Spatiotemporal Consistency Checking Of Passive Range Data

Robert C. Bolles, SRI International, bolles@ai.sri.com
John Woodfill, Interval Research, woodfill@interval.com

November 11, 1993

Abstract

A spatiotemporal technique for checking the consistency of stereo range results and integrating them over time is presented. This technique is designed as part of a passive ranging system whose goal is to produce range images with as high a resolution as possible in order to support the detection of as small objects as possible. The basic approach is to minimize the application of spatial aggregation operations, which reduce resolution, and to apply a set of multiple-match consistency checks over space and time to filter out mistakes. We present the basic approach, describe two implementations of it (one of which is a research-oriented system that runs on a Connection Machine and the other of which runs on a Sparc10 and provides real-time feedback for SRI's indoor robot, Flakey), present an initial characterization of the effectiveness of the technique, and conclude with ideas for future work.

Introduction

The ultimate goal of this research is to develop passive range sensing techniques that provide the spatial and depth resolutions required to detect small, but dangerous, navigation obstacles, such as holes and medium-sized rocks. Current feature-based techniques do not provide dense enough results to detect these objects and correlation-based stereo and motion systems typically smooth over them. Correlation systems apply spatial aggregation operations in three places, in image preprocessing (e.g., performing Gaussian smoothing to reduce image noise), in matching (e.g., using large correlation windows to provide an ample statistical footing), and in post processing (e.g., eliminating results that differ significantly from their neighbors). These operations reduce the chance of errors, but they also dramatically reduce the resolution of the results. Our approach, on the other hand, is to minimize the use of spatial aggregation in order to maximize the resolution and to provide an alternate set of filtering techniques to eliminate mistakes.

We propose a class of filtering strategies based on checking the consistences of multiple in(ter)dependent matches. The idea behind this class of strategies is the same as that behind the consistency checking techniques used in Moravec's "slider stereo" [10], Hannah's left-to-right and right-to-left doublechecking [7], INRIA's trinocular filtering [1], and Kanade's multi-image summing of SSD's [16]. In this paper we introduce a natural extension of these techniques to include spatiotemporal consistency checking. Figure 1 shows the basic approach. Each arrow represents an independent match from one image to another. The system performs conventional stereo disparity estimation from left to right and optical flow estimation from present to past. The disparity estimates are corroborated by performing an addtional right to left stereo match and verifying that the left to right result is the inverse of the right to left (the "left-right check"), as in [7] and [6]. The optical flow estimates are similarly corroborated by estimating the past to present flow field (the "forward-back check"). If these checks fail the matches are marked as invalid. In addition, since an optical flow estimate is available for each pixel in both cameras, the spatiotemporal transitivity of two stereo estimates and two optical flow estimates can be checked to insure that the four sided "loop" of matches is consistent (the "loop check"). If the loop check fails, the confidence in the individual matches is decreased. Validity information is associated with what we term "pixel features," instead of "pixels" or "features," since the information is associated with a grid of pixels, but remains in correspondence with moving scene features.

This spatiotemporal matching strategy provides a natural way of integrating local depth images over time. As shown in Figure 2, the model of the scene is image-centered. Each pixel-feature has assocated stereo disparity estimates, optical flow estimates and auxiliary information describing the numbers and types of consistency checks passed by the pixels. An advantage of this type of scene modeling is that it avoids the need to explicitly compute the 6 degree-of-freedom transforms that relate one local 3D model to another, as shown in Figure 3. Computing these transforms can be tricky when, for example, there is motion in a scene, and computationally expensive (e.g., see [13]). Our approach to integration differs from most other approaches, however, in that it does not provide a persistent *global* model of the scene. Rather it provides an image-centered model of the cur-

Figure 1: Spatiotemporal multiple-match consistency checking.

rently visible surfaces of the scene objects. As objects leave the field of view, they are lost. The information produced by this type of modeling system could be provided as data to a more conventional global modeling system.

The remainder of this paper is organized as follows. In Section 2 we describe several techniques for attaining credible stereo disparity estimates and discuss how they relate to our consistency-checking approach. In Section 3, we describe an experimental system that makes use of spatiotemporal consistency checking, and present examples of its use within a sensor system to support cross-country navigation. In Section 4, we briefly describe a second implementation of these ideas in a real-time system for SRI's indoor robot, Flakey. And finally, in Section 5, we draw some conclusions and discuss ideas for improvements and future work.

Attaining Credible Stereo Estimates

Producing stereo disparity estimates is simple — for each pixel on one image, find the best match under some metric on the other image. Attaining credible disparity estimates is a complex problem, to which there are many approaches.

System Engineering: Constrain the environment and/or the sensing system to eliminate or minimize the impact of as many factors as possible. For example, restrict the lighting to be from a well-modeled source. This approach also includes techniques to reduce the search region by calibrating the cameras relative to one another. Smaller, more focused search regions reduce the computation required to find a match, and more importantly, reduce the chance that the search might accidently locate an image feature that happens to look more like the desired feature than the real one.

Selective matching: Only attempt matches for features in the imagery that are highly distinctive. In so far as features are distinctive, the chances of a false match are minimized.

Multiple cameras: By using N calibrated cameras, one can obtain better depth estimates. The correlation surfaces for $(N-1)$ matches can be merged at each point, and the best "global" match can be selected. To merge the results, the raw disparities are converted into a common representation. Moravec implemented a system of this type by sliding a camera to nine different positions along a bar [10]. Kanade et al. have developed a technique of this type, using an inverse disparity representation to integrate multiple results [8]. They have applied their technique to camera configurations with three or more cameras in a line. Kanade is currently developing another version of this type of system using seven cameras arranged in an L-shaped configuration.

Match evaluation: Perform matching everywhere in the image, but attempt to winnow out bad estimates.

In many real situations, it is prohibitive, or impossible to tightly constrain the environment and sensing system. If there are few distinctive features in a scene, selective matching will not provide sufficient detail for detecting small scene elements. In addition, if the environment is constrained, matching is selective and multiple cameras are used, erroneous matches arise. Therefore, we pursue the match evaluation tack of performing stereo matching everywhere on relatively uncalibrated imagery and attempting to evaluate the matches in order to cull out erroneous estimates.

A number of techniques have been used to evaluate stereo matches, some with more success than others. Probably the first technique to be tried was simply a threshold on the correlation value computed for the best match. Unfortunately, although this value is related to the validity of a match, the range of values associated with correct matches significantly overlaps the range of values for incorrect matches. This means that for any reasonable threshold there is a large number of correct matches labelled as "bad" and incorrect matches labelled as "good." This situation occurs because the correlation value is a function of several interrelated factors, such as the change in perspective from one viewpoint to another, the reflectance properties of the scene feature, the amount of noise in the images, the overall change in intensities from one image to another, and the number of nearby features that have a similar appearance.

Given this complex interrelationship of factors, many approaches to evaluating stereo matches have been tried:

Factor Analysis: Develop computational models of as many of the factors as possible, implement a matching technique that estimates the parameters of these models, and then set thresholds on these parameter values. For example, Baltsavias has implemented an iterative matching technique that can estimate such things as the surface normal of a scene feature and the gain and offset between two image windows [2].

Correlation Surface Analysis: Examine the correlation surface near the best match and compute properties such as the height of the highest peak relative to the "background" or the number of alternative matches (i.e., significant peaks) within a certain distance of the highest peak. For example, Nishihara has implemented an evaluation procedure that only

Figure 3: Global map formation from a sequence of stereo pairs.

accepts a match if its peak is significantly higher than the second best one and the width of the peak is greater than some threshold [3].

Object Surface Constraints: Invoke constraints derived from assumptions about the types of surfaces in the scene. For example, Pollard et al apply a threshold of 1 pixel on the disparity gradient between two measured points [12]. As another example, Hannah has implemented an outlier rejection process that examines a large region around each result and marks points as inconsistent when they are more than 3 or 4 standard deviations away from the mean of the disparities in the region [3].

Multiple In(ter)dependent Matches: Perform multiple matches for each feature and compare the positions of their results. If the positions do not agree, mark the results as inconsistent. For example, Ayache and Lustman have implemented a trinocular stereo system in which two matches are made for each point, one from image1 to image2 and one from image2 to image3. Then, as shown in Figure 4, if the point in image3 is close enough to the epipolar line corresponding to the point in image1, mark the points as consistent [1].

Existing stereo systems have typically used combinations of these techniques to winnow out mistakes. In this paper we focus on the multiple-match approach and extend it to include temporal consistency for multi-camera sensors. One reason we concentrate on multiple-match techniques is that it is easy to set a threshold for "good" matches. We use a threshold of one pixel for all tests. For some of the other tests, such as correlation surface analysis and outlier rejection, it is difficult to find a principled way of setting the thresholds.

Several multiple-match techniques are possible and many of them have been incorporated into existing stereo systems to filter out mistakes. Some examples are:

Left-right Check: Perform each match twice, once from the left image to the right and once from the right image to the left (see Figure 5); if the distance between the initial point in the left image and the final right-to-left match is small enough, mark the point as consistent (e.g., see [7] or [6]).

Compare Different Techniques: Apply two or more matching techniques and compare the positions of their results. For example, different search strategies and/or different correlation metrics could be used. As far as we know, no one has implemented this approach. Several stereo systems use different sized windows within a hierarchical search strategy, where one match guides the search for a higher-resolution one, but nobody has applied two completely different techniques (e.g., a correlation-based technique and an edge-based technique) and then merged their results.

Compare Multiple Depth Estimates: Given N calibrated cameras (where N is 3 or more), (1) select one camera as the pivot camera, (2) for each point in the pivot camera's image, compute $(N-1)$ depth estimates by analyzing pairs of images, one of which is from the pivot camera, (3) mark points as consistent if the depth estimates are approximately equal. Yoshida and Hirose have implemented a five-camera sensor based on this approach [16].

Epipolar-line Check: Given three calibrated cameras, there is a way to use two matches for each point to check the results, as shown in Figure 4: Match points from image1 in image2 and points from image2 in image3; If the distance between image3's point and the epipolar line corresponding to image1's point is sufficiently small, mark the point as consistent [1].

Transitivity Check: Given three uncalibrated cameras (or three crudely calibrated cameras), another way to identify possible mistakes is to perform three matches for each point, one from image1 to image2, one from image2 to image3, and one from image3 to image1 (see Figure 6). If the distance from the starting point in image1 to the final point produced by traversing the loop is small enough, mark the point as consistent. This requires more matches than the epipolar-line check, but it can be applied without knowing a precise calibration.

Vehicle-Relative Motion Check: Given stereo images taken over time from a vehicle making a known motion (or a motion that can be computed), candidate matches can be evaluated by (1) computing a vehicle-relative x-y-z location from one image pair, (2) tracking the point into a second image pair, (3) computing another location for the point, adding in the known motion, and finally (4) checking to see if the two estimates are approximately equal. If not, the point is either a mistake or on a moving object.

Object-Relative Motion Check: If the motion of the vehicle is not known (and cannot be easily computed), then pairs of images taken at different times can be used to filter out mistakes by (1) tracking points over time, (2) selecting a point in the scene as a reference point, (3) computing the x-y-z locations of all points relative to the reference point, and (4) marking points with stable relative distances as consistent. Moezzi et al have implemented a system based on this approach [9].

two cameras that ar

Figure 5: Left-right consistency check.

Figure 6: Trinocular consistency check for uncalibrated cameras.

We are experimenting with a version of the transitivity check that compares stereo matches over time (see Figure 7). In keeping with our goal of minimizing smoothing, we work directly with the raw intensities at full field resolution and use small correlation windows to perform both the stereo matching from left to right and the optic-flow matching over time. If the two disparity maps D_n and D_{n-1} and the two flow fields M_l and M_r are viewed as maps from pixels to pixels, then for each pixel P, the spatiotemporal transitivity check measures the distance between $M_r(D_n(P))$ and $D_{n-1}(M_l(P))$. If the distance between these two points is within one pixel, P is marked as consistent.

We are still exploring ways of characterizing the effectiveness of this type of filter. As discussed in the next section, we have found that the loop test catches a significant number of mistakes not detected by the left-right check for example.

However, some erroneous matches pass both tests. To catch these mistakes, we are developing a version of the vehicle-relative motion check that uses an estimate of the vehicle's motion and a history of a feature's measured distances to detect mistakes. For example, consider the two scene features A and B in Figure 8a which are viewed by two one-dimensional cameras. The point A projects into aL in the left image and aR in the right image. If the matching system erroneously identifies bR as the match for aL (as shown in Figure 8a), the system produces a fictitious scene point, shown as the hollow point in the upper right corner of the diagram. If the pair of cameras is moved forward (i.e., from left to right) and the matching system persists in matching B to A, as shown in Figure 8b, then the fictitious point appears to move a distance d in the world. If d is significantly larger than the change potentially caused by inaccurate disparity measurements, then the aL-to-bR match is either a mistake or the corresponding point is moving in the scene.

In the next section we describe our experimental system, which was designed to explore these tests and their interactions.

The MIME System

Our purpose in implementing the MIME system was to explore the idea of maximizing the resolution of range data produced by a passive sensor. Our approach has been to minimize the use of explicit or implicit spatial aggregation operations and to recover the beneficial filtering effects of spatial aggregation by using a set of tests that compare the results of multiple matches.

The MIME system is implemented on a Connection Machine. Since it is a research system it is not optimized for speed, rather it is designed to facilitate the comparison of different matching and filtering strategies. As such, it has 20 or 30 top-level switches and parameters for specifying a particular processing configuration. The switches include a switch to apply left-right checking or not, a switch to apply the spatiotemporal check, and a switch to fill in missing data by using the results of a pixel's neighbors to predict a narrow search region. The parameters include the number of y disparities to search for each match (typically between 1 and 3 lines away from the epipolar line), the sizes of the correlation windows, and the sizes of the search regions for motion analysis. (In the next section we briefly describe an implementation of a real-time stereo system that was carefully engineered to run efficiently on SRI's indoor robot,

Figure 8: (a) aL-to-bR matching mistake and the deduced scene point. (b) Implied motion of the deduced scene point caused by a reoccurrence of the mistake.

Flakey.)

Assumptions and Experimental Constraints

Our experimental data was obtained by mounting a pair of black-and-white cameras on a HMMWV vehicle, aligning them manually so their optical axes were approximately parallel, recording the data on VHS videotape as the vehicle was driven on and off road, and finally digitizing sequences of video fields from the tapes. As a result, the epipolar geometry was not known precisely, partly because the relative position of the cameras was not known precisely and partly because there was no attempt to compute lens distortions and the like. Therefore, our matching technique could not rely on precise epipolar geometry, and yet it could constrain the searches to a small number of lines relative to the predicted epipolar line. In general, we and others have found it difficult to maintain precise calibrations as the vehicle bounces along over rocks and ditches, making it desirable to have a system that is capable of working with less constrained imagery.

The overall image intensities vary from one image to the next for several reasons. First, some of the data was gathered with auto-iris lenses, which are designed to cover a wide dynamic range of lighting conditions by automatically adjusting the apertures of the lenses. These lenses work well, except that they are not linked together, so one might be opening while the other one is closing. In addition, their control systems tend to "hunt" for the best settings, causing the intensities to fluctuate continuously. A second reason for intensity differences is that there is no way to completely turn off the automatic gain control (AGC) on our COHU cameras, even though there is a switch on the cameras that says "off." A third reason is that our method of synchronizing the two cameras involves using the video from one camera to drive the second one. This works well, except that it tends to drop the intensity of the initial camera a little because its output is doubly terminated. And finally, the lenses occasionally got dirty, making some parts of an image darker than others.

The bottom line is that we could not rely on the absolute intensity levels, forcing us to use a normalized correlation metric. The normalized metric does a good job of factoring out the gain and offset between the two correlation windows, however, it introduces mistakes because it eliminates the ability to discriminate between possible matches on the basis of absolute intensities. In other words, if the intensity transforms were better constrained, a tighter requirement could be placed on potential matches, which would reduce the chances of incorrect matches. This line of reasoning is identical to the argument for using as much geometric information (e.g., epipolar constraints and a limited range of acceptable object distances) as possible to limit the search areas for matches. There are two benefits. First, the smaller the search area, the faster the system can find the best match. And second, the smaller the search area, the higher the probability is of finding the correct match. Similarly in the intensity domain. The better the image-to-image intensities are known, the faster the matches can be computed and the more likely the correct match will be found.

One last point about our data acquisition, we did not have instruments to measure and record the dynamic motion of the vehicle as the data was obtained. We only knew the approximate speed of the vehicle. As a result, we are not able to analytically align results computed at different times.

System Description

The system is based on three multiple-match filters: the left-right check, the 4-way spatiotemporal check, and the forward motion version of the vehicle-relative check. The last filter has been implemented, but not thoroughly tested.

Given a new image pair, the "complete" system performs the following sequence of operations:

1. For each pixel in the left image, locate the best match in the right image (within the search region, which is typically 85 pixels wide by 3 or 7 pixels high for our camera configuration).

2. For each pixel in the right image, locate the best match in the left image, using the same size search areas as in step 1.

3. Mode filter the y disparities computed for the pixels with matches.

4. Select the left-to-right matches and right-to-left matches, corresponding to the y disparity produced in step 3.

5. Perform the left-right check, marking pixels in the left image as invalid if their left-to-right and right-to-left matches are not inverses of each other (to within one pixel).

6. For each pixel in the left image, locate the best match in the previous image from the left camera and mode filter.

7. For each pixel in the previous left image, locate the best match in the current left image and mode filter.

8. Perform a left-right-type check on the results of steps 6 and 7.

9. Perform steps 6 through 8 on the two most recent right images.

10. Perform the 4-way spatiotemporal check, marking pixels as invalid if the sequence of matches around the loop does not agree (to within one pixel).

11. For each valid pixel in the left image, perform the vehicle-relative check, marking pixels invalid if their motion differs significantly from their expected motion.

12. For each invalid pixel, use the average of the disparities of its neighbors as the center of a small search window within which to look for a possible match.

13. Map the previous statistics into the current left image (or all the images) by applying the motion vectors computed in steps 6 through 9.

14. Update the statistical arrays that have one entry per pixel and keep track of such things as the number of times the pixel feature has been completely consistent in a row.

The first 3 steps provide a way of reducing the height of the search regions to a single line. We introduced the option to perform mode filtering in this refinement process, even though it is a type of smoothing, for two reasons. First, we expect the y disparities to change slowly in an image. And second, the smoothing is not directly being applied to the raw image or the results, rather it is being used to compute an intermediate result that is used to locate the final matches. As mentioned earlier, if the epipolar constraints are known precisely, these steps would not be necessary.

Figure 9 shows an example of the type of sequence processed by the MIME system. The images have a rectangular aspect ratio because they are individual fields digitized from a videotape. They are a sequence of even fields, which are taken 1/30th of a second apart. Figure 10 shows the disparities computed and filtered from 4 image pairs. In this figure, lighter points are closer to the sensor. To emphasize the heights of objects, we transform these raw disparities into ones relative to a horizontal plane, as shown in Figure 11.

The computation of optical flow is performed using SSD correlation followed by a mode filtering step. The use of SSD correlation is justified since the images are taken from the same camera one thirtieth of a second apart, and absolute intensity levels are not expected to change drastically between frames. Mode filtering makes sense in that the flow field resulting from forward motion is expected to change slowly in an image. Figure 12 shows the unfiltered y disparities computed for one of the pairs of images in Figure 9. Figure 13 shows the mode-filtered version of these disparities, which are used to select the row for the best match.

We have introduced a slightly different left-right check than used by previous researchers. The first versions of this filter required that the left-to-right and right-to-left matches to be exact inverses (i.e., to the pixel). However, because of quantization effects, the commonly used version of the test allows the inverse to be within one pixel of the starting pixel, as shown in Figure 14. Our version loosens that constraint a bit more by taking into account the possibility that the right-to-left match may land on pixel that doesn't have a valid right-to-left match. It accepts a pixel in the left image if the matching pixel in the right image *or one of its two neighbors* maps back to within one pixel of the initial point (see Figure 15). This change makes the test more symmetric. It also accepts a few more pixels in the left image as valid. Figure 16 shows the results after the left-right test has been applied. Figure 17 shows the results after both the left-right and spatiotemporal loop tests have been applied.

Figure 18 illustrates a situation in which the left-right test fails to filter out a mistake. Two events conspire to produce this erroneous result. In Figure 18, aL is matched to bR, instead of aR, because something has altered A's appearance in the right image (e.g., A may be partially occluded). Similarly, bR is incorrectly matched to aL because B's appearance in the left image is different. As a result of these two mistakes, the left-right test erroneously accepts the aL-to-bR match.

Figure 19 and Figure 20 show an example of this type of mistake. Figure 19 shows the context of the mistake. It occurs on the front edge of a deep rut. Figure 20 is a blow-up of the images around the mistake. The correlation window on the right in the left image is mistakenly matched to the left window in the right image, instead of the right one. The happens because the window straddles an occlusion edge between two regions at different depths, the front edge of the rut and the back of the rut. In the right image, these two subwindows have different disparities, so that there is no coherent window matching the one in the left image. As a result, the matching system finds a completely new window in the right image that looks like the one in the left. This new window is also along the edge of the rut, causing the same problem for the matching procedure when it tries to match from right to left. Unfortunately, but not too surprisingly, this right-to-left match happens to find the original window in the left image as its best match (instead of the left window in the left image). As a result, the mistake passes the left-right test.

Figure 21 shows another example of how a mistake can pass the left-right test. The X on the left of the left image is not in the field of the view of the right camera. Therefore, the best match for it is the only visible X in the right image. If the search from right to left happens to prefer the left X, as indicated in the diagram, then this pair of mistakes leads to a mismatch that is not caught by the left-right check. Again, it took two events to produce the problem.

Figure 22 shows an example of how a mistake that is missed by the left-right check can also be missed by

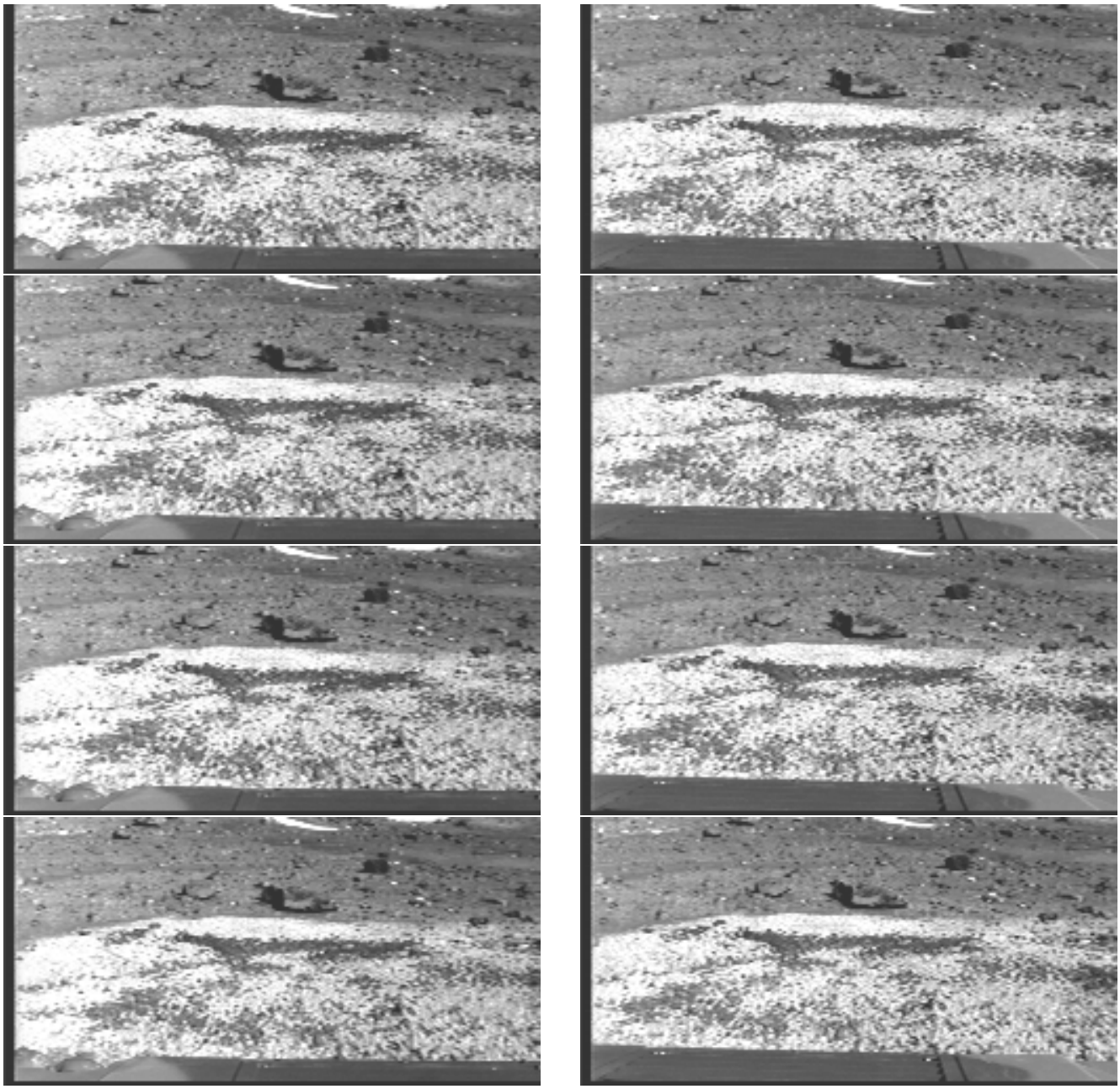

Figure 9: Sequence of stereo pairs of a Martian-type scene.

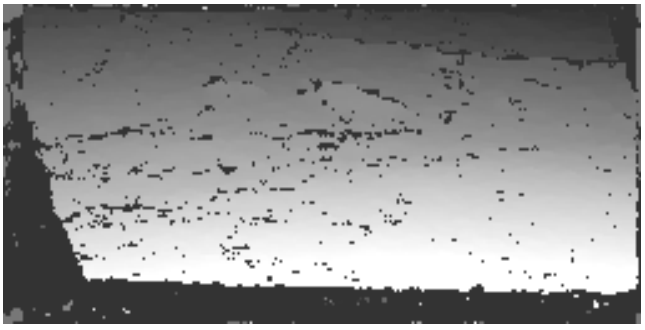

Figure 10: Temporally filtered stereo disparities for a pair of images from Figure 9.

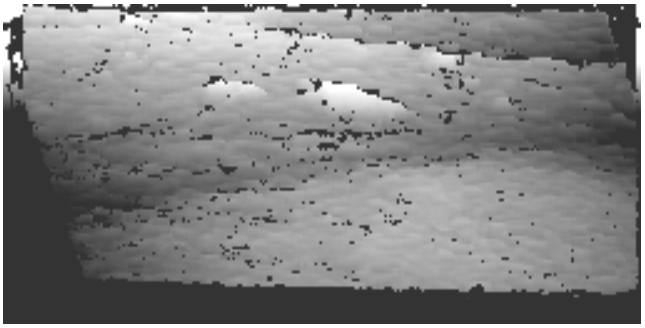

Figure 11: A skewed version of the disparities shown in Figure 10.

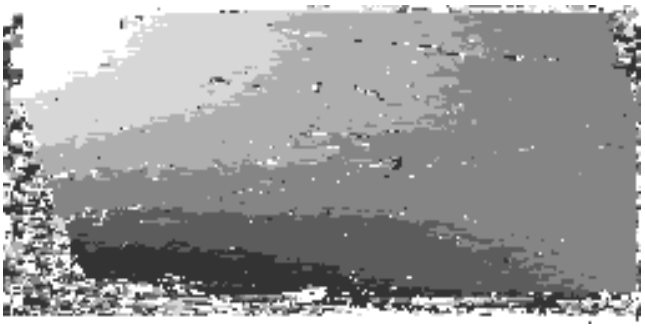

Figure 12: Unfiltered y disparities.

Figure 13: Mode filtered y disparities.

Figure 15: New left-right consistency check.

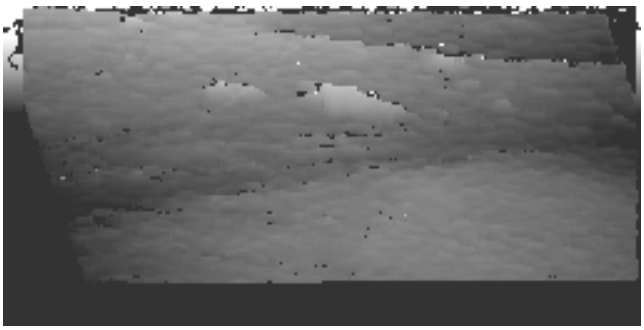

Figure 16: Disparity results after the left-right check has been applied.

Figure 18: A pair of mistakes that conspire to pass the left-right check.

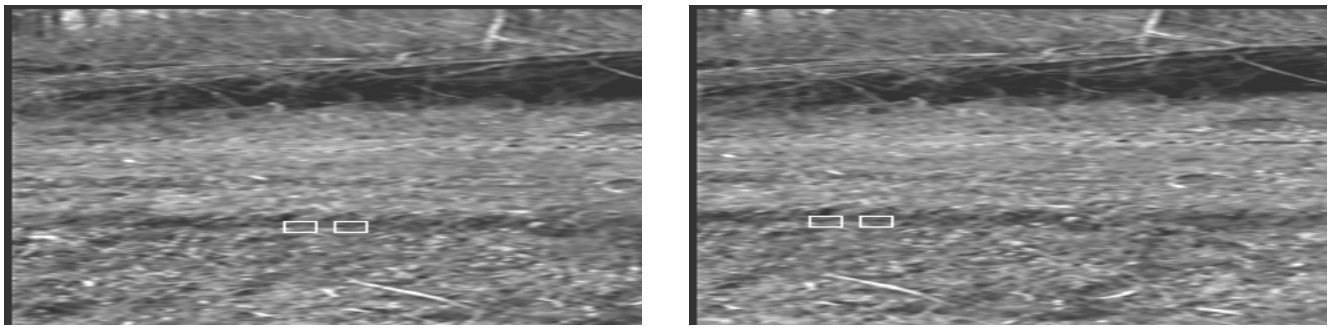

Figure 19: A pixel feature that erroneously passes the left-right check.

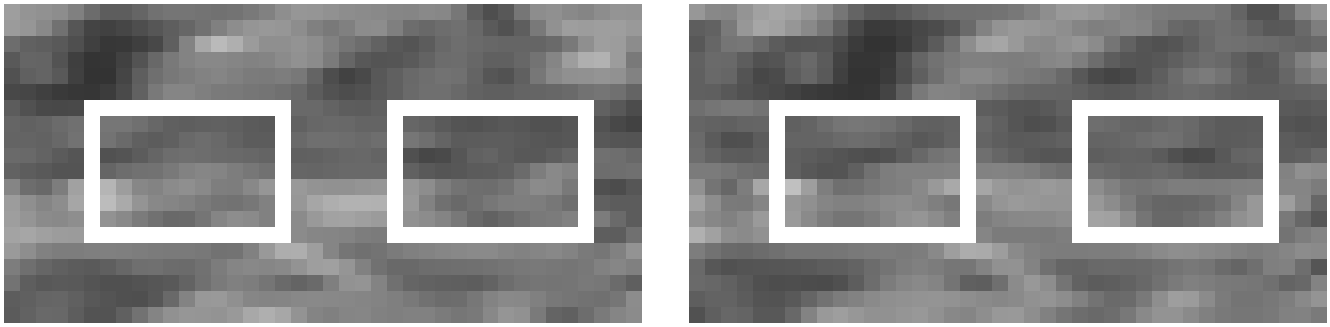

Figure 20: A blown-up version of the mistake shown in Figure 19.

Figure 21: A pair of mistakes, one of which is caused by a feature being out of the field of view of the other image.

the spatiotemporal check. This example is similar to the example in Figure 18. If the cause of the mistakes in first pair of images (e.g., partial occlusions) persists over time, the spatiotemporal test may also miss the mistake. However, if the relative positions of the cameras and scene features change significantly, the test is likely to pick up the inconsistency and mark the results as invalid.

We added the option to use the results of the neighbors of an invalid point to give it a second chance to find a compatible match. Given an estimate of its disparity, the system searches a small region about that estimate for the best match. If that match passes all the filters, it is marked as valid and included in the reported results. In our images, this process fills in 15 to 20% of the pixels initially marked as inconsistent by the left-right or spatiotemporal tests.

An open question about this process is the selection of the size of the search region about a suggested disparity. If the region is as large as the region used in step 1, the system will probably return the same matches that failed the test in step 5. At the other extreme, if the region is reduced to a single pixel, the system would report that pixel as the match and it would automatically pass all tests because, by definition, it is the best match in the region. So the question is how to reduce the size of the search region in a principled way so that it limits the search to an appropriate sized region without invalidating the evaluation procedures. We arbitrarily used regions that were 10 pixels wide in our experiments.

Experimental Results

In order to characterize the effectiveness of the various tests within

the MIME system, we have applied it to several different image sequences with several different parameter and switch settings. Figure 23 shows a typical set of statistics produced by the system when both the left-right and spatiotemporal tests are applied. For this particular sequence, the vehicle was turning to the left as it approached a deep rut in a relatively flat field (see Figure 24). Figure 25 shows the region of the image in which we gathered statistics. The rest of the image is out of the field of view of the right image. The left-right test marks an average of 12.1% of the pixels in the left image as inconsistent.

The motion tracking procedure is virtually perfect and the up-down version of the left-right check has no trouble verifying the flow vectors. The up-down check marks fewer than 3 pixels in every 1000 as inconsistent.

The spatiotemporal check marks an additional 4.3% of the points as possible errors, reducing the average number of "consistent" points in the left image to be 83.5%. When these tests are convolved together over 4 image pairs, the number of completely compatible points is 70.0%. When the suggestion procedure is used to fill in missing data, this number increase a few percentage points to about 74.3%. The number of gross errors passing all the tests is on the order of 10 to 20 pixels per image.

Flakey's Stereo system

For several years, Flakey, SRI's indoor robot, has used ultrasonic sensors and a structured-light sensor to locate potential obstacles in its path. These sensors, however, have several limitations. For example, sonar cannot detect thin objects, such as table legs. And the structured-light sensor can only measures distances to points that are in a particular plane and are close to the sensor. Therefore, in order to increase both Flakey's sensing resolution and sensing range, we have implemented a streamlined version of the MIME stereo system on the on-board Sparc10 processor. The resulting system produces a 105-by-240 range image in .4sec. In addition, Flakey's control system can select horizontal stripes from this image to be recomputed at a higher rate. For example, it can compute 20 rows of the range image at 30hertz.

Flakey uses a dual lens system to project a pair of images into a single video field. We originally installed

Figure 22: A mistake that passes the spatiotemporal test due to recurring errors.

Image Pair	Left-right Consistent	Forward-back Consistent	Loop Consistent	Completely Consistent for last four pairs	Completely Consistent w/ prediction
1	90.27	–	–		
2	88.83	100.00	82.36		
3	87.57	99.96	80.25		
4	88.11	99.97	77.39	70.04	70.04
5	89.55	99.98	77.54	66.32	66.71
6	89.83	100.00	84.43	66.68	67.55
7	88.33	99.99	82.62	67.41	69.13
8	89.92	100.00	82.05	69.82	72.30
9	89.32	100.00	85.22	75.13	78.91
10	87.19	100.00	77.76	70.03	74.94
11	86.41	99.89	77.77	66.88	72.83
12	86.43	100.00	75.37	65.16	71.28
13	87.37	99.98	75.87	62.91	68.70
14	86.81	100.00	82.82	64.72	70.74
15	86.72	99.82	78.55	64.00	69.93
16	86.32	100.00	82.29	67.52	72.82
17	86.45	100.00	77.49	70.51	74.77
18	84.76	100.00	76.89	67.15	72.84
19	85.37	100.00	78.67	67.80	74.15

Figure 23: Table of consistent pixels over time.

Figure 24: A scene with a deep rut crossing from left to right.

Figure 25: Region from the left image in Figure 24 from which the statistics in Figure 23 were computed.

the optical 2-to-1 lens with the hope that it would minimize the overall change in intensities from one image to the next. However, the lens has such strong vignetting problems that both half images are significantly darker at the edges than they are at the middle. We tried two separate cameras, but our inability to turn off their automatic gain control has made them difficult use, especially in buildings with a number of specular fixtures and bright lights.

In order to run the stereo matching algorithm as fast as possible, we did the following:

- Subsampled the images from left to right, reducing 315 columns down to 105. This reduced the range of disparities from about 50 to 16, which can be computed efficiently in the Sparc10's registers.
- Simplified the correlation metric to be the sum of squared differences, which can be computed significantly faster than normalized cross correlation. We computed the sums incrementally by sliding the correlation window over the search region.
- Deleted the spatiotemporal consistency test, but kept the left-right check to validate matches. Therefore, each match is performed twice. In addition, we added an interest-type operator to flag points in the left image that are unlikely to produce reliable results, because they lack texture.

Flakey merges the stereo data with its sonar data, and then plans its paths in the same way it always has. It uses a two-dimensional, robot-centered map to keep track of potential obstacles and landmarks.

In the future, we plan to add behaviors to Flakey so that it can plan data-gathering maneuvers to examine unmeasured regions or closely inspect points of special interest.

Conclusion and Future Work

In this paper we have (1) introduced a spatiotemporal consistency check for evaluating stereo results and (2) incorporated it into a system for integrating range data over time. We are in the process of characterizing the utility of this approach and its relationship to other similar techniques.

One observation we've made is that outdoor natural scenes contain sufficient texture to support dense correlation matching. For example, our matching technique locates matches for 70 to 80% of the pixels in most images, even though we use relatively small correlation windows. The 20 to 30% mistakes and no-data regions are caused by such things as bland areas, repeated patterns, and occlusions. Even though the mistakes represent a relatively small fraction of the results, most tasks require significantly more complete and more reliable data. For example, a navigation system cannot recommend driving over areas containing a unmeasured regions or unexplained points floating above the ground. Therefore, there is a need for evaluation techniques to assign con-

fidences to individual pixel features and for higher-level sensor control strategies to reexamine no-data or questionable regions.

The multiple-match consistency checking procedures discussed in this paper provide a form of "structural filtering" that we prefer over such techniques as thresholding correlation values because they are based on distance measurements for which it is relatively easy to determine appropriate thresholds—and there are always thresholds! We view the spatiotemporal filtering technique that we discussed to be one of several techniques from which a stereo system can be constructed. One benefit of it is that it provides a natural way to integrate range information over time, which opens up the possibility of additional temporal analysis of stereo results.

In the future we plan to complete the characterization of this approach, explore higher-level explanations of the pixels marked invalid by the consistency checks (e.g., produce explanations in terms of occlusions and bland areas), and investigate techniques for combining the results of multiple "binary" consistency checks to form scenes models capable of answering such questions as "What are navigable areas in front of the vehicle?" and "Where are there preliminary indications of a possible obstacle that should be examined more closely?"

References

[1] Ayache, N., and F. Lustman, "Fast and Reliable Passive Trinocular Stereovision," *Int'l Conf. on Computer Vision,* June 1987.

[2] Baltsavias, E.P., "Multiphoto Geometrically Constrained Matching," Institute for Geodosy and Photogrammetry, Zurick, Switzerland, December 1991.

[3] Bolles, R.C., H.H. Baker, and M.J. Hannah, "The JISCT Stereo Evaluation," SRI International Report, January 1993.

[4] Bolles, R.C., H.H. Baker, and M.J. Hannah, "The JISCT Stereo Evaluation," *Proc. ARPA Image Understanding Workshop*, Washington, D.C., pp. 263-274, April 1993.

[5] Dhond, U.R., and J.K. Aggarwal, "A Cost-Benefit Analysis of a Third Camera for Stereo Correspondence," *Int'l Jnl. of Computer Vision,* Vol. 6, No. 1, pp. 39-58, April 1991.

[6] Fua, P.V., "A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features," *Machine Vision and Applications*, 1991.

[7] Hannah, M.J., "A System for Digital Stereo Image Matching," *Photogrammetric Engineering and Remote Sensing,* Vol. 55, No. 12, pp. 1765-1770, December 1989.

[8] Kanade, T., M. Okutomi, T. Nakahara, "A Multiple-baseline Stereo Method," *Proc. Image Understanding Workshop,* San Diego, Ca, pp. 409-426, January 1992.

[9] Moezzi, S., S.L. Bartlett, and T.E. Weymouth, "The Camera Stability Problem and Dynamic Stereo Vision," *Proc. Computer Vision & Pattern Recognition Conf.,* pp. 109-113, 1991.

[10] Moravec, H.P., "Visual Mapping by a Robot Rover," *Proc. Int'l Joint Conf. on Artificial Intelligence,* Tokyo, Japan, pp.598-600, August 1979.

[11] Nishihara, H.K., "Practical Real-Time Imaging Stereo Matcher," Opt. Eng., **23**, 5, 536-545, Sept.-Oct. 1984. Also in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, edited by M.A. Fischler and O.Firschein, Morgan Kaufmann, Los Altos, 1987.

[12] Pollard, S.B., J.E.W. Mayhew, and J.P. Frisby, "PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit," *Perception,* Vol. 14, pp. 449-470, 1981.

[13] Szeliski, R., "Bayesian Modeling of Uncertainty in Low-Level Vision," *Int'l Jnl of Computer Vision,* Vol. 5, No. 3, pp. 271-301, December 1990.

[14] Woodfill, J., *Motion Vision and Tracking for Robots in Dynamic, Unstructured Environments.* PhD thesis, Stanford University, August 1992.

[15] Woodfill, J. and R. Zabih, "An algorithm for real-time tracking of non-rigid objects," *Proceedings of AAAI-91, Anaheim, CA.*, pages 718–723. The MIT Press, 1991.

[16] Yoshida, K. and S. Hirose, "Real-time Stereo Vision with Multiple Arrayed Camera," *IEEE Conf. on Robotics & Automation,* 1992.