2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Techniques and Applications for Persistent Backgrounding in a Humanoid Torso Robot

David Walker Duhon, Jerod J. Weinman, Erik Learned-Miller

***Abstract*— One of the most basic capabilities for an agent with a vision system is to recognize its own surroundings. Yet surprisingly, despite the ease of doing so, many robots store little or no record of their own visual surroundings. This paper explores the utility of keeping the simplest possible persistent record of the environment of a stationary torso robot, in the form of a collection of images captured from various pan-tilt angles around the robot. We demonstrate that this particularly simple process of storing background images can be useful for a variety of tasks, and can relieve the system designer of certain requirements as well. We explore three uses for such a record: auto-calibration, novel object detection with a moving camera, and developing attentional saliency maps.**

I. Introduction

Background subtraction is a simple technique for detecting changes in image or video data, and hence for detecting the appearance of novel objects, the disappearance of objects, the motion of objects, or changing imaging conditions. Background subtraction is among the most basic and widespread techniques used in computer vision, and the basic algorithm is easy to implement. As such, it has found widespread use in robotics and other systems, especially in systems with stationary cameras [4], [1].

While many robotics systems make use of background subtraction in some fashion, our experience suggests that most robotics platforms do not use backgrounding to its fullest potentional. In particular, when it is used, it frequently comes with some or all of the following limitations:

- it is only used when cameras are in a fixed position;
- the user of the robot is required to explicitly and manually re-acquire a background image prior to each use of background subtraction (any background images are thrown out at the end of an experimental session, or upon powering down the robot); and
- backgrounding is not used for any purpose beyond the immediate goal of detecting motion or novel objects.

We argue in this paper that backgrounding techniques can form a widely useful subsystem in robotics systems, especially non-mobile systems such as humanoid torsos and in mobile systems that spend large amounts of time in the same environment, such as a particular lab.

The motivation for our system stems from the human visual system–when humans awake from sleep, they can immediately orient themselves by the familiar structures and objects which they see around themselves. The visual memory of the environment around them gives them immediate information about their current situation including their physical orientation, the time of day (estimated by differences in lighting), new people or objects that have appeared in the scene, and a visual "index" into prior memories, visual and otherwise, associated with their current location.

Our goal is to endow our humanoid torso with similar capabilities. In particular, our goals in this paper are to use a backgrounding system to

- perform certain simple callibration procedures for the robot on power-up;
- perform basic novel-object detection tasks, *even in the presence of camera motion*; and
- gather useful statistics and build a model of areas of the probable appearance of new objects in the environment.

While we focus on just these three tasks in this paper, we believe a well-designed backgrounding system can also provide many other basic capabilities, such as determining time of day, playing a role in color constancy (by providing a consistent "reference" for relative color measurements under varying lighting conditions) and self-diagnosis (e.g. detecting sensor drift). We refer to the storage of a collection of background images on a permanent storage device and software for using these images to accomplish various tasks, collectively, as *persistent backgrounding*.

The following section introduces the hardware setup and discusses the first of the three applications: the use of SIFT features to localize the position of the BiSight head of our robot. Section III then discusses a technique that enables the differentiation of foreground and background using a moving camera. Finally, section IV introduces an application that allows us to model stable versus unstable regions in the humanoid torso's visual space. All of these applications operate on the same set of data, the persistent background of images.

II. Localization of the BiSight Head

Our application platform is the UMass humanoid torso robot, Dexter (Figure 1). Dexter is equipped with two seven degree-of-freedom Barrett Whole Arm Manipulators (WAMs), each with a three-fingered Barrett hand mounted at its wrist. More relevant for our purposes is Dexter's TRC BiSight head (Figure 2). The BiSight camera system has four degrees of freedom, two of which lie in the pan and tilt angles for the mount, with the other two consisting of the horizontal verge angles for each camera. The horizontal verge angles for each camera are fixed at 0 radians so that the optical axes are parallel, and so the task is reduced to

The authors are with the Department of Computer Science, University of Massachusetts Amherst, Amherst, MA 01002. Email wduhon,weinman,elm@cs.umass.edu.

determining the pan and tilt angles of the mount given the current image.

For our first application the task is to utilize a visual background model to infer the position of the BiSight mount given a fresh image. We have two motivations for solving this problem.

First, the process of zeroing the BiSight is tedious and slow. Upon powering up the robot, a special program must be invoked which slowly moves the BiSight until it is at the extreme of its tilt angle, and subsequently moves until it is at the extreme of the pan angle. This must be done sufficiently slowly so that there is no risk of gear damage when the BiSight reaches its limit of motion and encounters a physical stop. After reaching the physical limits of its pan and tilt range, the counters for these variables are set to their initial values. This process takes more than 30 seconds and is done many times a day when the robot is in regular use. Our first motivation is to bypass this procedure completely and initialize the pan-tilt variables from the background images by simply "recognizing" the position and angle of the head from what it is looking at.

Second, while robotic video systems can be built with encoders that sense absolute position and hence do not need explicit initialization, there are typically trade-offs in cost and accuracy that come with such devices. In addition, encoders and other hardware with moving parts are prone to failure (we have had multiple encoder failures in our own lab). Thus, a second motivation is to replace the functionality of encoders with parts that serve multiple purposes, namely video cameras with pan-tilt capability. This reduces the number of moving parts of the system and will hopefully lead to more robust design in the long term.

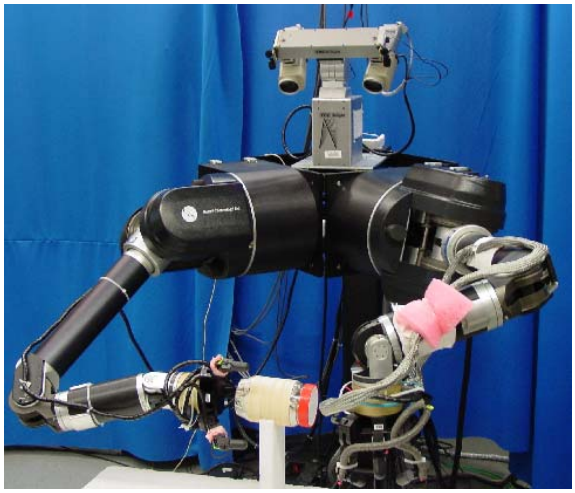

Fig. 1. Dexter

We start with a set of background images taken over a range of known BiSight pan and tilt values. These values are determined using the existing calibration procedure. Now, when given a new image, the idea is to utilize feature correspondences between the new image and the background images to infer the new image's position. While there are many choices of features that can be used effectively for this purpose, we have chosen SIFT features. SIFT features are a suitable choice both because they demonstrate a high degree of affine invariance and because they are highly distinctive, reducing the possibility of false matches [6].

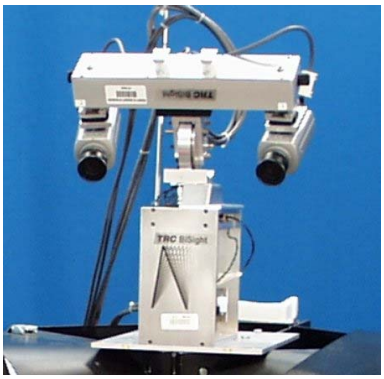

Fig. 2. BiSight Head

As a first approximation, by varying pan and tilt on the BiSight we would seem to be mapping out an image sphere. The relationship between BiSight pan/tilt angle and pixel column/row of a particular feature would then be very simple. By computing the spherical warp of the background images and the new image we should be able to position the new image with a high degree of precision among the panorama of background images.

The set of acquired background images, however, does not form a neat spherical image in our system. In particular, the rotational center of the BiSight is offset from the optical center of each camera causing the camera to translate as the BiSight pans. However, helping to some extent is the fact that the tilt axes of the BiSight and of the cameras are approximately aligned. For the analysis that follows, we make the assumption that this alignment is exact.

Despite the translation of the camera during panning, it is easy to derive from stored background images and a single newly acquired image reasonably accurate estimates of the "true" pan/tilt angles. The basic algorithm is simple: when given a new image, find the two closest background images in the set as judged by the number of feature correspondences. Next, find SIFT features that are present in all three images (the two background images and the new "query" image). In all three images, a given feature will be at a certain horizontal and vertical pixel offset, corresponding to a certain horizontal and vertical angle displacement from the camera's optical axis. To determine pan angle, we simply linearly interpolate the pan of the two known background images to get the pan of the new image, weighting the contribution of each background image by its horizontal angle distance to the new image. Determining tilt is similar, though an extra step is needed, as will be explained below. First to justify the algorithm for pan, examine Figure 3.

Figure 3 shows a bird's eye view of the BiSight. In the figure, A represents the rotational center of the BiSight, F

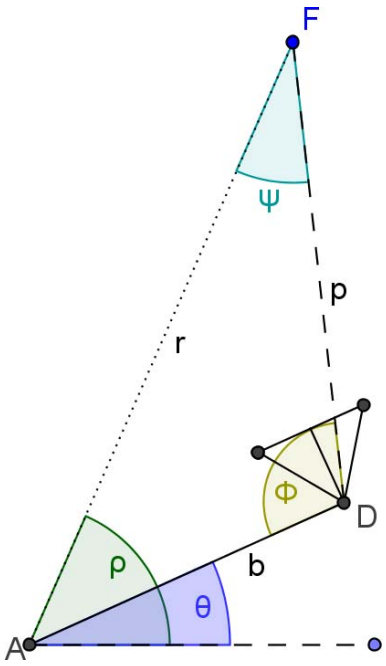

Fig. 3. A model of BiSight panning and its relation to the optical center of the camera used to acquire images. See text for details.

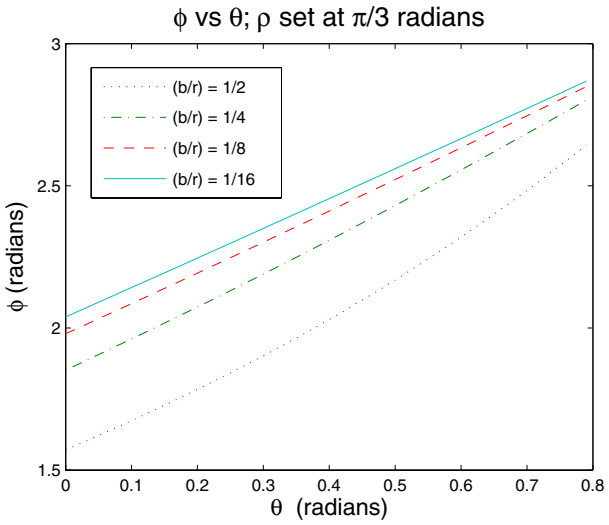

Fig. 4. Graphs showing the nearly linear relationship between ϕ and the bisight mount's pan angle (θ), justifying our linear interpolation method of identifying pan angle from horizontal image position of a located feature.

represents the position of a feature in space projected onto the rotational plane, while AD represents the offset from the rotational center to the optical center of the right camera of the BiSight. Letting θ represent pan, by the Law of Sines:

$$\begin{aligned}\frac{r}{\sin\phi} &= \frac{b}{\sin\psi} \\ &= \frac{b}{\sin(2\pi-(\rho-\theta)-\phi)} \\ &= \frac{b}{\sin((\rho-\theta)+\phi)} \\ &= \frac{b}{\sin(\rho-\theta)\cos(\phi)+\sin(\phi)cos(\rho-\theta)}.\end{aligned} \tag{1}$$

ϕ, which is approximately linearly related to the horizontal position of a feature in an image (see Figure 3), can then be expressed as a function of θ:

$$\begin{aligned}\left(\frac{b}{r}\right)\sin\phi &= \sin(\rho-\theta)\cos(\phi)+\sin(\phi)\cos(\rho-\theta) \\ 0 &= \sin(\rho-\theta)\cos(\phi)+\left(\frac{b}{r}-\cos(\rho-\theta)\right)\sin\phi \\ \tan\phi &= \frac{\sin(\rho-\theta)}{\left(\frac{b}{r}-\cos(\rho-\theta)\right)} \\ \phi &= \arctan\left(\frac{\sin(\rho-\theta)}{\left(\frac{b}{r}-\cos(\rho-\theta)\right)}\right).\end{aligned} \tag{2}$$

Figure 4 shows that for $r >> b$, pan varies almost linearly with ϕ, justifying the use of linear interpolation. Also, for relatively small r, local linearity is a fair approximation.

For tilt the situation is slightly different because the vertical angle displacement of a feature depends on both pan and tilt.

Referring to Figure 5, by the Law of Cosines:

$$w^2 = b^2+r^2-2br\cos(\rho-\theta). \tag{3}$$

Letting σ represent tilt, we also have:

$$\sigma+\lambda = \arctan(\frac{h}{w}). \tag{4}$$

Combining these two:

$$\sigma = \arctan\left(\frac{h}{\sqrt{b^2+r^2-2br\cos(\rho-\theta)}}\right)-\lambda. \tag{5}$$

Tilt is clearly a linear function of λ, the vertical angle displacement, but tilt also depends nonlinearly on changes in pan. Letting $\psi = \arctan\left(\frac{h}{\sqrt{b^2+r^2-2br\cos(\rho-\theta)}}\right)$, we can determine ψ for the background images using the sum $\psi = \sigma+\lambda$. Using both background images we can linearly interpolate to find an estimate of ψ_{new} for the new image. With ψ_{new} in hand we can then use the verticle angle displacement to determine σ_{new} for the new image. Thus, the tilt estimate for the new image becomes:

$$\sigma_{new} = \psi_{new}-\lambda. \tag{6}$$

Because ψ is approximately a linear function of pan Figure 6, the linear interpolation technique yields reasonable estimates.

A. Results for Pan/Tilt Determination

To demonstrate the technique, we took 20 images across a range of random BiSight pan and tilt values. We wanted to see how close the pan/tilt values fed to the controller would match those inferred from a small database of background

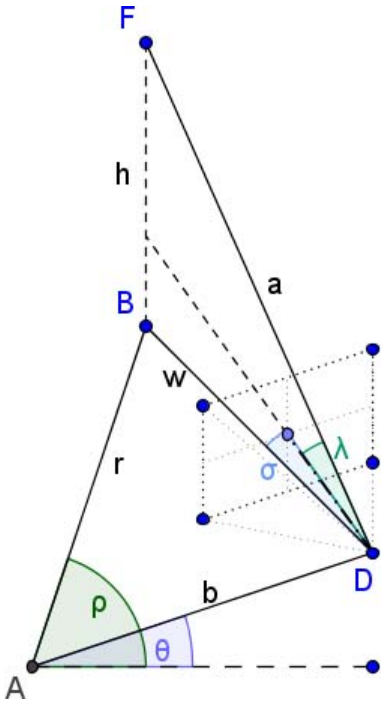

Fig. 5. The geometry of the bisight tilt. σ represents the angle of tilt, while λ represents the additional vertical angle displacement of the ray to the feature F. ψ (not shown) is the sum of σ and λ.

Ground		Found		Difference	
Pan	Tilt	Pan	Tilt	Pan	Tilt
-40.5	-8.8	-40.0	-8.9	-0.4	0.1
-42.0	17.2	-41.8	17.2	-0.2	0.0
-37.2	-3.7	-36.9	-3.7	-0.3	-0.0
-38.5	1.8	-38.3	1.9	-0.2	-0.1
.....	.....	.....	.....	.....	.....
32.4	23.8	32.7	23.9	-0.3	-0.1
			mean:	0.19	0.08

TABLE I

TABLE SHOWING GROUND TRUTH AND ESTIMATED PAN AND TILT ANGLES, USING COMPUTATIONS VIA BACKGROUND IMAGES. ANGLES ARE SHOWN IN DEGREES.

images. Our database consisted of 56 images with the range for pan from -0.8 to 0.75 radians and the range for tilt from -0.4 to 0.75 radians, with each image spaced apart by approximately 0.2 radians in both pan and tilt directions. The range for the 20 random images was the same. This range represents the typical working range of our BiSight.

Using our technique we were able to come within one degree of ground truth in all 20 test images. Typically the difference in both pan and tilt was on the order of a few tenths of a degree (see table below).

III. Backgrounding with a Moving Camera

Many applications of computer vision call for a division of a scene into a background and a foreground. The background consists of areas in the scene that either are stable or vary in a predictable way. The foreground areas of a scene are those that cannot be explained from the background model.

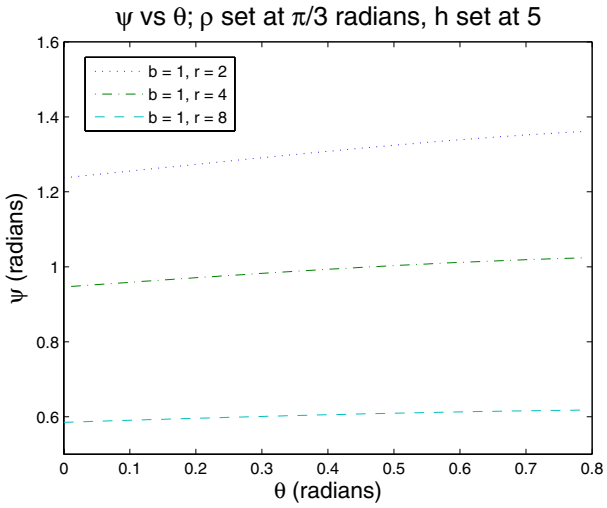

Fig. 6. Graph showing approximately linear relationship between ψ and the bisight mount's pan angle (θ) (see text).

In a typical implementation the camera is fixed so that each image pixel corresponds to the same ray in space. While this simplifies the problem considerably it also places artificial limitations on robotic platforms equipped with a moving camera. The camera may be needed to perform some other task other than backgrounding, such as guiding a reach or tracking a moving object, thus making it impossible to use standard techniques. At the same time, knowledge of changes in the visual scene is no less desirable. What we want is the ability to move the camera during a task while simultaneously monitoring changes in the visual scene.

The basic idea behind the technique is to use a background consisting of many images taken at given camera positions. In principle, the idea is simple, but there are two difficulties that can arise. One, the given camera position is only known to a certain precision. This may be due to fundamental precision limits in the BiSight, calibration errors in the camera, a time lag between the reporting of camera position and capturing the image, and in our case a shaky camera verge controller. Two, even if the camera position were known precisely, in our setup the changes in the visual scene as the camera moves depend upon the depth of objects in the scene. Such a difficulty would not arise if our camera could pan and tilt about its optical center [7], [8], but alas that is not the configuration we are working with. In other words, we must address the issue of parallax caused by the translation of the camera as it rotates.

The first step of the algorithm is to approximately align the new image (Figure 7, first column) with a stored background. To do this we pick the background image that has pan/tilt coordinates nearest to those of the new image (Figure 7, second column). Given the respective pan/tilt values of the background image and the new image, a rough estimate can then be made of the horizontal and vertical pixel offsets that will align the images as closely as possible. Next, we try to refine our estimate by searching for correspondences

using normalized cross correlation between random patches in the new image and those of the background image. This step is very quick, as our initial estimate should be fairly good, so the search for matches can take place in a restricted range; moreover it is sufficient to use a sample of only 1000 patches. Of the 1000 matches found, the mode of horizontal and vertical pixel displacement pairs is chosen as the best global estimate.

After alignment, the new image is compared to the background image. Because the new image and the background image were taken from slightly different camera positions, there will be some disparities in the images due to differences in depths of objects in the scene, even if no new objects are present in the scene. The third column of Figure 7 shows binarized difference images after the best global match of the images in the first two columns has been made. In each row, the only new object in the image is the human figure near the center of the image. The simple difference image contains both gaps that should be shown as forground but are not (missing portions in silhouette of the person) and extraneous detections that represent false detected novel objects.

There are a number of approaches for dealing with these disparities before performing simple background subtraction. One approach is to use stereo or other cues to estimate the depth of objects in the image, and use this depth and the stored images to try to estimate the proper appearance of an image from the newly acquired imaging position. If this image can be generated accurately (in other words, if we can infer what the background image should look like from a new position), then traditional background subtraction can be applied to understand image changes. Unfortunately the process of estimating the appearance of an image from a new point of view is extremely difficult. We take an alternative approach which largely avoids the problem of the exact computation of an image from a new point of view.

Our method, which depends upon detecting differences in a scene based upon images from two slightly different points of view, relies on the following observation: even though there is no single simple transformation on one image which will accurately reproduce the other image (due to disparities), *each patch of a new image is likely to have a close match in one of the stored background images*, since disparities frequently do not disrupt local correspondences.

To implement a matching technique based upon this observation, for each pixel in the new image, we search within a 3x3 pixel area of the background image for the closest match in terms of sum of squared differences (SSD) of local patches. After this search, the lowest valued SSD found is set to be the distance between the new image and the background image at that pixel. If this SSD is low then there is agreement between the images at that pixel and it is likely to be background, while a high SSD indicates incompatibility between the images at that pixel making it likely that the pixel is part of a foreground object. The binarized version of these images is shown in the fourth column of Figure 7.

Finally, we take the results of this "local matching" strategy and use a discriminative Markov random field, as described below, to improve the results by leveraging the spatial continuity of foreground and background patches. This is a principled approach for incorporating information about the mismatch of the images at each pixel and its neighbors to reassess whether a pixel is a match or not.

Let $\mathbf{w}$ represent the stored background image and $\mathbf{x}$ represent the aligned input observation. The corresponding labels $\mathbf{y}$ indicate whether a particular image location is predicted to be foreground. To make such predictions, we use a discriminative Markov field [12], [11],

$$p\left(\mathbf{y} \mid \mathbf{w}, \mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\gamma}\right) = \frac{1}{Z\left(\mathbf{w}, \mathbf{x}\right)} \exp\left\{U\left(\mathbf{y}, \mathbf{w}, \mathbf{x}; \boldsymbol{\theta}, \boldsymbol{\gamma}\right)\right\} \quad (7)$$

where $Z\left(\mathbf{w}, \mathbf{x}\right)$ ensures the probability distribution is normalized. Inside the exponent is an energy function U representing the compatibility between the segmentation hypothesis and the observations. We decompose this into two terms: local terms, which measure the compatibility between image features and label for an individual pixel, and interaction terms which measure the compatibility between a pair of neighboring labels. Specifically,

$$\begin{aligned} U\left(\mathbf{y}, \mathbf{w}, \mathbf{x}; \boldsymbol{\theta}, \boldsymbol{\gamma}\right) &= \sum_i \boldsymbol{\theta}\left(y_i\right) \cdot F_i\left(\mathbf{w}, \mathbf{x}\right) + \\ & \quad \sum_{i \sim j} \boldsymbol{\gamma}\left(y_i, y_j\right). \end{aligned} \quad (8)$$

The vector-valued function F_i returns features of the background and input image at pixel location i. Here, the feature is the SSD between the intensity channel of the two images, calculated as described above, as well as a constant bias term. The class-specific parameters $\boldsymbol{\theta}\left(y\right)$ act as weights on these features. For contextual interaction, the parameters $\boldsymbol{\gamma}$ model a bias for certain neighboring labels y_i and y_j. If these interaction parameters $\boldsymbol{\gamma}$ are all zero, only local information is used; this is equivalent to classification by logistic regression.

Estimating the parameters for the model involves finding the values of $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ that maximize their posterior probability, given a prior $p\left(\boldsymbol{\theta}, \boldsymbol{\gamma}\right)$ and some labeled training data $\mathcal{D} = \left\{\left(\mathbf{y}^{(k)}, \mathbf{w}^{(k)}, \mathbf{x}^{(k)}\right)\right\}_k$. The log posterior objective function is

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\gamma}; \mathcal{D}\right) &= \sum_k \left[U\left(\mathbf{y}^{(k)}, \mathbf{w}^{(k)}, \mathbf{x}^{(k)}\right) - \right. \\ & \quad \left. \log Z\left(\mathbf{w}^{(k)} \mathbf{x}^{(k)}\right)\right] - \quad (9) \\ & \quad \alpha \left\|\boldsymbol{\theta}\right\|_1 - \beta \left\|\boldsymbol{\gamma}\right\|_1, \quad (10) \end{aligned}$$

corresponding to using a Laplacian prior for $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, which we assume are independent *a priori* with parameters α and β. The objective is convex and thus may be optimized with standard techniques guaranteed to find a global optimum. Given the parameter values and and the image features, we predict the most likely labeling

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\gamma}\right) \quad (11)$$

Unfortunately, the learning step and the prediction are intractable due to exponential sums or search spaces. We

approximate these using the standard sum-product and max-product loopy belief propagation algorithms [10].

A. Results of Backgrounding

The fifth column of Figure 7 shows the final results of the foreground/background processing resulting from the discriminative Markov random field. In the first two rows, each step, both the local matching analysis and the Markov random field method, seem to help improve the foreground/background segmentation. The results in the first two rows are good, with only minor extra detections in the second case, and good filling of the foreground figure in both cases. The third row shows a more problematic example, caused by a dramatic lighting change due to sunshine coming in the window. The raw input to our local matching method is simply not good enough, and we need to incorporate specific techniques to deal with strong lighting changes before handling cases such as this. Still, we feel our techniques are quite good given the extremely simple nature of the stored data, namely a simple set of background images taken without special preparation of the environment.

IV. Environment Over Time

The final use of our persistent backgrounding system is to develop maps of activity in the visible environment of our humanoid torso robot. Such maps have been developed based upon detections of specific types of objects, as in [5] and for foreground/background segmentation with stationary cameras [3]. We are not aware of this having been done with a translating camera. Since we have a method for easily placing each acquired image in a global framework and classifying images into regions of foreground and background, it is a trivial matter to accumulate statistics about frequency of appearances of foreground objects in the environment.

Figure 8 shows the results after about an hour of monitoring background and foreground in the lab. Few objects in the scene were moved during the period so that the foreground consists mostly of people as they move about the lab. As might be expected, foreground objects are clustered around the middle of the scene rather than at the extreme pan angles, since users typically wish to interact with the robot in its central workspace. Also, because The BiSight platform is elevated, more foreground activity occurs at the higher tilt angles. At low tilt angles the BiSight is gazing above most of the activity in the room, and this results in a lack of changes in that portion of the scene.

One of our immediate goals is to incorporate these activity frequency maps back into the estimate of foreground/background segmentations. We hope to eliminate false detections by weighting our detections, so that higher thresholds are required in areas of infrequent activity. This should be fairly straightforward in the framework of Markov random fields.

V. Conclusion

We have shown that a simple set of images taken of the environment in which a humanoid robot works is enough information to perform a variety of interesting tasks, including

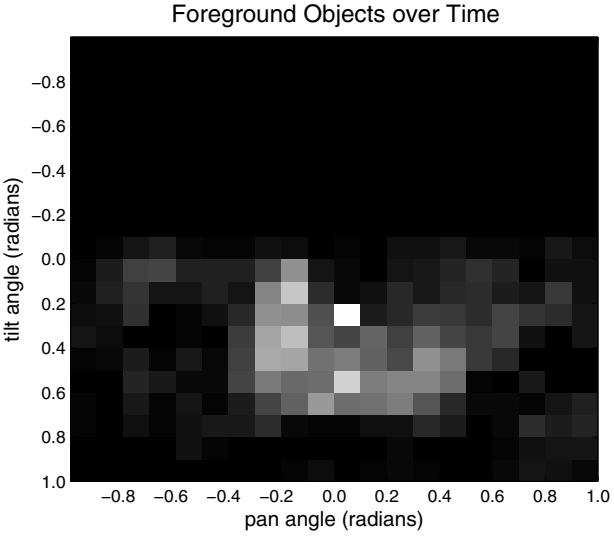

Fig. 8. Likely positions of foreground objects in pan/tilt space.

intialization of pan/tilt coordinates, foreground-background segmentation, and background activity modeling, even in the presence of a moving camera. It is easy to update background pictures, simply by waiting for a period of time during which there are no detections in the foreground to increase the probability that a stored set of background images does not accidentally contain foreground objects. Fortunately, the SIFT based methods for setting the intial pan/tilt settings are not affected by a partially changed background anyway, as spurious matches in SIFT features are quite unlikely.

There are a number of exciting directions for future work, particularly in the area of backgrounding with a moving camera. Among these are adapting these techniques to background models that use mixture distributions, as discussed in [2], [9]. While these models were originally designed to deal with small changes in the imaging environment (due to effects like the waving of trees in the wind), we believe they can also be used to add robustness to backgrounding techniques when there are small changes in camera position under rotation, as is common in many robotic vision setups. Second, the initial global alignment between the new image and background image can be improved by using a more sophisticated motion model and point-based feature matching. This addition should help reduce false positives due to misregistration. Third, we are experimenting with using local features in addition to image intensity with the expectation of reducing errors due to camouflage and lighting changes. Fourth, performance was not a primary concern in our investigations but must be for any practical system. The results shown were produced in less than 500ms, but, if desired, faster results are possible by switching to a faster discriminative Markov field inference algorithm such as ICM (Iterative Conditional Modes). Finally, as mentioned above we are interested in incorporating activity frequency maps back into our foreground/background segmentation.

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

Fig. 7. Three examples: (a) input image with foreground object, (b) closest background image from stored set of persistent background images, (c) image differencing and threshold after global alignment, (d) image differencing after local matching, (e) classification of foreground/background by discriminative Markov random field.

VI. Acknowledgements

This work is supported in part by the National Science Foundation (NSF) under grant IIS-0546666 and MIT/NASA cooperative agreement NNJ05HB61A.

References

[1] K. Toyama, J. Krumm, B. Brumitt and B. Meyers. Wallflower: Principles and practice of background maintenance. In *International Conference on Computer Vision*, pp. 255-261, 1999.

[2] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, pp. 246-252, 1999.

[3] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 22, No. 8, pp. 747-757, 2000.

[4] S. Hart, S. Ou, J. Sweeney and R. Grupen. A framework for learning declarative structure. In *Workshop on Manipulation for Human Environments, at Robotics: Science and Systems*, 2006.

[5] B. Scassellati. Foundations of a Theory of Mind for a Humanoid Robot. Ph. D. Thesis. Massachusetts Institute of Technology, 2001.

[6] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision*, 60, volume 2 , pp. 91-110, Jan 2004.

[7] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. tech. report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May, 2000.

[8] E. Hayman and J. Eklundh. Statistical background subtraction for a mobile observer. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 2003.

[9] Y. Ren, C. Chua, and Y. Ho. Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24, 1-3 (Jan. 2003), pp. 183-196. 2003.

[10] F. Kschischang, B. Frey, and H.-A.Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47, volume 2:498-519, Feb. 2001.

[11] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings 2003 IEEE International Conference on Computer Vision (ICCV '03)*, volume 2, pp. 1150-1157, 2003.

[12] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conference on Machine Learning*, pp. 282-289. Morgan Kaufmann, San Francisco, CA, 2001.