

Perceptual Cost Function for Motion Detection and Object Tracking

Thomas B. Kinsman, *Student Member, IEEE*,

Roxanne L. Canosa

Abstract — A perceptual distance metric, the EMD cost function, has been developed to track moving objects across pairs of images in a sequence. This perceptual metric, based on The Earth Mover’s Distance, was incorporated into a block-motion estimation program. Additional heuristics were incorporated based on probabilistic cognitive models of human motion perception.

Synthetic motion image sequences were generated to develop and stress test the metric. Using these synthetic motion image sequences, the metric is compared to the standard sum-of-absolute-differences [SAD] block-motion estimation method.

This new distance metric successfully detects and tracks subtle perceptible changes in illumination, texture, and contrast, even when the SAD method fails to estimate any motion. This new motion estimation technique could be developed into a drop-in replacement for standard SAD block-motion estimation method for improved motion estimation.

Index Terms—Image motion analysis

Earth mover distance, motion analysis, motion compensation, optical tracking, object recognition, pattern recognition.

I. INTRODUCTION

Block motion estimation is usually estimated using a Sum-of-Absolute Differences [SAD] technique. The SAD technique is the basis of most MPEG motion encoders. Nevertheless, it is surprisingly easy to demonstrate motion sequences which are clearly perceptible, but for which the SAD technique fails. Essentially, SAD has difficulty detecting the correct motion in several situations where motion is clearly perceptible to humans:

- Illumination changes – an object moves into a shadow, or a flash of light moves over a scene.
- An object with similar statistics as the background moves across the background, i.e., the “camouflage” problem.

Manuscript received Aug. 30, 2007. This work was part of an independent study at Rochester Institute of Technology.

T. B. Kinsman, MS-ECE, is an independent researcher, and Graduate Student at Rochester Institute of Technology. He spent 19 years in Kodak Research Labs, is a graduate of the Kodak Image Science Training Program, and holds an MS-ECE from Carnegie Mellon University. (e-mail: thomaskinsman at netacc.net).

R. L. Canosa, Ph.D., is at the Rochester Institute of Technology, One Lomb Memorial Drive, Rochester, NY 14623., USA. (e-mail: rlc at cs.rit.edu).

- A cloud whose statistics differ only slightly from the background. For example, the “white cat on a white rug” problem.
- A change in texture which moves over the image. For example, the “green golf ball on a green lawn” problem.

Test sequences were generated to simulate these cases, and were used to compare and contrast the robustness of the SAD method with the EMD cost function. In most cases, 5 out of 256 counts of noise was added to stress the system.

A. Generic Block Matching Motion Estimation

The general steps in block motion (SAD) estimation are:

1. Tessellation of the original image into tiles.
2. Local motion estimation on a tile by tile basis.
3. Background or Global motion removal.
4. Local Motion Estimation.

B. Why SAD Fails:

The SAD technique uses an 8x8 block from one image as a template in the subsequent image, and convolves the block over all possible locations. Typically, this fails because:

1. Not enough change has occurred to detect a change.
2. Movement of the object itself has caused the convolution to fail.
3. Change occurred in such a way that the best match is the current location.
4. Multiple contradictory solutions are found, and no rule exists to reduce the ambiguity.
5. The resulting motion is of such a low magnitude that it is indistinguishable from noise, and rejected.

C. Testing Approach

To reliably reproduce the failures, a synthetic motion test suite was developed. The hypothesis for the tests is that a block-matching technique based on the Earth Mover’s Distance (EMD) [1] would better match the perceptual motion experience by the human visual system (HVS). A single component version of the EMD was developed, and enhanced into the EMD Cost Function.

II. SYNTHETIC TEST SEQUENCES

Several types of synthetic sequences were generated:

A. The Square

The first sequence consisted of a square 8 pixel x 8 pixel light-colored block moving over a medium grey background, as shown in Figure 1. The SAD algorithm was applied to this sequence. This test sequence was used for development and for fundamental understanding of the issues, such as aliasing.

B. The Lump

The second sequence consisted of a Gaussian “lump” of lighter region which is added to the background noise, as shown in Figure 2. For these sequences, the background was a randomly generated noise pattern. This simulates the “flash of light in the dark” problem.

C. The Blob

The third sequence consisted of a region of high contrast superimposed over the background, with a uniform mean value across the image, as shown in Figure 3. This simulates the “white cat on the white rug” problem. Note that the mean value is 128 for all regions.

D. The Disk:

The fourth sequence consisted of a region of different texture superimposed over the background, with a uniform mean value across the image, as shown in Figure 4. This simulates the “green golf ball on a green lawn” problem.

To demonstrate that the EMD cost function is not a mean shift algorithm, the mean value of each region of this image is set to the same value; thus, the foreground and background have the same mean values. The object differs from the background only by the amount of contrast, the texture, or possibly both.

Note that the contrast of Figures 1-4 is greatly enhanced for illustrative purposes. Each region is also magnified in size; the same level of magnification for each figure. Also, for the “Lump”, “Blob”, and “Disk” sequences, 5 counts (on average) of uniform noise was added to each frame.



Fig. 1. The Square, an 8x8 square, one tile, moving from left to right.

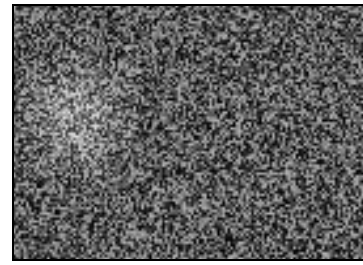


Fig. 2. The Lump, a brighter region (on left).

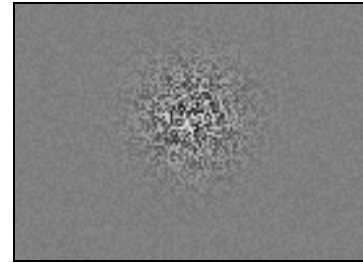


Fig. 3. The Blob, a region of higher contrast with uniform mean.

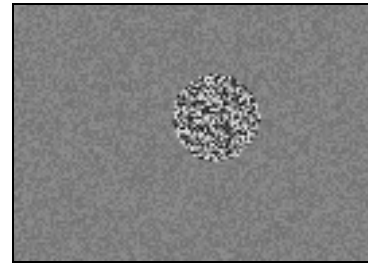


Fig. 4. The Disk, circular region of rougher texture and higher contrast (seen at lower right).

E. Discussion on the Use of Synthetic Sequences

It should be emphasized that while real-world image sequences were used periodically to assure the validity of the results, the development and testing of the synthetic image sequences was important for understanding how and when the algorithms failed. On the other hand, real-world sequences are easier for the algorithm to track objects in because objects in the real world tend to have higher contrast edges, more shadows, reflective surfaces, and tend to be more unique. However, with real world objects, a correct motion estimate might come about by chance rather than by design.

In practice, some erratic vectors (failures) were noticed in the real-world sequences. In those cases the use of simple, repeatable, synthetic sequences was essential for understanding the cause of the errors. Extracting knowledge about failures out of real world sequences is very difficult. Running real world sequences in a debugger, and waiting for the erratic vector to happen would be time consuming, and would not yield any useful information about the relative success of the algorithm.

III. TECHNICAL CHALLENGES

A. Tile Sampling Rate and Need for Over-Sampling

Typically, tessellation, as used with MPEG encoding, is exhaustive and will yield a unique solution; however, the result may be insufficient, depending upon the application. Since the EMD cost function is based on the histogram of the sampled tile and contains no texture information (for computational speed) additional sampling is necessary to find the best motion estimate.

The synthetic square sequence revealed that temporal aliasing occurs in some situations when the object being tracked starts half in the tile and moves to being half out of the tile. Since the background is uniform in this situation, no statistical change occurs to the tile. This situation is shown in Fig. 5. The center tile has the same statistics from one frame to the next, meaning that when looking at two possible frames, the algorithm cannot select between the tile the object is entering and the tile the object is leaving. As a consequence, the “no change” ruling measured no motion for the aliased tile. To compensate for this, the image was over-sampled by a factor of two both horizontally and vertically. This sampling is exhaustive and complete, but not unique.

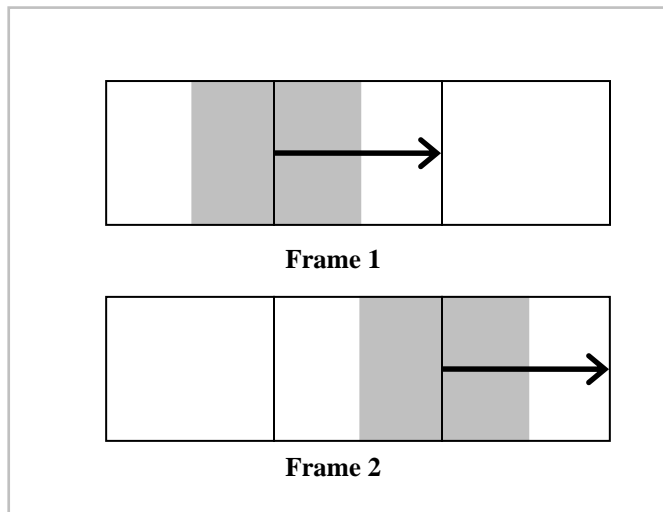


Fig. 5. Temporal aliasing of the central tile.

B. Computational Time

The original Earth Mover’s Distance, as described by Rubner [1], included the computation of Gabor wavelets, and terms for geometric orientation. To perform such a time consuming operation on each 8x8 tile would be computationally prohibitive. Further, Rubner allowed for multiple color planes, and used linear programming to solve for the shortest distance between two images. Again, such computation is prohibitive for most motion detection algorithms.

Fortunately Rubner also anticipated the use of his work for one color plane. The use of motion estimation based only on

one color is not unreasonable, especially when mimicking the human visual system, which is biased towards luminance information.

For one color plane, the EMD is calculated from the 1D histogram. In this case, computing the EMD is computationally tractable [1] [2]. The formula for computing the 1D EMD is given in Eq 1, where d_{ij} is the distance between values, and f_{ij} is the frequency of each value.

$$EMD_{dist}(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad (1)$$

Eq 1 - The 1D Earth Mover's Distance [3]

IV. DEVELOPMENT OF THE EMD COST FUNCTION

A. When the EMD Distance Failed

Initially, the EMD distance was used to locate the tile. This worked very well for the single square of 8x8 pixels that was moving through a flat field. Unfortunately, when searching for a Gaussian blob against the background, it fails. The reason for the failure is that on a Gaussian noise background, finding a Gaussian noise object is difficult using the distance between the histograms. For example, consider the case where the average value of the texture sought is identically equal to the average value of the texture of the background. The EMD distance alone does not discriminate the difference, while the human visual system certainly can.

B. The EMD Error Metric

To compensate for failure due to a Gaussian noise object, the concept of the *EMD error metric* was developed. EMD error is the difference between two histograms, taking the EMD distance into consideration, as shown in Equation 2.

$$EMD_{error} = \sum |f_i - g_j| : j=i - EMD_{dist} \quad (2)$$

Eq 2 - The 1D EMD Error

To extend the Earth Mover analogy, consider this “error” as either:

- The left-over dirt, after each individual pile of dirt has been moved into its respective hole, or the dual situation –
- The amount of dirt missing from each hole, after each respective pile has been filled into it.

From a pattern recognition point of view, the EMD error could be considered a local computation of the Bhattacharyya distance – a measure of discrimination between Gaussian distributions[4]. The Bhattacharyya distance is used as a metric of class separability.

C. When the EMD Error Failed

The EMD error successfully tracked the Gaussian “blob”. Unfortunately, it failed to track the simple square tile moving against a flat background. This is because once the EMD distance has been taken into consideration, the distribution of the tile and the distribution of the background are exactly the same: both are flat, uniform, images.

D. Best of Both Worlds – The EMD Cost Function

To incorporate the best of both techniques, a cost function was created to minimize over the search region. This cost function is given in Equation 3, where EMD_{dist} is the EMD distance, and EMD_{error} is the EMD error.

$$EMD_{cost} = 8 \times EMD_{dist}^2 + 7 \times EMD_{error}^2 + \sqrt{\Delta x^2 + \Delta y^2} \quad (3)$$

Eq 3 - EMD cost function

The inclusion of the length of the motion vector allows the system to find the “minimum motion” in cases where both the EMD distance and the EMD error are zero. The minimum motion term matches perceptual models of the human visual system [5].

V. ALGORITHM

A. Algorithm steps:

The implementation of the EMD motion estimate is an enhancement of the generic block matching algorithm, with several heuristics added to increase accuracy. The steps are given as follows:

1. Convert each image in the sequence to luminance values using openCV’s `cvCvtColor(... CV_BGR2GRAY)`.
2. Local change detection:
On a tile-by-tile basis, the SAD is computed to detect local changes. Tiles which change by an amount below a threshold are classified as “background” tiles. A threshold of 256 code values per tile is used.
3. Over-Sampling:
The image is tessellated into 8x8 pixel tiles, over sampling by 2 both vertically and horizontally.
4. Morphology:
Individual foreground tiles which are surrounded by background are rejected. This rejects real world changes such as white-caps, point light sources coming on, etc...
5. Local Motion Estimation:
Remaining foreground tiles in the original image are searched for in the subsequent image. A 49x49 region is searched for a match. The best match is selected, based on the EMD cost function (or the SAD method for comparison).

6. Edge Cleaning:

Some outlying edge vectors are rejected as being highly unlikely, even if they do minimize the cost function.

7. Non-Maximal Suppression:

To mimic human perception, for each remaining tile motion vector, the surrounding neighborhood of tiles are searched to see if it is the local maximum. Only local maxima are kept. Motions below the threshold of 1.4 pixels per frame are treated as noise and suppressed.

8. Vector Averaging:

The vectors which remain are averaged together and an “average vector” is returned. Lacking segmentation, this presumes only one object is in motion.

B. Algorithm Block Diagram

Figure 6 shows a diagram of the motion estimation phases.

VI. EXPERIMENTAL METHODOLOGY

A. Simulation

The test suite of synthetic sequences started with a simple moving 8x8 block. Development focused on the image sequences which gave the worst results. After successfully tracking the motion in the initial sequences, the test suite was grown to include more difficult sequences, the worst cases were then identified, and failure modes of those test cases were again examined for further improvements.

B. Why add noise?

The ability to add noise enabled a crude simulation of noise that is added in the image processing chain of a camera, or noise resulting from the CMOS/CCD sensor itself. In both cases, the ability to add noise enabled the possible study of how much noise the methods could tolerate. Contrast control also enabled testing the robustness of the algorithm.

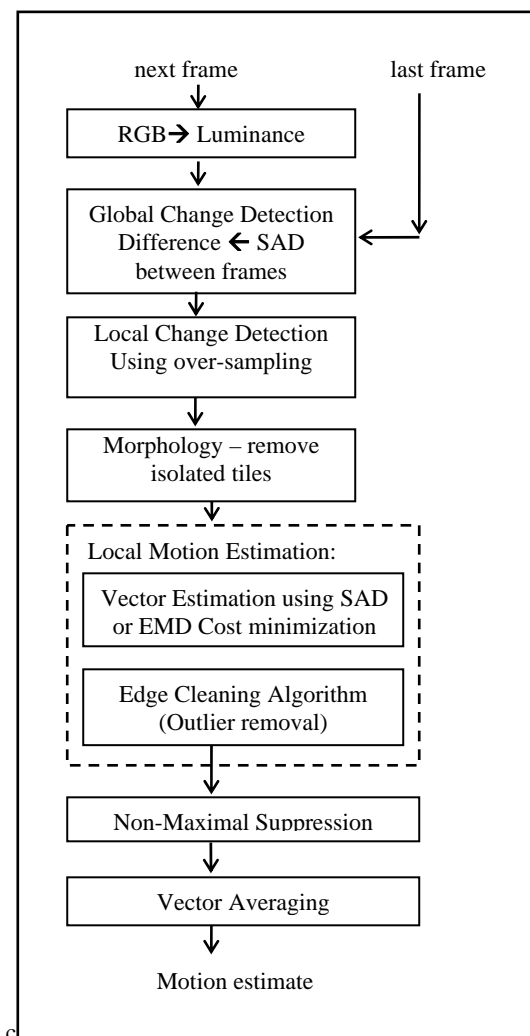


Fig. 6. - Motion Estimation Phases

VII. RESULTS

A. Changing Speed

The first experiment used the “Lump” sequence with a maximum boost set to 30 code values above the background pattern. No noise was added. At this level, the visual difference is barely discernable. The speed was varied from 0 to 12 pixels per frame, and the algorithms were run.

Results from this experiment found that the SAD method completely failed to register any motion. Additionally, the EMD cost method failed to detect any change until the velocity reached 4 pixels per frame. The average velocity over the entire sequence of 32 frames is shown in Figure 7.

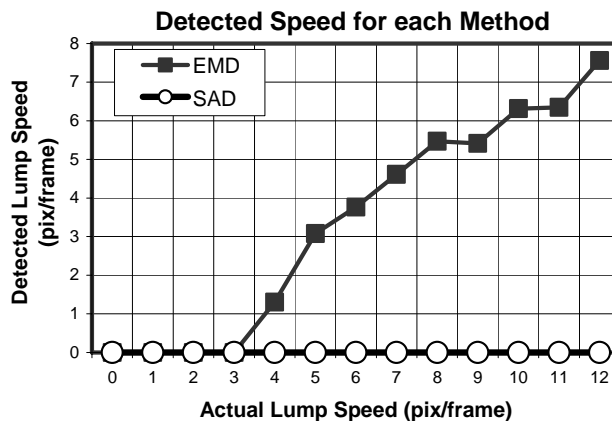


Fig. 7. Detection speed of the Lump at various actual speeds.

These results are typical of the current instance of the EMD detection algorithm - the speed detected is below the actual speed. However, compared to the SAD method, which detected no motion, this is an obvious improvement. Also, consider that the EMD method is currently set to ignore vectors whose magnitude is less than 1.4 pixels per frame. This causes small motions to be ignored as noise.

Fig. 7 shows speed detection rising through the background pattern. At this low of a boost, the local region did not change enough frame-to-frame to create a detected signal for the motion estimation phase to look for, at slow speeds. Clearly, faster motions cause more change, which is easier to detect.

B. Detection through Increasing Background Noise:

For the second experiment, the lump boost was fixed at 40 counts above the background, and the speed was fixed at 8 pixels per frame. The background noise level (noise added as a change in the background from frame to frame) was increased for each sequence. Results found that the SAD method again completely failed to notice any motion; the EMD detected speeds are shown in Fig. 8.

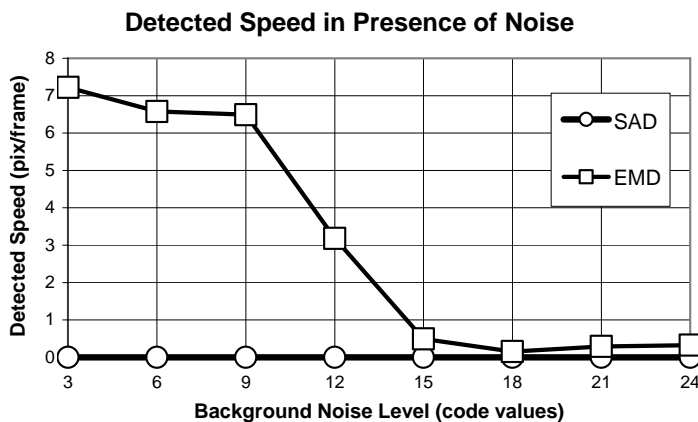


Fig. 8. Detected speed of The Lump in the presence of background noise.

Here the SAD method noticed a change, but produced no motion vector. In this case, the object was detected with a

zero motion vector – the tile which best represented the sought tile was the original tile.

Clearly, the EMD cost method is susceptible to background noise. Too much noise prevents the EMD cost function from finding the correct motion vector, although a change is detected.

C. Results for Rising Foreground Noise:

Figure 9 shows the result when the foreground noise was increased, while the lump boost was fixed at 40 counts. Again the SAD method failed to detect any motion.

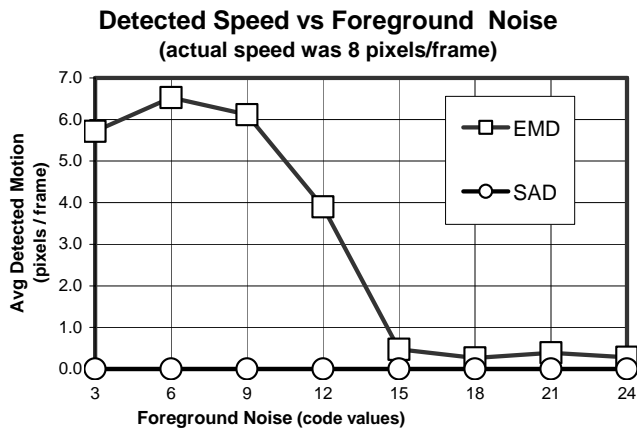


Fig. 9. Detected lump speed in the presence of increasing foreground noise.

In agreement with the earlier results, the EMD method suffers from the presence of noise – less is better. More importantly, the SAD method once again failed to notice any motion.

D. Contrast Stress Test For The Blob:

In this experiment, the “Blob” sequence was used and the contrast was changed between the foreground and background. The contrast is expressed in terms of code values. A value of 32 means that the data will roughly range over 32 code values, centered about a code value of 128. The mean for the image was 128.

As the background contrast was increased, the foreground contrast was also decreased – so that in the mid ranges the contrasts were very similar, and difficult to detect. On the other hand, this should make it easy to detect change when the background contrast is low.

As might be expected, when the contrasts were nearly equal, no motion vectors were detected. However, at higher contrast values, the EMD method detected more motion vectors than the SAD method, as shown in Figure 10.

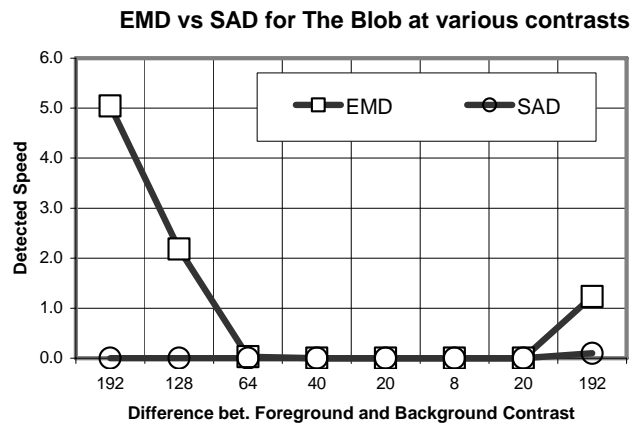


Fig. 10. Detected speed of The Blob as a function of difference between foreground and background contrast.

Clearly, the EMD method was a better motion estimation technique than the SAD method for this case.

E. Contrast Stress Test For The Disk:

In this experiment, the disk was used, and the contrast was changed between the foreground and background. Again, contrast is expressed in terms of code values, so a value of 32 means that the data will roughly range over 32 code values – centered about 128.

The “Disk” sequence has a completely different background texture than the foreground, consisting of random dots that are roughly twice the size as the dots in the foreground. In addition, the dots do not move with respect to the image location, so the changes across the image results in a good edge for the change detector to detect. When the contrasts were nearly equal, fewer motion vectors were detected. However, at higher contrast values, the EMD still detected more motion vectors than the SAD.

Of particular interest is that when the difference in contrast between the foreground and background was 0 (no contrast difference) the EMD cost function still was able to estimate more motion than the SAD motion estimator. Figure 11 depicts these results.

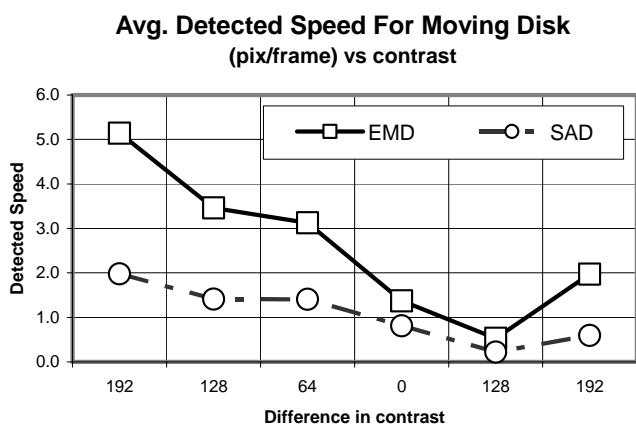


Fig. 11. Detection speed of the Disk, as a function of change in contrast between foreground and background

Again the EMD Cost Function estimates more motion than the SAD motion estimation method for a change in texture.

As a point of clarification, in Fig. 11, the horizontal axis is the *difference* in contrast. At the value of zero here, the contrast of both the foreground and background were set to 128 counts.

VIII. CONCLUSIONS

This study introduces the Earth Mover's cost function – a combination of a one dimensional Earth Mover's Distance metric with the newly developed EMD error function. The Earth Mover's cost function can be used to robustly track objects in motion.

The biologically inspired EMD cost function provides a simple, straight-forward technique for perceptual local motion estimation. The use of the Earth Mover's error, combined with the Earth Mover's Distance, creates a valid cost function to minimize error and resolve motion vectors in the presence of background noise. The introduction of the EMD cost function, as a new perceptual metric, obviates the need for sophisticated modeling which might only match a particular data set.

These initial results seem promising. It is hoped that the addition of other techniques, such as physical modeling (momentum or Kalman filtering) will help improve the results.

ACKNOWLEDGMENT

T. B. Kinsman thanks Yossi Rubner for answering various e-mails regarding his ground-breaking thesis.

REFERENCES

- [1] Yossi Rubner, Carlo Tomasi, and Leonidas Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval,"

International Journal of Computer Vision, 40(2), 99-121, 2000. Kluwer Academic Publications, Netherlands.

- [2] Yossi Rubner, "Perceptual Metrics for Image Database Navigation," Stanford University Ph.D. Dissertation, May 1999, p. 29.
- [3] Yossi Rubner, Carlo Tomasi, and Leonidas Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," International Journal of Computer Vision, 40(2) 99-121, 2000, p. 105, Kluwer Academic Publications, Netherlands.
- [4] Theodoridis and Koutroumbas, Pattern Recognition 3rd ed., 2006, Elsevier, Academic Press, New York, p.227.
- [5] Rao, Olshausen, & Lewicki (editors), "Probabilistic Models of the Brain: Perception and Neural Function," 2002, MIT Press, Cambridge, Massachusetts, p. 79

Thomas B. Kinsman (M'07) BS-EE University of Delaware, Newark, DE, MS-ECE Carnegie Mellon University, Pittsburgh, PA.

Mr. Kinsman worked with the Kodak Research Labs from 1987 until 2006, on Image Science and digital photography. He was the first one to format digital images correctly for writing to CD-ROM at Kodak. He worked on noise cleaning, image compression, mobile imaging, performance optimization, embedded development, and many other fields during his tenure with Kodak. He left Kodak to pursue ideas related to pattern recognition, and motion imaging, and is currently pursuing graduate training at the Rochester Institute of Technology.

Mr. Kinsman is a lifetime member of Tau-Beta-Pi. He has earned awards in the study of biology, photography, and in the field of cognitive science commonly called "magic". Knowledge of these fields contributed to this work.

Roxanne L. Canosa Ph.D. Imaging Science, Rochester Institute of Technology, Rochester, NY.

Dr. Canosa is a professor of Computer Science at the Rochester Institute of Technology, and conducts research on computer vision and image understanding algorithms. Her primary interests are in simulating human visual perceptual capabilities and human eye movements in artificial vision systems.