2008

Machine vision: a survey

David Phillips
Western Washington University

Machine Vision: A Survey

David Phillips
Western Washington University, Computer Science Department
chiaco@gmail.com

Abstract

This paper surveys the field of machine vision from a computer science perspective. It is written to act as an introduction to the field and presents the reader with references to specific implementations. Machine vision is a complex and developing field that can be broken into the three stages: stereo correspondence, scene reconstruction, and object recognition. We present the techniques and general approaches to each of these stages and summarize the future direction of research.

Keywords: machine vision; stereo correspondence; scene reconstruction; 3d object recognition

1. Introduction

Decades ago machine vision was considered science fiction, but has since grown into a complete field of research in computer science. The work of past researchers has built a solid foundation on which modern approaches may be constructed. This paper presents an overview of the history and modern approaches to machine vision, as well as a proposed structure for future machine vision techniques.

Machine vision is important because it allows programs to perform automated tasks that previously required human supervision. Such examples include assembly line part recognition, satellite reconnaissance, face recognition, unmanned aerial vehicles, crime scene reconstruction, and even unmanned automobiles. The current goal of vision is to develop a generic framework that may be implemented to solve a wide range of problems, and ultimately become a foundational system in artificial intelligence and robotics.

There are dozens of approaches to different subsets of the larger goal, but machine vision lacks a universal process for generic object recognition. There are two categories of vision: active vision and passive vision. Active vision involves interacting with the environment to receive information. Techniques such as range finding, RADAR, LIDAR, and SONAR all emit signals such as light, sound, or radio waves and will listen for reflected signals to resolve an image. This paper focuses on the techniques of passive vision, such as gathering light from the environment like human eyes. Passive vision is important because it is stealthy and doesn't fill the air with cluttered emissions, and the hardware is significantly less expensive to implement.

The goal of vision is to perform scene reconstruction and object recognition in an automated way. Research can be broken into three stages: stereo correspondence, scene reconstruction, and object recognition. The paper is organized into sections to discuss each of these stages and concludes with a current assessment of the state of the field.

2. Stereo Correspondence

Stereo correspondence is the process of taking input from two cameras and identifying shared features in each image. The result of correspondence is a disparity map of the image (see Figure 5). A disparity map is essentially an inverse depth map that will depict how far away every pixel is from the plane of the camera[11]. The cameras in the system must be a known fixed distance apart and at known orientations to each other for computations to work effectively. Researchers will often perform calibration tests to determine these distances and orientations automatically [5]. The input of the image is raw data from the camera that will typically need to undergo preprocessing called image rectification. Image rectification is the process of transforming an image onto a common surface such that distorted images are normalized. This is particularly important in the case of fish-eye camera lenses or camera height mismatches. The resulting output is a normalized image from each camera that we can run through matching correspondence algorithms.

Correspondence is typically done between two cameras, but some experiments may use dozens to improve their output [14]. Using more cameras will yield more accurate results but will take significantly longer to perform matching. An analogy to this would be taking the testimony of fifty witnesses to a traffic accident to discover what occurred, as opposed to only asking two bystanders. The two bystanders can quickly tell you what they saw and agree upon it, while the fifty will take significantly longer to come to an agreement. However the results from the many witnesses will be more accurate than the two bystanders. Economic considerations may dictate that using six inexpensive cameras is more efficient than two high quality cameras.

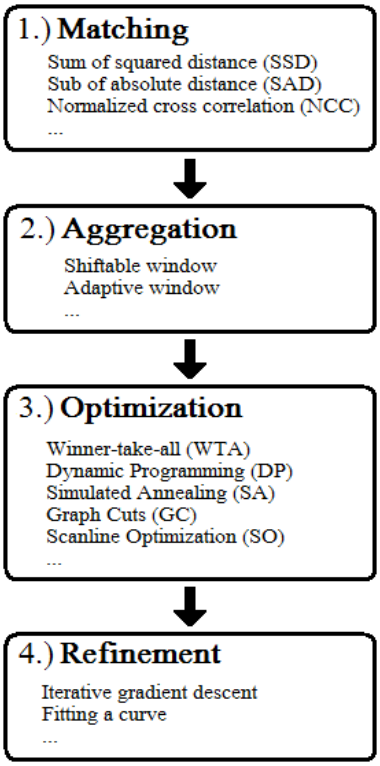

Figure 1: The steps of stereo correspondence algorithms

Typical stereo correspondence algorithms can be classified by their implementations of matching cost, aggregation method, disparity optimization, and disparity refinement [11].

- **Matching cost** refers to the algorithm used to estimate similarity between pixel intensities in the image set.
- **Support cost aggregation** is the method of generating the matching cost estimates from a local support region.
- **Disparity optimization** is the process of determining the disparity (depth) at a given pixel.
- **Refinement** is the process of smoothing out disparities to continuous values from common techniques that yield discretized intervals.

Correspondence techniques can be classified into feature-based or intensity-based matching techniques. Feature-based approaches attempt to match edges or corners between the images. However this requires extensive pre-processing on the images such as blurring and gradient detection, and post-processing to interpolate the disparities between the features. Feature extraction is not a preferred method due to noise and image occlusion (a visual obstruction), which leads to sparse and unreliable disparity maps.

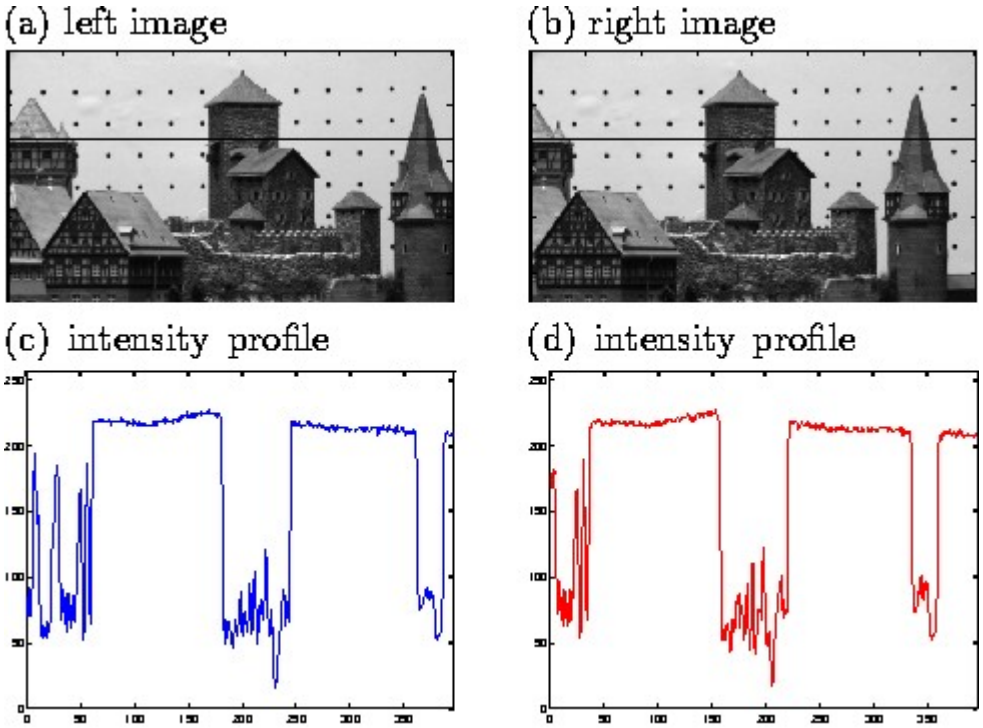

Figure 2: An intensity profile is developed for each image using the coloration intensity from each pixel

Intensity-based matching uses scanlines to develop intensity profiles (see Figure X). A scanline approach relies upon rectified images and operates on one horizontal line at a time. Image correspondence is established by sliding the offset camera's intensity profile along the horizontal, and minimizing the difference (maximizing the similarity) in

intensities. When comparing color images it is customary to develop an intensity profile for each color component and average the resulting intensity.

The scanline approach has been adapted to create sliding windows which perform the same calculations but on a local scale [11]. A sliding window will develop an intensity profile for a window size of N by N around the pixel in question. An equal sized window is then slid along the other image, and attempts to minimize the difference in intensity profiles.

Typical measurements to determine similarity include Normalized cross correlation (NCC), Sum of squared differences (SSD), and sum of absolute differences (SAD)[6]. Let X and Y represent the intensities in two windows, and there exists N tuples (X_1, Y_1) … (X_N, Y_N), for a window of size N. Then NCC is given by

$$R = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\left[\sum_{i=1}^{n} (X_i - \overline{X})^2 \sum_{i=1}^{n} (Y_i - \overline{Y})^2\right]}}$$

where $\overline{X}$ and $\overline{Y}$ represent the sample means of the corresponding windows. The value of NCC will be between -1 and 1, where a value of 1 indicates perfect correlation. The Sum of square differences is

$$SSD = \sum_{i=1}^{n} (X_i - Y_i)^2$$

which measures the squared Euclidean distance between X and Y. Sum of absolute differences is

$$SAD = \sum_{i=1}^{n} |X_i - Y_i|$$

and measures the absolute distance between intensity values. Other methods of similarity measurement exist but are not as popular as these.

There are a number of factors that can contribute to poor matching conditions such as occlusion and specular lighting [6]. Specular lighting refers to reflections from non-Lambertian surfaces. A Lambertian surface is a diffuse surface that reflects light equally in all directions. These factors become problematic when the cameras see different images of the same scene because of their differing perspectives.

Disparity optimization attempts to construct an accurate disparity map by determining the best set of disparities that represent the scene surface. Common approaches include winner-take-all (WTA), dynamic programming (DP), simulated annealing (SA), graph cuts (GC), and scanline optimization (SO)[11].

- **Winner-take-all** is the simplest and most common approach for window-based aggregation methods. It chooses the disparity that minimizes the cost value.
- **Dynamic Programming** attempts to find a globally optimized minimum-cost path through the matrix of all pairwise matching costs between two corresponding scanlines (Figure 3)[3].

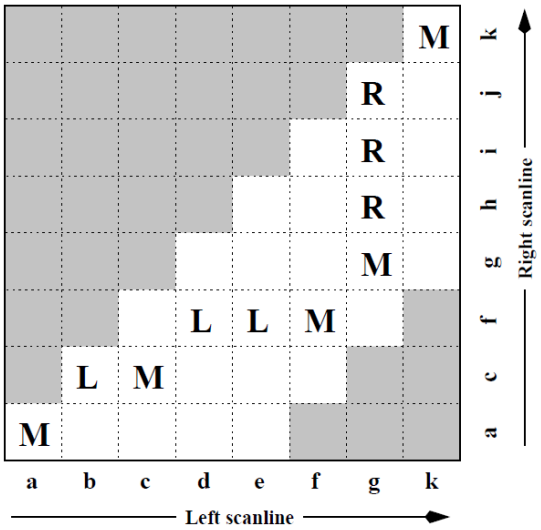

Figure 3: Stereo matching using dynamic programming. For each pair of corresponding scanlines, a minimizing path through the matrix of all pairwise matching costs is selected. Lowercase letters (a-k) symbolize the intensities along each scanline. Uppercase letters represent the selected path through the matrix. Matches are indicated by M, while partially occluded points (which have a fixed cost) are indicated by L and R, corresponding to points only visible in the left and right image, respectively.

- **Scanline Optimization** is similar to DP except that there is no occlusion cost.
- **Simulated Annealing** randomly chooses a disparity and evaluates the cost at that pixel and will slide left or right in an attempt to reach the global minimum.
- **Graph cuts** implement the α-β swap move algorithm [4].

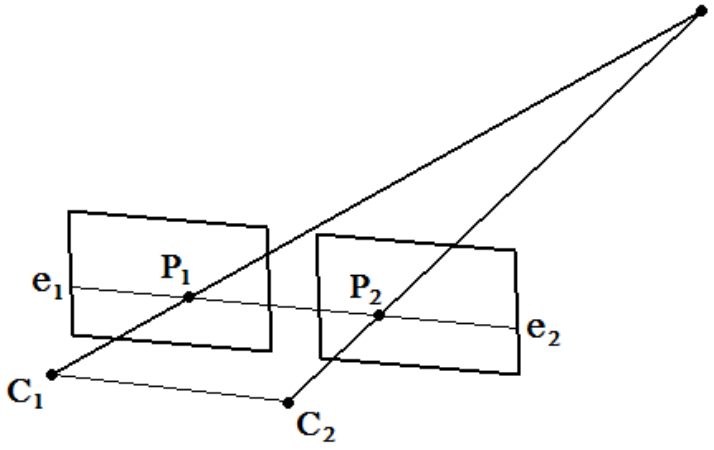

Figure 4: Epipolar geometry used to calculate the depth of a projected pixel. C_1 and C_2 represent the left and right cameras, while e_1 and e_2 represent the epipolar line, and P_1 P_2 are the projected coordinates in the image.

The disparity map is calculated using epipolar geometry after correspondence has been determined. Epipolar geometry is a form of projective geometry specifically designed to determine the three dimensional position of an object from two cameras (see Figure 4) [10]. In an ideal world with perfect camera resolution we would be able to precisely calculate the exact position of an object. However, using cameras with limited resolutions the triangulation angle limits depth resolution. This leads to an inaccurate projected position, and an inherent degree of error that can be predicted within some tolerance. This is why having many cameras allows for more precise estimates of location because there are more cameras to vote on the final position. After the position of correspondence has been determined it is just a matter of calculating the depth of the position from the plane of the camera to determine a disparity map (see Figure 5).

Figure 5: The reference image (left) and obtained disparity map (right)

Disparity refinement is the process of smoothing the final disparity map from discretized intervals into continuous values. This is done by smoothing the disparity between the top best disparity values, or between the neighbors. It is often assumed that the disparity of a neighboring pixel is equal to the current one, and thus smoothing across them will yield accurate results.

Modern implementations offload epipolar geometry calculations to the graphics card and use the projective texturing hardware to accelerate the process. This lightens the load on the CPU significantly and leads to increases a hundred fold increase in 3d reconstruction.

3. Scene Reconstruction

Scene reconstruction is the process of building a three dimensional model that represents the image from the cameras. Examples of this include reconstruction of a crime scene environment for analysis, constructing three dimensional maps of buildings for virtual worlds, or even rebuilding a precious moment from a family vacation. Reconstruction of a scene creates ordered information in a familiar format to humans, which allows us create algorithms to act upon it in intuitive ways.

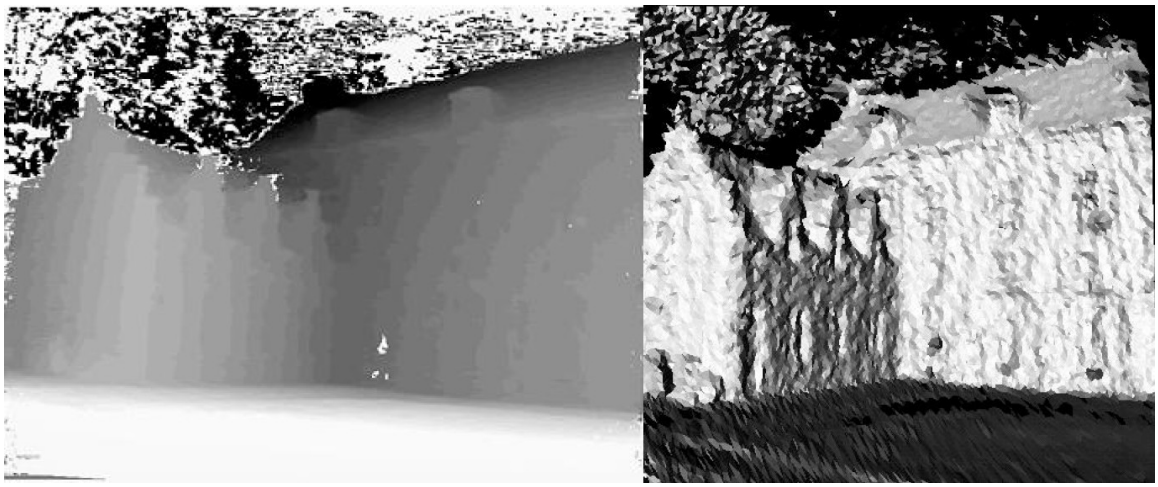

Figure 6: The disparity map (left) and reconstructed scene (right)

There are two camera models that need to be considered when performing reconstruction: calibrated and uncalibrated. A calibrated camera has a known focal point, field of view, and lens type. An uncalibrated camera has unknown attributes that can be experimentally discovered when picturing known images [5]. Uncalibrated cameras tend to use structure from motion [13] techniques to derive position information. Significant amounts of research have been contributed to scene reconstruction from a set of uncalibrated camera images, but that is beyond the scope of this paper. Calibrated cameras that are a known distance and orientation relative to each other allow epipolar geometry calculations in the stereo correspondence stage.

The disparity maps obtained from stereo correspondence can be used to determine the three dimensional position of the pixel. A pixel that occupies a three dimensional position is referred to as a voxel (volumetric pixel). By projecting thousands of voxels into Euclidean space we are able to build a scene that a human can view and understand (see Figure 6). The scene can be textured by coloring the voxels with the average color from the original pixels.

However, the voxel-based data storage has not always been feasible due to hardware memory constraints decades ago. Older methods tend to reduce the scene complexity by performing interpolation across local regions to form surface meshes [10]. These fused geometry representations would eliminate many voxels by merging them into one approximated mesh. Smoothing algorithms would then be run on the meshes to reduce the effect of noise. Texture can be estimated by combining the coloration of neighboring voxels. This has the effect of reduced accuracy with the data and loss of small features, but requires much less memory.

Modern approaches take the opposite approach and interpolate sub-voxel information from surrounding neighbors to give greater detailed to a scene [5]. A scene is sparsely populated with voxels and thus sparse data representations are used, except in a few experiments where voxels are locked into a grid for dense scene reconstruction. Feature information such as edges and corners can be stored in a scene for fast camera calibration when position and orientation have been changed slightly.

Scene detail can be improved by increasing camera resolutions, or adding more pictures to the set of data. An analogy would be picking up an unknown object and turning it around in your hand so you can better build a mental image of its structure.

These methods are described as structure from motion[13]. Moving the camera system will slightly change its position and orientation and allow a new set of stereo images to be taken. The goal is to incorporate the new set of images into the scene reconstructed by the previous set. The first step is to determine the camera's new position and orientation relative to the previous state. This is done by comparison of prominent sparse features (high probability matched corners and edges) to extrapolate new position and orientation. When the new camera parameters have been established, the projected voxel set can be added into the reconstructed scene. The consecutive stereo images can be obtained by mounting two cameras into one chassis and recording a video stream from each and panning around a room (see Figure 7).

Figure 7: Reconstructed scene from consecutive image sets

Deriving individual objects from a reconstructed scene is simple for a human but complex for a machine. Voxels can be clustered using coloration and Euclidean distance into logically unique objects. Human interaction, learning algorithms, or experience can aid the machine with association between objects (i.e. a chair is composed of the chair's feet and the upholstery).

It should be noted that these methods assume a static scene (that the scene does not change between images), but this is not a realistic assumption for the real world. This can be countered by increasing the sampling rate of the cameras and periodically flushing the scene of voxels that change. The sampling rate is increased because as the change in time between scenes approaches zero, the change in object movement between scenes approaches zero – and it becomes an instantaneous static scene. While this creates problems in the present, it will not be a problem in the future.

4. Object Recognition

Object recognition is the process of identifying an object and its location within a picture or scene. Recognition has historically been performed on two dimensional images using a set database for reference. These approaches often encountered problems with partial occlusion, variable lighting conditions, and cluttered backgrounds. Using a three dimensional scene eliminates occlusion as a problem during this stage of analysis (because it is handled earlier in stereo correspondence). Cluttered scenes only cause concern when it comes to scene reconstruction and clustering groups of voxels together into an object.

Object recognition must go through the two phases of acquisition (training) and recognition. The program must first acquire objects into its memory that it can later identify and recognize. Traditionally object model databases have stored sets of images from numerous angles and lighting conditions – but this has proven memory inefficient. Constructive Solid Geometry (CSG) is a more memory efficient method used to store 3D information about an object. CSG uses solid primitives combined through intersection, union, and set difference operations to form an object. This can be logically represented in a binary tree, where leaf nodes are solid primitives and parent nodes are the operations.

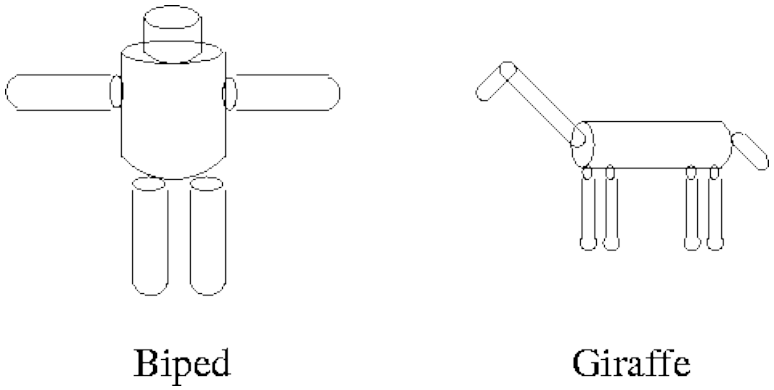

Figure 8: Constructive Solid Geometry (CSG) representation of objects

Wire frames have also been used to depict the outline of a 3D model. However a wire frame model can have the same projective appearance from many perspectives, and thus is not a preferred method. Voxel-based data stores tend to use methods of spatial occupancy, which lock voxels into a discretized 3D binary array. The object is volumetrically constructed by combining structural information from the binary array. A generic version of CSG was developed into the A Cone Representation of Objects Not Yet Modeled (ACRONYM) system [2]. ACRONYM uses the concept of "swept volumes." A swept volume is a 3D object that has a 2D shape for an end point, an axis that the 2D shape is "swept" down, and a "sweeping rule" that describes what happens to the shape as it moves down the axis. A cylinder, cube, pyramid, and a bottle are all swept objects. ACRONYM constructed swept objects and attempted to cluster them together to form a whole object. It ran into problems with generation of complex sweeping rules, and was later abandoned.

Object recognition techniques can be broken into two general approaches of pattern recognition and feature-based geometric recognition. Pattern recognition operates on low-level data (voxels) and attempts to match texture and coloration patterns against a database. Because many objects share the same texture this is typically not a preferred matching technique. Feature-based geometric recognition seeks invariant features that are unique to an object. An invariant feature does not change when transformations are applied to the object. Standard transformations that must be considered are Euclidean transformations (translation, rotation, or reflection), and affine transforms (sheering and scaling). Ratios, edges, and corners are examples of invariant features. The scale-invariant feature transform (SIFT) methodology offers a generic technique of identifying invariant features and comparing them between objects. SIFT has proven highly successful, even when faced with partial occlusion.

5. Experimental Results

Stereo correspondence algorithms have been researched for many decades. Experimentally it has been determined that larger sized sliding windows function better in textureless areas, and all methods perform poorly near discontinuities (occluded areas). Sliding windows out-performed all other aggregation methods, and adaptive windows offered the best results against discontinuities [11]. Sliding windows and SSD performed the best of all local methods, and graph cuts performed the best overall. In terms of efficiency the sliding windows performed best in mere seconds, while graph cuts would take up to 30 minutes.

Scene reconstruction techniques are directly based on the results of stereo correspondence algorithms and thus it is difficult to compare them directly. Smoothing algorithms achieve good results with interpolating new voxels. Estimated texturing methods on newly created voxels also create realistic looking scenes. Construction of object meshes using the nearest neighbor approach currently fails to account for individual object clustering. This leads to the appearance of “snow” across the entire scene and object blend together.

Scale-invariant feature transform (SIFT) methodology has proven highly effective for matching objects in variable lighting conditions and perspective transformations. The older techniques of incrementally constructing 3D models offer a logical hierarchy, but are highly susceptible to noise and error.

Conclusion

Machine vision is advancing through the refinement of individual techniques and incorporating ideas from previous researchers. There are many other paths of research, including active vision, structure from motion, and scene reconstruction from uncalibrated camera pictures which contain subset problems that overlap with problems presented in this paper.

Possible avenues of research include investigating the relative efficiency of using a large number of lower-quality cameras for correspondence compared with two high quality cameras. The stereo correspondence disparity refinement methods of fitting a curve to the top few disparity values to obtain a sub-pixel disparity needs more research

to develop an alternative approach (the old approach has been raising questions within the community [11]). Research also needs to be performed on voxel-based memory representation of a generic scene, with possible storage of important scene features such as edges and corners. In scene reconstruction work needs to be done to investigate clustering techniques to form objects based on coloration and proximity.

In the future we can expect to see hardware upgrades that increase camera resolution, CPU and GPU power, and memory availability. These increases in capability offer better voxel position resolution and more detail for object recognition. Because of these increases we can expect the efficient correspondence algorithms of window based dynamic programming to become the most popular. Methods such as graph-cuts will be popular for long-term offline calculations that require accuracy such as modeling of important buildings, famous landmarks, or other planets. Eventually an international standard will exist for model representation and online object repositories will exist for public use.

References

1. Arman, F. and Aggarwal, J. K. 1993. Model-based object recognition in dense-range images—a review. *ACM Comput. Surv.* 25, 1 (Mar. 1993), 5-43. DOI= http://doi.acm.org/10.1145/151254.151255
2. Besl, P. J. and Jain, R. C. 1985. Three-dimensional object recognition. *ACM Comput. Surv.* 17, 1 (Mar. 1985), 75-145. DOI= http://doi.acm.org/10.1145/4078.4081
3. Bobick, A. F. and Intille, S. S. 1999. Large Occlusion Stereo. *Int. J. Comput. Vision* 33, 3 (Sep. 1999), 181-200. DOI= http://dx.doi.org/10.1023/A:1008150329890
4. Boykov, Y., Veksler, O., and Zabih, R. 2001. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 11 (Nov. 2001), 1222-1239. DOI= http://dx.doi.org/10.1109/34.969114
5. Debevec, P. E., Taylor, C. J., and Malik, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of the 23rd Annual Conference on Computer Graphics and interactive Techniques* SIGGRAPH '96. ACM, New York, NY, 11-20. DOI= http://doi.acm.org/10.1145/237170.237191
6. Kumar, S., Srinivas, Chatterji, B. N. 2004. *Robust Similarity Measures for Stereo Correspondence*. Institution of Engineers (India). Volume 18. 2004.
7. Kutulakos, K. N. and Seitz, S. M. 1998 *A Theory of Shape by Space Carving*. Technical Report. UMI Order Number: TR692., University of Rochester.
8. Lowe, D. G. 1999. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the international Conference on Computer Vision-Volume 2 - Volume 2* (September 20 - 25, 1999). ICCV. IEEE Computer Society, Washington, DC, 1150.
9. Rodrigues, R. and Fernandes, A. R. 2004. Accelerated epipolar geometry computation for 3D reconstruction using projective texturing. In *Proceedings of the 20th Spring Conference on Computer Graphics* (Budmerice, Slovakia, April 22 - 24, 2004). SCCG '04. ACM, New York, NY, 200-206. DOI= http://doi.acm.org/10.1145/1037210.1037241
10. Sabharwal, C. L. 1992. Stereoscopic projections and 3D scene reconstruction. In *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing:*

Technological Challenges of the 1990's (Kansas City, Missouri, United States). H. Berghel, G. Hedrick, E. Deaton, D. Roach, and R. Wainwright, Eds. SAC '92. ACM, New York, NY, 1248-1257. DOI= http://doi.acm.org/10.1145/130069.130155

11. Scharstein, D. and Szeliski, R. 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. J. Comput. Vision* 47, 1-3 (Apr. 2002), 7-42.
12. Stich, T., Linz, C., Wallraven, C., Cunningham, D., and Magnor, M. 2008. Perception-motivated interpolation of image sequences. In *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization* (Los Angeles, California, August 09 - 10, 2008). APGV '08. ACM, New York, NY, 97-106. DOI= http://doi.acm.org/10.1145/1394281.1394299
13. Tomasi, C. and Kanade, T. 1992 *Shape and Motion from Image Streams: a Factorization Method Parts 2,8,10 Full Report on the Orthographic Case*. Technical Report. UMI Order Number: CS-92-104., Carnegie Mellon University.
14. Wilburn, B., Joshi, N., Vaish, V., Talvala, E., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. 2005. High performance imaging using large camera arrays. In *ACM SIGGRAPH 2005 Papers* (Los Angeles, California, July 31 - August 04, 2005). J. Marks, Ed. SIGGRAPH '05. ACM, New York, NY, 765-776. DOI= http://doi.acm.org/10.1145/1186822.1073259