Feature Matching Across 1D Panoramas

Amy Briggs
Middlebury College
Middlebury, VT 05753
briggs@middlebury.edu

Yunpeng Li
Cornell University
Ithaca, NY 14853
yuli@cs.cornell.edu

Daniel Scharstein
Middlebury College
Middlebury, VT 05753
schar@middlebury.edu

Abstract— This paper presents a new method for feature matching between pairs of one-dimensional panoramic images for use in navigation and localization by a mobile robot equipped with an omnidirectional camera. We extract locally scale-invariant feature points from the scale space of such images, and collect color information and shape properties of the scale-space surface in a feature descriptor. We define a matching cost based on these descriptors, and present a novel dynamic programming method to establish globally optimal feature correspondences between images taken by a moving robot. Our method can handle arbitrary rotations and large numbers of missing features. It is also robust to significant changes in lighting conditions and viewing angle, and in the presence of some occlusion.

I. INTRODUCTION

Vision-based robot navigation and localization is challenging due to the vast amount of visual information available, requiring extensive storage and processing time. To deal with these challenges, we propose the use of features extracted from one-dimensional panoramic images. In prior work we presented methods for extracting stable features from the scale space of 1D panoramic images [1]. In this paper we extend this line of work with several contributions: (1) feature descriptors that allow the local matching of features, (2) a novel global matching method based on circular dynamic programming, and (3) initial results that demonstrate the applicability to robot navigation. The long-term goal of our research is to use natural landmarks extracted from omnidirectional imagery for probabilistic path planning and navigation [2].

Scale-invariant interest operators and feature detectors [3], [4], [5], [6], [7] work by computing the *scale space* of an image and finding the extrema of simple operators. Potential interest points are then augmented with descriptors invariant to local (rigid, affine, or perspective) transformations. Such descriptors can be stored and later used to identify scene locations observed from different points.

Here we propose a slightly different approach. Building on the methods presented in [1] we extract stable features from the scale space of 1D panoramic images. Instead of using only highly distinct features, we select a large number of features in each frame, many of which will remain stable over changes in viewpoint. Localization then requires a global matching method robust to outliers and missing features.

Figure 1 shows a sample panoramic image taken by the robot, the 1D circular image formed by averaging the center scanlines, and an epipolar-plane image (EPI) [8], i.e., the evolution of the 1D image over time as the robot travels.
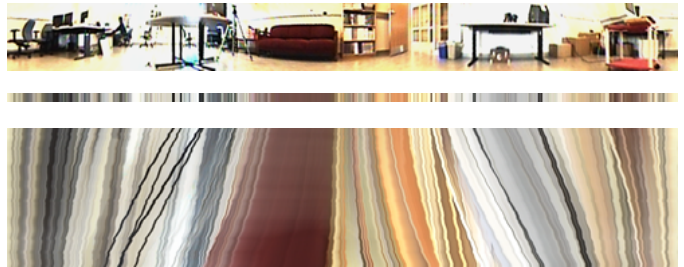

Fig. 1. A sample panoramic view from the robot's omnidirectional camera, the circular 1D image formed by averaging the center scanlines of the panoramic view, and the epipolar plane image (EPI), a "stack" of one-dimensional images over time as the robot travels.

A. Motivation

One-dimensional images can be processed quickly with low storage requirements, enabling dense sampling and real-time analysis of views. The reduced dimensionality also aids greatly in image matching, since fewer parameters need to be estimated. However, there are also some factors that make it difficult to extract stable, globally invariant features from 1D omnidirectional images.

First, for global invariance to viewpoints, the imaged scene has to lie in the plane traversed by the camera (the epipolar plane). This requires that the robot travels on a planar surface, which limits the applicability to indoor environments. Even then, extracting a single scanline from an omnidirectional view is problematic since it is difficult to precisely maintain the camera's orientation due to vibrations [9].

Instead, we form our 1D images by averaging the center scanlines of the cylindrical view, typically subtending a vertical viewing angle of about 15 degrees. We thus trade true distance-invariant intensities for robustness. This is not a problem in practice since intensities still change smoothly with distance which in turn causes smooth changes in the scale space. We demonstrate below that we can still robustly match features in the presence of such smooth changes.

A second difficulty of the 1D approach is that one-dimensional images do not carry very much information. Distinct features that can be matched reliably and uniquely over wide ranges of views are rare. A unique descriptor would have to span many pixels, increasing the chance of occlusion. We thus forego global uniqueness of features in favor of a large number of simple features. This requires a global matching technique that not only matches individual features, but also

considers their spatial relation. The appeal of a scale-space approach is that interest points correspond to scene features of all sizes, ranging from small details such as chair legs to large features, such as entire walls of a room.

B. Related Work

There has been much recent work on invariant features in 2D images, including Lowe's SIFT detector [3], [7], and the invariant interest points by Mikolajczyk and Schmid [4], [5]. Such features have been used for object recognition and image retrieval, as well as robot localization and navigation [10], [11], with a comparison of local image descriptors in [6].

The classic epipolar-plane image (EPI) analysis approach [8] has been applied to panoramic views by Zhu et al. [9] with the application of dense matching and 3D reconstruction. Camera vibrations are compensated for using explicit image stabilization.

Ishiguro and Tsuji [12] describe a method for robot localization from memorized omnidirectional views, which are stored using Fourier coefficients; similarly, Pajdla and Hlaváč [13] use the image phase of a panoramic view for robot localization. Cauchois et al. [14] present a method for robot localization by correlating real and synthesized omnidirectional images, but they can only handle small viewpoint changes. Matsumoto et al. [15] present a similar method based on simply comparing cylindrical gray-level images. None of the above methods computes explicit feature correpondences or can tolerate large changes in viewpoints or partial occlusion.

The idea of matching two panoramic images (or, more generally, circular feature sequences) using dynamic programming originates with the work by Zheng and Tsuji [16], who coined the term *circular dynamic programming*. They match vertical line segments across two panoramic views, and do not model unmatched features explicitly, but rather allow a line segment in one image to match multiple segments in the other image. Vertical edges in omnidirectional images are also used by Yagi et al. [17]. Variants of circular dynamic programming have also been used in dense stereo matching, for example in the work by Sun and Peleg [18], who find a circular shortest path in the cylindrical disparity space. In contrast, our dynamic programming method appears to be the first to match two circular sequences of sparse features while explicitly accounting for unmatched features.

C. Organization of the Paper

The remainder of the paper is organized as follows. Section II reviews scale space and interest point selection. Section III presents our local feature descriptors and matching cost definition. Section IV then presents our global matching method. Experimental results evaluating the quality of global matching are presented in Section V, and we conclude in Section VI.

II. SCALE-SPACE FEATURES

We start with a brief review of the feature detection method introduced in [1]. The key idea is to compute the scale space $S(x, \sigma)$ of each 1D omnidirectional image $I(x)$, $x \in [0, 2\pi]$,
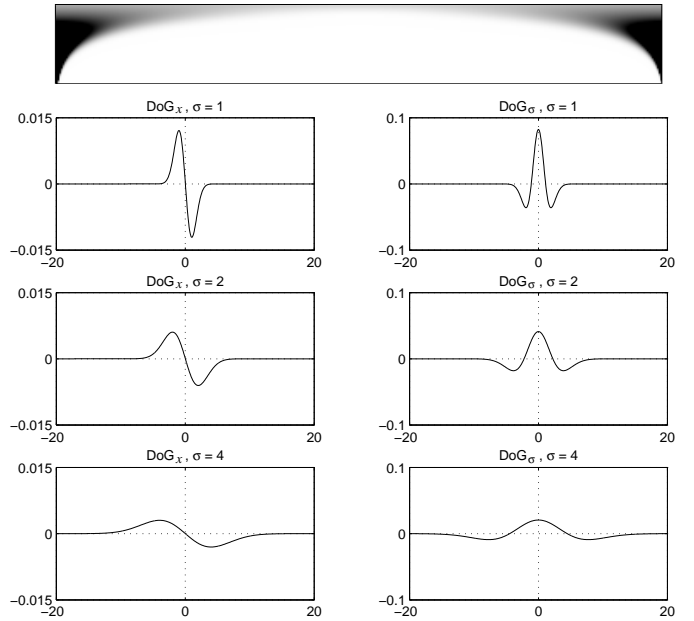

Fig. 2. Top: Gray-level plot of the circular convolution kernel G. The horizontal axis is the image position $x = 0 \ldots 2\pi$; the vertical axis is the scale σ on a logarithmic scale, ranging over nine octaves: $\sigma = 2^{-1} \ldots 2^8$. Bottom: Effective difference of Gaussian (DoG) kernels resulting from computing column (x) differences (left), and row (σ) differences (right), for smoothing scales $\sigma = 1, 2, 4$. To obtain comparable responses the x kernels (left) are scaled by σ.

over a range of scales σ, and to detect locally scale-invariant interest points or "keypoints" in this space. The scale space is defined as the convolution of the image with a circular Gaussian kernel $G(x, \sigma)$.

Similar to 2D scale-space approaches, we represent the scale space using a logarithmic scale for σ, so that neighboring values of σ in the discrete representation of S are a constant factor k apart. Theoretical and empirical motivation for this representation can be found in [4], [7]. For the results shown in this paper, we use $k = 2^{1/3}$, i.e., 3 samples per octave (doubling of σ). Figure 2 shows a gray-level representation of the convolution kernel G, and Figure 3b shows a sample scale space image. In this paper we compute the scale space of the luminance (gray-scale) image, unlike in [1] where each of the three color bands of the image was treated separately. We have found that computing keypoints in each color band yields little additional information, since color changes in the real world are almost always accompanied by intensity changes. Computing the scale space of only the luminance image also yields a significant speedup. Color information is still utilized, however, by storing it in each keypoint's descriptor as described below.

Given a discretized scale space, differences are computed both vertically (between neighboring smoothing scales σ), and horizontally (between neighboring image locations x), resulting in the difference scale spaces D_σ and D_x, respectively. Differencing the scale space between neighboring values of σ (which differ by a constant factor k) is also done in the 2D
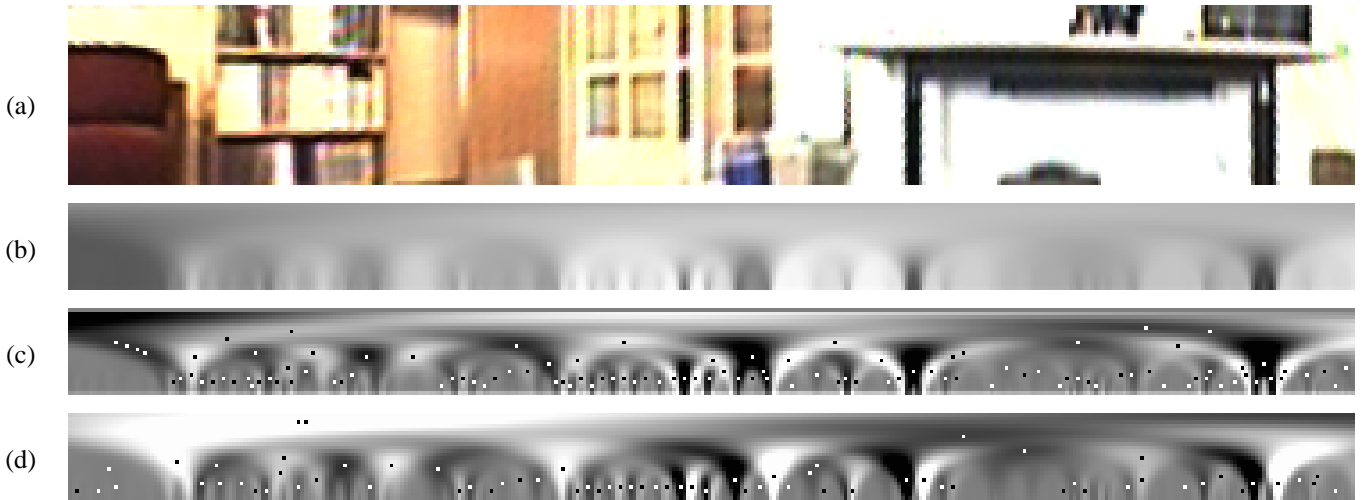
Fig. 3. (a) Part of the circular panorama shown in Figure 1. (b) The scale space S of the average of all 50 scanlines in (a). (c) Differences of rows D_σ with marked minima and maxima (candidate features). (d) Differences of columns D_x with marked minima and maxima.

scale-space approaches [7]. It is equivalent to convolving the original image with a difference-of-Gaussian (DoG) operator ("Mexican hat operator"), as shown in the right-hand column of Figure 2. This provides an approximation of the Laplacian of the image, which is an attractive 2D invariant due to its rotational symmetry. In our one-dimensional case, however, we assume a fixed orientation, and can thus also utilize the directional derivatives by subtracting horizontally neighboring image locations. Unlike in the vertical case, however, we have to multiply the differences by the corresponding value of σ to compensate for the decreasing height of the Gaussian. The resulting equivalent DoG operator is shown in the left-hand column of Figure 2.

Finally, interest point selection proceeds by finding the minima and maxima of D_σ and D_x. To do this, we consider all 3×3 neighborhoods in both images, and check whether the center value is an extremum. We obtain subpixel estimates of both location x and scale σ of all extrema by fitting a quadratic surface to the 3×3 neighborhood. This also provides estimates of the local curvature. The entire process of scale space computation, differencing, and interest point selection is illustrated in Figure 3.

The appeal of finding extrema in scale space is that it provides automatic estimates of both position and scale of features. Intuitively, at each image location, the DoG kernel (Figure 2) that best matches the underlying image intensities is selected. Depending on its vertical (σ) position in scale space, an extremum can thus represent an image feature of any size, ranging from small details, such as table legs, to large features, such as a couch or a wall of a room.

For features to be useful, however, we need to be able to reliably match them across views. In [1] we only demonstrated the features' stability by tracking them through various image sequences. The main focus of this paper are reliable feature matching techniques. We start by describing how to compute local feature descriptors.

III. LOCAL FEATURE MATCHING

The extrema in both difference scale spaces D_σ and D_x are our candidate features. While many of them correspond to real physical features in the original image and persist over changes in viewpoints, others are caused by image noise or by minor, unstable variations of intensities and colors. Some of the unstable features can be identified by their local properties, in particular by a small absolute value at the extremum, or low curvature around it. We exclude these features by imposing lower bounds $v_{\min} = 0.1$ for the extremum's absolute value $|v|$, and $c_{\min} = 0.05$ for its scale-invariant curvature c as measured by the product of σ and the geometric mean of the two quadratic terms q_{xx} and $q_{\sigma\sigma}$ of the quadratic fit:

$$c = \sigma\sqrt{q_{xx}q_{\sigma\sigma}}. \tag{1}$$

This process typically removes about 10% of the features. It was shown in [1] that thresholding can only be used to identify a small fraction of the unstable features, and that there is no clear predictor of the stability of features over different views. Thus, any matching technique must tolerate large numbers of features that cannot be matched.

A. Feature Descriptors

A typical 1D frame, a circular scanline of 1000 pixels depicting a cluttered indoor scene, has about 200–400 features after removing the clearly unstable features. There are four feature categories (minima and maxima in each of D_σ and D_x), each containing about 50–100 features that we want to match with their corresponding features from a different frame. To do this, we compute a matching score for every pair of features within each category. The score should indicate the likelihood that the two scale-space features correspond to the same physical feature in the world.

We use a vector, the *feature descriptor*, to collect information about both the shape of the scale space at that feature and the original intensities and colors of the corresponding
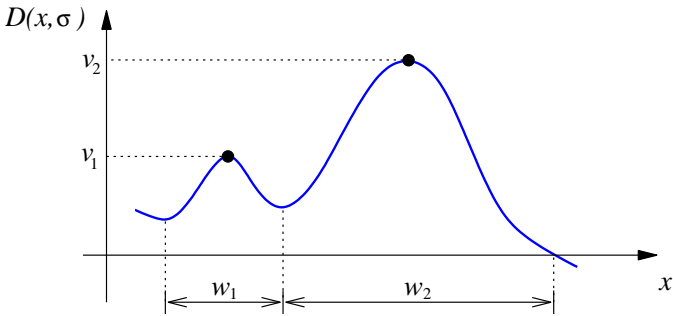
Fig. 4. The horizontal span w of maxima in a difference scale space $D(x,\sigma)$ (shown here for a fixed σ) is their extent in the x direction between the nearest minima or zero-crossings. The span, together with an extremum's absolute value v and curvature c, comprise the shape descriptors of a scale-space feature.

image location. Properties associated with the shape of the scale space at the extremum include:

1) The value v at the extremum, which indicates the contrast of a physical feature. The descriptor value is given by $\log(|v|)$.
2) The scale-invariant curvature c at the extremum (equation 1), which indicates the sharpness of a physical feature. The descriptor value is given by $\log(c)$.
3) The horizontal span w of the extremum before the slope becomes flat or the surface reaches zero (see Figure 4). This indicates the importance and extent of a physical feature. It proves to be a better (less local) descriptor than the extreme value and the curvature alone. The descriptor value is $\log(w/\sigma)$.

The reason the logarithm is used in each of the above measures (all of which are strictly positive) is to achieve a more uniform distribution of each feature property.

The descriptor also contains color information from the original 1D image at the feature location and its two neighboring pixels. Colors are normalized for invariance to illumination changes. We found that including color information is necessary for reliable matching, but that a small neighborhood of 3 pixels is sufficient. Since x-features are caused by intensity edges, while σ-features occur in the center of objects, we handle them slightly differently: for σ features, we simply store the RGB components of the average (normalized) color in the feature vector, while for x-features we store the three pixels' colors separately. We give a higher weight to the color components of the descriptor to balance their relative importance with the shape descriptor values.

The dissimilarity of two features of the same type can now be computed as the Euclidean distance between their feature vectors. We define the matching score of two features to be the inverse of this distance.

B. Performance of Local Matching

Our choice of feature descriptors and matching cost are the result of extensive experiments. We have used two methods to assess the performance of local matching (solely based on matching cost without any global consistency or smoothness constraints): (1) to compute statistics based on tracked features, and (2) to generate plots of 2-way consistent matches.

If the robot moves slowly, tracking features from one frame to the next (as was done in [1]) is a simple way to establish true correspondences across more distant frames, and thus more distant viewpoints. It allows assessing a proposed matching cost measure by comparing the costs of tracked features (which we assume to be correctly matched) with those of other feature pairs (which we assume to be not correctly matched). This allows a quantitative evaluation of different matching costs, which is useful for tuning the parameters, such as the respective weights of the descriptor. A disadvantage, however, is not all features can be tracked uniquely, and thus our ground-truth correspondences contain false negatives.

A different, qualitative performance assessment can be obtained by observing correspondence plots of 2-way best matches, i.e., feature pairs for which the matching cost is minimal for both. In the absence of occlusion, a 2D plot relating the two 1D image locations of matched features should consist of a single (roughly diagonal) line. Figure 5 shows sample plots. Note that the shape of this line is related to the robot's motion: no movement yields the diagonal $y = x$, rotation of the robot shifts the line, and translation yields an S-curve since features on either side of the robot move in opposite directions. The points not on the line, however, are clearly mismatches and indicate that many features cannot be matched locally (i.e., without considering their relation to other matches). Based on our experiments, local matching becomes unreliable for viewpoint changes of more than 15 degrees and in the presence of occlusion.

IV. GLOBAL FEATURE MATCHING

In the absence of narrow occluding objects, the features visible from two different locations will have the same relative ordering. This observation, known as the *ordering constraint*, enables an efficient algorithm for finding the globally optimal solution to the feature matching problem. Our algorithm is based on dynamic programming (DP), and is related to DP scanline algorithms that have been used in stereo matching [19], [20], [21] which also use the ordering constraint. There are several differences between our scenario and the one commonly assumed in stereo matching.

1) In stereo, both the disparity range (allowable shift) and the scanlines are bounded, while we have circular scanlines and arbitrary shifts.
2) In stereo, unmatched pixels are always a direct result of occlusion (or surface foreshortening), while in our case unmatched features are common due to their instability.
3) Most DP stereo algorithms match individual pixels (perhaps after some preprocessing), which are evenly spaced, while we match features at arbitrary locations. In other words, DP stereo methods perform *dense* matching with the goal of surface reconstruction, while we perform *sparse* matching with the goal of robot localization.
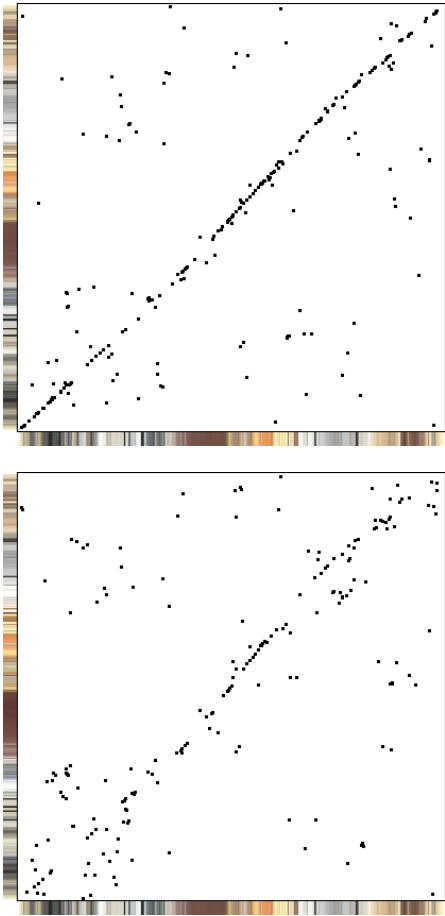
Fig. 5. Local matching patterns resulting from matching two scanlines separated by 50 frames (top) and 120 frames (bottom). Only 2-way best matches are shown (i.e., those with mutually maximal matching scores). The correct matches lie roughly along the diagonal. It can be seen that the local matching performance quickly degrades as the distance between viewpoints increases. In the examples shown, the maximum changes in viewing angle are roughly 10 degrees (top) and 20 degrees (bottom).

Given two circular images with features at locations $\{x_i | i = 1 \ldots N_x\}$ and $\{y_j | j = 1 \ldots N_y\}$ respectively, the goal of the algorithm is to find a set of matches $\mathcal{M}$ that obeys the ordering constraint and maximizes the total matching score

$$\sum_{(i,j) \in \mathcal{M}} S(i,j), \qquad (2)$$

where $S(i,j)$ denotes the score of matching features i and j.

We delay addressing the problem of handling circular scanlines by assuming for now one known match. Since the images "wrap around" and can be rotated arbitrarily, we can assume without loss of generality that $(1,1) \in \mathcal{M}$ and that $x_1 = y_1 = 0$.

In this case, finding the optimal assignment consists of finding a monotone increasing path through the grid of feature locations (x_i, y_j), starting at $(0,0)$ and ending at $(2\pi, 2\pi)$, such that the total matching score of all visited grid locations is maximized. See Figure 6 for illustration, where the maximizing path is shown dashed.
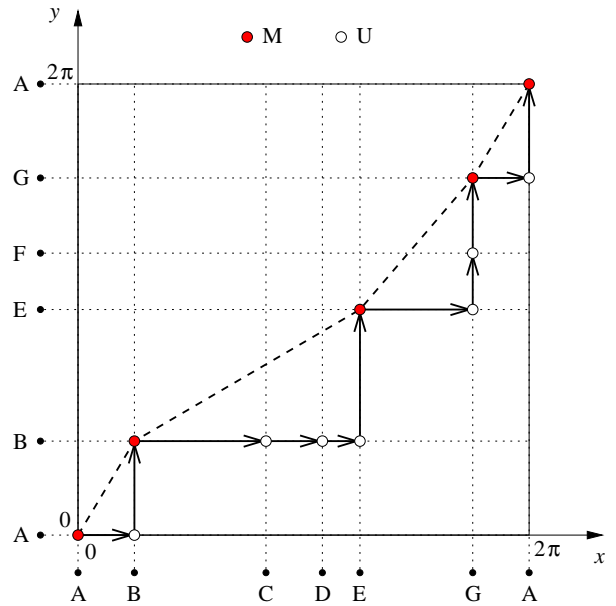

Fig. 6. Global feature matching using dynamic programming. The figure shows the matching space of the features on two circular scanlines. 7 distinct features A–G are present, of which only 4 are visible in both images. (C and D stay unmatched in scanline 1, while F stays unmatched in scanline 2.) The path connecting the matches M (filled circles) is shown dashed, which increases monotonically due to the ordering constraint. The algorithm instead traverses the path indicated with arrows, which includes unmatched feature pairs U (open circles). It can be seen that M states can only be reached from U states below, while U states can be reached from M or U states to the left, or from U states below. This motivates equation 3.

Under the ordering constraint we can find the optimal path using dynamic programming. We maintain two tables M and U, whose entries correspond to all feature pairings. Entry $M(i,j)$ represents the maximum score of all monotone paths from $(1,1)$ to (i,j), where (i,j) is matched, while entry $U(i,j)$ represents the maximum score of all such paths where (i,j) remains unmatched. We need the second array to account for features that are only present in one view but not the other. Referring to Figure 6, it can be seen that an optimizing path (shown dashed) can be traversed using only horizontal and vertical moves between neighboring table entries (or "states"), which motivates the following update rules:

$$\begin{aligned} M(i,j) &= U(i,j-1) + S(i,j) \\ U(i,j) &= \max(M(i-1,j), U(i-1,j), U(i,j-1)) \end{aligned} \qquad (3)$$

Note that even when consecutive features are matched (e.g., features A and B in the figure), we pass for convenience through a U state. This keeps the update rule for an M state very simple: it inherits the matching score from the U state below, in addition to the score of the current match. A U state can be reached from an M state to the left, or from a U state to the left or below, and inherits the maximum of these three scores. We also keep track of the state that yielded the maximum value, so that the optimal path can be backtracked at the end.

Thus, given starting values $M(1,1) = S(1,1)$ and $U(1,1) = 0$, and assuming zeros in all cells outside the

tables, all entries in both tables can be computed consecutively from left to right and bottom to top. The top-right score then indicates the global maximum of the total matching score, and the optimal correspondences can be recovered by backtracking the best path. Assuming the same number of features N in both frames, the total run time of the algorithm is $O(N^2)$.

Now we turn to the problem of handling arbitrary rotations. The algorithm as discussed so far will find the optimal path only if the initial match $(1,1)$ is indeed on the optimal path. Since we cannot know whether a point is on the optimal path or not when we start the algorithm, one way to find the globally optimal path is to run the algorithm with every possible match (i,j) as starting point, which would yield an overall run time of $O(N^4)$. Note, however, that any path will include feature 1, either in a matched or in an unmatched state. Thus, we only need to run the algorithm with $2N$ starting states $M(1,j)$ and $U(1,j)$, yielding a run time of $O(N^3)$.

In practice, the optimal path passes with high certainty through many of the 2-way best matches (as described in Section III-B), in particular those with high matching scores. We can therefore use a limited number of such matches as starting or "seed" points, and still find the optimal path, or at least a path very close to it which is sufficient in practice. This idea is similar to the use of "ground control points" in scanline stereo algorithms [21]; however, we only require one of these points to lie on the globally best path. We have found through experiments that using 20 seed points yields a good compromise between quality and performance.

Since our matching scores are strictly positive, the algorithm will sometimes match stray features in otherwise featureless areas. These incorrect matches result in "jags" in the path that usually can be easily detected based on the slope of their adjoining segments. We remove such erroneous matches in a postprocessing step.

The entire matching process is quite fast. Our current implementation on a 3 GHz Pentium 4 takes about 70 ms to match two frames (35 ms for feature extraction and local matching cost computation, and 35 ms for the global matching). Taken together with 25 ms for unwarping the original image, averaging the scanlines, and computing the scale space, the total processing rate is about 10 Hz.

V. Experimental Results

Figure 7 shows the matching curve computed by our global matching algorithm for the same scenario as shown at the bottom of Figure 5. The correct matches have clearly been recovered.

Once the global matches have been computed, the matching curve can be used to map (warp) the second scanline into the first. Applying this techniqe to an entire image sequence, i.e., stabilizing an EPI with respect to the first line, yields an effective way of visualizing the performance of the global matching method. The resulting EPI, which we call the *mapped line image* (MLI), should now consist only of vertical stripes if the global matching is correct for all frames.
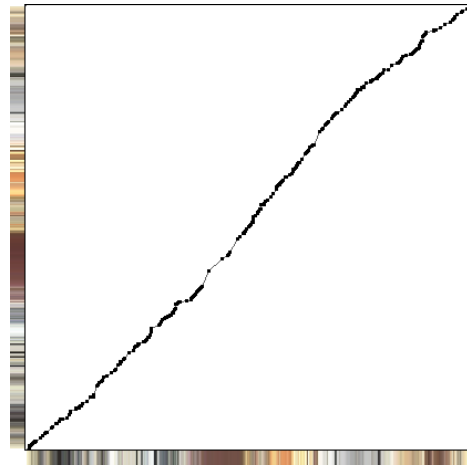

Fig. 7. The global matching curve for two scanlines separated by 120 frames (same scenario as Figure 5 bottom).
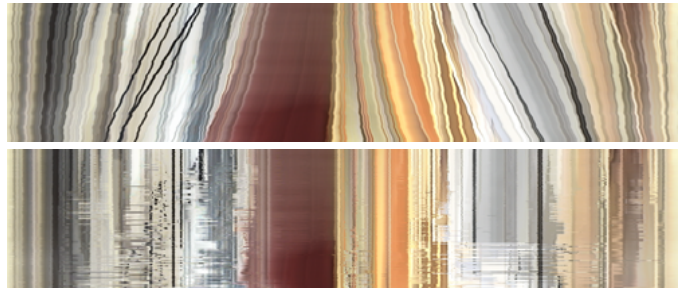

Fig. 8. Top: EPI from Figure 1, which contains some occlusion. Bottom: Mapped line image (MLI), where each row has been stabilized with respect to the first row based on the global matching curve between the two.

Figure 8 shows an example of an MLI for the EPI from Figure 1. Some matching errors are noticeable as horizontal streaking, in particular in the bottom portion of the MLI where the interframe distance is quite large (and thus also the change in viewing angle and viewing distance). Many of these streaks, however, are also caused by occlusion and by violations of the ordering constraint. In particular, note the thin black lines on the left corresponding to the legs of the table, which is much closer than the background. The occlusion is clearly visible in the EPI as the black lines are more slanted than the background. Under the ordering constraint, the algorithm can only recover the motion of either the foreground or the background, but not both. Still, the recovered global matches correctly align most of the surroundings.

Figure 9 demonstrates that our method can handle arbitrary rotations (circular shifts) as well as significant lighting changes. It shows two images taken by a robot from the same location but at different orientations. In addition the ceiling lights were turned off in one of the images. The observed matching curve is very close to the expected curve, a straight line at a 45-degree angle (which wraps around since the frames are circular), despite the fact that only few features could be matched. This is impressive given that the lighting change also results in a significant change in color. Since the color
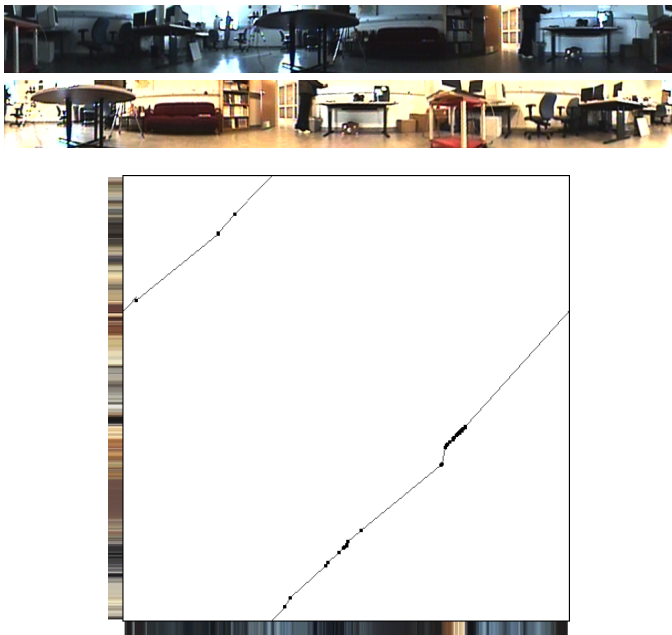
Fig. 9. Top: Two onmidirectional images from the same viewpoint under different rotations and in different lighting conditions. Bottom: Global matching results.
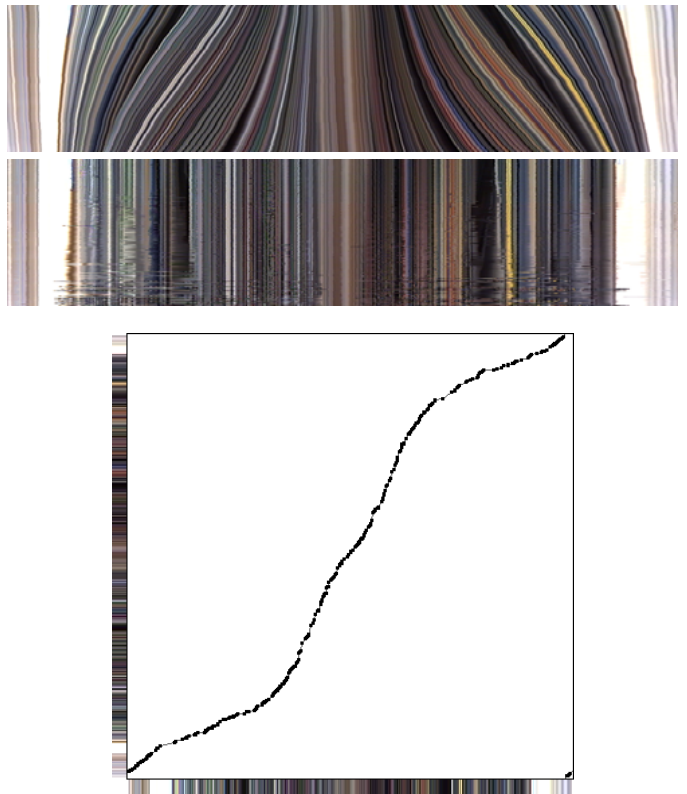

Fig. 10. Top: EPI and MLI of an image sequence with repetitive patterns taken by a translating robot. Bottom: The global matching curve between the first frame and one of the last frames.

descriptors are only normalized with respect to luminance, the matching scores are strongly affected by a color change.

Next, Figure 10 demonstrates that our method can deal quite effectively with repeating patterns, which are impossible to match locally. The image shows the EPI of an image sequence taken as the robot was driving between bookshelves. The MLI contains few errors, most of which occur only in the last portion of the sequence. The global matching curve shows that the significant change in viewing angle can still be recovered.

Note that MLIs only serve to visualize the matching results, but that our goal is neither to perform dense matching of the original images, nor to reconstruct the exact geometry of the surroundings. In fact, for our intended application of robot localization and navigation, we want to use the global matching results between the robot's current view and a set of stored reference views.

We close this section with an experiment that illustrates the potential of our method for robot localization. We compute a measure of distance between two views as follows. We first estimate the global rotation ϕ between the views by fitting a straight line with slope 1 to the points (x_i, y_i) on the "unwrapped" global matching curve: $y_i = x_i + \phi$. We then measure the straightness of the matching curve by computing the robust residual (the average of the middle two quartiles) of this fit. Note that this average residual measures the average absolute difference in viewing angle over all features.

To test the measure, we choose a small set of reference frames from one sequence taken by the robot, typically every 100-th or 200-th frame. We then select for each frame of a different sequence the closest reference frame based on the computed distance. Figure 11 shows the results for two sequences taken along similar paths, but under different lighting conditions and occlusions. It can be seen that the minimum distance between frames and reference frames seems to correlate well with the actual distance, and that the selected closest reference frame only briefly oscillates for views halfway between reference frames. We have performed such experiments for other pairs of sequences and for sparser sets of reference frames, with similar results. Overall we have found that the distance measure is typically not affected by lighting changes and in the presence of some occlusion.

In summary, our experiments demonstrate that our method is quite promising for robot localization and navigation. If little occlusion is present, changes in viewing direction of about 45 degrees can be tolerated. With more occlusion the accuracy of the individual matches degrades, but the overall correspondences can still be recovered. In conjunction with a robust distance function, the closest reference frame can be reliably selected from a collection of stored views.

VI. Conclusion

We have presented a new method for feature matching between pairs of one-dimensional panoramic images for use in mobile robot navigation and localization. Our method extracts locally scale-invariant feature points from the scale space of such images, with color information and shape properties of the scale space surface encoded in a feature descriptor. We
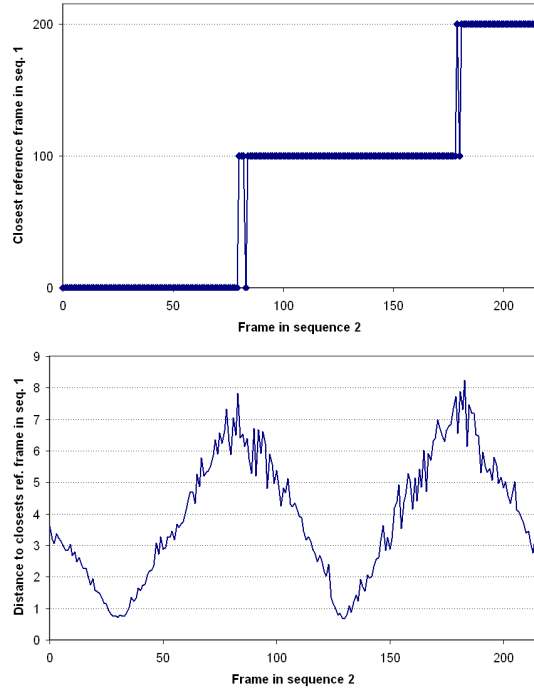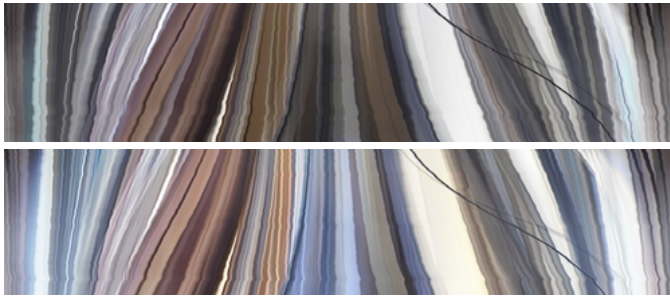
Fig. 11. Top: Two EPIs of sequences taken along similar paths with different lighting and occlusion. Every 100-th frame of the first sequence was stored as a reference frame. Middle: The index of the closest reference frame for each frame of the second sequence. Bottom: The distance to the closest reference frame, as measured by the (robust) average difference of viewing angle in degrees.

have also presented a novel dynamic programming method to establish globally optimal correspondences between features in different images. Experimental results show that our method handles arbitrary rotations, large numbers of missing features due to occlusion and noise, and is robust to significant changes in lighting conditions and viewing angle.

We are currently designing more sophisticated distance measures that extract the angle of heading of the two views (i.e., the epipoles), and allow some reasoning about the geometry of the observed scene. The long-term goal of our work is to reliably navigate large indoor environments based on features extracted from stored reference views.

REFERENCES

[1] A. Briggs, C. Detweiler, P. Mullen, and D. Scharstein, "Scale-space features in 1D omnidirectional images," in *Omnivis 2004, the Fifth Workshop on Omnidirectional Vision, Prague, Czech Republic*, May 2004, pp. 115–126.

[2] A. Briggs, C. Detweiler, D. Scharstein, and A. Vandenberg-Rodes, "Expected shortest paths for landmark-based robot navigation," *International Journal of Robotics Research*, vol. 23, no. 7–8, pp. 717–728, July 2004.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision, Corfu, Greece*, Sept. 1999, pp. 1150–1157.

[4] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the International Conference on Computer Vision, Vancouver, Canada*, July 2001, pp. 525–531.

[5] ——, "An affine invariant interest point detector," in *Proceedings of the European Conference on Computer Vision, Copenhagen*, vol. 4, May 2002, pp. 700–714.

[6] ——, "A performance evaluation of local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin*, vol. II, June 2003, pp. 257–263.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] R. C. Bolles and H. H. Baker, "Epipolar-plane image analysis: A technique for analyzing motion sequences," AI Center, SRI International, Tech. Rep. 377, Feb. 1986.

[9] Z. Zhu, G. Xu, and X. Lin, "Panoramic EPI generation and analysis of video from a moving platform with vibration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado*, June 1999, pp. 531–537.

[10] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland*, Oct. 2002, pp. 226–231.

[11] ——, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research*, pp. 735–758, Aug. 2002.

[12] H. Ishiguro and S. Tsuji, "Image-based memory of environment," in *Proceedings of the IEEE International Conference on Intelligent Robotics and Systems*, 1996, pp. 634–639.

[13] T. Pajdla and V. Hlavac, "Zero phase representation of panoramic images for image based localization," in *Proceedings of the Eighth International Conference on Computer Analysis of Images and Patterns, Ljubljana, Slovenia, Springer LNCS 1689*, Sept. 1999, pp. 550–557.

[14] C. Cauchois, E. Brassart, L. Delahoche, and A. Clerentin, "3D localization with conical vision," in *Omnivis 2003, the Fourth Workshop on Omnidirectional Vision, Madison, Wisconsin*, June 2003.

[15] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue, "Visual navigation using omnidirectional view sequence," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, Korea*, 1999, pp. 317–322.

[16] J. Y. Zheng and S. Tsuji, "Panoramic representation for route recognition by a mobile robot," *International Journal of Computer Vision*, vol. 9, no. 1, pp. 55–76, 1992.

[17] Y. Yagi, Y. Nishizawa, and M. Yachida, "Map-based navigation for a mobile robot with omnidirectional image sensor COPIS," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, pp. 634–648, 1995.

[18] C. Sun and S. Peleg, "Fast panoramic stereo matching using cylindrical maximum surfaces," *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, vol. 34, no. 1, pp. 760–765, Feb. 2004.

[19] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 542–567, May 1996.

[20] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, Dec. 1999.

[21] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, Sept. 1999.