is simply defined as $M + \alpha P$, where α has a typical value from 0.1 to 0.5. Experiments show that *only about 100 randomly selected feature points* are needed in calculating P and M.

Because of the constraints of the physical dynamics, we can safely consider only the neighboring templates equaling the current rotation and scale. The current template and its neighbors are used in the template selection process. The template with the lowest matching cost is chosen, its matching result is recorded, and the rotation θ and scale γ are updated based on the following smoothing model, in which α is typically 0.9:

$$(\theta, \gamma) = \alpha(\theta_{estimate}, \gamma_{estimate}) + (1 - \alpha)(\theta_{last}, \gamma_{last})$$

The proposed scheme is resistant to drifting because we do not track the object sequentially based on the features in the previous frame, and thus avoid template errors accumulated during the tracking process. But we do use parameters such as the rotation angle and scale estimated from previous frames to reduce the searching complexity. A model selection error may still possibly spread to future frames and make the tracker fail. In our scheme, tracking failure can be detected by comparing the minimum matching error with a threshold. When the matching error is too large, we infer a tracking failure and apply a restart process. In this process, all the templates in a rotation and scale range are used to match the target image and the best template is chosen.

5 Experimental Results

We start by comparing the proposed successive convexification LP scheme with the GC, BP, and ICM methods for local deformation estimation. We use the same energy formulation for all methods. GC uses the α-expansion scheme for symbol updating and a fixed order symbol sequence in the iteration. BP is the baseline belief propagation algorithm. BP was not conducted for all the images because of its high complexity. Fig. 4 shows comparison results for matching synthetic grayscale images with ground truth

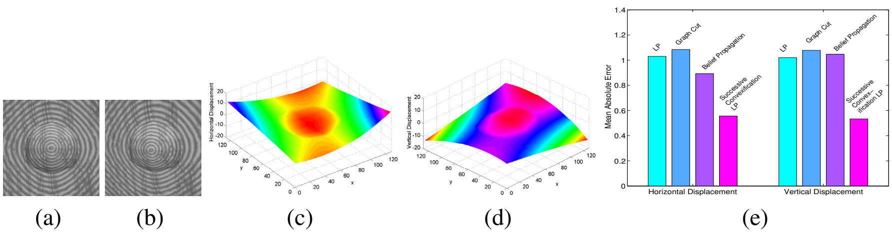


Fig. 4. (a, b): Reference and target image; (c, d): Ground truth horizontal and vertical displacement; (e): Lowest MAE is achieved for the different methods by adjusting the smoothing factors so that they each perform optimally

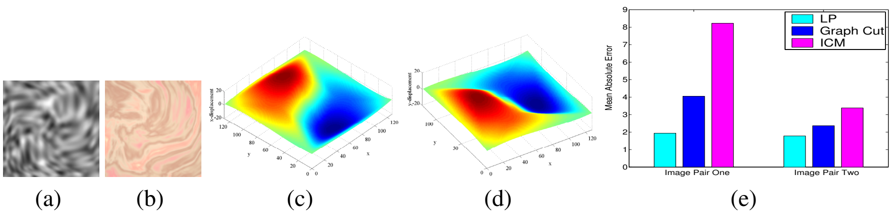
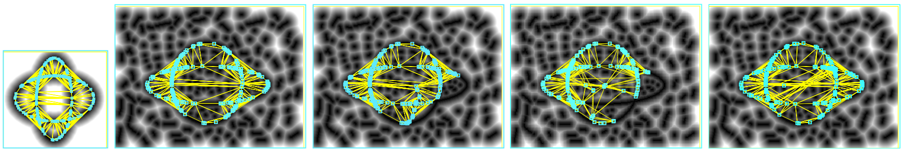
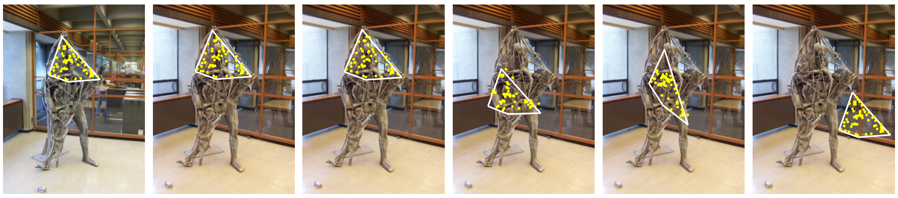


Fig. 5. (a, b): Two synthetic images. (c, d) Ground-truth x-motion and y-motion for images (a) and (b); (e): Mean absolute errors of LP, ICM and Graph 3Cut.



(a) (b) (c) (d) (e)

Fig. 6. (a): Template model showing distance transform; (b): Matching result of proposed scheme; (c): Matching result by GC; (d): Matching result by ICM. (e): Matching result by BP.



(a) (b) (c) (d) (e) (f)

Fig. 7. Color template matching. (a): Template; (b): Matching result of successive relaxation LP with range the whole target image; (c): Matching result for GC with range $[-50, 50] \times [50, 50]$; (d): Matching result for GC with range the whole target image; (e, f): Matching results for ICM and direct sweeping search.

Fig. 8. Tracking result for video tape sequence. Selected from 600 frames.

displacement. The search window is [-20,20]$\times$[-20,20]. The proposed method achieves substantially better results. Fig. 5 shows another experiment result based on synthetic images. Two images and their warped versions based on a known displacement field are used in the experiment. Then, the proposed LP scheme, ICM, and GC are used to solve the L_1 norm consistent labeling problem. The target labels are all the pixels in the target image. The mean absolute error of the estimated matching for each method is then calculated based on the ground truth matching. The smoothing factor for each method is then adjusted such that a minimum matching error is obtained. These match errors for different methods are listed in Fig. 5. The proposed scheme achieves the minimum matching error. Fig. 6 shows another matching result for synthetic images. In this experiment, the pixel block size is 5 by 5 and the smoothing factor equals 1.5. The search range is the whole target image. The sites are selected on the zero-value pixels and the matching candidates for each site are all the zero-value pixels in the target image. The proposed scheme again performs the best among different schemes compared. Fig. 7 shows a template matching experiment for a color image of a wire-metal sculpture. The direct search scheme uses a fixed template shape and sweeps a search window over the target image. As shown in the figure, only our proposed scheme finds the correct target when the search range increases to the whole image. The matching energy for LP is 6.12e3, lowest compared to GC 7.45e3, ICM 11.5e3, and direct search 12.73e3.

Fig. 8 shows an experiment result for tracking a planar object under indoor lighting conditions. The glossy surface makes robust tracking a challenging task. In this experiment we use the first frame as the template image and the region of interest is indicated and a graph template is generated automatically based on random selected edge pixels and Delaunay triangulation. The scale and the rotation status of the template are adaptively updated based on the matching result. In the experiment, the smoothing factor α

Fig. 9. Tracking result for car sequence. Selected from 1000 frames.

Fig. 10. Tracking result for hand sequence. Selected from 500 frames.

for scale and rotation update is set to 0.9. The proposed scheme successfully tracks the object over a long video sequence. Fig. 9 is another tracking result based on a single template, this time for a cluttered outdoor scene. The proposed scheme robustly and accurately follows the moving car in a video sequence with 1000 frames.

Fig. 10 shows a tracking result in which two exemplars are used. The hand undergoes dramatic shapes changes between the two gestures. There are also large scale and rotation changes of the hand involved in this sequence. The proposed scheme successfully tracks the movement of the hand in a poor, low-contrast video. Fig. 11 shows a result for tracking a walking person, using three exemplars. The posture of the person walking in the scene is accurately recovered. The template follows the object successfully in this very complex-background setting.

6 Conclusions

In this paper, we present a robust multiple-template object tracking scheme based on a robust linear programming based matching scheme — successive convexification linear programming. The proposed optimization method can be used for solving consistent labeling problems with L_1 regularity term, and is found to be able to converge to the global optimal solution with high probability. The successive convexification idea can also be generalized to other convex smoothing term problems. The proposed scheme is shown to be more efficient than the graph cut and belief propagation schemes for object matching problems. Based on the proposed optimization scheme, we study an object tracking framework in which multiple templates can be defined. Templates are updated

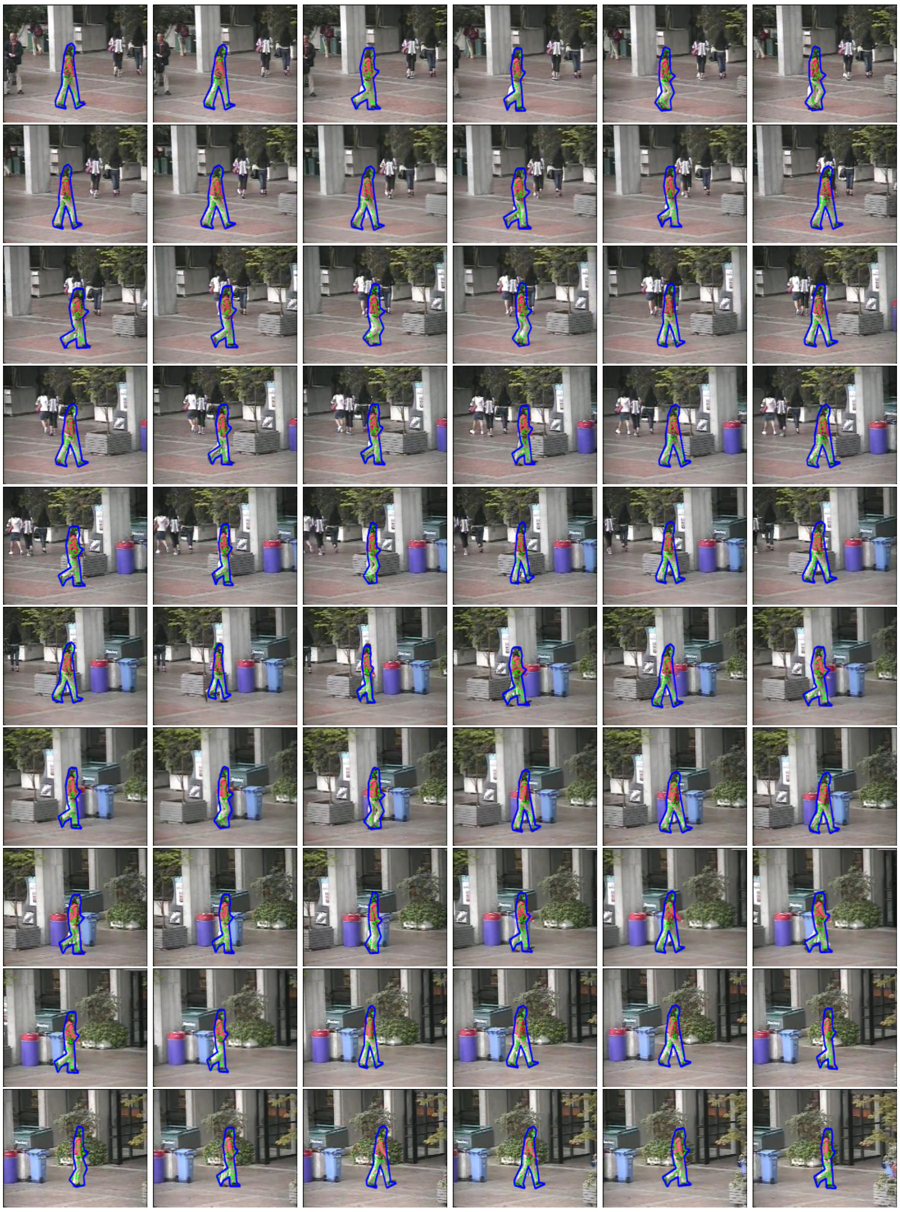


Fig. 11. Tracking result for walking sequence. Selected from 150 frames.

based on the estimation of the global transform resulting from mean square estimation of the matching patterns of the previous stage, and template matching is solved by the proposed LP based scheme. We further present measures to quantify the similarities of the template with target objects. By choosing the best template to its corresponding matching patterns, the proposed scheme can be used to robustly track objects changing appearance dramatically, using only a few templates. Since no training, but only several exemplars are needed, the proposed scheme is easier to deploy than appearance-model based schemes. Experiments show robust tracking results in cluttered environments. The proposed matching scheme can also be applied to other applications such as large scale motion estimation, wide baseline matching and 3D object reconstruction.

7 Appendix

Proposition 1: The energy with consistent rounding is bounded above by $3\varepsilon_{opt} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}(||\mathbf{m_p}-\mathbf{f_p}|| + ||\mathbf{m_q}-\mathbf{f_q}||)$, where ε_{opt} is the optimal energy.

Proof: The proof is simple but lengthy:

$$\sum_{\mathbf{s}} c(\mathbf{s},\mathbf{r_s}) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{r_p}-\mathbf{r_q}||$$
$$\leq \sum_{\mathbf{s}} c(\mathbf{s},\mathbf{r_s}) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}(||\mathbf{r_p}-\mathbf{f_q}|| + ||\mathbf{r_q}-\mathbf{f_p}|| + ||\mathbf{f_p}-\mathbf{f_q}||)$$
$$= \sum_{\mathbf{s}} c(\mathbf{s},\mathbf{r_s}) + \sum_{\mathbf{s}} \sum_{\mathbf{p}\in\mathcal{N}(\mathbf{s})} \lambda_{\mathbf{s},\mathbf{p}}||\mathbf{r_s}-\mathbf{f_p}|| + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f_p}-\mathbf{f_q}||$$

Recalling the anchor selection rule,

$$c(\mathbf{s},\mathbf{r_s}) + \sum_{\mathbf{p}\in\mathcal{N}(\mathbf{s})} \lambda_{\mathbf{s},\mathbf{p}}||\mathbf{r_s}-\mathbf{f_p}|| \leq c(\mathbf{s},\mathbf{m_s}) + \sum_{\mathbf{p}\in\mathcal{N}(\mathbf{s})} \lambda_{\mathbf{s},\mathbf{p}}||\mathbf{m_s}-\mathbf{f_p}||$$

Therefore

$$\sum_{\mathbf{s}} c(\mathbf{s},\mathbf{r_s}) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{r_p}-\mathbf{r_q}||$$
$$\leq \sum_{\mathbf{s}} c(\mathbf{s},\mathbf{m_s}) + \sum_{\mathbf{s}} \sum_{\mathbf{p}\in\mathcal{N}(\mathbf{s})} \lambda_{\mathbf{s},\mathbf{p}}||\mathbf{m_s}-\mathbf{f_p}|| + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f_p}-\mathbf{f_q}||$$
$$= \sum_{\mathbf{s}} c(\mathbf{s},\mathbf{m_s}) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}(||\mathbf{m_p}-\mathbf{f_q}|| + ||\mathbf{m_q}-\mathbf{f_p}||) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f_p}-\mathbf{f_q}||$$
$$\leq \sum_{\mathbf{s}} c(\mathbf{s},\mathbf{m_s}) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}(||\mathbf{m_p}-\mathbf{m_q}|| + ||\mathbf{f_q}-\mathbf{m_q}||$$
$$+ ||\mathbf{m_q}-\mathbf{m_p}|| + ||\mathbf{m_p}-\mathbf{f_p}||) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f_p}-\mathbf{f_q}||$$
$$\leq \sum_{\mathbf{s}} c(\mathbf{s},\mathbf{m_s}) + 2\sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{m_p}-\mathbf{m_q}|| + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f_p}-\mathbf{f_q}||$$
$$+ \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}(||\mathbf{f_q}-\mathbf{m_q}|| + ||\mathbf{m_p}-\mathbf{f_p}||)$$

Noticing that $\sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}}||\mathbf{f_p}-\mathbf{f_q}|| \leq \varepsilon_{opt}$, the proof is complete.

References

1. H. Ishikawa. Global Optimization using Embedded Graphs, Ph.D. Dissertation. May 2000, NYU.
2. T.M. Breuel. A comparison of search strategies for geometric branch and bound algorithms. ECCV, III:837-850, 2002.
3. A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. IEEE Trans. Systems, Man, and Cybernetics, 6(6):420-433, 1976.
4. J. Besag. On the statistical analysis of dirty pictures. J. R. Statis. Soc. Lond. B, 48:259-302, 1986.
5. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. PAMI, 23:1222-1239, 2001.
6. J. Pearl. Probabilistic reasoning in intelligent systems – Networks of plausible inference. Morgan-Kaufmann, 1988.
7. Y. Weiss and W.T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. IEEE Trans. on Information Theory, 47(2):736-744, 2001.

8. P.F. Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. CVPR, I:261-268, 2004.
9. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. ECCV, III:82-96, 2002.
10. V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. ICCV, II:508-515, 2001.
11. J. Sun, H.Y. Shum and N.N. Zheng. Stereo matching using belief propagation. PAMI, 25(7):787-800, 2003.
12. J.M. Coughlan and S.J. Ferreira. Finding deformable shapes using loopy belief propagation. ECCV, III:453-468, 2002.
13. B. Luo and E.R. Hancock. Structural matching using the EM algorithm and singular value decomposition. PAMI, 23:1120-1136, 2001.
14. H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. CVPR, II:44-51, 2000.
15. A. Rangarajan, H.L. Chui and F.L. Bookstein. The softassign procrustes matching algorithm. Information Processing in Medical Imaging, pages 29-42, Springer, 1997.
16. P. D. Tao, T. Q. Phong, R. Horaud, and L. Quan. Stability of Lagrangian duality for nonconvex quadratic programming solution methods and applications to computer vision. Mathematical Modelling and Numerical Analysis, 31(1):57-90, 1997.
17. X. Bai, H. Yu, E. Hancock. Graph matching using embedding and semidefinite programming. BMVC, 2004.
18. M. Ben-Ezra, S. Peleg, and M. Werman. Real-time motion analysis with linear programming. ICCV, pages 703-709, 1999.
19. J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. IEEE Symposium on Foundations of Computer Science, pages 14-23, 1999.
20. C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. Symp. on Discrete Algs, pages 109-118, 2001.
21. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. CVPR, II:142-149, 2000.
22. M. J. Black and A. D. Jepson. Eigentracking: robust matching and tracking of articulated objects using a view-based representation. ECCV, pages 329-342, 1996.
23. L.P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance models. CVPR, I:803-810, 2003.
24. H. Jiang, Z.N. Li, and M.S. Drew. Optimizing motion estimation with linear programming and detail-preserving variational method. CVPR, I:738-745, 2004.
25. H. Jiang, Z.N. Li, and M.S. Drew. Posture recognition with convex programming. ICME, 2005.