Matching Bags of Regions in RGBD images

Hao Jiang
Boston College, USA
hjiang@cs.bc.edu

Abstract

We study the new problem of matching regions between a pair of RGBD images given a large set of overlapping region proposals. These region proposals do not have a tree hierarchy and are treated as bags of regions. Matching RGBD images using bags of region candidates with unstructured relations is a challenging combinatorial problem. We propose a linear formulation, which optimizes the region selection and matching simultaneously so that the matched regions have similar color histogram, shape, and small overlaps, the selected regions have a small number and overall low concavity, and they tend to cover both of the images. We efficiently compute the lower bound by solving a sequence of min-cost bipartite matching problems via Lagrangian relaxation and we obtain the global optimum using branch and bound. Our experiments show that the proposed method is fast, accurate, and robust against cluttered scenes.

1. Introduction

Region matching between images is an important task in computer vision. It is also challenging because of the difficulty of extracting regions consistently from one image to another. Even with the same set of parameters, a segmentation method may give quite different partitions on two similar images. Instead of relying on segmentation algorithms to give consistent partitions, we resort to a large set of region candidates from different segmentation methods. By using a large number of region proposals, we have a better chance to find a subset that has a one-to-one matching between two images. Since these region candidates do not have specific relations to each other, we treat them as *a bag of regions* on each image. In this paper, we propose a novel method to optimize the region selection and matching between a pair of RGBD images given a large bag of overlapping region proposals in source and target images. Fig. 1 shows one example of our method on matching the regions between two RGBD images.

The fundamental problem of our region selection and matching task is min-cost bipartite matching with global region constraints as shown in Fig. 1(e). Traditional bipartite matching minimizes the matching costs with the constraint that each site from one image at most matches one site on the other image. To match a bag of overlapping regions, the optimization needs to satisfy more conditions such as the max-covering constraints, overlapping penalty constraints, and the number constraints. Previous min-cost bipartite matching methods, such as the Hungarian algorithm, cannot be directly used any more. Finding the global optimal solution of the proposed region matching problem is a challenging combinatorial problem, which to our knowledge has not been studied before.

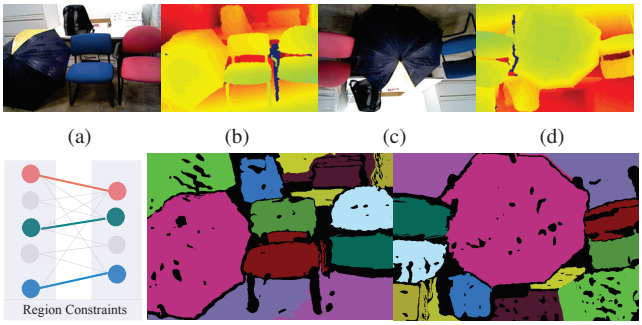


Figure 1. Finding region correspondence between RGBD images using bags of region candidates. We extract candidate regions using different methods to form bags of candidates on the source (a, b) and target (c, d) RGBD images. Our method optimizes the region selection and matching using the graph model in (e) and gives the matching result in (f, g). The same color in (f, g) indicates matched regions.

1.1. Related methods

Matching non-overlapping regions between two images has been intensively studied. A many-to-one region matching method is proposed in [1]. This method matches non-overlapping regions in the source and target color images. To handle region splitting and merging, partial region matching method is proposed. Inexact matching [5] has also been proposed to handle region splitting and merging. Apart from matching in a single level, a region hierarchy can be generated by organizing the regions and successively merged ones into a tree. Matching regions in the tree has be

studied in [2, 3]. Matching tree structured attributes has also be studied in [6]. Organizing a single partition into a tree by region merging may still not be able to capture the correct segmentation for matching. Selecting a good merging plan to generate the tree is also a challenging task. In this paper, we propose to take advantage of a large set of region candidates. Our method does not restrict the ways to generate region candidates. Any region generation methods such as graph based method [16], category independent proposals [20], CPMC [19], RGBD region classification method [13], k-means and many others can be used. By using different methods and a large set of proposals, we have a better chance to find consistent regions between images. These region proposals cannot be organized into a tree. We simply treat them as bags of regions. We propose a novel method to select and match bags of regions between two RGBD images.

Our method is related to graph matching methods [9, 10, 11, 12], which have been used to match single deformable object across images. To deal with multiple objects in two images, we need to define a graph model for each object and then match each model to the target image. The difficulty is that we do not have a partition of the image to start with and therefore it is difficult to define these graph models automatically. Our proposed method does not require any knowledge about the object partition in images. It also does not assume the movement of regions and how large the movement is. Our method thus can handle region matching with very large displacement, scaling and rotation.

In [14], video object segmentation and tracking also uses a large set of region proposals. This method handles each consistent object proposal across a video separately using motion, abjectness and graph cuts. In contrast, our proposed method finds the optimal regions with small overlap and their matching simultaneously between two RGBD images. In [4], superpixel matching is established between successive RGBD video frames using bipartite matching at each level of the region partition tree. Our method deals with two static RGBD images or two frames that may be far apart in video, for which a consistent superpixel partition is hard to obtain. Bipartite matching as shown in our experiments is not the best option for our application. Bipartite matching also cannot be used to solve the overlapping region matching problem directly. These video processing methods are hard to extend to match static images in our application.

Multi-model feature matching methods such as sequential RANSAC [22, 21] and progressive mode seeking [15] have been proposed to find the model and feature matching at the same time. These methods require strong features on the objects, for example SIFT features on textured objects. Our proposed method tackles the matching problem from the point of region correspondence and it works for targets without texture.

Our contribution in this paper is manyfold: (1) we propose a new problem of matching bags of regions across two RGBD images; (2) we propose a novel integer linear formulation of the problem; (3) we solve the problem efficiently by reducing it to min-cost bipartite matching using Lagrangian relaxation. Our experiments show that our method gives promising results.

2. Method

2.1. Overview

We extract a large set of candidate regions from RGBD images by using both color and 3D shape. Each candidate region corresponds to a 3D surface patch in a scene. These candidate regions may be overlapping. Since they can be generated from different region proposal methods, there is no tree hierarchy for the patches. By extracting a large set of regions, we expect that there are subsets of regions in the images that can be matched.

In more details, we find a subset $\mathcal{X}$ from region candidate set I in RGBD image one, a subset $\mathcal{Y}$ from region candidate set J in image two and a one-to-one mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ that matches the regions, so that the following energy function is minimized.

$$\min_{\mathcal{X},\mathcal{Y},f:\mathcal{X}\rightarrow\mathcal{Y}} \{C(\mathcal{X},\mathcal{Y},f) + T(\mathcal{X},\mathcal{Y}) + H(\mathcal{X},\mathcal{Y}) + N(\mathcal{X},\mathcal{Y}) - U(\mathcal{X},\mathcal{Y})\} \quad (1)$$
$$\text{s.t. } \mathcal{X} \text{ and } \mathcal{Y} \text{ are region subsets of } I \text{ and } J \text{ and } f \text{ is a one-to-one mapping from } \mathcal{X} \text{ to } \mathcal{Y}.$$

Here $C(\mathcal{X},\mathcal{Y},f)$ is the cost of matching regions in $\mathcal{X}$ in image one to regions in $\mathcal{Y}$ in image two using f. C is small if f correlates regions that have similar appearance, size and shape. $T(.)$ penalizes the intersection among selected regions in each RGBD image. $H(.)$ penalizes the concavity of 3D surface patches in $\mathcal{X}$ and $\mathcal{Y}$. The more a surface extrudes away from the camera, the more concave a surface is. $N(.)$ quantifies the number of selected regions. $U(.)$ represents the coverage of the selected regions in both source and target images. By minimizing the above objective function, we find a subset of regions in two RGBD images and the correspondence between them so that the matched ones share similar color, size and shape, the overall concavity is low, regions have small overlaps in both images and we use a small number of region candidates to cover both images.

Optimizing the region matching is a combinatorial problem. It involves more constraints than traditional bipartite matching. For example, there is an exclusion constraint among the regions in each image: if one region is selected, other regions that are overlapping with the region in a large portion cannot be selected for matching. The global constraints on the region coverage and total number also complicate the problem. Efficient min-cost bipartite matching cannot be directly applied to this problem. Naive exhaustive search is too complex for real problems with a lot of

candidate regions. In the following, we propose a linear formulation. We then propose a fast method to compute the lower bound by reducing the problem to a sequence of min-cost bipartite matching via Lagrangian relaxation. Then we show how to search for the global optimal solution quickly by using branch and bound.

2.2. Linear formulation

We formulate the optimization in Eq. (1) into an integer linear optimization.

Local matching cost term

We use binary variable $z_{i,j}$ to indicate the matching from region i in source image to region j in target image; if the matching is true $z_{i,j} = 1$ and otherwise 0. The total matching cost is $\sum_{i\in I,\, j\in J} c_{i,j} z_{i,j}$, where $c_{i,j}$ is the cost of matching region i to region j, and recall that I and J are the region candidate sets in source and target images respectively. $c_{i,j}$ is a combination of the color histogram distance and the shape descriptor distance between region i and j.

We use a depth weighted color histogram to quantify region appearance. A depth weighted color histogram for region i is defined as $h_i(n) = \sum_{k\in G_i} \Pi_n(v_k) d_k^2 / \cos^2(\theta_k)$, where $h_i(n)$ is the frequency in color bin n, v_k is pixel k's color value, G_i is the pixel set of region i, $\Pi_n(v)$ is 1 if v is in bin n and otherwise 0, d_k is the depth of the pixel k, θ_k is the angle between the pixel normal vector and the camera optical axis. We treat each image pixel as the projection of a small surface patch in 3D and $d^2/\cos^2(\theta)$ is proportional to the small surface patch area. Depth weighted color histogram of a 2D region is proportional to the color histogram of the corresponding surface patch in 3D and is invariant to the distance of cameras to targets. For color images, we compute the histogram on RGB channels separately and concatenate the vectors. In this paper, the bin number is 16 for each color channel. The color histogram distance is computed using the χ^2 distance between the weighted color histograms. We use scale and rotation invariant moments [18] to quantify the shape of the projected regions on 2D image planes. L_1 norm is used to compute the distance between two shape descriptors. Local matching cost is the linear combination of the color and shape distances.

Intersection term

Apart from selecting region pairs with low matching costs, the regions selected on both images should have small overlaps. We introduce variables x_i and y_j, which are the region selection variables on image one and two for region i and j respectively. If region i is selected in the matching, $x_i = 1$ and otherwise 0. Similar conditions hold for y. The selection variables are related to the matching indicator variable z by $x_i = \sum_{j\in J} z_{i,j}$, $y_j = \sum_{i\in I} z_{i,j}$. To penalize the overlaps between selected regions, we introduce the term $\sum_{i\in I} l_i x_i + \sum_{j\in J} r_j y_j$, where l_i is the pixel number of region i and r_j is the pixel number of region j, into the objective function. When the objective is minimized, the algorithm tends to select non-overlapping regions, because using overlapping regions to cover a given area will increase the objective.

Covering term

Simply minimizing the local matching cost term and the region intersection penalty term would give a trivial all zero solution because all the coefficients in the objective are non-negative. We introduce a term to encourage the selected regions to cover both the source and target images. We partition the source and target images into small tiles. We use p_m to indicate whether tile m in source image is covered by a selected region; $p_m = 1$ if the tile is covered and otherwise $p_m = 0$. Similarly we denote q_n as the indicator variable for tile n in target image. The covering term in the objective is $-(\sum_{m\in\mathcal{M}} p_m + \sum_{n\in\mathcal{N}} q_n)$, where $\mathcal{M}$ and $\mathcal{N}$ are sets of tiles in source and target images respectively. And we let $\sum_{i\in P_m} x_i \geq p_m$, $\forall m \in \mathcal{M}$, and $\sum_{j\in Q_n} y_j \geq q_n$, $\forall n \in \mathcal{N}$ where P_m and Q_n are the region sets covering tile m in image one and tile n in image two respectively. If all the regions that cover a tile are not selected, the tile selection variable will be zero. If at least one region that covers a tile is selected, the corresponding tile selection variable has to be 1 to minimize the objective. Therefore the summation of all the tile selection indicator variables indeed represents the coverage of the selected regions in the matching. With a proper weight, the covering term tend to spread out the selected regions during the matching to give a desirable result.

Convexity term

The above terms are still not enough, because there is no constraint on the level of details to partition the scene. The optimization may give a trivial solution in which both source and target images are treated as a single region in the matching. To avoid the degenerated case, we introduce the concavity penalty term to control the level of region details. For each 3D surface patch, its concavity is defined as the average distance of each point to the frontal hull. The frontal hull is defined as follows. We compute the convex hull of the points on a 3D surface patch. We then send rays from the camera center to all the directions. The first intersection points of these rays with the convex hull form the frontal hull. Apparently, a "convex" surface patch that extrudes towards the camera center has low concavity and a surface patch that extrudes away from the camera center has larger concavity. Let f_i be the concavity of region i in image one and g_j be the concavity of region j in image two, and l_i and r_j be pixel numbers. The concavity penalty term in the objective is $\sum_{i\in I} l_i f_i x_i + \sum_{j\in J} r_j g_j y_j$. The concavity term represents the total distance between the points in the scene and the front hull of each selected region. It thus penalizes the selection of large patches that stretch across several objects and helps control the detail level of region partition.

The number term

There are in fact many ways to partition a scene into roughly convex patches which have similar total concavity. If we simply minimize the concavity, we have a bias towards partition with small regions. We thus introduce another term to penalize the number of regions selected in both images. We prefer a relatively small number of large regions because large regions have more distinguishing power for matching. We strike the balance by introducing the number term $\sum_{i\in I} x_i + \sum_{j\in J} y_j$ into the objective function. By properly selecting the weight, the number term and the convexity term automatically select the details of the region partition in the matching.

Combining all the terms, we obtain the following mixed integer linear program:

$$\begin{aligned} \min \sum_{i\in I,\, j\in J} c_{i,j} z_{i,j} + \sum_{i\in I} (\phi l_i + \mu l_i f_i + \gamma) x_i + \\ \sum_{j\in J} (\phi r_j + \mu r_j g_j + \gamma) y_j - \eta \Big(\sum_{m\in\mathcal{M}} p_m + \sum_{n\in\mathcal{N}} q_n \Big) \\ \text{s.t.} \quad x_i = \sum_{j\in J} z_{i,j},\ y_j = \sum_{i\in I} z_{i,j} \\ \sum_{i\in P_m} x_i \geq p_m,\ \forall m \in \mathcal{M},\ \sum_{j\in Q_n} y_j \geq q_n,\ \forall n \in \mathcal{N} \\ x, y, z = 0 \text{ or } 1,\ 0 \leq p, q \leq 1 \end{aligned} \tag{2}$$

The mixed integer linear program is a combinatorial problem. A naive exhaustive search approach determines the 1 or 0 assignment to each of the matching variable $z_{i,j}$ by explicit enumerating. Such an approach is infeasible for practical problems that involve hundreds or thousands of candidate regions in the images. We propose an implicit enumeration approach by branch and bound. In this paper, we manually set $\phi = 0.1, \mu = 1, \gamma = 20000, \eta = 400$, weight for color to 2 and weight for shape to 1000. Here depth unit is meter. These parameters are fixed in all the experiments. Learning the parameters is also possible by maximizing the gap between the energy of positive matching exemplars to that of negative exemplars; the optimization is still linear.

2.3. Lower bound and Lagrangian relaxation

We compute the lower bound of Eq. (2). Even though we can obtain the lower bound by relaxing the binary variables in the mixed integer program to floating point variables in [0,1], the linear program's complexity is high for problems with very large number of region candidates. We propose a more efficient method to obtain a lower bound using Lagrangian relaxation. Without directly solving the linear program, we solve a sequence of easier min-cost bipartite matching problems. The lower bound of our approach is the same as the linear program relaxation.

Our integer program has the following format:

$$\begin{aligned} \min(c^T z - e^T w) \\ \text{s.t. } Az \leq 1,\ Bz \geq w,\ 0 \leq w \leq 1,\ z \text{ is binary,} \end{aligned} \tag{3}$$

in which c, for which we abuse the notation a bit, is determined by the local matching cost, the area intersection cost, the concavity cost and the number cost, z denotes the vector of matching variables, e is the weight for the covering variables w, which are p and q in the original notation. The x and y terms have been absorbed into the z terms. $Az \leq 1$ is the bipartite matching constraint, $Bz \geq w$ is the max-covering constraint, and other constraints set the bounds for variables.

In the above optimization, the max-covering constraint complicates the problem. We use the Lagrangian dual to obtain a lower bound. After moving the complicated constraints into the objective, the Lagrangian dual of the integer program is

$$\begin{aligned} \max_{\lambda} \min[c^T z - e^T w + \lambda^T (w - Bz)] \\ = \max_{\lambda} \min[(c^T - \lambda^T B) z + (\lambda^T - e^T) w] \\ \text{s.t. } Az \leq 1, 0 \leq w \leq 1, \text{ and } z \text{ is binary.} \end{aligned} \tag{4}$$

Here $\lambda \geq 0$ is the vector of Lagrange multipliers. For any feasible solution of the original integer program, $\lambda^T(w - Bz)$ is non-positive and therefore the internal minimization gives a lower bound of the original integer program for any non-negative λ. The dual problem is much easier to solve than the original integer program. For each given λ, the internal minimization can be separated:

$$\text{[P1]: } \min(c^T - \lambda^T B) z, \ \text{ s.t. } Az \leq 1, z \text{ is binary.} \tag{5}$$

$$\text{[P2]: } \min(\lambda^T - e^T) w, \ \text{ s.t. } 0 \leq w \leq 1\ . \tag{6}$$

For P1, it is easy to verify that all the z variables with non-negative coefficients have to take value 0. If not, we can always zero their value and the objective function does not increase. To solve P1, we build a network with the edges that correspond to the variables z whose coefficients are negative; the two ends of the edges correspond to regions in the source and target images that z variables link. The costs on these edges are the corresponding coefficients of the z variable. Each edge's capacity has a lower bound 0 and an upper bound 1. Min-cost max-flow of this network gives the solution of P1. We use the augmented path method to solve the min-cost matching problem. The time complexity of this method is $O(n^3)$ where n is the number of nodes in the bipartite graph. The Hungarian algorithm can also be used, which also has the complexity $O(n^3)$ if properly implemented. In summary, the element of z is 0 if its coefficient is non-negative; other elements of z with negative weights are determined by the min-cost bipartite matching. P2 can be minimized simply by setting w to the upper or lower bound based on the sign of the coefficients; if the coefficient $\lambda_k - e_k \geq 0$, $w_k = 0$, otherwise $w_k = 1$.

Since $\min[(c^T - \lambda^T B) z + (\lambda^T - e^T) w]$ is a concave function of λ, $\max_\lambda \min[(c^T - \lambda^T B) z + (\lambda^T - e^T) w]$ can be solved using the standard subgradient method that alternates between the optimization of P1 and P2 and updating the λ.

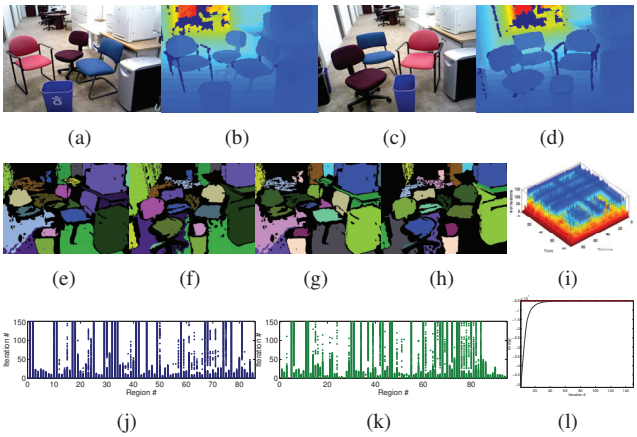


Figure 2. Matching RGBD image pair using Lagrangian relaxation. (a, b): Source color and depth image. (c, d): Target color and depth image. (e, f): Matching result from linear relaxation. (g, h): Matching result from the Lagrangian relaxation. (i): Matching cost matrix evolving during subgradient iteration. (j, k): x and y selection during iteration. A dot indicates a selection. (l): Energy of the Lagrangian dual during iteration.

We initialize λ to a vector of large numbers so that most of the elements in $(c^T - \lambda^T B)$ is negative. After optimizing P1 and P2 to obtain solution to z and w using the current λ, we let $\lambda \leftarrow \max\{0, \lambda + \delta(w - Bz)\}$. The iteration goes on until the relative energy increment of the Lagrangian relaxation is less than a threshold. In this paper, we use a fixed step size δ and it gives good results.

The optimum of the Lagrangian relaxation of our problem equals that of the linear program relaxation. Notice that $\min[(c^T - \lambda^T B)z + (\lambda^T - e^T)w]$ s.t. $Az \leq 1, 0 \leq w \leq 1$, and $z = 0$ or 1, is totally unimodular. Therefore, in the Lagrangian relaxation the binary constraint is superfluous and can be replaced by floating point bound in [0,1]. The Lagrangian dual is thus also the dual of the linear relaxation of the original problem. It has been proved [17] that in such case Lagrangian dual's energy equals that of the linear program relaxation. Note that even though the energy of the Lagrangian relaxation is the same as the linear program, the z and w from the dual are not necessarily the optimal solution to the primal problem. However, if z and w are primal feasible and satisfy the complimentary slackness condition $\lambda^T(w - Bz) = 0$, then they also give the global optimal solution to the primal problem.

Example 1: We show an example of the Lagrangian relaxation of the RGBD image matching in Fig. 2. In this example, we have 85 overlapping candidate regions in the first RGBD image and 94 candidate regions in the second RGBD image. We construct the integer linear formulation using the proposed method. We use 1536 covering indicator variables. We first directly solve the linear program relaxation that replaces the binary variable constraints with the [0,1] bounds. The solution gives the integer values for all the binary variables and thus gives the global optimum of the mixed integer problem. We illustrate the region selection and correspondence in Fig. 2(e, f), in which regions with the same color are matched. We construct the proposed Lagrangian relaxation of the problem. In the subgradient method, initially all the elements in λ are set to 9500 and this makes most of the coefficients in $c^T - \lambda^T B$ to be negative. Minimizing problem P1 is equivalent to solving a min-cost bipartite matching problem corresponding to the negative elements in $c^T - \lambda^T B$. In stage one, since all the elements in $c^T - \lambda^T B$ are negative, all the regions in image one and two take part in the min-cost matching. P2 is then solved by assigning w the lower bound or upper bound based on the coefficient sign. Then λ is updated. Here we use a fixed $\delta = 100$ to update λ. The negative coefficient in $c^T - \lambda^T B$ becomes sparser and sparser as the iteration goes on as Fig. 2(i) shows. Fig. 2(j, k) show the region selection in the first and second image during the iteration. As shown in Fig. 2(l) the energy of the Lagrangian relaxation quickly increases and approaches the red line, which shows the minimum energy of the linear relaxation. In fact, the energy of the Lagrangian dual never equals the LP relaxation due to numerical errors. Lagrangian dual also gives a matching configuration because it computes the matching in z. The result is shown in Fig. 2(g, h). Note that the region matching is close to but not the same as the LP solution. However, this does not pose a problem because we only require the lower bound from the Lagrangian relaxation to be tight enough in our branch and bound procedure.

2.4. Branch and bound

We first need to determine on what variables we generate the search tree branches. One apparent choice is that we can branch on the matching variables z. However, there can be huge number of z because it is a quadratic function of the number of region candidates. We branch on the region selection variables x and y on image one and two instead. By fixing x and y to 1 or 0, we enforce that some regions have to be part of the matching and some have to be excluded. This has a big advantage because their number is much smaller than that of the matching variables.

The Lagrangian dual can still be computed efficiently for each search tree node. Each tree node fixes some x and y to 1 or 0, and introduces extra constraints $Dz = d$, where matrix D is determined by $x_i = \sum_j z_{i,j}$ and $y_j = \sum_i z_{i,j}$, and d is a vector of 1 and 0s. If we treat this as a complicated constraint, the Lagrangian dual of the search tree node is

$$\max_{\lambda,\xi}\{\min_{z,w}[(c^T - \lambda^T B + \xi^T D)z + (\lambda^T - e^T)w - \xi^T d]\} \quad (7)$$

$$\text{s.t. } Az \leq 1, 0 \leq w \leq 1, \text{ and } z \text{ is binary.}$$

We can still decompose the problem into P1 and P2. P1 can be reduced to a min-cost matching problem and P2 can be solved by assigning the upper or lower bound. We use the subgradient method to determine ξ, which is similar to how we deal with λ. For ξ, we update it using $\xi \leftarrow \xi + \delta_\xi(Dz -$

d), where δ_ξ is a positive step size. Note that the Lagrangian relaxation still gives the same bound as the linear program relaxation of the original problem at each search tree node.

The branch and bound procedure is as follows. We get the first feasible solution of the integer program and the upper bound from the z of the Lagrangian relaxation (w can be determined from z to minimize the primal objective). We branch on region variables based on the size of the region; the bigger ones have higher priority because they determine more about the matching energy. After choosing a variable to branch on, without losing generality, assuming the variable to be x_i, we generate two branches: one with $x_i = 0$ and the other with $x_i = 1$. We then need to calculate the lower bounds on both branches. The Lagrangian relaxation of the modified optimization can still be computed efficiently by using min-cost bipartite matching. For each branch, if the Lagrangian dual solution is primal feasible and satisfies the complimentary slackness condition, we obtain a global optimal candidate on that branch. We update the upper bound if the new candidate has smaller objective. If the lower bound is bigger than the current upper bound, the branch is pruned. The branch is also pruned if the dual does not converge which means the primal problem is infeasible. If a branch is not pruned, it is active. During the branching, we always select the active branch with the lowest lower bound first. To speed up the subgradient method, for each new search branch, the λ and ξ corresponding to the parent node constraints are reused as initial values. The extra element in ξ corresponding to the new constraint is initialized to be 1. The branch and bound terminates if the ratio of the gap between the lowest active lower bound and current upper bound to the current upper bound is less than 0.001. The branch and bound converges fast, thanks to the tight bound of Lagrangian relaxation. In a pair of RGBD images with 500 candidates in each image, it takes tens of seconds to find the global optimal solution.

3. Experimental results

Matching bags of regions on two RGBD images is a new problem. There are no previous dedicated methods that we can compete with. However, there are generic matching methods that can be used. Bipartite perfect matching has been widely used to match two sets of non-overlapping regions. Here we try three different methods to generate the non-overlapping or near non-overlapping regions of an image: the graph based method [16], k-means on normals to extract planar patches, and approximate convex shape method [8]. We also compare the proposed method with a greedy approach, which always matches the pair of regions with the lowest matching cost and small overlap with the already chosen ones in both the source and target images. Such a greedy scheme does not guarantee to give the global optimal result because our problem does not have optimal substructure.

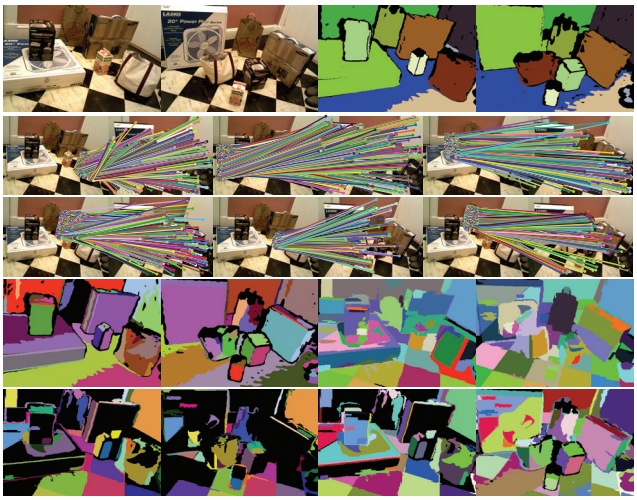


Figure 3. Row 1: Source and target images (left two) and region matching results (right two) of the proposed method. Same colors indicate matched regions. Row 2 and 3: Pixel level matching results using PatchMatch [7]. Row 4: Bipartite matching results using PatchMatch on k-means segmentation (left two) and graph based segmentation [16] (right two). Row 5: Greedy method results using local matching cost of this paper (left two) and PatchMatch costs (right two).

Figure 4. Row 1: Source and target images (left two) and the region matching results (right two) of the proposed method. Row 2: k-means superpixels on source and target images (left two) and bipartite matching results (right two). Row 3: Superpixels on source and target images (left two) using [16] and bipartite matching results (right two).

Fig. 3 Row 1 shows the matching results of the proposed method on two RGBD images. The region proposals are from successively merged k-means superpixels [8] and superpixels from graph based method [16]. The matching is successful even with the cluttered scene, partial occlusion and objects with little texture. Pixel level matching using PatchMatch [7] is shown in Fig. 3 Rows 2 and 3. The pixel level matching result is noisy and does not give the target region directly. We test whether bipartite matching and the greedy method can be used to aggregate the PatchMatch results by using large regions. Bipartite matching uses superpixels from k-means and graph based method [16]. We use the same set of parameters to extract superpixels in the source and target images. We use the average PatchMatch

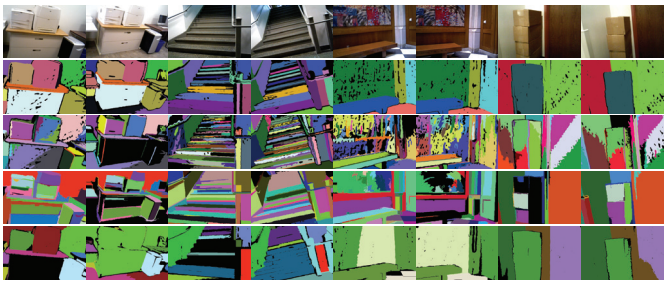


Figure 5. Comparison with bipartite matching on the ground truth dataset. Row 1: Source and target image pairs. Row 2: Matching results of the proposed method. Row 3-5: Matching results of bipartite matching on k-means superpixels (Row 3), superpixels using [16] (Row 4) and regions from [8] (Row 5).

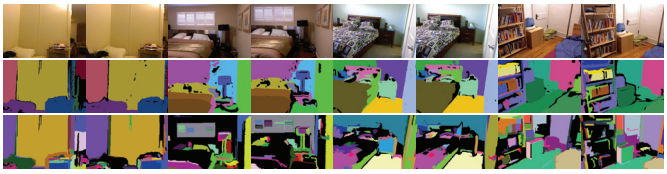


Figure 6. Comparison with the greedy method on the ground truth dataset. Row 1: Source and target image pairs. Row 2: Matching results of the proposed method. Row 3: Matching results of the greedy method.

	This paper	BipartiteC	BipartiteS	BipartiteK	Greedy
PrimeSense	0.6164	0.5396	0.4046	0.3617	0.2502
NYU Kinect	0.5997	0.5179	0.4416	0.2844	0.3348

Table 1. Average matching scores in ground truth test.

	This paper	BipartiteC	BipartiteS	BipartiteK	Greedy
PrimeSense	10.0185	9.6414	38.9781	39.0808	51.7643
NYU Kinect	11.5423	10.3429	48.9713	47.0032	45.0239

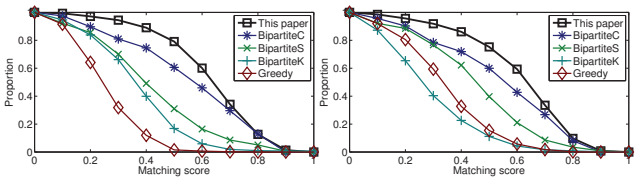
Table 2. Average number of matching pairs.

cost from source to target regions as the local region matching cost. As shown in Row 4 of Fig. 3, bipartite matching fails to match the two images. The greedy method uses the same region candidates as the proposed method. Fig. 3 Row 5 shows the greedy method results using the same local matching cost as the proposed method and the PatchMatch matching cost respectively. In both cases, the greedy method fails. The proposed method gives superior results.

Fig. 4 shows another comparison result. Row 1 shows the matching result of the proposed method. Row 2 and 3 show bipartite matching results using the same local matching cost as the proposed method. Row 2 shows the bipartite matching results with k-means superpixels and Row 3 with the graph based method superpixels [16]. The graph based method uses a linear combination of the color image and depth image to extract superpixels. The inconsistency of region partition using existing segmentation methods is the main reason that bipartite matching approach gives inferior result. Our method is able to handle a large set of overlapping region proposals in the global optimization and gives more reliable results.

We further test the performance of the proposed method on ground truth data. To simplify the ground truth extraction, we apply the proposed region matching method on rigid scenes, whose pointwise matching can be reliably obtained by estimating the rigid transformation of a whole scene using SIFT features on color images and RANSAC on 3D point clouds. The region matching methods do not have the knowledge of rigid scene and do not use the neighbor smoothing constraints in the matching. In this condition, matching bags of regions in two RGBD images of a rigid scene has the same difficulty as matching a dynamic scene. It is thus a sufficient test to show how different methods compare.

The ground truth dataset includes 627 image pairs from NYU V2 Kinect raw dataset [13] and 594 image pairs from our own PrimeSense dataset. In NYU dataset the camera movement is mostly panning and in ours the camera rotates around a target scene. Pairs of images are extracted from videos. These images are 50 frames apart in our PrimeSense dataset and 30 frames apart in the NYU raw video dataset. Sample comparison results of the proposed method against bipartite matching and the greedy method are shown in Fig. 5 and 6. The quantitative results are shown in Fig. 7

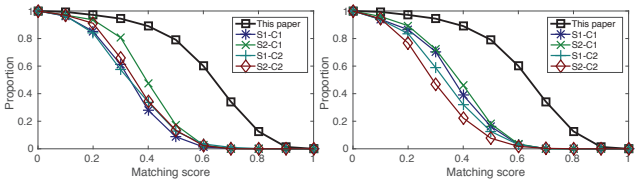


(a) (b)

Figure 7. Region matching score curves for PrimeSense dataset (a) and NYU Kinect dataset (b). Proportions of tests with scores above different thresholds are shown. BipartiteC uses regions from [8], BipartiteS uses regions from [16] and BipartiteK uses regions from k-means.

	This paper	BipartiteC	BipartiteS	BipartiteK	Greedy
PrimeSense	0.9007	0.8607	0.8110	0.8841	0.6274
NYU Kinect	0.9258	0.8230	0.7385	0.9147	0.6171

Table 3. Average coverage.



(a) (b)

Figure 8. Comparison with hierarchical tree matching [3] on PrimeSense (a) and NYU Kinect (b) datasets. Proportions of tests with scores above different thresholds are shown. The tree method uses regions from two methods. In case one (C1), regions are from 3D normal k-means, and in case two (C2) regions are from [16]. For the tree method, two matching scores S1 and S2 are computed in each case.

	This paper	S1-C1	S2-C1	S1-C2	S2-C2
PrimeSense	0.6164	0.3310	0.3872	0.3390	0.3539
NYU Kinect	0.5997	0.3582	0.3747	0.3324	0.2989

Table 4. Average matching scores. Comparison with tree matching method [3].

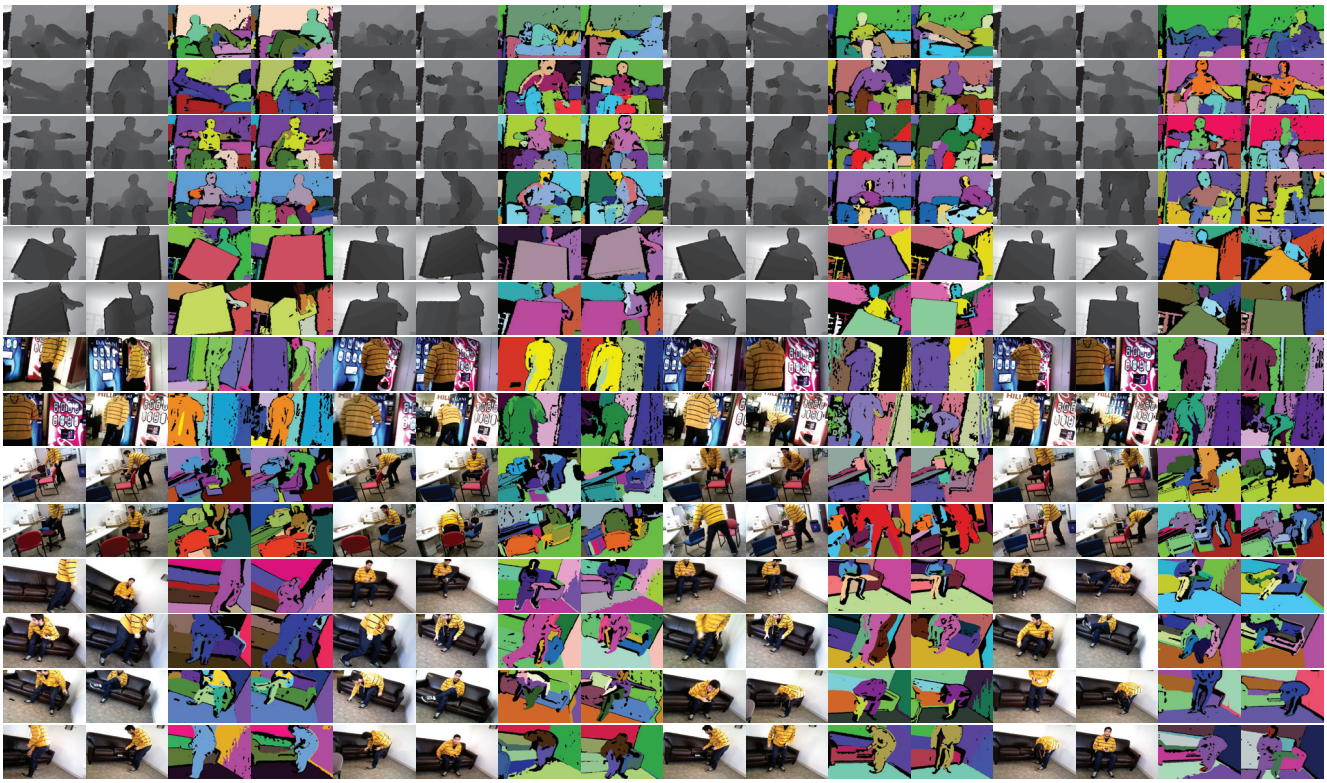


Figure 9. Sample results on more challenging dataset. These image pairs are extracted from five videos and images are 100 frames apart. These matching tests involve deformable and articulated human subjects, moving rigid objects, and occlusion between objects.

and Table 1, 2 and 3. We quantify the matching with a score, which is defined to be a weighted score of matched regions from source to target image. The region matching score is computed as the ratio of the projected source region intersection with the matched target region to the projected region union with the target region. The weights are the ratios of the selected region areas to total area of the selected regions in source image. We compute the weighted sum of the region matching scores to obtain the matching score for an image pair. In ideal case, a perfect matching score is 1. In practice, the score is in [0,1]. As shown in Fig. 7 and Table 1, the proposed method has higher matching score than the bipartite matching and the greedy method. The number of selected regions in matching is shown in Table 2. Table 3 shows the coverage of matched regions in source and target images from different methods. The average region coverage of the proposed method is higher than competing methods.

We also compare with the hierarchical tree matching method [3]. We use two methods to generate region hierarchical trees. In case one (C1), regions are from k-means of 3D normals, and in case two (C2) regions are from [16]; we use successive merging method in [8] to generate region trees. We compute region matching scores for tree matching in two settings: S1 for matching non-overlapping regions at source and target tree levels with the largest mean match pair similarity [3] and mean concavity less than 0.5 meter, and S2 for all matched regions with similarity greater than half the largest similarity. The comparison results are shown in Fig. 8 and Table 4. Our method gives higher matching scores than the tree method. Tree matching methods require regions to have a hierarchy; our proposed method allows any region proposals.

More experimental results sampled from 259 image pairs with 100 frames apart in five videos are shown in Fig. 9. These tests involve more challenging cases. Different targets including human subjects are involved. Our proposed method gives reliable results. Our method is also fast. Typical running time to solve the branch and bound problem on an image pair takes a few seconds. In this paper, we use simple color histogram to quantify the region similarity. The failure cases are mostly due to brightness and color changes of captured images. By using better region appearance and shape descriptor, the performance of the proposed method can be further improved.

4. Conclusion

We study the new problem of matching bags of regions in RGBD image pairs. We propose an effective mixed integer linear formulation. A fast dual method is proposed to compute the lower bound. We obtain the global optimal solution using branch and bound. Our proposed method gives promising results on challenging test images.

Acknowledgments

This research is supported by the United States NSF funding 1018641.

References

[1] V. Hedau, H. Arora, N. Ahuja, "Matching images under unstable segmentations", CVPR 2008. 1

[2] S. Todorovic and M.C. Nechyba, "Dynamic trees for unsupervised segmentation and matching of image regions", TPAMI, 27(11), 2005. 2

[3] Sinisa Todorovic, Narendra Ahuja, "Region-Based Hierarchical Image Matching", IJCV, vol.78, no.1, pp. 47-66, 2008. 2, 7, 8

[4] S. Hickson, S. Birchfield, I. Essa, and H. Christensen, "Efficient hierarchical graph-Based segmentation of RGBD videos", CVPR 2014. 2

[5] C. Wang and K. Abe, "Region correspondence by inexact attributed planar graph matching", ICCV 1995. 1

[6] M. Pelillo, K. Siddiqi, and S. W. Zucker. "Many-to-many matching of attributed trees using association graphs and game dynamics". IWVF, pp. 583-593, 2001. 2

[7] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman "PatchMatch: A randomized correspondence algorithm for structural image editing", SIGGRAPH 2009. 6

[8] H. Jiang, "Finding approximate convex shapes in RGBD images", ECCV 2014. 6, 7, 8

[9] M. Zaslavskiy, F. Bach, and J.-P. Vert, "A path following algorithm for the graph matching problem", IEEE Trans. on PAMI, Vol.31, No.12, 2009. 2

[10] F. Zhou and F. De la Torre, "Deformable graph matching", CVPR 2013. 2

[11] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching", TPAMI 1996. 2

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

[13] N. Silberman, D. Hoiem, P. Kohli and R. Fergus, "Indoor segmentation and support inference from RGBD images", ECCV12. 2, 7

[14] Y.J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation", ICCV 2011. 2

[15] C. Wang, L. Wang, L. Liu, "Progressive mode-seeking on graphs for sparse feature matching", ECCV 2014. 2

[16] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation", IJCV, vol.59, no.2, 2004. 2, 6, 7, 8

[17] G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, 1999. 5

[18] J. Flusser and T. Suk, "Rotation moment invariants for recognition of symmetric objects", IEEE Trans. Image Proc., vol. 15, pp. 3784-3790, 2006. 3

[19] J. Carreira, C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation", CVPR 2010. 2

[20] I. Endres, D. Hoiem, "Category independent object proposals", ECCV 2010. 2

[21] J. Rabin , J. Delon , Y. Gousseau , L. Moisan "MAC-RANSAC: a robust algorithm for the recognition of multiple objects", 3DPTV 2010. 2

[22] M. Zuliani, C. S. Kenney, and B. S. Manjunath, "The multi-RANSAC algorithm and its application to detect planar homographies", ICIP 2005. 2