# Robust Sequence Proximity Estimation by Radial Distance Hashing

Michael Kertesz Yehezkel Yeshurun

Department of Computer Science
Tel-Aviv University
Ramat Aviv, ISRAEL
{kertesz,hezy}@math.tau.ac.il

## Abstract

*There is a recent growing interest in image analysis of multiple views of a scene, often involving aspects of reconstruction, mosaicing and new view generation.*

*As the availability of multiple camera systems augments, we suggest that such tasks could be carried out where the image source is a set of unsynchronized and uncalibrated cameras moving arbitrarily in a 3D scene. In order to make efficient use of this data, it is necessary to define a measure of inter-sequence proximity.*

*In this paper we suggest such a measure, based on pure 2D analysis, namely the ratios between image-space distances among a set of feature points. We show this measure to be sound, and propose a simple iterative method to robustly estimate the relative positions of the set of moving cameras, even in the presence of substantial amount of noise, and without computing egomotion.*

## 1 Introduction

As the availability of low-cost video cameras augments, multi-camera systems find applications in various fields, including surveillance, military and surface- or volume-mapping applications. The huge amount of images captured by such multi-camera systems usually contains redundant information, both within each video sequence, and in-between sequences (as cameras volumes of view often partially overlap). Conversely, it is probable that new information can be extracted by integrating images coming from different video sources. Thus, the need to unite multiple video sequences into a more compact representation arises.

Several systems, which are aimed at enhancing or compressing video sequences coming from a *single* camera, have been developed over the years. These algorithms can be roughly categorized into three groups. The algorithms in the first group attempt to reconstruct the 3D structure of the scene from calibrated or uncalibrated video sequences [16, 17, 9]. Algorithms in the second group, known as Arbitrary View Generation algorithms [2, 15], exploit certain invariants in the geometry of the scene in order to interpolate an intermediate view of the scene. Finally, mosaicing algorithms [7, 12] combine a set of input images into a single larger representation.

A common feature of these algorithms is that they rely on the proximity of frames in the sequence, i.e. the assumption that two successive frames were taken from relatively close camera positions. Clearly, in order to generalize these single-camera algorithms onto multiple video sequences, it is necessary to recognize parts of different sequences, that are "close" enough. Such parts can then be handled by the algorithms, as if they were taken by the same camera, with respect to the probability that they contain redundant or complementary information.

In this paper we address the problem of recognizing parts of sequences that cover overlapping view volumes and were taken from proximal camera locations, in the case of multiple uncalibrated cameras moving in a stationary 3D world. We propose a distance measure based on 2D geometry of feature points, as well as a robust algorithm to evaluate this measure, denoted Radial Distance Hashing. In this respect, it is important to note that we do not assume a given cross-sequence feature point correspondence. Rather, using robust statistics methods, Radial Distance Hashing is used to obtain this correspondence. The proposed technique functions as a pre-processing stage for the algorithms described above.

The paper is organized as follows: in Section 2, we discuss various sequence proximity factors and show why camera-space proximity is a good choice for defining the distance between sequences. Section 3 introduces Radial Distance Hashing. In Section 4, the application of Radial Distance Hashing to camera proximity estimation and cross-sequence correspondence is described. We then construct an algorithm that com-



0-7695-0164-8/99 $10.00 © 1999 IEEE

putes the distance between two uncalibrated image sequences. Finally, in Section 5, we present experimental results obtained by applying the proposed algorithm on synthetic and real-world image sequences.

## 2 Sequence Proximity Factors

Consider a stationary 3D world in which $N$ cameras are independently moving. The internal and external parameters of the cameras are unknown, nor is the *relative* position of the cameras. A likely scenario of this sort would be a set of MRPVs surveying a common area of interest, or a set of autonomous robots.

Let us now look at two of those $N$ cameras, $C$ and $C'$, from which two video sequences, $S$ and $S'$ were acquired.

Given $S$ and $S'$, our goal is to define a proximity factor that will indicate how "close" those two sequences are at different points in time. Specifically, we attempt to identify those images in $S$, $S'$ that are at maximum proximity. We now describe three different approaches to define the distance between the sequences: Image-Space, Scene-Space and Camera-Space. We argue that the latter is the most adequate for multi-view algorithms.

**Image-Space.** The most straightforward approach to define the distance between two images is by directly comparing properties of their image features, such as their grayscale histograms, apparent textures, edge maps, etc.

The main advantage of these methods is that they are easy to compute and do not require any knowledge of the actual scene structure. However, Image-Space methods are very sensitive to camera parameters, occlusion and a changing amount of mutual overlap in view fields, and are therefore not suitable for the case of arbitrarily moving cameras.

**Scene-Space.** The second approach lies at the opposite end of the complexity scale. Scene-Space methods attempt to reconstruct, up to a scale factor, the complete three-dimensional representation of the scene from each image sequence. Assuming that the image sequences have been captured in the same scene, the constructed 3D models can be aligned. The proximity between images is then defined as the degree of congruence between the view volumes. As the reconstruction process ultimately results in retrieval of external camera parameters, the proximity between given images can be retrieved.

Obviously, a 3D reconstructed model is inherently the most complete way of scene-representation. Yet, 3D reconstruction methods are computationally expensive and extremely liable to errors. These problems call for an approach capable of determining camera position without complete scene reconstruction.

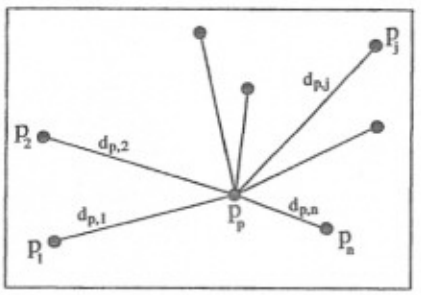


Figure 1: The set of image space distances between a pivot point and the remaining points.

**Camera-Space.** This approach consists of defining the proximity measure between images as the proximity *between camera locations from which those images were taken.* This approach is more apt for multi-view algorithms, as camera proximity directly affects the amount of visible parallax between the two images. Mosaicing, for example, requires little or no parallax; reconstruction and new view generation algorithms call for a controlled amount of parallax.

A naïve solution for estimating camera-space proximity consists of calculating the egomotion [1, 5] of each camera, aligning the retrieved egomotion tracks into one coordinate system and measuring the distance between those camera tracks. However, egomotion estimation from uncalibrated sequences is very sensitive to noise, and exhibits accumulative errors over long image sequences. Therefore, this solution is not suitable for long uncalibrated sequences.

In the following sections, we develop an alternative method, based on 2D geometric relationships between feature points, that does not require absolute egomotion retrieval. Rather, only the *relative* location of cameras is estimated.

## 3 Radial Distance Hashing

In this section we define "Radial Distance Hashing", a transformation from 2D locations of tracked feature points to a vector space. This transformation lies at the base of our approach. Suppose a set of feature points has been tracked throughout a given image sequence and $n$ points, $p_1, p_2, \ldots, p_n$ have been detected in frame $f$. The number of points may vary from frame to frame, as points enter and leave the view field or are occluded by other objects in the scene.

We assume that the selection of feature points has been done by the same feature extractor across all image sequences, such that at least some of the points

are consistently chosen across sequences. As the image formation model, we use perspective projection on the plane. The image plane is parallel to the XY plane and the viewing direction is along the positive Z-axis.

Let us denote by $d^f_{i,j}$ the Euclidean ($L_2$) image-space distance between a pair of interest points, $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ in frame $f$:

$$d^f_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

We now select one of the interest points, $p_p$, (a pivot point), and represent the distance between this point and the remaining $(n-1)$ points in frame $f$ as a $(n-1)$ dimensional vector (Figure 2),

$$D^f_p = (d^f_{p,1}, d^f_{p,2}, \ldots, d^f_{p,j}, \ldots, d^f_{p,n}) \tag{2}$$

(for all $j \neq p$)
Let us denote $D^f_p$ the "Radial Distance Hash" (RDH) of frame $f$ with respect to pivot $p_p$. Using this notation, each image can be represented as a single point in $\boldsymbol{R}^{n-1}$, for any given pivot. This measure somewhat resembles the well known Cross-Ratio invariant [11], yet it is to be noted, that, as opposed to the Cross-Ratio of a set of feature points, the Radial Distance Hash is *not* invariant to translation under perspective projection. In fact, this variance is its most useful property for the set of algorithms in discussion, enabling it to be used for distinguishing between different views of the same scene.

### 3.1 Comparing RDH vectors

Suppose two sequences, $S$ and $S'$ have been captured by two different cameras, and let $D^f_p = (d_1, d_2, ..., d_{n-1})$ and $D^{f'}_p = (d'_1, d'_2, \ldots, d'_{n-1})$ be the RDH vectors of frame $f \in S$ and $f' \in S'$ with respect to a correlated pivot point $p_p$. The case in which the number of extracted points in the two images, $n$, is not identical is dealt with later. Note, that in order to compare two RDH vectors obtained from distinct image sequences, it is required that the vector coordinates are ordered. In other words, if $d_k$ represents the image space distance in image $f$ between two scene points $P_i$ and $P_j$, then $d'_k$ represents the distance in image $f'$ between the same two scene points. Fully satisfying this ordering constraint, which in practice requires full cross-sequence correspondence between feature points, is practically impossible. In the next section, we show how a subset of corresponding points can be retrieved from a noisy first estimate.

Let $M^f_p = M^f_p(D^f_p, D^{f'}_p)$ denote the Manhattan ($L_1$) distance between two normalized Radial Distance Hash vectors $D^f_p$ and $D^{f'}_p$, thus

$$M^f_p = \sum_{i=1}^{n-1} \left| \frac{d^f_{p,i}}{\left\| D^f_p \right\|} - \frac{d^{f'}_{p,i}}{\left\| D^{f'}_p \right\|} \right| \tag{3}$$

where $\left\| D^f_p \right\|$ is the $L_1$ norm of vector $D^f_p$,

$$\left\| D^f_p \right\| = \sum_{i=1}^{n-1} \left| d^f_{p,i} \right| \tag{4}$$

$L_1$ was chosen as the distance measure, since it minimizes the effects of outliers, especially when working with normalized vectors. As shown below, normalization is required in order to preserve invariance to camera scaling. Thus, if all cameras are known to have a common and fixed focal length, this normalization is not obligatory.

### 3.2 Overcoming Point Occlusion

As a result of camera motion in a 3D scene, it is very likely that tracked points will temporarily disappear, group, enter or leave the view field. Furthermore, noisy input to the feature extractor may result in inaccurate tracking of the points and "feature flicker", as some points might momentarily be missed by the feature extractor. Although some of those tracking problems can be solved at feature extractor level by heuristic search algorithms [6] or by using the smoothness property [8] to interpolate estimated tracks in case the expected direction and maximal velocity of tracked points is known [13, 14], some unhandled feature noise usually endures.

In order to handle the common situation in which not all feature points are visible in the two frames to be compared, $f$ and $f'$, we project the two RDH vectors, $D$ and $D'$ onto their lowest common subspace. In practice, this means that only the distances between corresponding feature points seen in both frames are used when computing $M^f_p(D, D')$. An algorithm for achieving the required cross-sequence feature point correspondence is described in the next section.

In Section 5, we present results obtained on a simulated scene in which a high level of occlusion was artificially induced.

## 4 Proximity Estimation by RDH

We now show how RDH can be used to estimate camera-space proximity and simultaneously determine cross-sequence feature point correspondence. This is accomplished in the following iterative fashion: First a crude initial guess for cross-sequence correspondence is established. Then, RDH vectors are computed with respect to every feature point, serving as a set of estimated measures for camera-space proximity. Deviations from the mean measure are used to discard badly

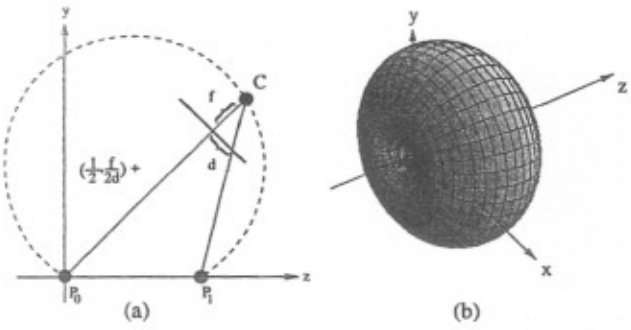


Figure 2: (a) Given the image-space distance between $P_0$ and $P_1$, and the focal length $f$, camera location is restricted to the surface of revolution of the dashed circle over the z-axis, as seen in (b).

correlated feature points, thus improving the proximity estimation, and simultaneously refining the set of corresponding feature points.

### 4.1 Estimating camera space proximity

First, let us see why $M_p^f$ is a good estimate for the camera-space distance between $f$ and $f'$. The full analytical proof is shown in [10] and is only briefly outlined below. The proof consists of three parts: Invariance to camera rotation and scaling, uniqueness and smoothness [3].

**Invariance.** Since the RDH vector represents a set of distances between image-space points, it is trivial that it is invariant to z-axis rotation, and that normalizing this vector yields scale invariance. Due to the foreshortening effect, RDH is not invariant to x- and y-axis rotation, assuming perspective projection. However, this change in RDH vector value induced by camera rotation is extremely low compared to the change caused by camera translation. Indeed, experimental results have shown that in natural scenes, x- and y-axis camera rotations have no noticeable effect on the estimated distance between cameras.

**Uniquness.** The mapping from camera-location to RDH vector values is shown to be unique. Hence, images taken from different camera locations will necessarily yield different vectors in $R^{n-1}$. This claim is based on the fact that for a given pair of scene points, each measured distance between the corresponding points as projected on the image plane, restricts the world-coordinates of the camera location to a single surface. This surface can be described as the surface of revolution of a circle around the line connecting those two scene points. In a coordinate system built such that the two scene points in discussion lie at $P_0(0,0,0)$ and $P_1(0,0,1)$, the center of the circle is shown to be at $(\frac{1}{2}, \frac{f}{2d})$, where $f$ is the focal length of the camera and $d$ is the measured Euclidean distance between the two points in image space (Figure 2). We show that four noncoplanar tracked points restrict the location of the camera to a single point.

**Smoothness.** Finally, it is easy to see that a smooth change in camera location will yield smooth change in the image-space distances between feature points, and therefore smooth change in $D_p^f$.

As a consequence of the three mentioned properties, $M_p^f$ monotonically increases with the increase in distance between cameras, and is very little affected by 3D camera rotation or scaling. Thus, it is a suitable estimate for the camera-space distance between $f$ and $f'$.

### 4.2 Obtaining cross-sequence correspondence

As mentioned in section 3, the ordering constraint must be satisfied in order to compare two RDH vectors. Clearly, satisfying this constraint throughout long image sequences is impossible in real-world scenarios. In our experiments we have shown, though, that even from a set of largely uncorrelated feature points (70% outliers), a well correlated subset could be retrieved using the following algorithm.

To formulate the problem as a classification problem, let us denote by $P = \{p_1, p_2, \ldots, p_n\}$ the set of presumably corresponding interest points detected in the two images, and by $B$ and $C$ the classes of badly and correctly correlated points, respectively. Hence, our goal is to classify each point $p_i \in P$ as either a badly correlated point ($p_i \in B$) or as a correctly correlated point ($p_i \in C$).

The algorithm consists of two stages. In the initialization stage, a-priori probabilities of match are computed using a measure of similarity between the feature points in the two sequences [8]. Each point $p_i$ is assigned a match probability $0 \leq w_i \leq 1$. In the experiments described in Section 5, we used blobs of distinct colors as the points of interest, and the difference between the average blob color as the similarity criterion. The optimal match can be modeled as a minimal match in a bipartite graph. The second stage is an iterative stage, which is aimed at improving this initial correspondence by detecting and discarding badly correlated points.

Let us look at the simple case where out of the $n$ interest points, only one of the points, $p_1$, is badly correlated. Thus, we have $B = \{p_1\}$; $C = \{p_2, p_3, \ldots, p_n\}$.

Our algorithm is based on the following interesting property: the fact that $p_1$ is badly correlated will on-

ly affect the Radial Distance Hash with respect to $p_1$ ($M_1^f$), whereas $M_2^f, M_3^f, \ldots, M_n^f$ will be only slightly affected by the erroneous correspondence. To see why this happens, recall the definition of $M_p^f$ in Equation 3. A failure to correlate the pivot point causes a complete violation of the ordering constraint, thus resulting in $n-1$ erroneous distance comparisons, namely the comparison of $d_{1,i}^f$ and $d_{1,i}^{f'}(i = 2 \ldots n)$. In contrast, in all other $M_p^f (p \neq 1)$ functions, only one coordinate out of the $n-1$ coordinates of $D_p^f$ will be wrongly matched, and the overall Manhattan distance will not be affected much.

We now use the above-mentioned property to classify the points into the $B$ and $C$ classes. For all $p_c \in C$, the distance function $M_c^f$ will issue correct and uniform camera-space distance estimates, whereas for all $p_b \in B$ the distance function $M_b^f$ will yield inconsistent distance estimates. Hence, the correct values, $M_c^f$, will create a cluster of estimates around the actual distance between the cameras, whereas the wrong hash values, $M_b^f$, yield randomly scattered estimates.

Since the set of distance estimates $M_c^f$ has been calculated in respect to different pivot points, there is no reason to believe that the actual values of the estimated distances will be uniform. However, experimental results have shown that the inconsistency of the distance estimates that were generated from $M_c^f$ is negligable in comparison to the inconsistency of badly correlated points. A method that does not assume uniformity of $M_c^f$ distance estimates is described later in this section.

The identification of the badly correlated points can be done using a variant of the Least Median Squares method, which takes into account the initial correspondence probability $w_i$. This method was chosen for its robustness, even under a substantial amount of outliers. The overall estimated distance at frame $f$, $M_*^f$, is chosen to minimize the median of the weighted squares of residuals:

$$M_*^f = \arg\min_M med(w_i(M_i^f - M)^2) \qquad (5)$$

Theoretically, the set of badly correlated points (outliers) can now be identified based on the processing of only one pair of frames, $f$ and $f'$. However, in order to further improve the detection robustness, we inspect the deviation of each feature point from $M_*^f$ along a set of frames ($F = \{f_1, \ldots, f_m\}$). We then identify the point $p_E$ which is most likely to be erroneously corellated, using the sum of squared errors along the *whole* set of frames:

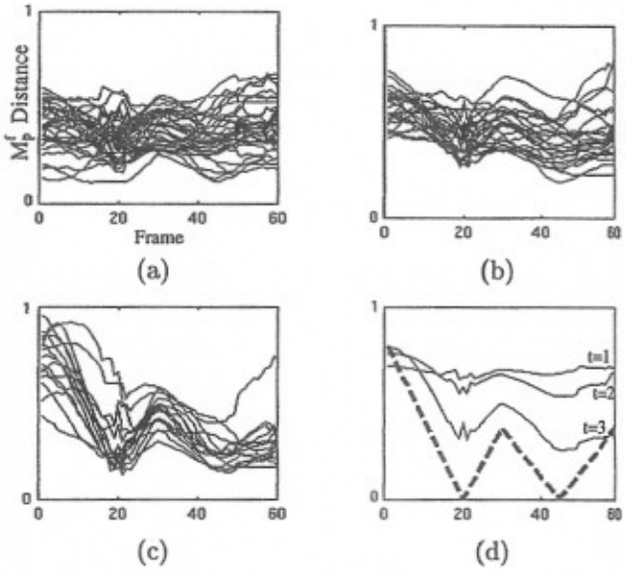


Figure 3: *(a-c) Three successive RDH iterations on a set of 35 feature points, out of which 24 were outliers. Each line represents the estimated distance obtained by selecting a different pivot. (d) The overall distance estimate after each iteration, and true distance between cameras (dashed line). Points of minimal distance can easily be identified.*

$$E = \arg\max_{1<i<n} \sum_{f \in F} (M_i^f - M_*^f)^2 \qquad (6)$$

To conclude, the iterative stage of the algorithm consists of locating $p_E$, discarding it (by ignoring this point from this stage on) and recalculating the RDH vectors. Note, that after a badly correlated point is discarded, the accuracy of the remaining hash values ($M_p^f$) increases. This process is repeated until the variance of the hash values, $\sum_{f \in F}(var(M_i^f))$ drops below a given threshold, indicating that most of the badly correlated points have been detected (Figure 3).

An alternative method that allows the comparison of the estimated distances without assuming that all $M_c^f$ values are uniform consists of inspecting the *relative* change in the estimated distance along the image sequence ($F = \{f_1, \ldots, f_m\}$). At any given transition between frame $f_i$ and frame $f_{i+1}$ the distance between the two cameras either increases, decreases or remains constant. This change in distance between the cameras will necessarily yield a corresponding change in $M_c^f$ distances. Assume, for example, that two cameras are moving towards each other at the time two

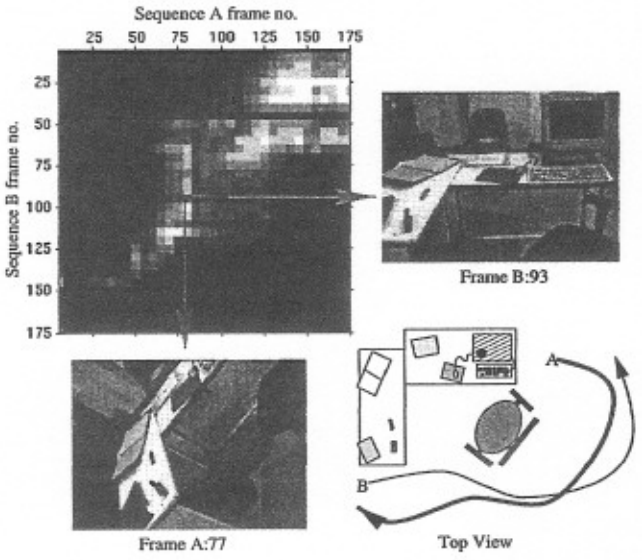


Figure 4: Proximity estimation between two video sequences. Bright spots indicate lower $M_*^f$ value, i.e. greater proximity.

frames, $f_i$ and $f_{i+1}$ are taken. In that case we obtain $M_c^{f_i} > M_c^{f_{i+1}}$ for all $p_c \in C$. In other words, the sign of the derivative of $M_c^f$ over $f$ will be unifom.

We can therefore identify the point $p_E$ by inspecting the number of frames in which the sign of the derivative of $M_E^f$ does not conform with the majority of derivative signs of all $M_p^f$.

### 4.3 Distance Search

Once cross-sequence correspondence has been obtained, the estimated distance between the image sequences can be computed. As we have shown above, $M_*^f$ monotonically increases with the increase in camera distance and can therefore be used to obtain the distance estimate between any pair of frames.

For many applications, the exact distance between the cameras at each frame is not required. Rather, only the frames in which the cameras were at maximal proximity are of interest. Since the distance between cameras is known to be smooth and continuous, linear search methods can be used to find those points without having to calculate the distances between any two frames.

## 5 Experimental Results

The feasibility of the described algorithm was tested on both simulated and real-world images. The main advantage of simulated images is that the exact location of each "camera" is precisely known along the

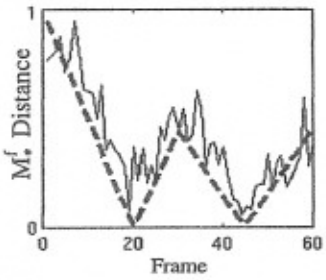


Figure 5: Results of RDH on simulated data in which 50% of the theoretically visible feature points are occluded at every frame, and 30% of the points are inaccurately tracked.

sequence. Thus, the results obtained by applying the algorithm can easily be compared to true camera location. Moreover, a controlled amount of noise can be introduced into the simulated data. Once the feasibility of the algorithm has been proven on simulated data, it was tested on a pair of uncalibrated real image sequences.

### 5.1 Simulations

The two image sequences used in these experiments were "captured" in a simulated 3D scene in which 35 interest points have been tracked. Each sequence consists of 60 frames. The paths of the two cameras are independent, and intersect at two points (frame 20 and frame 45). The 3D location, focal length and 3D rotation of the cameras vary simultaneously along the sequence. In Figure 5 the robustness of the algorithm to 50% feature flicker combined with inaccurate feature tracking can be seen. Figure 3 shows three iterations of the RDH vector calculations with nearly 70% erroneous initial cross-sequence correspondence.

### 5.2 "Desktop" Sequence

The two sequences used in this experiment were acquired using a standard hand-held video camera. Each sequence consists of 175 color frames, in which a maximum of 13 interest points were tracked. Interest points were identified as blobs of uniform color. Figure 4 shows the matrix of estimated distances between the two sequences. The bright area along the secondary diagonal suggests that the trajectories of the cameras were inversely aligned (see top view in Figure 4). Thus, the first frame of one sequence was captured from relatively close position to where the last frame of the other sequence was captured, and vice versa. Two frames captured from relatively close

positions (frame 77 in sequence A and frame 93 in sequence B) are shown.

The full video sequences and the estimated distance can be seen in Movie 1.

## 6 Conclusions

We have proposed a camera-proximity measure, which can be viewed as an essential building block in the rapidly evolving field of multi-camera analysis. A new frame hashing function, denoted Radial Distance Hashing has been shown to be a correct and efficiently computable measure to detect frames that were taken from proximal positions by two uncalibrated cameras.

A method for simultaneously estimating camera proximity and feature point correlation using a statistical comparison of RDH vectors was described. This method has been shown to be robust to substantial noise, thus qualifying for the analysis of real-world situations.

RDH offers a novel view on geometric invariance in image analysis. It does so by restricting the functional operation to the cost-effective analysis of 2D relations, yet relating directly to observations regarding the three-dimensional world.

### Acknowledgments

We would like to thank S. Peleg and G. Hoffman for valuable comments and discussions.

## References

[1] G. Adiv, "Determining 3d motion and structure from optical flow generated by several moving objects," *IEEE Trans. on Pattern Anal. and Machine Intell.*, 7:384-401, 1985.

[2] S. Avidan and A. Shashua, "Novel View Synthesis by Cascading Trilinear Tensors," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 4(4), 1998.

[3] O. Faugeras, "Three-Dimentional Computer Vision: A Geometric Viewpoint," 1993.

[4] J. Heel, "Direct Estimation of Structure and Motion from Multiple Frames," A.I. Memo No. 1190, March 1990.

[5] B. Horn, "Relative orientation," *Int. Journal of Computer Vision*, 4:59-78, 1990.

[6] V. Hwnag, "Tracking feature points in time-varying images using an opportunistic selection approach", *Pattern Recognition*, 22:247-256, 1989.

[7] M. Irani and P. Anandan, "Video Indexing on Mosaic Representations," *Proceedings of the IEEE*, 1998.

[8] R. Jain, R. Kasturi and B. Shunck, "Machine Vision," 1995.

[9] T. Kanade, P. W. Rander and P. J. Narayanan, "Virtualized Reality: Constructing Virtual Worlds from Real Scenes," IEEE MultiMedia, 4(1), May 1997.

[10] M. Kertesz and Y. Yeshurun, "Radial Distance Hashing," Technical Note No. 0799, March 1999.

[11] Joseph L. Mundy and Andrew Zisserman, "Geometric Invariance in Computer Vision," 1992.

[12] S. Peleg and J. Herman, "Panoramic Mosaics by Manifold Projection," June 1997.

[13] V. Salari and I. K. Sethi, "Feature point correspondence in the presence of occlusion," *IEEE Trans. on Pattern Anal. and Machine Intell.*, 12:87-91, 1990.

[14] I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Trans. on Pattern Anal. and Machine Intell.*, 9:56-73, 1987.

[15] G.P. Stein and A. Shashua, "Model-based Brightness Constraints: on Direct Estimation of Structure and Motion," *CVPR*, June 1997.

[16] Phil Torr, Andrew W. Fitzgibbon and Andrew Zisserman, "Maintaining Multiple Motion Model Hypotheses Over Many Views to Recover Matching and Structure," *Proc. 6th International Conference on Computer Vision*, pp. 485-491, 1998.

[17] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Roayal Society, London B*, 203:405-426, 1979.