Maximum-weight bipartite matching technique and its application in image feature matching*

Yong-Qing Cheng, Victor Wu, Robert T. Collins, Allen R. Hanson, Edward M. Riseman

Department of Computer Science
Lederle Graduate Research Center
University of Massachusetts
Amherst, MA. 01003-4610

ABSTRACT

An important and difficult problem in computer vision is to determine 2D image feature correspondences over a set of images. In this paper, two new affinity measures for image points and lines from different images are presented, and are used to construct unweighted and weighted bipartite graphs. It is shown that the image feature matching problem can be reduced to an unweighted matching problem in the bipartite graphs. It is further shown that the problem can be formulated as the general maximum-weight bipartite matching problem, thus generalizing the above unweighted bipartite matching technique.

Keywords: image feature matching, bipartite matching, image sequence analysis, computer vision

1 INTRODUCTION

A fundamental and important problem in computer vision is to acquire 3D models of objects and scenes from a set of images. The basic principles involved in 3D model acquisition are feature matching and triangulation, with the two commonly used types of image features being points and lines. Usually, $2D$ features such as corners, curvature points, and lines are extracted first from each image. Then, the correspondence of these features is established between any pair of images, usually referred to as "the correspondence problem". Finally, the location of each $3D$ point or line is recovered from these $2D$ correspondences in the set of images. So far, extensive research has been devoted to developing robust algorithms in this area[1–3,8–10] including processing of monocular motion sequences, stereo pairs, and sets of distinct views. Although point-based triangulation and line-based triangulation are the two most extensively used triangulation algorithms, more attention has been paid to line-based triangulation since it provides more accurate reconstructions.

Unfortunately, the above triangulation process assumes the correspondence problem has been resolved. In many applications, this information is not available and mechanisms to achieve correspondences are unreliable. This has caused criticism and doubts about feature-based methods because the process of finding $2D$ image feature correspondences can be computationally expensive and is difficult to implement reliably, requiring subsequent algorithms to employ robust mechanisms to detect outliers due to mismatches.[8]

*This work was funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041 and by ARPA/TACOM contract DAAE07-91-C-R035.

In recent years, a large amount of research has been carried out on a variety of correspondence problems[4−7,11]. Many researchers have worked on the problem of motion estimation without correspondences[4−7]. Aloimonos, et. al.[6] presented an algorithm to estimate $3D$ motion without correspondences by combining motion and stereo matching. Recently, Huang and his research group[4] presented a series of algorithms to estimate rigid-body motion from $3D$ data without matching point correspondences. Goldgof et. al.[4] presented moment-based algorithms for matching and motion estimation of $3D$ points or lines sets without correspondences and applied these algorithms to object tracking over image sequences. Lee et. al.[7] proposed an algorithm to deal with the correspondence problem in image sequence analysis.

The main disadvantages of the above approaches include sensitivity to noise and to missing or false points or lines. Methods based on tracking image points or line segments through a sequence of closely spaced image frames, or based on small-motion approximations, are not relevant in our present domain, because we are presented with a set of disparate, monocular views. Furthermore, heuristic measures based on similarity of point or line segment appearance across multiple images will also fail, since widely disparate viewpoints, taken at different times of day and under different weather conditions can lead to corresponding image points or segments of significantly different appearance. What is needed for correspondence matching among disparate, monocular images is a description of the "affinity" between image features, i.e. a measure of the degree to which candidates from different images consistently represent the projection of the same 3D point or the same 3D line, and an efficient matching mechanism.

Traditionally, two separate processing phases have been employed to reconstruct 3D model structure: feature matching and 3D triangulation. With this division of function, it is very difficult to employ 3D information to measure the affinity between image features during the matching phase. We argue that it is better to combine matching and triangulation in an integrated manner in which the affinity measures for image features are easily described.

Recently, graph theoretic methods[11−13] have been applied in the areas of image feature matching and grouping. Unlike many other matching techniques, the unweighted and weighted bipartite matching techniques ensure that a maximum matching can be found. From the point of view of implementation, they can be implemented efficiently, and can be applied to matching problems involving graphs of reasonably large size (about 100,000 vertices).[14] It was shown[14] that the unweighted matching problem can be solved in the running time complexity $O(|V||E|log|V|)$. The extremely important advantage of the unweighted bipartite matching methods is that they can be implemented in a simple, efficient, and parallel way. It was also shown[14] that there is the current best algorithm $O(|V|(|E|+|V|log|E|))$ for a general non-bipartite graph. For the weighted bipartite matching problem, it is considerably easier. Here, we address how to use the bipartite matching techniques to deal with the image feature matching problem.

In this paper, two kinds of new affinity measures for image points and lines from different images are first presented, which are used to construct unweighted and weighted bipartite graphs. Then, the image feature matching problem is reduced to an unweighted matching problem in a bipartite graph. Finally, it is shown that the problem can be further formulated as the general maximum-weight bipartite matching problem, which is the generalization of the above unweighted bipartite matching.

2 MEASURING 2D IMAGE POINT AND LINE AFFINITY

2.1 Measuring 2D point affinity from two images via a 3D pseudo-intersection point

Given a point p_1 in image I_1, we seek its match p_2 in another image I_2. Point p_2 necessarily belongs to a *epipolar line* of image I_2 determined completely by p_1, and vice versa. Most of the existing matching algorithms[9] directly utilize this 2D epipolar line constraint to determine the image point correspondences from two images. However, it is very difficult to employ 3D information to measure the affinity between image features from this

2D epipolar line constraint.

The key observation is that for any pair of image points p_1 and p_2 from two images, there exists a $3D$ *pseudo-intersection point* with the smallest sum of squared distances from it to the two projection lines of p_1 and p_2. The physical meaning of the pseudo-intersection point is that ideally, if p_1 and p_2 are the corresponding image points from two images I_1 and I_2, then their pseudo-intersection point is the real $3D$ point recovered by the traditional triangulation algorithm; if the pseudo-intersection point is located far from the two projection lines, then p_1 and p_2 are not corresponding image points. Given any pair of image points, which are chosen at random, one from each image, the pair may or may not truly correspond to a single 3D point in the scene. The linear algorithm discussed in[11] still yields a 3D pseudo-intersection point in either case, but when this 3D point is projected back into each image it will generally coincide reasonably well with the original pair of 2D image points only if they are in correspondence, and will yield a very poor fit otherwise. This is the basis of our affinity measure for 2D image points: given a pair of possibly corresponding image points (p_i^1, p_j^2), we project the pseudo-intersection point into the two images I_1 and I_2 to get the two projected image points $\bar{p}_i^1(\bar{u}_i, \bar{v}_i)$ and $\bar{p}_j^2(\bar{u}_j, \bar{v}_j)$. We then measure the distance between the observed and projected image points in each image. A $2D$ *similarity function* $sfp(p_i^1, p_j^2)$ is defined as

$$sfp(p_i^1, p_j^2) = e^{-(\|p_i^1 - \bar{p}_i^1\|_2 + \|p_j^2 - \bar{p}_j^2\|_2)} \tag{1}$$

The criterion underlying $sfp(p_i^1, p_j^2)$ is that the best estimate for any $3D$ pseudo-intersection point is the point that minimizes the sum of the least-squares distances between the predicted image location of the computed $3D$ point and its actual image locations in the first and second images. if $sfp(p_i^1, p_j^2) = 0$, it means that p_i^1 is not compatible at all with p_j^2; if $sfp(p_i^1, p_j^2) = 1$, it means that p_i^1 is perfectly compatible with p_j^2.

2.2 Measuring 2D line affinity from three images via a 3D pseudo-intersection line

Similar to the 2D affinity measure between image point features in the last subsection, given the poses of three images, a 2D affinity among image line segments will be developed for the problem of determining image line correspondences while simultaneously computing the corresponding $3D$ lines. The challenge in combining matching and triangulation for image line features is that it is more difficult to describe the affinity between image lines. Line segment endpoints are not meaningful since there may exist significant fragmentation in image line data, and therefore only the position and orientation of the infinite image line passing through a given line segment can be considered reliable. Moreover, this implies that at least three images are necessary to describe affinity, since the projection planes for any pair of image lines in two images always intersect in a 3D line (if parallel, the planes are said to intersect at infinity), and thus no conclusive evidence about possible correspondences between infinite image lines may be derived from only two images.

Given three images and their corresponding camera poses, now we assume that three line segments are chosen at random, one from each image, so that the set may or may not truly correspond to a single 3D line in the scene. For any triplet of image lines from three images, there exists a 3D *pseudo-intersection line* with the smallest sum of squares of the mutial moments of L with respect to the two projection lines of two endpoints for each image line. As this computation will be performed many times as we search for correct correspondences, we need it to be as fast as possible, even if this speed is achieved at the expense of accuracy. Thus, a linear line reconstruction algorithm is needed to achieve a closed-form solution for the best 3D pseudo-intersection line.

For any triplet of image line segments from three images, we wish to compute an affinity value that measures the degree to which these lines are consistent with the hypothesis that they are all projections of the same linear 3D scene structure. To do this, we first compute their pseudo-intersection line L, and then project L back into each image to get three infinite image lines $l_i (i = 1, ..., 3)$.

Suppose l_i is represented by the equation

$$f_i u + g_i v + h_i = 0$$

in pixel coordinates (u, v), and that the endpoints of the original 2D line segment in image i are (u_a, v_a) and

(u_b, v_b). A natural measure of the distance from the line segment to the projected pseudo-intersection line l_i is the sum of absolute pixel distances from the line segment endpoints to l_i, that is

$$r_i = \frac{\mid f_i u_a + g_i v_a + h_i \mid + \mid f_j u_b + g_i v_b + h_i \mid}{\sqrt{f_i^2 + g_i^2}} \quad . \tag{2}$$

If the three image line segments actually do correspond to a single linear 3D structure, we can expect all of them to lie "close" to their respective reprojections of the pseudo-intersection line, where closeness is judged based on our knowledge of the error characteristics of the line segment extraction process and the level of noise in the image. On the other hand, if the image line segments do not correspond to a linear scene structure, their distance from the projected pseudo-intersection line will be large (barring accidental alignments), the distance being greater to the extent that the chosen lines are truly geometrically incompatible.

Based on the above distance measure, the $2D$ affinity or *similarity* value $sfl(l_1, l_2, l_3)$ for a triplet of image line segments from three images is defined as

$$sfl(l_1, l_2, l_3) = e^{-(\sum_{i=1}^{3} r_i)/6} \tag{3}$$

where $\sum_{i=1}^{3} r_i/6$ can be interpreted as the average distance from the set of image line segment endpoints to their projected pseudo-intersection line. Similarly, if $sfl(l_1, l_2, l_3) = 0$, it means that l_1,l_2, and l_3 are not compatible at all; if $sfl(l_1, l_2, l_3) = 1$, it means tha l_1,l_2, and l_3 are perfectly compatible.

3 FORMULATION AS A MAXIMUM-WEIGHT BIPARTITE MATCHING PROBLEM

In this section, we show how the problem of image feature matching is formulated as a maximum-weight bipartite matching problem by introducing the 2D image point affinity function in equation (1) and the 2D image line affinity function in equation (3).

3.1 Reduction to a weighted bipartite graph for the image point matching problem

Given the two sets of image points L $= \{p_i^1 \mid i = 1, 2, ..., n_1\}$ from image I_1 and $R = \{p_j^2 \mid j = 1, 2, ..., n_2\}$ from image I_2, then an undirected weighted graph $G = (V, E)$ can be constructed as follows: $V = L \cup R, E = \{e_{ij}\}$. Each edge e_{ij} $(i = 1, 2, ..., n_1;\ j = 1, 2, ..., n_2)$ corresponds to a weighted link between p_i^1 in I_1 and p_j^2 in I_2, whose weight $w(e_{ij})$ is equal to the "distance" between p_i^1 and p_j^2, i.e. $w(e_{ij})$=$sfp(p_i^1, p_j^2)$. Obviously, the graph arising in such a case is a weighted bipartite graph by construction, since two points in the same image cannot be linked.

3.2 Reduction to two bipartite graphs for image line segment matching problem

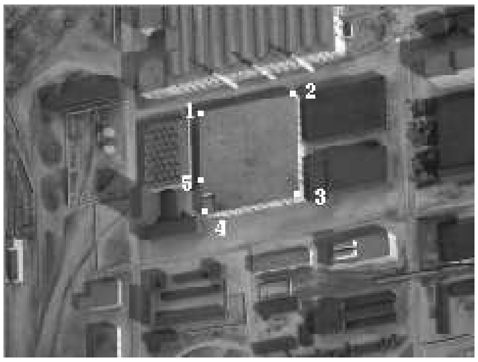
Given a set of line segments l_α,l_β, and l_γ in a triplet of images I_1, I_2 and I_3, two undirected bipartite graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can be constructed as follows. First, generate two vertex sets V_1 and V_2 such that $V_1 = I_1 \cup I_2$ and $V_2 = I_2 \cup I_3$. Next, for all feasible matches among any three image lines l_α, l_β and l_γ, one from each image, generate their edges $e_{\alpha\beta}^1 \in E_1$ and $e_{\beta\gamma}^2 \in E_2$ whose weights are equal to the affinity measure $sfl(l_\alpha, l_\beta, l_\gamma)$, as defined in the last section, i.e. $w(e_{\alpha\beta}^1) = w(e_{\beta\gamma}^2) = sfl(l_\alpha, l_\beta, l_\gamma)$. Note that this in general could involve taking all triplets of image line segments, one from each image, unless domain specific information is used to prune the set of possible matches down to a smaller feasible set.

It should be noted that due to the fragmentation of image lines, multiple competing edges could exist between the same two nodes in either graph. For example, suppose there exists a possible correspondence among the line segments α, β, and γ from the three images respectively, and another possible correspondence between α, β, and

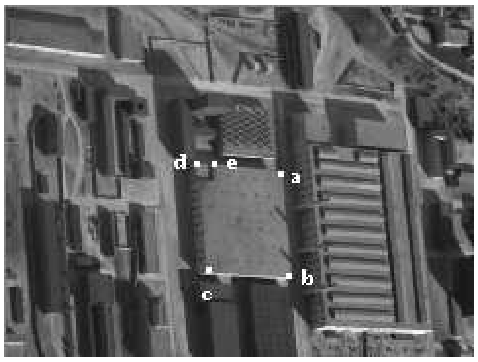
γ'. It would seem then, that two edges between α and β are needed, one to store the weight for $sfl(\alpha,\beta,\gamma)$ and one for $sfl(\alpha,\beta,\gamma')$. In practice, we remove these trivial conflicts at graph creation time by checking if an edge already exists between two nodes before adding a new one. If the affinity value of the new edge is larger than the edge already there, then the old edge is replaced by the new one, otherwise it is left alone.

3.3 Formulation as a maximum-weight bipartite matching problem

As discussed in Subsections 3.1 and 3.2, we know that for any pair of image points p_i^1 and p_j^2, there is a weighted link e_{ij} between p_i^1 and p_j^2 in the weighted bipartite graph G; for any triplet of image line segments l_α, l_β, and l_γ, there is a weighted link $e_{\alpha\beta}^1$ between l_α and l_β in the first bipartite graph G_1 and a weighted link $e_{\beta\gamma}^2$ between l_β and l_γ in the second bipartite graph G_2. Ideally, if they are the corresponding image features (points/lines), then their weights $w(e_{ij})$, or $w(e_{\alpha\beta}^1)$ and $w(e_{\beta\gamma}^2)$ should be equal to the maximum weight, i.e. 1, thus they significantly contribute to the final matching to be determined, and the number of total image feature (point/line) correspondences is equal to the size of the matching. Due to the errors in some of the camera poses and the locations of the extracted image points or line segments, however, the weights $w(e_{ij})$, $w(e_{\alpha\beta})$, and $w(e_{\beta\gamma})$ are not exactly equal to 1, but approach 1. On the other hand, from graph theory, we know that given an undirected graph, a *matching* is a subset of edges $M \subseteq E$ such that for all vertices $v \in V$, at most one edge of M is incident on v. A vertex $v \in V$ is matched by M if some edge in M is incident on v; otherwise, v is unmatched. The *maximum-weight matching* is a matching M_w of size $\mid M_w \mid$ such that the sum of the weights of the edges in M_w is maximum over all possible matchings. Therefore, the image feature correspondences to be determined correspond to the maximum-weight matching in the bipartite graphs G, or G_1 and G_2. For example, Figure 1 gives a pair of aerial images from the RADIUS image sequence which is shown here as a simple illustrative example. An example of a weighted bipartite graph representing the pair of images shown in Figure 1 is shown in Figure 2(a), where $L = \{1,2,3,4,5\}$, $R = \{a,b,c,d,e\}$, and the weights are computed based on the affinity function of image points, as defined in equation (1).



(a)



(b)

Figure 1: Example: a pair of RADIUS images with 5 points each as possible correspondences.

4 ALGORITHMS BASED ON THE UNWEIGHTED AND WEIGHTED BIPARTITE MATCHING TECHNIQUES

As discussed from Section 3, the correspondence problem of image points and lines can be considered as the problem of finding the maximum-weight matching in the weighted bipartite graphs. The remaining question is how to find the maximum-weight matching in the weighted bipartite graphs.

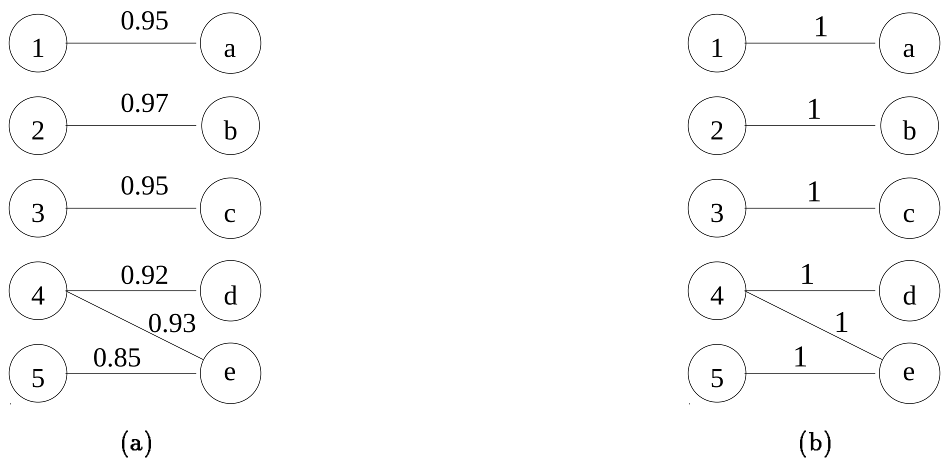


Figure 2: The weighted and unweighted bipartite graphs: (a) weighted bipartite graph; (b) unweighted bipartite graph.

4.1 Algorithm based on the unweighted bipartite matching technique

If each edge has a unit weight in the bipartite graph, then we get the unweighted bipartite matching problem, which is to find a matching of maximum cardinality. The above image feature matching problem for image points and lines can be reduced to the unweighted matching problem by setting all the weights in the bipartite graph to be 1 if $sfp(p_i^1, p_j^2) \geq T_p$ for image points p_i^1 and p_j^2, or if $sfl(l_\alpha, l_\beta, l_\gamma) \geq T_l$ for image line segments l_α, l_β, and l_γ. Here, the thresholds T_p and T_l are chosen empirically. For the weighted bipartite graph shown in Figure 2(a), its unweighted bipartite graph is shown in Figure 2(b) by setting all the weights in the bipartite graph to be 1 if $sfp(p_i^1, p_j^2) \geq T_p = 0.8$.

The problem of image feature matching seems on the surface to have little to do with flow networks, but it can in fact be reduced to a maximum-flow problem. By relating the unweighted matching problem for bipartite graphs to the max-flow problem for simple networks, the matching problem becomes simpler, and the fastest maximum flow algorithm can be used to find the maximum matching. In order to reduce the problem of a maximum matching in the bipartite graph G to a maximum flow problem in the flow network G', the trick is to construct a flow network in which flows correspond to correspondences. We build a corresponding flow network $G' = (V', E')$ for the bipartite graph G as follows: Let the source s and sink t be new vertices not in V, let $V' = V \cup \{s, t\}$, and let the directed edges of G' be given by

$$\begin{aligned} E' = \quad & \{(s, v_i') : v_i' \in L\} \cup \{(v_i', v_j') : v_i' \in L, v_j' \in R, \\ & (v_i', v_j') \in E\} \cup \{(v_j', t) : v \in R\} \end{aligned}$$

and finally, assign unit flow capacity to each edge in E.

Further, it has been shown that a maximum matching M_w in a bipartite graph G corresponds to a maximum flow in its corresponding flow network G'. Therefore, the image feature correspondence problem is exactly equivalent to finding the maximum flow in $G' = (V', E')$, and we can compute a maximum matching in G by finding a maximum flow in G'. The main advantage of formulating the image feature correspondence problem as the unweighted bipartite matching problem is that there exist very fast algorithms (Goldberg's algorithm is $O(|V||E|log|V|)$), which can be implemented in an efficient and parallel way to find the maximum matching in the unweighted bipartite graph.

4.2 Algorithm based on the weighted bipartite matching technique

The main disadvantage of the unweighted bipartite matching formulation is that it is crucial to choose an appropriate value for the threshold T_p for the image point correspondence problem and T_l for the image line segment correspondence problem. If T_p or T_l is too small, more outliers will be created; if T_p or T_l is too large, it will filter too many correct correspondences. For example, as shown in Figure 2(a), if we choose $T_p = 0.9$, then the correct correspondence $(5, e)$ could be filtered. In this case, we would miss the matching $(5, e)$ and could not then disambiguate the matchings $(4, d)$ and $(4, e)$ for the left image point "4". Therefore, it is necessary to deal with the general maximum-weight bipartite matching problem, which is the generalization of the unweighted bipartite matching problem. Although the weighted matching problem is not characterized by maximum flows in terms of augmenting paths, it indeed can be solved based on exactly the same idea: start with any empty matching, and repeatedly discover augmenting paths. In the following, we focus on how to find the maximum-weight matching in the weighted bipartite graph.

Given a matching M in a bipartite graph $G = (V, E)$, a simple path in G is called an *augmenting path with respect to matching* M if its two endpoints are both unmatched, and its edges are alternatively in $E - M$ and in M.

Let p denote an augmenting path with respect to matching M, and P denote the set of edges in p, then $M \oplus P$ is called the *symmetric difference* of M and P. $M \oplus P$ is the set of elements that are in one of M or P, but not both, i.e. $M \oplus P = (M - P) \cup (P - M)$. It can be shown that $M \oplus P$ has the following properties: (1) it is a matching; (2) $\mid M \oplus P \mid = \mid M \mid +1$. For example, consider the matching M shown in Figure 3(a), the edges $(1, a)$, $(2, b)$, $(3, c)$, and $(4, e)$ are matched, and the edges $(4, d)$ and $(5, e)$ are unmatched. The augmenting path $P = \{(5, e), (4, e), (4, d)\}$ with respect to matching M is shown in Figure 3(b). The symmetric difference $M \oplus P$ is shown in Figure 3(c), i.e. $M \oplus P = \{(1, a), (2, b), (3, c), (4, e), (5, d)\}$, and $\mid M \oplus P \mid = 4+1=5$.

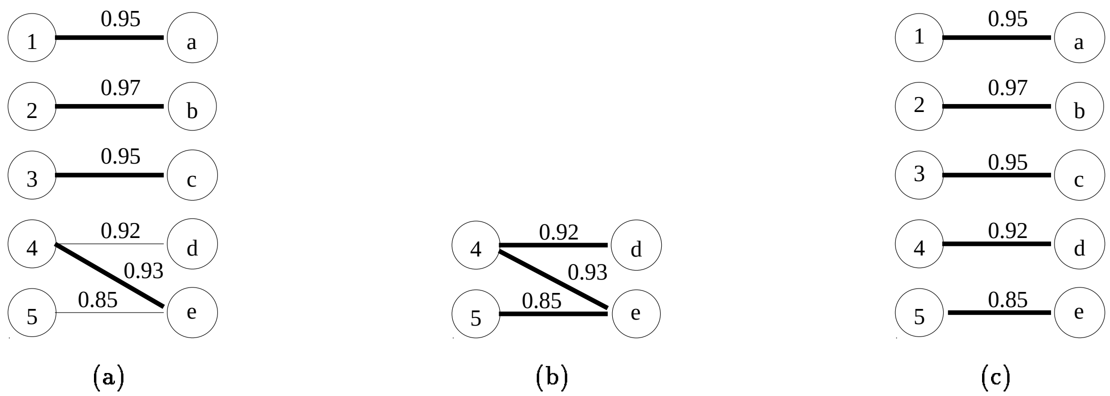


Figure 3: The symmetric difference operator of M and P: (a) matching M; (b) augmenting path P wst. M; (c) $M \oplus P$.

For the matching M, its *total weight of matching* M is defined as

$$w(M) = \sum_{e \in M} w(e)$$

Let M' be a set of edges, then an *incremental weight* $\Delta M'$ is defined as the total weight of the unmatched edges in M' minus the total weight of the matched edges in M':

$$\Delta M' = w(M' - M) - w(M' \cap M)$$

From the definition of incremental weight, we know that for an augmenting path p with respect to M, then ΔP gives the net change in the weight of the matching after augmenting p:

$$w(M \oplus P) = w(M) + \Delta P$$

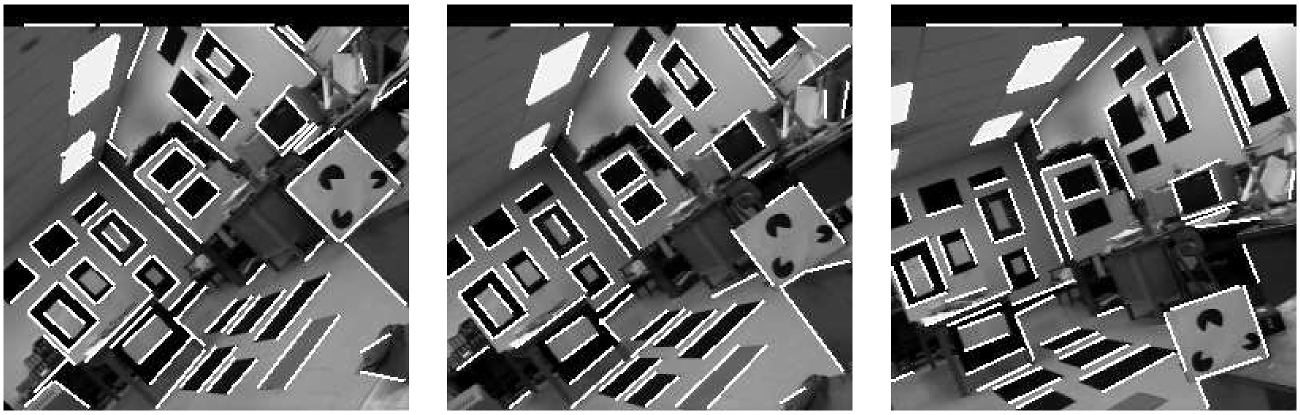
Intuitively, we can use an iterative algorithm to construct a maximum-weight matching. Initially, the matching M is empty. At each iteration, the matching M is increased by finding an augmenting path of maximum incremental weight. This is repeated until no augmenting path with respect to matching M can be found. Indeed, it was proved that if we repeatedly perform augmentations using augmenting paths of maximum incremental weight, this process yields a maximum-weight matching M_w[14].

The remaining problem is how to search for augmenting paths with respect to matching M in a systematic and efficient way. Naturally, a search for augmenting paths must start by constructing alternating paths from the unmatched points. Because an augmenting path must have one unmatched endpoint in L and the other in R, without loss of generality, we can start growing alternating paths only from unmatched vertices of L. We may search for all possible alternating paths from unmatched vertices of L simultaneously in a breadth-first manner. Here, the most efficient algorithm[14] is used to compute the maximum-weight matching in the weighted bipartite graph. This algorithm has two basic steps: (1) to find a shortest path augmentation from a subset of left vertices in L to a subset of right vertices in R; (2) to perform the shortest augmentation. Since a new initialization procedure and a technique to compute the shortest augmenting paths is used, it was shown that the algorithm is very efficient. More details about this algorithm are discussed in[14].

Since the number of matched vertices increases by two each time, this takes at most $\frac{n}{2}$ augmentations. It has been shown that for a matching M of size k of maximum weight among all matchings of size at most k, if there exists a matching M^* of maximum weight among all matchings in G, and $w(M^*) \geq w(M)$, then M has an augmenting path of positive incremental weight. Therefore, the image feature correspondence problem can be exactly reduced to finding the maximum-weight matching in the weighted bipartite graph.

5 EXPERIMENTAL RESULTS AND CONCLUSIONS

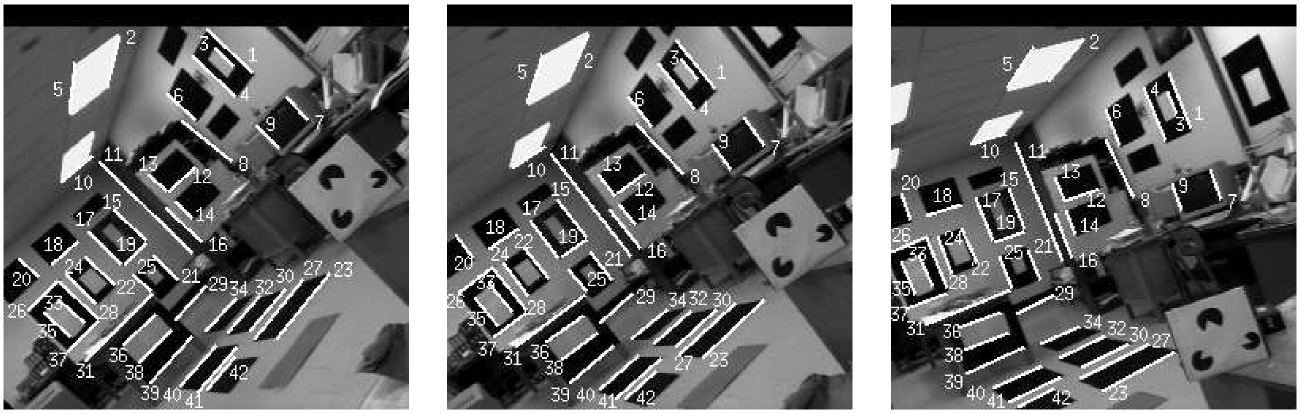
The proposed image point matching algorithm has been tested on the ARPA RADIUS Model Board 1 image set, and the experimental results were reported in[11].



(a) (b) (c)

Figure 4: The three image line sets from the PUMA image sequence: (a) the 1st frame; (b) the 3rd frame ; (c) the 7'th frame.

The proposed image line matching algorithm is tested here on an indoor image sequence. This image sequence was generated by fixing a camera to a PUMA arm and rotating the arm by 4 degrees between consecutive positions of the camera. The plane of rotation of the camera is approximately parallel to the image plane. Twenty frames were taken over a total angular displacement of 116 degrees. The maximum displacement of the camera in these twenty frames is approximately 2 feet along the world y-axis and 1 feet along the world x-axis. The camera pose from each view was determined by the pose determination algorithm in[8]. For each image, image lines were



(a) (b) (c)

Figure 5: The image line matching results: (a) the matched image lines in the 1st frame; (b) the matched image lines in the 3rd frame; (c) the matched image lines in the 7'th frame.

extracted by our line extraction algorithm. Figures 4(a), 4(b), and 4(c) show a triplet of the extracted image line sets from the 1st, 3rd, and 7'th frames in the sequence. Their image line matching results are shown in Figures 5(a)-(c).

The proposed algorithms are currently being tested on more images from other image sequences. Unlike other image matching techniques, the work presented in this paper has the following advantages: (1) it ensures that the maximum matching with maximum weights can be found; (2) from the point of view of implementation, the unweighted and weighted bipartite matching approaches can be implemented in an efficient and parallel way; (3) the thresholds T_p for image points and T_l for image lines are not needed by introducing the maximum-weight bipartite matching mechanism; (4) the proposed matching techniques can be applied to reasonably large matching problems.

6 REFERENCES

[1] Steven D. Blostein and Thomas S. Huang, "Quantization errors in stereo triangulation", Proc. of First International Conf. on Computer Vision, pp. 325-334, 1987.

[2] Z. Zhang and O. D. Faugeras, "Building a 3D world model with a mobile robot: 3D line segment representation and integration", Proc. of the IEEE International Conf. on Pattern Recognition, Atlantic City, New Jersey, June, 1990.

[3] Z. Zhang and O. D. Faugeras, "Tracking and grouping 3D line segemnts", Proc. of 3rd International Conf. on Computer Vision, pp. 577-580, Osaka, Japan, 1990.

[4] D. B. Goldgof, H. Lee and T. S. Huang, "Matching and motion estimation of three-dimensional point and line sets using eigenstructure without correspondences", Pattern Recognition, Vol.25, No. 3, pp. 271-286, 1992.

[5] Chia-Hoang Lee and Anupam Joshi, "Correspondence problem in image sequence analysis", Pattern Recognition, Vol. 26, No. 1, pp. 47-61, 1993.

[6] J. Aloimonos and I. Rigoutsos, "Determining 3-D motion of a rigid surface patch without correspondences under perspective projection", Proc. of AAAI, pp. 681-688, August 1986.

[7] H. Lee, Z. Lin and T. S. Huang, "Estimating rigid-body motion from three-dimentional data without matching point correspondences", Int. J. Imaging Systems Technol. 2, pp. 55-62, 1990.

[8] Kumar, R. *Model Dependent Inference of 3D Information from a Sequence of 2D Images.* Ph.D. Thesis, Computer Science Department, University of Massachusetts, February 1992. Also published as CMPSCI tech report TR92-04.

[9] N. Ayache, "Artificial vision for mobile robots: stereo vision and multisensory perception", The MIT Press, 1991.

[10] R. Deriche, R. Vaillant and O. D. Faugeras, "From noisy edge points to 3D reconstruction of a scene: a robust approach and its uncertainty analysis", Proc. of the Second Europeian Conf. on Computer Vision, pp. 71-79, 1992.

[11] Y.Q. Cheng, R. Collins, A. Hanson, and E. Riseman, "Triangulation without correspondences", *Arpa Image Understanding Workshop,* Monterey, CA, 1994, pp. 993–1000.

[12] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation", IEEE PAMI, 13(7), pp. 1101-1113, Nov. 1993.

[13] W. Kim and A. Kak, "3D object recognition using bipartite matching embedded in discrete relaxation", IEEE PAMI, 13(3), pp. 224-251, March 1991.

[14] D. S. Johnson and C. C. McGeoch, "Network flows and matching: first DIMACS implementation challenge", Series in DIMACS, Vol. 12, 1993.