Multiple Image Analysis at the Hebrew University: Motion, Structure, and Recognition*

Shmuel Peleg, Daphna Weinshall, and **Michael Werman**
Institute of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
contact email: daphna@cs.huji.ac.il

Abstract

Computer vision research at the Hebrew University focused last year on the following directions:

3D Object representation & recognition:

- The computation of generic views, by characterizing view likelihood and stability for general objects (Section 2).
- The complexity of database indexing (Section 3).

Image sequence analysis:

- The simultaneous use of all the information in multiple images, multiple points in three images (Section 4) or multiple images of 6 or more points [4].
- Video stabilization and compression using multiple frames (Section 5).

1 Introduction

In order to track and reconstruct objects robustly and precisely we would like to use all available information. This includes using information from multiple images and using multiple features on each object. The methods so far have used two pictures at a time and integrated the information using statistical filters. We plan to use the full information available from multiple pictures simultaneously before using statistical filters. We go about this in 2 ways:

Direct shape computation: compute shape without computing camera geometry using new shape representations such as the shape tensor [4].

Camera calibration from multiple frames: compute camera geometry without computing shape using new structures such as the trilinear tensor (Section 4).

More specifically, we developed a family of algorithms which compute camera geometry and object structure from multiple images, under the perspective projection assumption. These algorithms are linear and are easy to compute in a robust and incremental way. The "camera tensor", for example, which is based on three images, can be used to compute new views of a scene with high accuracy. Camera motion can be computed as well, with application to tracking and to image stabilization, image compression and 3D reconstruction (Section 5). Dual to the "camera tensor" is the "shape tensor", which can be used for direct shape reconstruction and object recognition [4].

In order to build a three-dimensional object recognition system, one needs first to decide how to represent the shape of the objects. In order to use a recognition approach like alignment, typically explicit descriptions of the $3D$ shape of objects is stored. In order to use invariant-based approaches such as geometric hashing, different representations should be used:

1. One possibility is to store a set of "representative views" of the object. But what are representative views? We provide an answer to this question by defining view likelihood and stability, and by showing how to compute these measures (Section 2).
2. Generally, a recognition process starts from a set of image measurements that provides access into a database. Depending on the particular set of measurements, the object defines a manifold in the database - more than one entry in the indexing table stores the object. We defined and analyzed the complexity of these manifolds (Section 3).

2 What are "canonical" views?

What are "good" two-dimensional pictures of three-dimensional objects? In other words, how can we measure how representative an image of a $3D$ object is? An answer to this question can help us do better in some difficult vision tasks such as the following:

- design better databases of $3D$ objects;
- select the best answer in an object recognition task when there is more than a single object in the database which may explain the image;
- obtain robust performance in ambiguous reconstruction tasks using noisy images.

In [3] we define measures of likelihood and stability, which indicate how representative a 2-D picture of a 3-D object is. These measures depend **only** on the $2D$

*This research was sponsored by ARPA through the U.S. Office of Naval Research under grant N00014-93-1-1202, R&T Project Code 4424341—01.

metric used by the vision system to compare images. We thus give a practical way to measure how "generic" views are, and identify "characteristic" views. For example, we find that for many $2D$ metrics, the "flattest" view of an object is the "best" picture of the object. We show how to use the likelihood measure to resolve ambiguities in difficult reconstruction tasks (see Fig. 1). One interesting consequence of our maximum-likelihood augmented shape from shading algorithm is the underestimation of depth. Finally, we show how to augment existing object recognition systems with maximum likelihood estimation, using view likelihood. The difference in performance between geometrical-based object recognition, and object recognition enhanced with maximum likelihood, is demonstrated in Fig. 2.

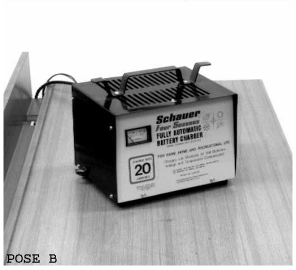

Figure 1: A battery charger. Its real dimensions are: depth - 22.5, length - 28, and height - 19 (all in cm.). Given a fixed noise tolerance, the dimensions of the charger cannot be unambiguously determined from the image, even when using the knowledge that it is a box-like object. Using maximum likelihood to choose among all the possible interepretations, we get that the most likely interpretation, when normalized by length, is: depth - 19.5, height - 21. Thus depth is underestimated.

3 Complexity of indexing

Object recognition starts from a set of image measurements (including locations of points, lines, surfaces, color, and shading), which provides access into a database where representations of objects are stored. In [5] we describe a complexity theory of indexing, a meta-analysis which identifies the best set of measurements (up to algebraic transformations) such that:

1. The representation of objects are linear subspaces and thus easy to **learn**;
2. Direct indexing is **efficient** since the linear subspaces are of minimal rank.

Index complexity is determined via a simple process, equivalent to computing the rank of a matrix. We readily rederive the index complexity of the few previously analyzed cases. We then compute the best index for new and more interesting cases: 6 points in one perspective image, 6 directions in one para-perspective image, and 2 perspective images of 7 points. With color we get the following result: 4 color sensors are sufficient for color constancy at a point, and the sensor-output index is irreducible; the most efficient representation of a color is a plane in 3-D space. For future applications with any vision problem where the relations between shape and image measurements can be written down, we give an algorithm to construct the most efficient database that can be directly obtained by learning from examples.

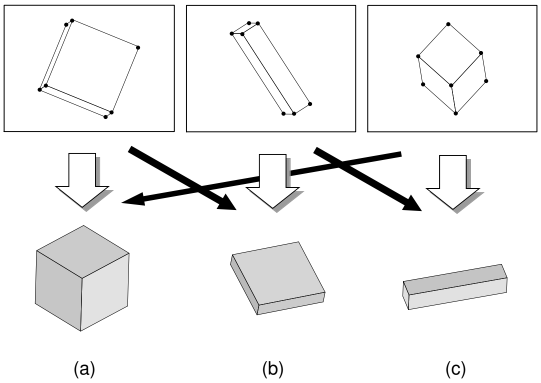

Figure 2: Three polyhedral objects: (a) a cube of dimensions $1 \times 1 \times 1$ cm., (b) a flat box of dimensions $5 \times 5 \times 1$ cm., (c) an elongated box of dimensions $5 \times 1 \times 1$ cm. Top: three images of these objects, obtained from special viewpoints; the extracted features are shown with dark circles. Bottom: the same polyhedral objects viewed from a "typical" viewpoint. A geometry-based recognition system would "correctly" recognize each image in the top row as an instance of the object illustrated in the bottom row (this matching is shown by thick white arrows). A maximum likelihood recognition system would recognize the images in the top row as: left = flat box, middle = elongated box, right = cube (this matching is shown by thin black arrows.)

4 Using multiple images for camera geometry

Background: It has previously been established that certain trilinear forms of three perspective views give rise to a tensor of 27 intrinsic coefficients. Other investigations have established the existence of quadrilinear forms (with total of 81 coefficients) across four views with the negative result that further views would not add any new constraints.

Our ultimate goal is to find the means for combining together the contribution of any number of views, not only three. To this end we suggest to start with investigating the space of *all* trilinear tensors and look for rank deficiencies in that space. Any finding of that sort is extremely useful because it readily allows a statistical way of putting together many views, simply by means of factorization. The main result is that trilinear tensors across $m > 3$ views are embedded in a low dimensional linear subspace.

There are two motivations to this work. First, is the practical aspect – if any additional view over the fourth view is redundant, what is the best and most efficient way of capturing that redundancy (in a linear manner)? Second, the existence of the quad-linearities is somewhat unsettling because the number of coefficients has risen from 27 to 81, whereas a view adds only 12 parameters. In other words, the quad-linearities may be too redun-

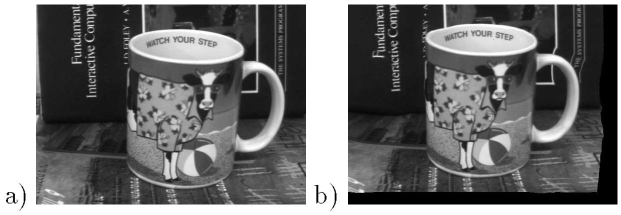

Figure 3: (a) One of the original four images. (b) The reprojection from the linearly combined tensors.

dant a representation of the constraints over four views.

Two main results are shown: (i) trilinear tensors across $m > 3$ views are embedded in a low dimensional linear subspace, (ii) given two views, all the induced homography matrices are embedded in a four-dimensional linear subspace. The two results, separately and combined, offer new possibilities of handling multiple views in a linear manner (via factorization), some of which are further detailed in [1]. One application of these results, where a new image is synthesized by linearly combining two tensors and doing forward mapping, is shown in Fig. 3.

5 Camera stabilization

Computing camera rotation from image sequences can serve many computer vision applications. One direct application is image stabilization, and when the camera rotation is known the computation of camera translation and 3D scene structure are much simplified. A new approach for recovering camera rotation is presented in [2], which proves to be much more robust than existing methods by avoiding the computation of the epipole. Another benefit of the new approach is that it does not assume any specific scene structure.

The rotation matrix of the camera is computed explicitly from three homography matrices, recovered using the trilinear tensor. The entire computation is linear for small angles, and is therefore fast and stable. Iterating the linear computation can then be used to recover larger rotations. Results obtained by using this method on real images are shown in Figs. 4, 5.

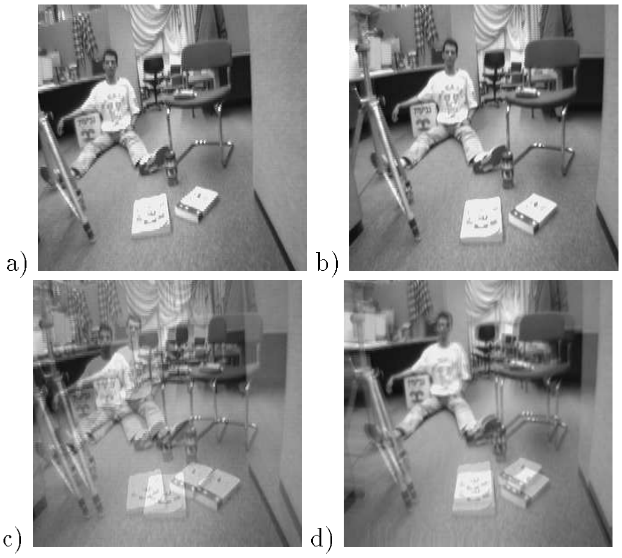

Figure 4: A sequence of indoor scene.
a) First original frame.
b) Second original frame. The unstable camera was moving towards the man.
c) average of the two original images. The rotations make the average image unclear.
d) average of the two images after rotation cancellation. The remaining motion is only due to the original translation. The sequence looks as if it was taken using a stabilized camera.

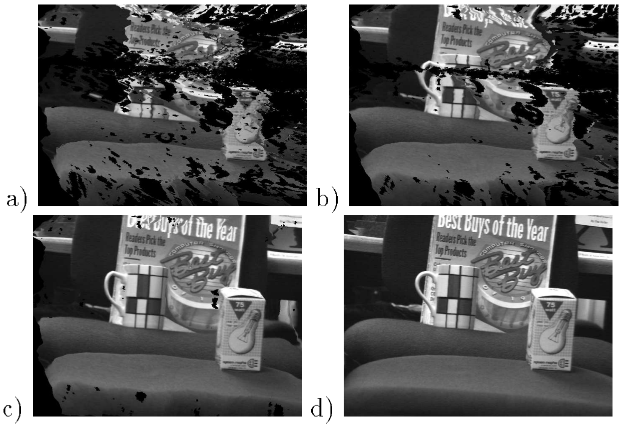

Figure 5: Reprojection in different methods.
a) Reprojection using epipolar line intersection. Fundamental Matrices computed with code distributed by INRIA.
b) Reprojection using epipolar line intersection. Fundamental Matrices computed from tensor.
c) Reprojection using the tensor equations.
d) Original third image. Presented for comparison.

References

[1] S. Avidan and A. Shashua. Tensorial Transfer: On the Representation of $N > 3$ Views of a 3D scene. In *Proc., APRA IU Workshop*, 1996.

[2] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. In *Proc., APRA IU Workshop*, 1996.

[3] D. Weinshall and M. Werman. A computational theory of canonical views. In *Proc., APRA IU Workshop*, 1996.

[4] D. Weinshall, M. Werman, and A. Shashua. Shape tensors for efficient and learnable indexing. In *Proc. of the IEEE Workshop on Representations of Visual Scenes*, Cambridge, Mass, 1995.

[5] M. Werman and D. Weinshall. Complexity of Indexing: Efficient and Learnable Large Database Indexing. In *Proc., APRA IU Workshop*, 1996.