Video Based System Monitoring

by

Brian W. Anthony

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author ....
Department of Mechanical Engineering
May 21, 2006

Certified by ....
Kamal Youcef-Toumi
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Professor Lallit Anand
Chairman, Department Committee on Graduate Students

Video Based System Monitoring

by

Brian W. Anthony

Submitted to the Department of Mechanical Engineering
on May 21, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

In this work we develop new algorithms for video comparison, for video alignment, and for determining the similarity between entire video clips or detecting similarities between sub-videos. The intent of this work is to develop video-based techniques for autonomous monitoring of systems in industrial, manufacturing, and research environments.

We develop an algorithm, Dynamic Time and Space Warping, to determine a model-free similarity between an example and an unknown video. The algorithm optimally shifts space and warps time according to local measures of video similarity. The resulting similarity measure is an optimal path of similarity versus space and time used to optimally align or compare the two video. We demonstrate the applicability of such similarity measures to industrial wear monitoring, failure prediction, and assembly-line feedback control and to non-industrial settings with examples in sports and entertainment.

We extend the similarity machinery and introduce a new technique for video event-detection. The local similarity is integrated along the optimal space-time path in order to determine a cumulative similarity. We demonstrate the applicability to content query and surveillance; we identify the temporal and spatial location inside of a large video stream which is similar to a query, or template, video. We explore applications in video classification.

We investigate the performance degradation and robustness of the algorithms to noise via distortion of real examples and simulation. We develop techniques to aid engineers in the selection of a video template that is relevant to their monitoring application and locally robust to noise. We explore the structure and computational complexity of the algorithms. We demonstrate that the algorithms are highly-parallelizable and that the fast processing rates necessary for many industrial monitoring applications are achievable.

Thesis Supervisor: Kamal Youcef-Toumi
Title: Professor of Mechanical Engineering

Acknowledgments

First and foremost, I thank Professor Kamal Youcef-Toumi. Without his support and understanding I would never have completed the journey that has culminated in the creation of this document. It is an honor to call him my advisor, my colleague, and my friend.

I thank my committee, Professors Kamal Youcef-Toumi, George Barbastathis, William Freeman, and Derek Rowell for their advice and insight. I thank my family: Linda (a.k.a Mom), Justin (a.k.a Sparky), Bao, and of course Tia. Thank you all for being you. You all have my love. Maria Carbone's friendship, love, patience, words, and understanding has helped me grow as a person, to you I am indebted. My two most dedicated proof readers were Linda Anthony - thank you mom - and Kim Curtis - thank you.

Though it has been years since I've talked with many of you I thank Craig Latimer, Vinay Shaw, Akhil Madhani, Gunter Niemeyer, Susanna Leverrone, Chris Tarr, Jesse Hong, Cynthia Breazeal, Ela Ben-Ur, Arrin Katz, Tom Stahovich, and Ken Salisbury, for fun in the early years. To old friends who I have not mentioned, I thank you. I also thank others who have altered the course of my life, Dipen Sinha, Pete Carellas, and Aaron Anthony.

The path towards my doctorate has been fraught with the detours and roadblocks of life. Several personal and professional hurdles have been as much of a learning experience as the knowledge creation that went into this document. I thank the philosopher Kenneth E. Winsome for his words of wisdom, "Life. A journey, not a destination."

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The determination of two videos similarity (or dissimilarity) is a central requirement for algorithms developed for video classification, video detection, and video alignment. Many applications for video-based system monitoring in industrial, manufacturing, and scientific environments are neither strictly classification nor detection applications. Industrial applications, such as lifetime testing, and failure detection, require a detailed determination of similarity between two videos (a template video and test video)[1]as a function of space and time. Applications in sports-training and physical-therapy will also benefit from a technique to spatially and temporally localize similarity between two videos. These applications require a comparison algorithm that can report detailed, local, similarity between two videos as a function of space and time before this similarity is integrated over the entirety of the video volume to determine a cumulative similarity.

Additional algorithmic requirements are motivated by the goal of creating versatile algorithms that can address the numerous applications that arise in system monitoring. Fast processing rates, the ability to compare videos of amorphous and flexible subject matter, and the ability to operate on pixel values, motion, or any

[1] Depending on context the known video will be called the “template” video or “query” video. The unknown will be called the “test” video.

volumetric data type are a few of the additional requirements.

To the best of our knowledge no algorithm exists that can report the detailed similarity while also satisfying these, and other, important requirements. We approach the problem of determining detailed, local, similarity by asking how one video must be modified in space and time to make it most similar to another.

1.1.1 Motivating Example - Heart Valve Cycle

A replacement heart valve for the human heart has been developed. The valve is made of tissue from the heart of a pig surrounded by a polymer ring.

As part of the valve development, in order to demonstrate its viability as a replacement valve for humans, it must undergo accelerated lifetime failure tests. The heart valve is placed in a cross section of a pipe. The pressure across the valve is cyclically varied; a clear fluid flows back and forth through the valve. This simulates the oscillating flow in a beating heart. The valve will open and close in response to the fluid flow. A camera is set to record the valve. The system runs unattended for months. We monitor the deterioration of the valve over time.

Deterioration is related to the change in the temporal profile of the opening and closing of the valve. We monitor this change by comparing a video of a cycle of an "ideal", or new valve, to a video of a much later cycle of the valve. Figure 1-1 shows two videos of the heart valve. **Q** is the ideal or new valve, **C** is the older or deteriorated valve.

1.1.2 Motivating Example - Karate Punch

A professional karate instructor trains to eliminate variability from a punching move that is part of the practice of karate. The instructor is recorded performing a punching move. As an example video we use a punch where the instructor was perfect in the execution of the move. Subsequently the instructor compares this ideal video of himself to other less perfect videos of himself and of his students. Perfecting the move entails replicating the combined spatial and temporal behavior of the ideal

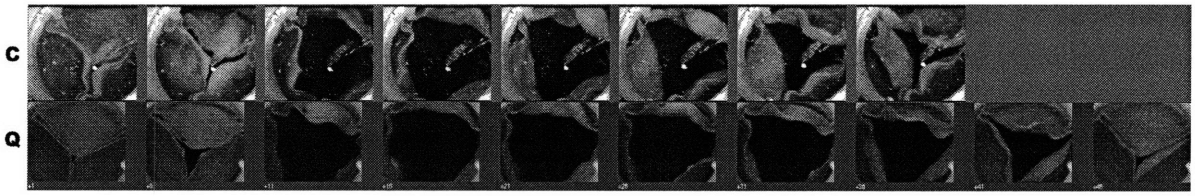


Figure 1-1: Motivating Example - Heart Valve Cycle. The example, template, or query video is labelled **Q**. The unknown or test video is labelled **C**. We monitor the deterioration of a heart valve over time. Deterioration is related to the change in the temporal profile of the opening and closing of the valve. We monitor this change by comparing a video of a cycle of an "ideal", or new valve, to a video of a much later cycle of the valve.

example. Two videos that we want to compare are depicted in Figure 1-2. **Q** is the video of the ideal execution, **C** is the unknown. This is similar to applications in baseball pitching, golf, diving, ice skating, and other sports where it is important to learn-and-perform or judge an ideal motion.

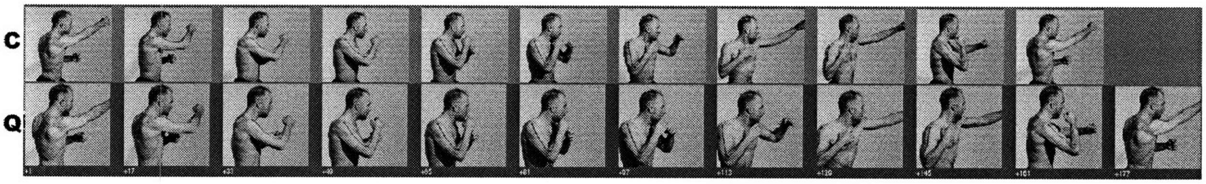


Figure 1-2: Motivating Example - Karate Punch. A professional karate instructor trains to eliminate variability from a punching move that is part of the practice of karate. The instructor is recorded performing a perfect punching move. Subsequently the instructor compares this ideal video of himself to other, imperfect, videos of himself and of his students.

1.1.3 Relationship to Classification and Detection

In the heart valve example, we want to know when and where similarity occurs during a cycle in order to intelligently monitor the degradation of the heart valve. In the karate example, we again, want to know when and where similarity occurs in order to help people learn the move. These applications are neither classification nor detection problems. However, research in both of those areas is relevant to this work. Further, the mechanism that we develop to determine such detailed similarity should generalize to both the standard classification and detection problems.

1.1.4 Video Based System Monitoring

A detailed description of the video-based system monitoring problem for industrial, manufacturing, and scientific environments is provided in Chapter 2. The basic idea, similar to the heart valve and karate examples, is that we compare ideal or template videos to unknown videos that may have deviated appearance, spatial, or temporal characteristics. We do this in order to make decisions about the equipment or process that we are observing. Conceptually this idea is depicted in Figure 1-3.

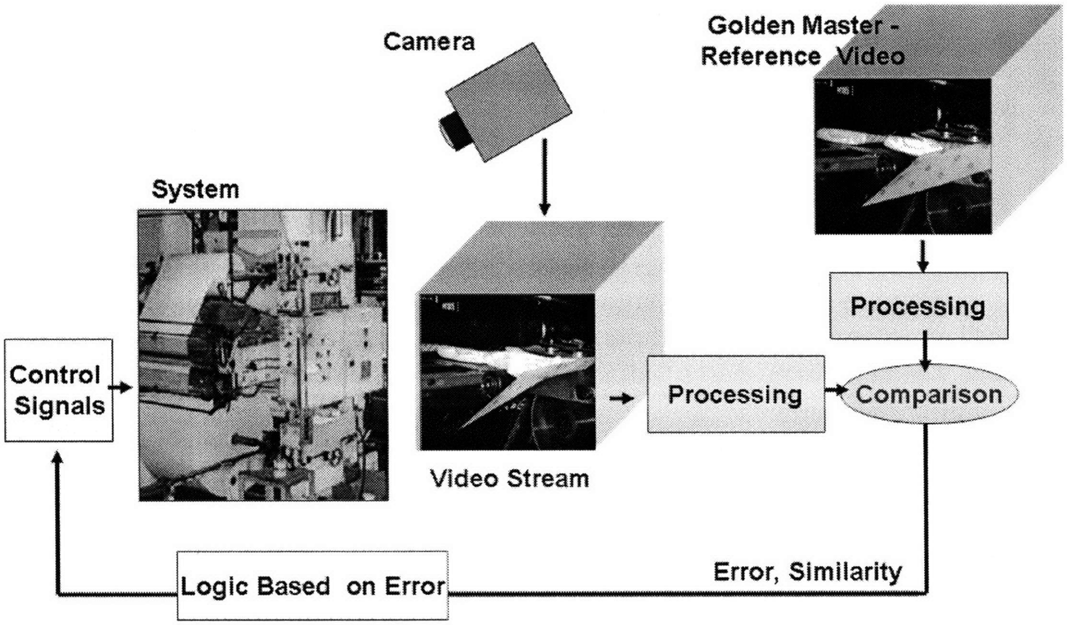


Figure 1-3: Motivation - System Monitoring. We compare ideal or template videos to unknown videos that may have deviated appearance, spatial, or temporal characteristics. We do this in order to make decisions about the equipment or process that we are observing.

1.2 Video Classification, Video Detection, Video Comparison

Video classification and video detection are important areas of research in computer vision. Both areas are built on a mechanism to determining a measure, often a single scalar value, of the similarity (dissimilarity) between a known video sequence [2]and an unknown video sequence. In **video detection** applications, we attempt to find a known (or query) video inside of a temporally and spatially larger unknown by comparing it to "every" subsequence. In **video classification**, known examples from several labelled sets of videos are compared to an unknown sequence in order to identify it as belonging to one of the sets. In a classification application, often, both example and unknown videos are temporally clipped and spatially framed tightly around the subject. The conceptual relationship between detection and classification is shown in Figure 1-4.

This question of similarity means different things for different applications. The types of videos, their expected content, or subject matter will constrain the complexity and dictate the versatility of the comparison mechanism. If, for example, we know that the input to a classification algorithm is always a video of a person walking, then we can customize the video comparison mechanism to that application. However, the resulting video comparison mechanism will not be easily extended to an application classifying videos of airplanes.

Approaches to the classification and detection applications, and therefore to the underlying video comparison problem, are broadly described as being model-based or appearance-based. Model-based approaches first estimate a set of low-order model parameters, such the location of the center-of-mass of a person, and then determine similarity between the model parameters. Appearance-based approaches operate directly on volumetric data, such as of pixel values, optical flow, normal flow, or some combination thereof. Appearance-based techniques, generally, are more computation-

[2] We note that the known videos can be a "composite" or representative video determined as an "average" of numerous training samples.

ally intensive, but are readily extended to different subject matter. In between model and appearance-based techniques are "structural" approaches; volumetric structures or features (that are not motion model or subject specific), such as interest points, [LL03], and contours are extracted and compared.

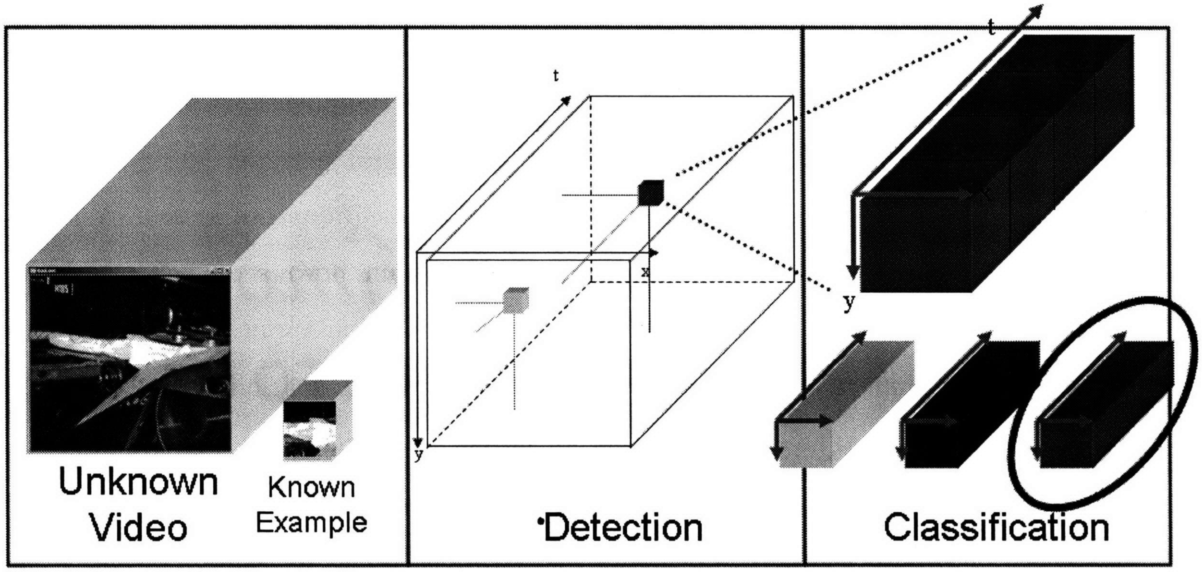


Figure 1-4: Detection and Classification. In **video detection** applications, we attempt to find a known (or query) video inside of a temporally and spatially larger unknown by comparing it to "every" subsequence. In **video classification**, known examples from several labelled sets of videos are compared to an unknown sequence in order to identify it as belonging to one of the sets.

In order to address a myriad of industrial monitoring applications we require :

- A video comparison mechanism that is versatile and applicable to varied subject matter; it is for this reason that we develop an appearance or volumetric-data approach.
- A video comparison mechanism that can be used in detection or classification applications.
- A video comparison mechanism that can provide the local and detailed similarity between two videos and not require numerous training samples.

Following is a partial list of the desirable properties of a video comparison mechanism. This list is a combination of the needs that we foresee for industrial applications, such as heart valve monitoring, and the properties that other researchers have built into their classification or detection algorithms. This list is not exclusive to appearance-based approaches; though we do focus on such approaches because of our need for general applicability to varied subject matter.

Desirable properties of a video comparison mechanism include :

- Robust to spatial and temporal variation between two videos. A desirable property of a video comparison technique is to be able to quantify similarity even if the similarity is subtle. A video comparison technique is not useful if it can only compare videos that are predominantly similar. For example, it should be able to compare a video of a grey horse running fast to a video of a white horse running slowly, not just two videos of the same horse running at approximately the same speed.

- Ability to control the algorithmic sensitivity to spatial and temporal deviation. We would like to be able to control the "search space" of similarity between two videos. Do we allow 20% variation deviation in the position vs. time path of objects found in two video or 80% variation?

- Spatial scale invariance. Changes in the optics, the physical distance of an object, or the size of the object that is video-recorded will cause differences in the spatial size (the number of pixels in height and width) of the object in a video frame. It is preferable if a video comparison technique can accommodate such variation. For example, if the horse running in an example video is 200 pixels tall we would like to be able to compare that video to videos of other horses that are from 150 to 250 pixels tall.

- Rotational invariance. Changes in camera orientation, or the orientation of the objects that are video-recorded will cause differences in the orientation of pixels in a video frame and in the apparent path that an object takes through a video sequence. Video comparison techniques are preferable if they can accommodate such variation.

- Ability to operate on pixels, motion, or any volumetric data. In some industrial applications pixels are the volumetric data that we want to compare, in others, motion is the quantity of interest. Much recent work in video detection and classification tends to discount the use of pixel values. This is done to make techniques robust to different "actors" that may be different in appearance but that exhibit similar motion. In order to address pixel and motion comparison applications in a general way we prefer a technique that is appearance (i.e. "volumetric-data") based. Volumetric-data is any (XYT) arrangement of values such as pixels, optical flow, combinations or moments of the two, or other such values.

- Able to accommodate non-static background. Background "clutter" may be static or dynamic; a comparison technique should be able to accommodate either. A video of a person walking in front of a brick wall should be comparable to a video of a person walking in front of moving traffic.

- Does not require spatially segmented videos. We may not be able to easily extract the subject matter of interest from its background. We prefer that the mechanism be able to ignore varied backgrounds.

- Accommodate subject matter that is amorphous, flexible, and lacking of distinct spatial or spatio-temporal features.

- Naturally accommodate comparing videos of different spatial and temporal sizes.

- Easy to scan. A technique that has been designed for classification of video clips that are spatially and temporally segmented around some subject matter of interest, may not be easily extended to the general detection problem. In video detection we compare an example video to every sub-video of a larger video by scanning the comparison mechanism through the space and time of the larger video.

- Fast computation. A video comparison technique isn't practical if it takes hours to compare two short clips.

- Require minimal training examples, preferably require only a single example video. In many industrial applications we may only be provided with 1 example or training video. We will not necessarily have the luxury of a large statistical sampling of videos with which to train a probabilistic model.

- Does not require body centered videos. The subject matter may be body centered, or move through a video due to its motion or due to camera motion; we want to accommodate either scenario.

- Able to report the detailed local similarity between two videos. We are interested in "when" and "where" similarity occurs (similarity along a time and space path) between two video as opposed to a single scalar measure of overall similarity.

- Preservation of time. Time and timing are very important for industrial applications. Much existing work in video detection and classification smooths or integrates over the time and space axes when comparing two videos. This is done in order to accommodate the expected natural variation between videos, but eliminates the possibility of extracting a more detailed comparison between videos. We prefer a method that accommodates and determines the detailed temporal variation through the entire sequence.

The way that we think about the similarity between two videos, is to ask how must one video be modified in order to make one look like the other. We do this by determining the local similarity between two videos and then by finding an optimal path through the local similarities, mapping one video to the other. In this work we allow time to non-linearly warp, and space to shift. We do not allow the spatial arrangement of information to warp or change relative positions. In future work we will modify our algorithm to allow such spatial distortion. The current work finds optimal paths aligning an entire video to another, a natural extension is to allow sub-video regions to be compared along their own paths parallel to the dominant warp path.

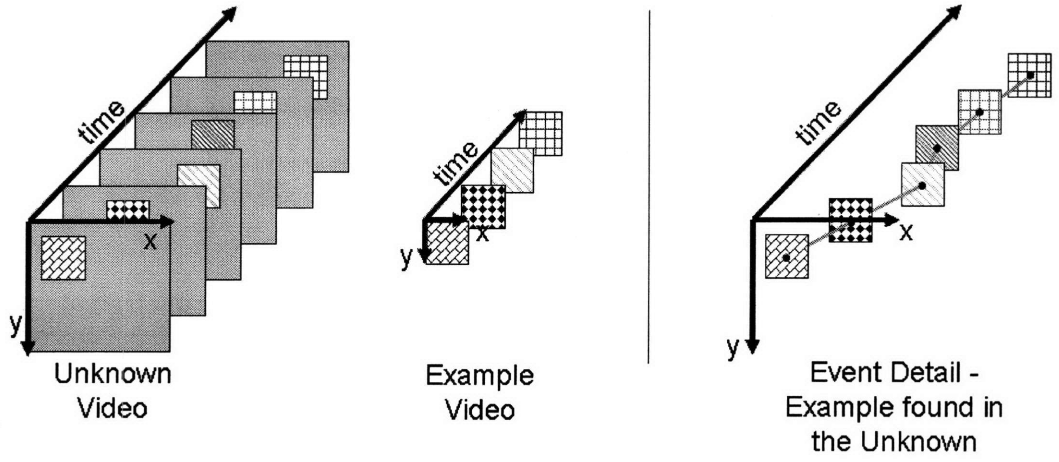


Figure 1-5: Video Event Detail. The video event detail describes the time and space coordinate transformations between the local regions of two videos. Alternatively, it can be viewed as the way that time and space must be warped and shifted to best align the two videos. Finally, it can be viewed as the (time and space) path along which one video is "found" inside of the other. This series of transformations is determined by finding a low cost path of "local" similarity between the two videos. The peaks and valley of the local similarity along this path tells us exactly when and where regions of high and low similarity occur.

The video event detail is illustrated in Figure 1-5. The video event detail describes the time and space coordinate transformations between the local regions of two videos. Alternatively, it can be viewed as the way that time and space must be warped and shifted to best align the two videos. Finally, it can be viewed as the (time and space) path along which one video is "found" inside of the other. This series of transformations is determined by finding a low cost path of "local" similarity between the two videos. The peaks and valley of the local similarity along this path tells us exactly when and where regions of high and low similarity occur.

In this work we develop a technique for determining this detailed spatial and temporal similarity and alignment between two videos. We explicitly ask when and where is the similarity between two videos as functions of space and time? How similar? We answer these questions without models of objects, without tracking features, and by using application-specific volumetric data (i.e appearance-based). We develop a novel framework that extends Dynamic Time Warping (DTW), [HCLR83], [MRR80], [RK04], [KPZG04] to include variation along the spatial axis. We call it Dynamic Time and Space Warping (DTSW).

When applying DTSW to two videos' volumetric-data values we find an optimal space-shifting and time-warping alignment between both videos with a measure of local similarity between each aligned frame. This alignment problem is stated as an optimization problem solvable via a dynamic programming algorithm. We show that the structure of the problem is such that it can be solved through the iterative application of two non-linear image filters. Further the structure of the DTSW algorithm allows massive parallelization, enabling fast solutions. As part of this framework we develop a compact, principal component representation of the known or query video; this facilitates efficient calculation of the data-structure to which we iteratively apply the two non-linear filters.

We demonstrate the use of the DTSW algorithm on several industrial and non-industrial applications. We generalize the application of DTSW to standard detection and classification problems.

The advantages of DTSW compared to existing appearance-based techniques include :

- DTSW can be used to determine detailed similarities between two videos as a function of space and time.
- DTSW is structured such that is can be implemented using parallel processing to increase the rate of operation.
- DTSW uses motion, pixels, or any other volumetric data that is application appropriate.

- DTSW is appropriate for subject matter that is amorphous and flexible.

- DTSW can be used for video comparison or alignment, video event detection, and video classification applications.

- Naively implemented, DTSW is computationally expensive. However, theoretical and practical considerations - and its parallel structure - allow it to be implemented at rates on the order of 100 Hz for video sizes that are industrially relevant.

1.3 Related Work

A video is variously interpreted as a series of frames, a volumetric (XYT) arrangement of data, a point in a multidimensional space, and others. Numerous approaches to the video comparison problem have been developed that combine application [e.g. classification, detection, other] requirements with the various ways of interpreting what a video is. Much existing work in some way internalizes, or otherwise implicity uses, e.g. averages over, the local similarity. This is done to make a technique robust to expected spatial and temporal variation between the two video. As indicated, we want to examine this local similarity. A composite measure of similarity, a single scalar value, may be determined by "integrating" the local similarities over space and time.

Yan Ke, R. Sukthankar and M. Hebert in 2005 , "Efficient Visual Event Detection Using Volumetric Features", [KSH05] represent and compare videos with a volumetric arrangement of one-box and two-box features. A box feature is an XYT region of a video, in which the cumulative sums of optical flow components are calculated. They align training sequences at their temporal beginning; their classifier learns that the beginning of a video is more discriminative than the end; this is inappropriate for industrial applications were the entire sequence is important. Scanning is efficient after the integral video structure is calculated but is not appropriate for very large or continuous video streams. However, this work does not provide a mechanism to examine the detailed local similarity between videos.

Eli Shechtman and Michal Irani in 2005, "Space-time behavior based correlation", [SI05], develop a technique for video detection. Their video comparison technique relies on an abundance of motion and does not allow appearance-based comparisons. To accommodate spatial and temporal variation between videos they integrate a consistency measure over the volumes of the example video. However, there is no mechanism by which one could examine the detailed inter-volume (or frame to frame) matching, only a final aggregate value averaged over a region is determined.

Moshe Blank, etal in 2005, "Actions as Space-Time Shapes" [BGS+05] use space-time volumetric shapes of actors to compare actions. This technique works well for static backgrounds and the simple detection problem. However, it doesn't allow for a detailed temporal and or spatial comparison between two videos. Their technique relies on an abundance of motion and does not allow appearance based comparisons.

Aaron F. Bobick and James W. Davis in 2001, "The recognition of human movement using temporal templates" [BD01] determine representative images that capture the presence and recency of motion through a video. Their process is done in two steps - construction of a motion-energy image (MEI), where motion has occurred - generation of a motion-history image (MHI), where intensity is a function of the recency of motion. Though very compact and efficient, the detailed temporal variation of the video is lost. Further, their work and others is best suited to static backgrounds and action that is both easily segmented from the background and body-centered through a sequence.

Other work also takes the approach of representing a video as a single representative "image". Niyogi and Adelson in 1994, "Analyzing and recognizing walking figures in XYT", [NA94a], use shapes on space-time slice images to model human-gait. BenAbdelkader, etal, in 2001, "EigenGait: Motion-based Recognition of People using Image Self-similarity", [BCND01], calculate a representative self-similarity matrix image from a tracked human in frontal-parallel sequence with static background. Cutler Davis in 2000, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications", [CD00], use self-similarity matrix images of the frames of a video as an approximate phase-portrait in order to detect periodic motion.

Work by Lihi Zelnik-Manor and Michal Irani in 2001, “Event-Based Video Analysis”, [ZMI01], accommodate temporal and spatial variation by representing and comparing videos via time varying histograms of video data. They use a simple statistical distance measure between video sequences based on their behavioral [motion] content. They cluster similar histograms to temporally segment a long video-sequence into event consistent sub-sequences.

Other work uses histogram based comparison for temporal change detection to detect pans, fades, and scene changes that are a natural part of video streams associated with television, movies, sports video streams [MZ02], [NPC99], [NPZ03], [NPZC00].

Michael J. Black in 1999, “Explaining optical flow events with parameterized spatio-temporal models” [Bla99] uses a spatio-temporal representation (frames of optical flow) to represent and classify videos that contain single “actors” that are temporally segmented, spatially framed, and body centered, with a relative static background. Within a Condensation framework, [IB98], [BJ98], a known video is fit to an unknown by determining the phase, rate, spatial position, and scale to account for the variation between the given flow sequence and an unknown sequence. The posterior distribution over this parameter space, conditioned on image measurements, is represented using factored sampling and is predicted and updated through time. The framework is used to classify and localize motion events.

This work is the most capable of the related work that we have found to determine the detailed temporal and spatial relationship between two videos. However, it is not apparent how to efficiently scan the algorithm in either space or time for the general detection problem. Further, the resulting fit (in time and space) is smoothed and less detailed in comparison to what we are able to obtain. This technique is limited to matching videos with single actors. We can simultaneously align an example video to multiple targets in an unknown video. The factored sampling approach is very computationally expensive.

Peter Sand and Seth Teller in 2004, ”Video Matching” [ST04] temporally align a pair of video by searching for complete frames that match best according to a robust

- regression based - image registration process. The authors combine elements of feature matching and optical flow based correspondence. This work requires that the recorded scene is mostly static, and that the cameras that recorded each sequence followed spatially similar trajectories. This work does not allow for spatial shifting in the frame-matching process and can not find an unknown template inside a larger unknown sequence.

Other appearance-based approaches to the detection and classification problem are domain specific, place strict limitations on a video's background, and limit the types of videos that can be compared.

Ivan Laptev and Tony Lindeberg in 2003 and 2004, "Space-time interest points" and "Velocity adaptation of space-time interest points" [LL03] and [LL04] identify interesting points in a video-volume from the eigenvalues of the second-moment of a spatio-temporal gradient matrix. They classify [compare and cluster] interest points [events] by using a feature vector of normalized spatio-temporal Gaussian derivatives. Interest points do no appear for smooth, continuous motions; for interest points to be useful a video must contain regions where an object rapidly changes direction of motion. False interest points can arise from lighting conditions and shadows. They use set of interest points is used to model human walking. Model matching is then used to detect walking in new video sequences. Conversely, DTSW is a very general way to compare any two videos. We neither require static backgrounds nor limit ourselves to comparing videos with "interest points" or other specific volumetric features.

Periodic motions and behaviors are detected in [DBR00], [PN97], [All91]. Much work customizes the video comparison mechanism to surveillance and human gait applications. These works generally incorporate models of human gait and behavior [YB99], [Bre97], [NA94b], [LB95], [NA94a], [Bla99], [SC02], [EBMM03], [LL03], [LL04], [ZN04], [BD01], [YOI92], [BCND01], [ZMI01], [BGS+05], [SI05], [KSH05], [PN97], [CD00], [MG96]. Others target applications of gesture recognition other than gait, [SK02], [AAYS05], [WPG00], [PGW02], [WB99], [MG98], [KSH05], and use video as aid in speech recognition [YB99], [Bla99].

Many model-based approaches use Hidden Markov Models (HMM's) to probabilistically accommodate temporal variation between two videos [YOI92], [PGW02], [WB99] , [KSRC04], [KCC02], [KCC03], [KRCK02], [ZN04]. These techniques do not expose the detailed temporal matching or similarity that we seek. Recent work, [SK02], has extended HMM's to allow multiple candidates "per frame" but nevertheless operate on top of a model-based extraction of these targets.

Tables 1.1 and 1.2 summarize the strengths and weaknesses of the most relevant video comparison techniques. This table compares various methods to one another and to our technique.

1.4 Comparing or Aligning Videos

Via a simple example we explain our approach to the video comparison problem, and then explain the set of natural constraints that we use when locally comparing and matching regions of two videos.

Consider the two short sequences illustrated in Figure 1-6. The example, template, or query video is labelled **Q**. The unknown or test video is labelled **C**. From a visual comparison we make the following observations :

- Frame Q_1 is most similar to (or best matches) a spatial portion of frame C_1.
- Frame Q_2 and Q_3 are most similar to a spatial portion of frame C_2.
- Frame Q_4 is most similar to a spatial portion of both frames C_3 and C_4.

This matching of frames is shown in Figure 1-7. The temporally aligned - and spatially extracted - output sequences are also shown. We have found a non-linearly warped, common temporal axis. The spatial locations of the smaller video frames have been found inside of the larger. The two resulting aligned video sequences that are similar in appearance. This is the conceptual basis of DTSW; determine the best temporal and spatial alignment of frames, based on a local evaluation of similarity while warping time and shifting space subject to constraints on these distortions.

Author(s) :	Ke, Sukth., Herbert 2005	Shect., Irani 2005	Blank, etal 2005	Bobick, Davis 2001	Zelnik-Manor, Irani 2001	Black 1999	Sand, Teller 2004	Anthony 2006
Technique :	Boxes, Integral Video [KSH05]	Gradient Rank Change [SI05]	Space-Time, Poisson Shapes $[BGS^+05]$	MEI MHI [BD01]	Temporal Histograms [ZMI01]	Optical Flow Condensation Models [Bla99]	Robust Frame Registration [ST04]	DTSW
Attribute								
Pixels, Motion, or Other Vol.Data	All	Motion	NO (c1)	Motion	NO (e1)	All (f1)	NO (g1)	All
Amorphous, Flexible Subject Matter	Yes / nd	Yes	NO	Maybe	Yes / nd	Yes / nd	Maybe / nd	Yes
Accommodates Temporal Variation	Some (a3)	Small	Small	Small	Yes	Yes	Yes	Yes
Accommodates Spatial Variation	Yes	Yes	Yes	Yes	NO	Yes	NO	Yes
Temporal and Spatial Sensitivity Algorithmically Controllable	NO	Yes (b5)	Yes - local	Yes (d5)	Yes	Yes	Yes	Yes
Single Example Video	Maybe (a6)	Yes	Yes	Maybe (d6)	Yes	Yes	Yes	Yes
Non-static background	Non-static / nd	Yes	Static	Static	Mostly Static	Static	Mostly Static	Non-static
Body Centered or Free	Body	Free	Body	Body	Body + Static	Body	NA (g8)	Free
Requires Segmentation or Free	Free	Free	Requires	Requires	Requires + Static	Requires	NA (g8)	Free
Scannable in Time for infinite length detection	Yes (a10)	Yes	Yes	NA	Yes	NO	NO	Yes
Simultaneously "find" or operate on multiple targets	Yes	Yes	Maybe / nd	NO	NO	NO	NO	Yes
Classification	Yes	Yes / nd	Yes	Yes	Yes	Yes	NO	Yes
Detection	Yes	Yes	Yes - simple	NO	Time Only	NO	NO	Yes
Overall Similarity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Detail Time	NO	NO	NO	NO	NO	Yes (f15)	Yes	Yes
Detail Space	NO	NO	NO	NO	NO	Yes (f15)	NO	Yes
Report Local Similarity Along a Path.	NO	NO	NO	NO	NO	Yes	Yes (g17)	Yes
Processing Rate	Medium (a18)	Slow	Medium	Fast	Fast	Slow	Medium	Medium to Fast (h18)
Scale	Iterative (a19)	Iterative (b19)	Iterative	Iterative	Iterative	Iterative	NA	Iterative (h19)
Coarse to Fine approach?	Not Clear	Yes	Not Clear	Maybe / nd	Yes	Yes / nd	Yes / nd	Yes / nd

Table 1.1: Video Comparison Techniques. See notes in Table 1.2

(a3)	Some, Training doesn't accommodate temporal variation. Falters toward end of template sequences. Erroneously learns that the temporal ends are noisy.
(a6)	Maybe - best with multiple training samples.
(a10)	Yes - Uses integral video structure. Has problems with long or continuous sequences.
(a18)	At "Frame Rate" on 160x120 pixel size videos.
(a19)	Iteratively adjust scale. Initial experiments suggest can accommodate 25% variation.
(b5)	Yes - via local averaging gaussian filters
(b19)	Initial experiments suggest can naturally accommodate small variation.
(c1)	Appearance (pixel) based space-time shapes (silhouettes) carved out of a static background
(d5)	Yes - temporal averaging duration.
(d6)	Maybe - best with multiple training samples.
(e1)	Space-Time Gradients.
(f1)	Yes, only demonstrated with motion.
(f15)	Yes (smoothed).
(g1)	No - Requires combination of pixel intensity and motion.
(g8)	Requires two sequences of same frame size.
(g17)	Yes - temporal alignment only.
(h18)	Parallelizable.
(h19)	Iteratively adjust scale. Initial experiments suggest can accommodate 25% variation.
Yes / ND	Yes but not demonstrated.

Table 1.2: Video Comparison Techniques Notes.

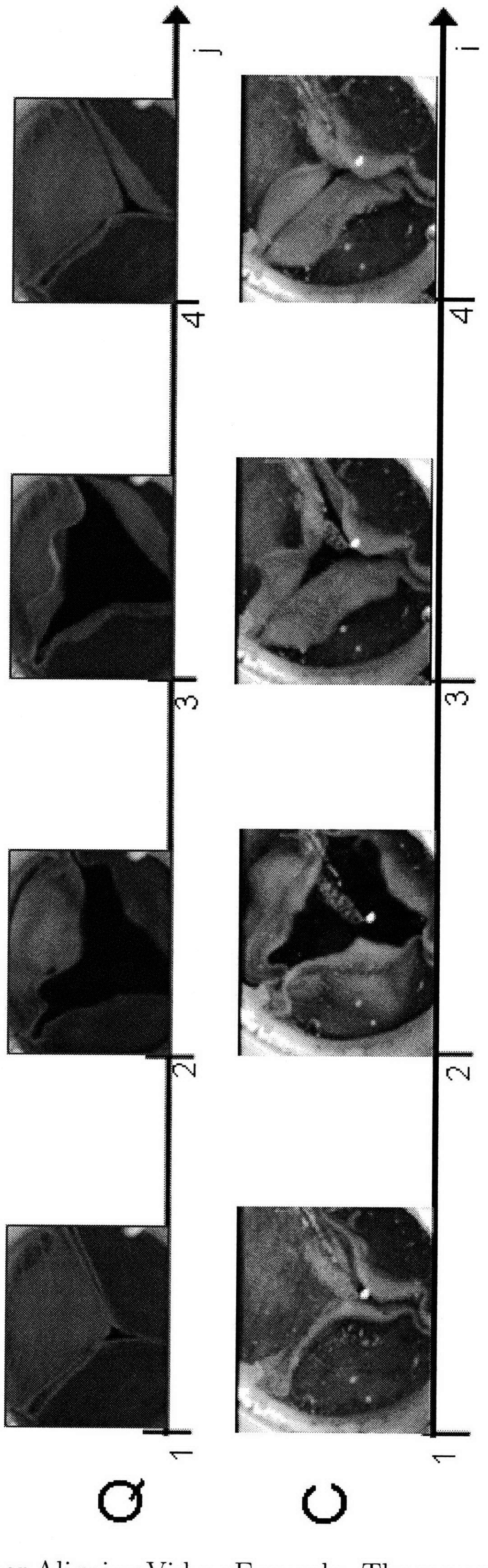


Figure 1-6: Comparing or Aligning Videos Example. The example, template, or query video is labelled **Q**. The unknown or test video is labelled **C**.

The overall similarity between these two videos is found by summing all the aligned frame-to-frame similarities.

This is the conceptual basis of our technique; determine the best temporal and spatial alignment of frames, based on a local evaluation of similarity and subject to local path constraints. The overall similarity between these two videos is found by summing all the aligned frame-to-frame similarities.

1.4.1 Constraints when Comparing or Aligning Videos

Spatial continuity. Figure 1-8. We expect that the frames of the query video are time sequential and that the position changes from frame to frame are bounded. For a particular time-isolated query frame, the best matched region in a test frame may be background clutter or far outlying objects instead of the region that an intelligent user would select (based on the context of information in adjacent frames). The spatial continuity constraint forces the pairing, of a frame from the query video with a sub-frame of the test video, to be within a small spatial radius for all frames over a period of time.

Spatial drift. Figure 1-9. Spatially distributed regions of information are matched in time and space - that it, we locally match a frame (a group of pixels at a single instant in time) of the query video to a sub-frame (a group of pixels at a single instance in time) of a test video. These spatial regions of pixels (or other data) do not accumulate or move for a fixed instance in time - that is, a frame corresponds to unique locations of objects at points in time. If it is necessary to replicate a frame of the test video in order to match to two or more frames of the template video, then the frame-to-frame matching must occur at the same spatial location (in the replicated frame of the test video). If frame exposures were long compared to the time scales of captured motion, then blurring or spatial drift would make physical sense, and this constraint would be relaxed.

Temporal continuity. Figure 1-10. Thought we do expect temporal variation between two videos we limit the local extent of that variation.

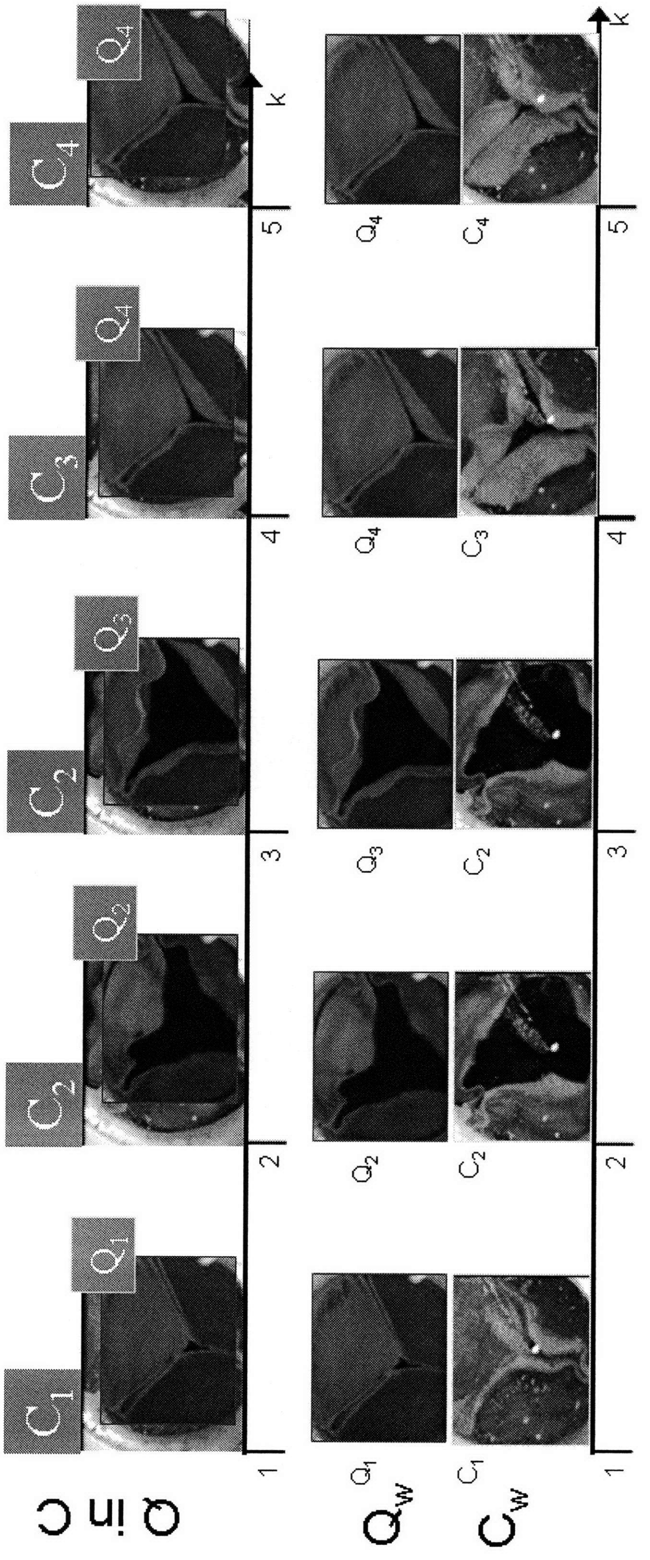


Figure 1-7: Comparing or Aligning Videos Example - Result. This is the conceptual basis of DTSW. Determine the best temporal and spatial alignment of frames, based on a local evaluation of similarity while warping time and shifting space subject to constraints on these distortions.

Bi-Temporal causality. Figure 1-11. We consider events that are progressing in time. Local time scales may vary; we don't expect moments in either video to be out of time sequential order.

End points. Figure 1-12. The first frame of the query is matched to the first frame of the test video. Likewise for the last frames. This constraint is easily relaxed in order to accommodate imprecise temporal clipping and to facilitate scanning (for detection).

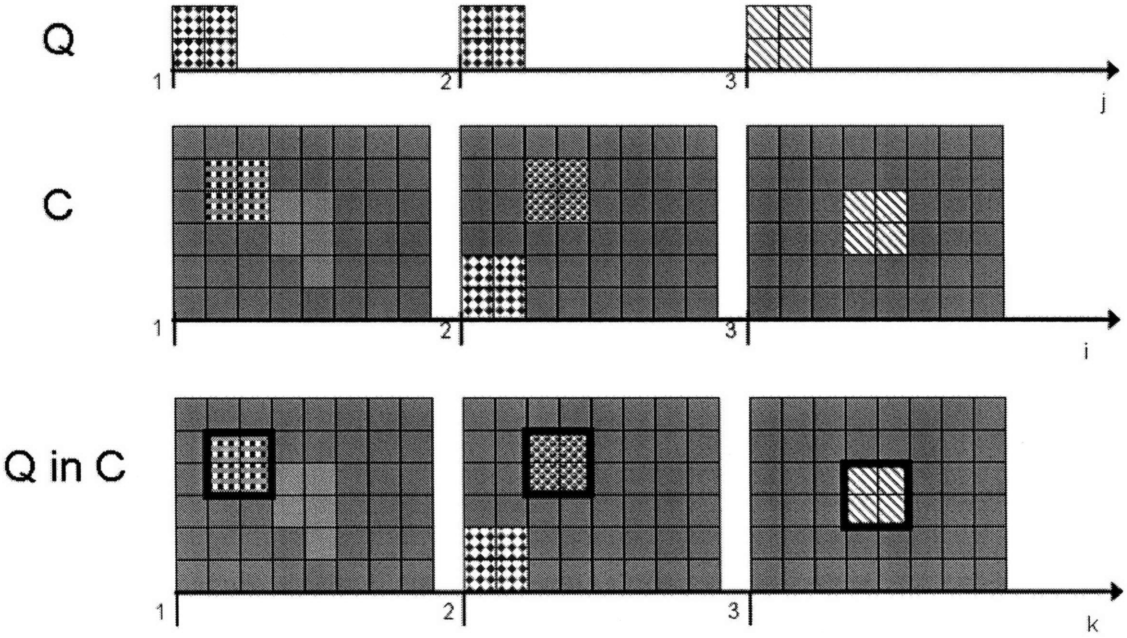


Figure 1-8: Constraints when Comparing or Aligning Videos - Spatial Continuity. The spatial continuity constraint forces the pairing, of a frame from the query video with a sub-frame of the test video, to be within a small spatial radius for all frames over a period of time.

Any of these constraints are easily relaxed if the physical motivations for each are not present in the videos that are to be compared.

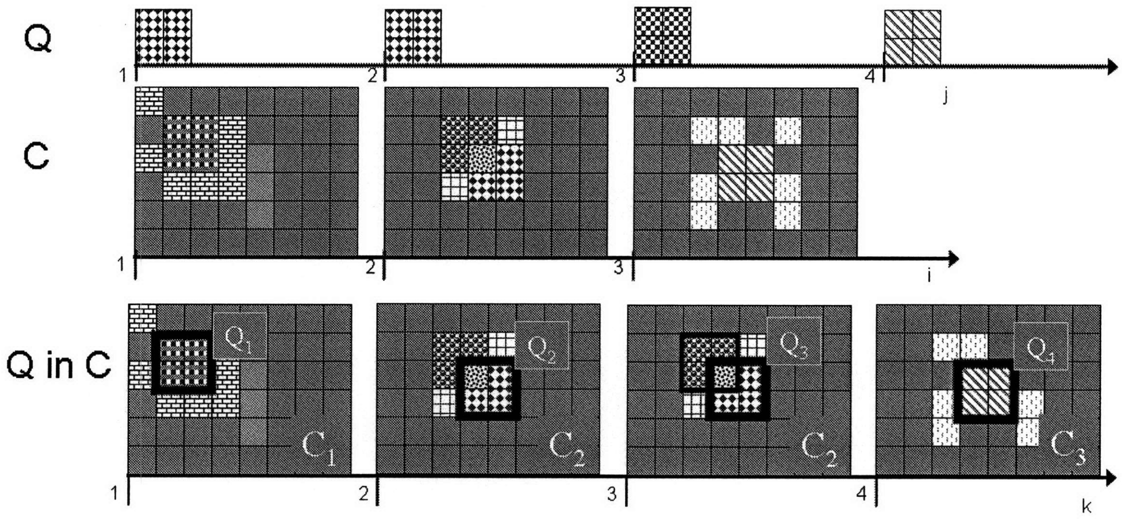


Figure 1-9: Constraints when Comparing or Aligning Videos - Spatial Drift. If it is necessary to replicate a frame of the test video in order to match to two or more frames of the template video, then the frame-to-frame matching must occur at the same spatial location (in the replicated frame of the test video).

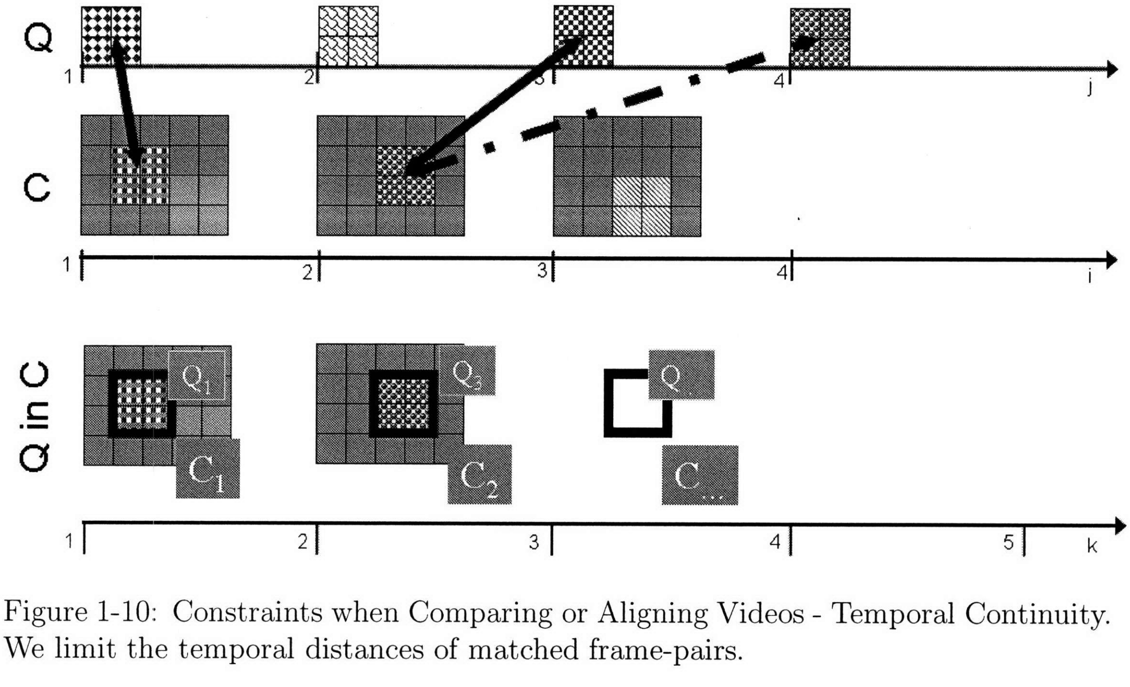


Figure 1-10: Constraints when Comparing or Aligning Videos - Temporal Continuity. We limit the temporal distances of matched frame-pairs.

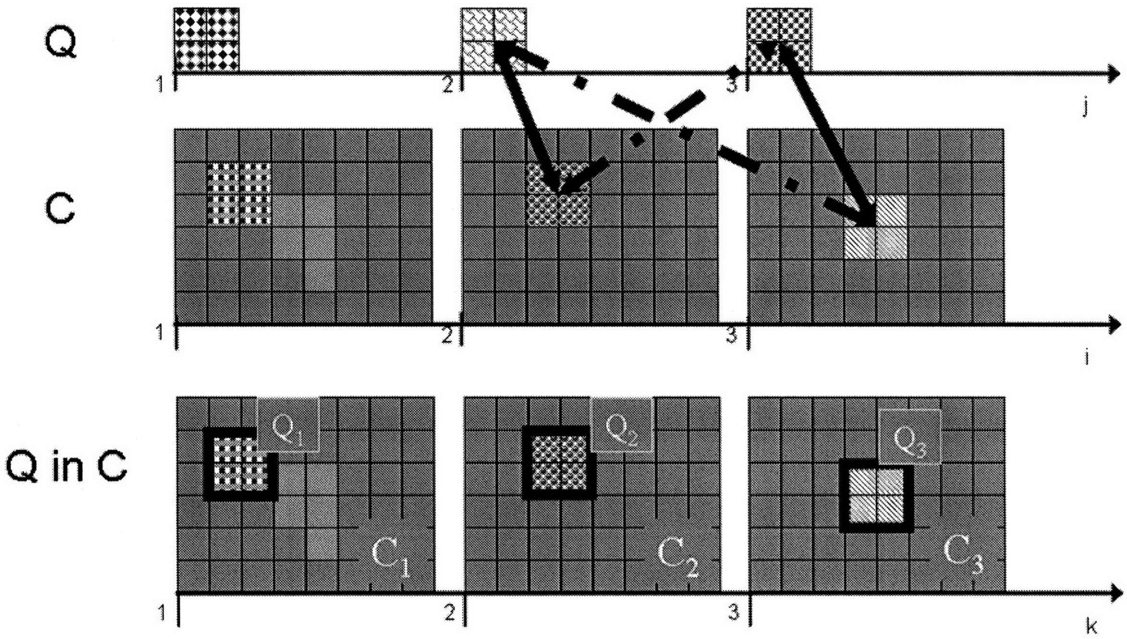


Figure 1-11: Constraints when Comparing or Aligning Videos - Bitemporal Causality. We consider events that are progressing in time. We don't expect the sequence of matched frames in either video to be out of time sequential order.

1.4.2 Conceptual Approach to Comparing or Aligning Videos

We approach the problem of comparing two videos by asking how the videos are best aligned, subject to a set of warping and shifting constraints. This problem can be cast as a minimum-cost or shortest-path problem.

One video, the template or known **Q**, is J frames long. The second video, the unknown **C**, is I frames long. The frame size of **Q** is generally smaller that the frame size of **C**. A frame of video **Q** can be positioned, relative to a frame of **C**, in X different spatial locations horizontally and Y different spatial locations vertically and still remain inside of a frame of **C**. We calculate a scalar similarity, or distance, between a frame of **Q** and any same-sized region from any frame of **C**. A complete enumeration of this local similarity defines a four-dimensional hypervolume, $[I \times J \times X \times Y]$. The problem of comparing or aligning is now one of finding a minimum cost path through this hypervolume such that the warping and shifting constraints are met.

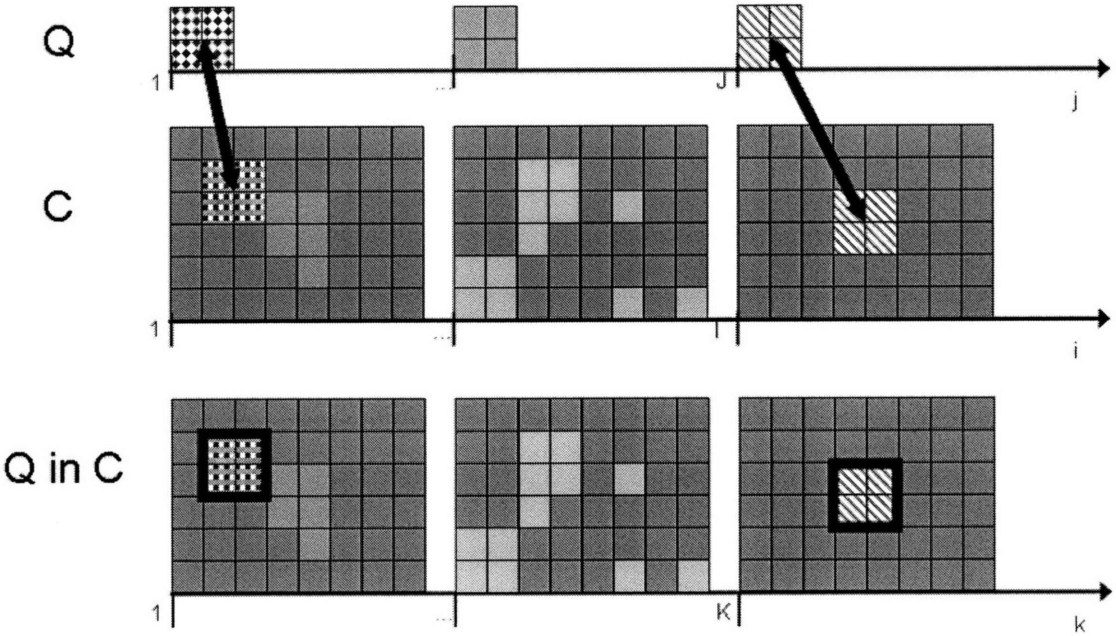


Figure 1-12: Constraints when Comparing or Aligning Videos - End Frames. The first frame of the query is matched to the first frame of the test video. Likewise for the last frames. This constraint is easily relaxed in order to accommodate imprecise temporal clipping and to facilitate scanning (for detection).

1.5 Contributions

- Detailed exploration of how to use video in automated industrial monitoring and control. This system taxonomy combines subject framing, subject properties, and desired process control parameters. A decision flowchart is developed to aid the non-video engineer when implementing a video monitoring solution.

- A new technique, Dynamic Time and Space Warping (DTSW) is developed.

  DTSW is developed to address both the subject matter and monitoring needs that can not be addressed via existing algorithms. It deterministically compares a single template video to a new test video. It does this by finding the minimum cost path of the template through the test video.

  DTSW determines a detailed local similarity in time and space between a single template video and an unknown video. The local similarity is accumulated (path dependent integration) over time and space to determine an overall similarity. The subject matter may be rigid-and-articulated or amorphous-and-flexible. The subject matter may lack distinct features in space and in time. It is a view-based algorithm; it uses a temporal-spatial representation of information, such as pixel intensity, optical flow, moments of optical flow, or intensity gradients.

- Compact representation of the Temporal Template. PCA is not new, but it enables a compact representation and is uniquely applied in the DTSW algorithm.

- An efficient filter bank based implementation of DTSW is developed.

- Techniques to aid the non-video engineer in the selection of a quality temporal Template Video are explored.

- A unique technique for video event detection is developed as an extension to DTSW.

  As mentioned previously, detection (specifically temporal detection) is not a significant requirement in most industrial applications. Outside sensory inputs such as logic switches or proximity sensors will provide the temporal segmentation. However, an intermediate stage of DTSW provides a data structure that is easily analyzed for the video event detections.

1.6 Summary

The determination of two videos' similarity (or dissimilarity) is a central requirement for algorithms developed for video classification, video detection, and video alignment. Many applications for video-based system monitoring in industrial, manufacturing, and scientific environments are neither strictly classification nor detection applications. Industrial applications, such as lifetime testing, or failure detection, require a detailed determination of similarity between two videos (a template video and test video) as a function of space and time. We address the concept of spatial and temporal similarity by asking how one video must be modified in space and time to make it look like another. This question led us to develop an algorithm called Dynamic Time and Space Warping. DTSW determines the optimal time-warping space-shifting alignment between two videos.

DTSW is an appearance-based algorithm that can use any volumetric data values, such as pixel intensity, optical flow, or moments of either. DTSW is unique in that it simultaneously accommodates both temporal and spatial variability, allows us to locally analyze two videos' similarity or alignment, only requires a single exemplar[3] video, accommodates flexible or amorphous subject matter lacking distinct features, can be easily scanned (for detection applications), and is computational scalable due to it parallel structure. As far as we have been able to determine no existing technique simultaneously provides all of these benefits especially while being as computationally efficient as DTSW.

[3] One that serves as a model or example, an ideal model, a typical or standard specimen. - Merriam-Webster Online Dictionary.

1.7 Document Outline

The remainder of this document is organized as follows. In Chapter 2 we discuss system monitoring in more detail. In chapter 3 we develop a series of definitions. In Chapter 4 we develop the Dynamic Time and Space Warping (DTSW) algorithm. In Chapter 5 we evaluate the performance of the DTSW algorithm on several nonindustrial application. In Chapter 6 we evaluate the performance of the DTSW algorithm on for system monitoring applications. In Chapter 7 we address video event detection and classification applications. In Chapter 8 we explore the computational structure of the DTSW algorithm, determine how to quantify the quality of a video template, and evaluate the performance of DTSW through simulation. In Chapter 9 we summarize the contribution of this work, and discuss directions for future work.

Chapter 2

Video Based System Monitoring

2.1 Introduction

There are two common uses of cameras in industry, automated image classification, and off-line video “motion capture”. The combination of these areas motivates the high-level goal of this work - to expand the notion and framework of industrial and scientific machine vision to include the use of video for automated monitoring.

We discuss the system level - lighting, cameras, subject preparation - particularities that are common in industrial and scientific environments. We discuss how these system level particularities make industrial video sequences different from the sequences found in human surveillance or vehicle mounted camera applications.

Industrial and scientific applications for automated video classification and event detection have been briefly mentioned in discussions in previous work, [PN97], but have not received explicit attention. The developed techniques and algorithms do not generally address the unique problems and requirements that must be addressed in order to use video for automated monitoring in manufacturing, industrial, and scientific applications.

We discuss the type of information that is important for automatic monitoring applications. Many types of monitoring, or process control, decisions can be made if we first determine the combined temporal and spatial location of a template video inside a newly acquired test video.

Work in the areas of Video Classification and Video Event Detection is relevant to this work. The Video Classification problem is to determine to which set or sets of training samples a newly acquired (relatively short) Video Segment belongs. The Video Event Detection problem is to determine when and where a video event occurs inside of another video stream. The particular needs of video-based automated monitoring applications have not been addressed in the literature or in the market place.

2.2 Image Comparison

The most common use of video in industry involves the *automatic* comparison of *single images*, not sequences. Machine vision as it is thought of in industry, involves the comparison of a Golden Master[1]image to a test image of a newly produced part. A *computer* algorithmically compares a stored ideal image - a template - to a newly acquired image - the test image - of a just produced part. The degree of similarity between the two images is determined and used to make an automatic accept-or-reject determination of the newly produced parts quality.

Cognex has become a 300 million-dollar company focusing quite intently on developing hardware and software for this type of problem for single static images. Machine Vision as it is most commonly used in industry, refers to the determination of the "quality" of a part or product determined by calculating a difference measure between a single image of a part under test with a single image of the ideal part.

The difference measure between the test image and the golden-master image may be based on orientation of a group of pixels, similarity in color, similarity in shape, or similarity in size, among other criteria. The difference measure is then passed through a threshold for a pass or fail determination of acceptability. This acceptability measure is used to direct the rejection or acceptance of a part, to generate an operator alarm, or otherwise provide some automatic feedback to the process under

[1] The Golden Master term is used in industrial **image** template matching. The Golden Master is the standard image that is the point of comparison for an image of a newly produced part.

observation. This idea is illustrated in Figure 2-1.

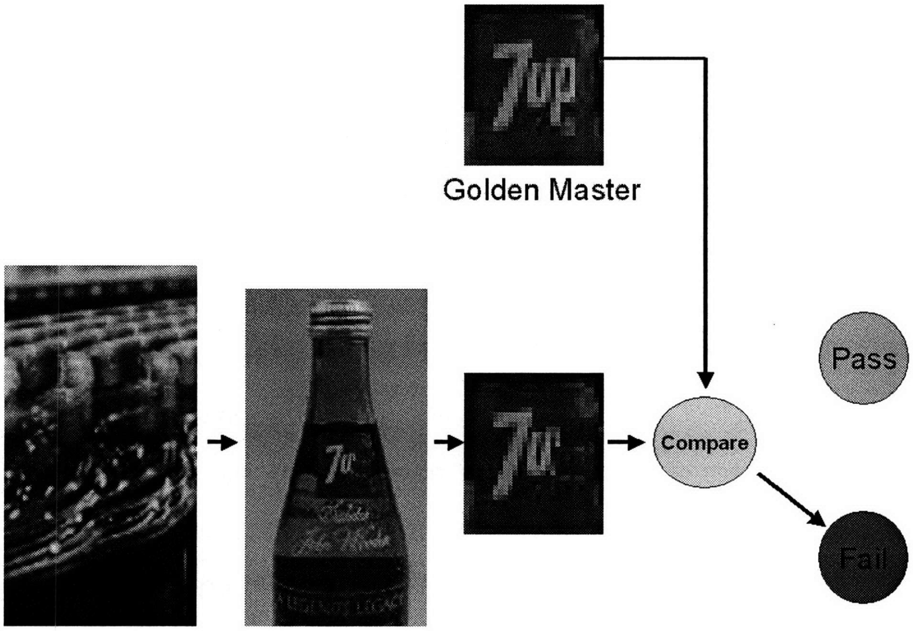


Figure 2-1: Example of Industrial Image Classification for Assemply Line Decisions. Machine vision as it is thought of in industry, involves the comparison of a Golden Master (template) image to a test image of a newly produced part. A computer algorithmically compares a stored ideal image - a template - to a newly acquired image - the test image - of a just produced part. The degree of similarity between the two images is determined and used to make an automatic accept-or-reject determination of the newly produced parts quality.

2.3 Motion Capture Video

The second most common use of video in industry is colloquially called "motion capture" video. In contrast, this use of video involves the *non-automatic* analysis of an *image sequence* by a *person.* A camera is used to record a video sequence of some motion event. A person then iteratively and interactively uses software tools to analyze the video. They determine motion as a function of time that will allow them to understand, or fix, the process they recorded.

In this way, a camera is used to visualize and identify a motion glitch, similar to how an oscilloscope is used to identify an electrical glitch[2]. This is an iterative, human-in-the-loop process used for trouble shooting a faulty process or understanding a product during its design phase. A process technician, engineer, or scientist is interested in capturing and later analyzing the motion of a process or piece of machinery. Lighting and camera setup is specified in order to make analysis easy. After capture the video is analyzed to extract feature positions, velocities, or acceleration as a function of time. The fast movements of the subject matter and the desire for highly accurate and dense measurements dictate video capture rates from hundreds of frames per second (fps) to tens-of-thousand frames per second. The fast frame rates eliminate the aliasing that is endemic to many computer vision applications, but produces gigabytes of raw video data in several seconds.

Examples of applications where video is used as a debugging tool to analyze faults include stamping, assembly, and packaging applications. Product and lifetime testing on a single part or product is another application area. These applications are often highly repetitive and cyclical. Figure 2-2 depicts a Motion Capture example; a video of a moving cam shaft is captured and analyzed to show its motion.

2.4 Video Comparison

The high-level motivation of this thesis is to expand the notion and framework of industrial machine vision to include the use of video (versus single images) for automated monitoring. This entails defining the application framework under which video - most commonly high-speed video - can be easily and practically applied to automated industrial and manufacturing process-monitoring applications; creating the computational tools to aid a user in selecting a quality template video; and creating the tools to automate video detection, classification, and detailed analysis for automated monitoring tasks. The application of this technology should be "easy" to

[2] A fault, flaw, or defect in a system or machine. Word History: It is first recorded in English in 1962 in the writing of John Glenn: "Another term we adopted to describe some of our problems was 'glitch.' " Glenn then gives the technical sense of the word the astronauts had adopted: "Literally, a glitch is a spike or change in voltage in an electrical current." It is easy to see why the astronauts, who were engaged in a highly technical endeavor, might have generalized a term from electronics to cover other technical problems. Since then glitch has passed beyond technical use and now covers a wide variety of malfunctions and mishaps. Source, http://www.thefreedictionary.com/glitch.

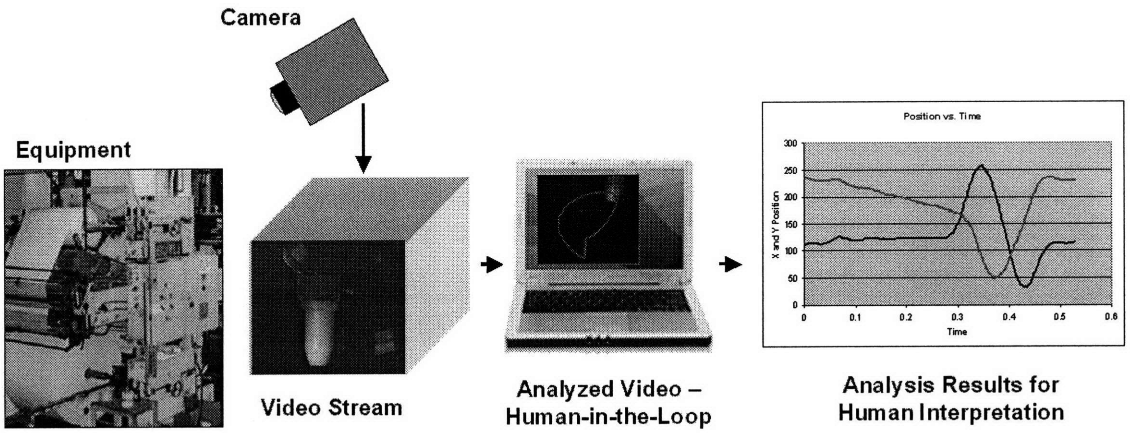


Figure 2-2: Industrial Video - Motion Capture. A camera is used to record a video sequence of some motion event. A person then iteratively and interactively uses software tools to analyze the video. They determine motion as a function of time that will allow them to understand, or fix, the process they recorded.

apply by a non-video engineer by providing the system guidelines and analysis tools for such tasks.

2.5 The Industrial or Scientific Environment

The industrial or scientific video stream is very different from the video acquired from mobile robotics, broadcast television, surveillance cameras, personal camcorders, or other common sources of video.

Some aspects of industrial and scientific video analysis are simplified by constraints on the subject matter and the control over the environment. Industrial and scientific video streams when compared to surveillance or human motion video streams tend to be highly engineered and structured. The subject matter or "scene" of industrial videos is much more predictable and repeatable. In industrial applications, the subject matter is carefully lit and framed to make analysis simple. Camera(s) are stationary and at known locations, lighting is fixed, and there is a possibility that the process or machinery that is to be monitored can be painted or otherwise prepared to create a good result.

We mentioned the need for extremely fast record rates in “motion capture” applications. The same need for speed exists for video-based automated monitoring applications, which will again result in large amounts of raw data acquired in a very short period of time.

2.6 Video Based System Monitoring

Consider the following physical systems :

- A section of a diaper-packaging machine, where diapers travel along a conveyor, are wrapped and fed into a package.
- A synthetic heart-valve during a simulated lifetime test.
- A race track around which horses with jockeys are running.
- A martial arts studio with various teachers and students practicing.

When attempting to explain or understand most any physical system that has moving parts we are interested in position and motion parameters (velocity, acceleration, jerk) as a function of time and space for the numerous rigid and flexible pieces that compose the system. We are interested in relative motion between parts as a function of time and space. We are interested in the relative timing and location of “events”. We are interested in detecting the occurrence of “events”. For physical systems that are immobile we are also interested in visible changes that are optical in nature, such as color.

We monitor the behavior or operation of one system to another by comparing Position, Motion, Appearance, and Relative Timing, (PMART) all versus time and space. We monitor a system by continuously monitoring PMART parameters[3]. We compare to the desired performance of a known good system and to the historical record. Deviation, or change beyond a threshold, is used to adjust control settings or to alert an operator that the system has deteriorated and should be repaired or taken off-line.

We implicitly monitor motion, appearance, and timing of a complex system, via explicit comparison of a series of video events. Process control decisions or performance degradation decisions are made based on the combined spatial and temporal similarity between a video event template and a video event found within a video sequence of the system under observation or test. This idea is shown in Figure 2-3. The measurement of motion is not an explicit need. The goal is to determine timing, positions, and appearance, deviations from the expected.

The following is a sample of the type of information that is of interest in the context of automated industrial-video event detection and comparison:

- Overall similarity.
- Temporal Shift between two different events.
- Repetition rate of event.
- Position of event along the event path.
- Relative time of event along the event path.

[3] Implicit Measurements. In manufacturing, scientific, and other controlled environments, time, relative timing, position and other motion parameters are frequently measured with encoders, accelerometers, proximity or contact sensors, and optical beam-break sensors. Appearance "measurements" require some optical sensor such as a video camera.

In harsh environments or due to other concerns, it is often not possible to use sensors that require close proximity to the subject matter. In such situations, it may be possible to record motions with a video camera and then determine the *perceived* timing and position information from video streams. Then, sometimes, it is possible to relate the perceived PMART measurements to the underlying physics of motion or other explanations of change. This the premise of industrial Motion Capture Video described previously.

Relating the video data to the real world physics of motion or geometry is application and problem specific. One generally would need to account for camera positions, optical distortions, lighting, etc. Even if a mapping to the real world is not possible, it is still frequently possible to identify, and/or segment, image regions of similar appearance or motion, and to determined their spatial and temporal path in video coordinates.

In many monitoring applications it is not necessary to explicitly determine an event's PMART parameters and to subsequently compare those parameters. An implicit comparison is often sufficient. It is often sufficient to compare data streams - such as video - directly and to set threshold on the change in the data stream instead of thresholds on change in the PMART parameters. System specifics will dictate if these measures of similarity can be related to the deviations of the motions or relative timing of the subject matter. It will often not be necessary.

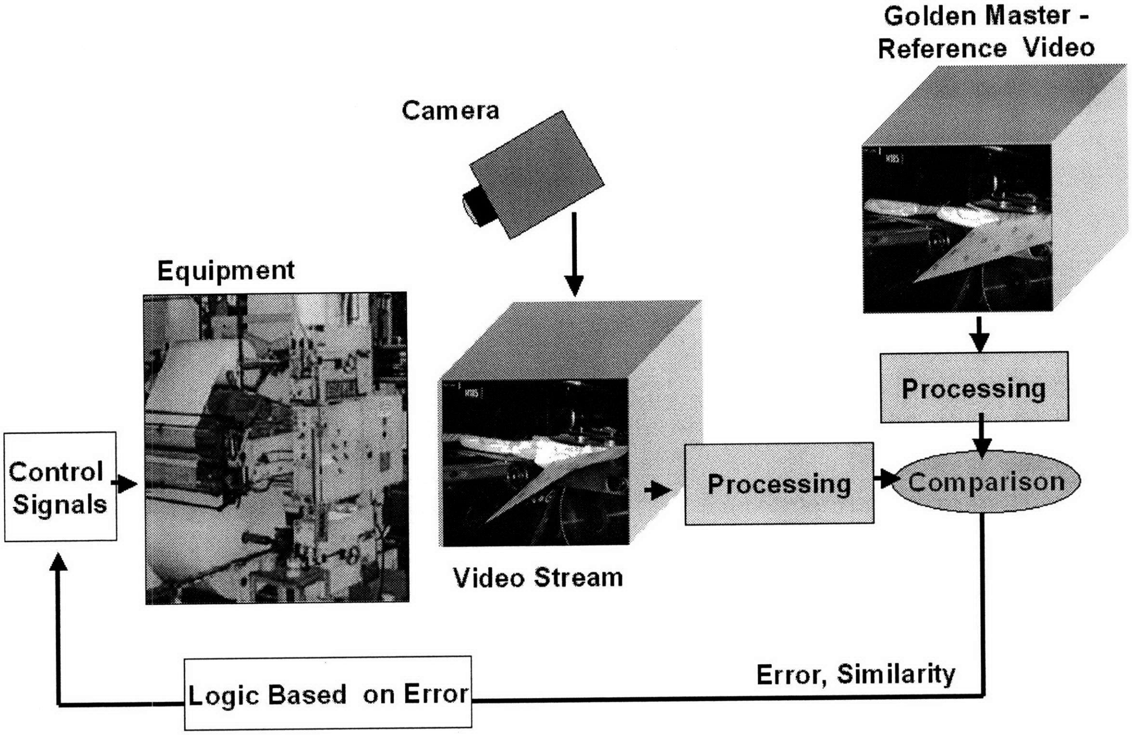


Figure 2-3: Video Comparison for Assembly Line Decisions. We monitor the behavior or operation of one system to another by comparing Position, Motion, Appearance, and Relative Timing, (PMART) all versus time and space. We monitor a system by continuously monitoring PMART parameters. We compare to the desired performance of a known good system and to the historical record. Deviation, or change beyond a threshold, is used to adjust control settings or to alert an operator that the system has deteriorated and should be repaired or taken off-line.

Our task may include components of standard detection and classification problems; detect the presence of a prescribed example in a video stream. Generally we will need to provide a detailed comparison between the detected/segmented sub-video sequence and the Golden Master video template. Figure 2-4 illustrates the detailed comparison concept. We identify system failure, abnormal motion, degradation, or other problems by determining if the local or overall similarity between two videos (in time and space) has fallen below a threshold. Algorithmic temporal segmentation will often be unnecessary, as control lines or other sensory events will provide the temporal segmentation of a video stream into well-defined event specific segments or clips.

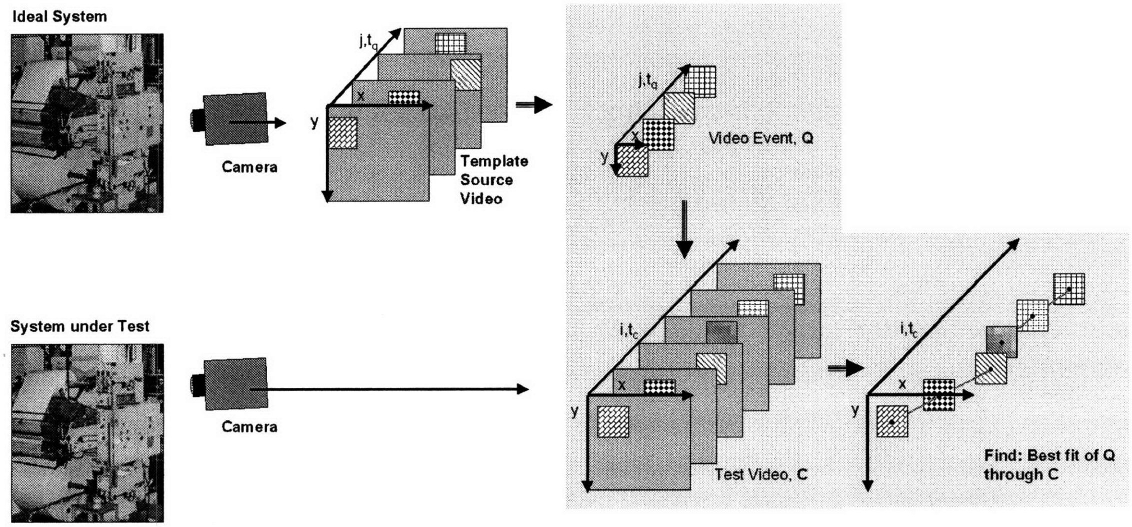


Figure 2-4: Video Event Detail in a Test Video. We will provide a detailed comparison between the detected/segmented sub-video sequence and the Golden Master video template. We identify system failure, abnormal motion, degradation, or other problems by determining if the local or overall similarity between two videos (in time and space) has fallen below a threshold.

2.6.1 Systems - A Collection of Events. Video Events.

We consider the complex spatial and temporal moving parts of a physical system to be a series of spatio-temporal events. An event is defined as a finite sized temporally and spatially coherent object moving, or visually changing, through a finite space over a finite time. An event can be the motion of a rigid object or the motion of a flexible amorphous object. This definition of an event is both application specific and open to interpretation.

For the previously mentioned physical systems events can de defined as :

- A single diaper transported down an assembly line, wrapped and fed into a packaging machine is an event.
- A single beat of synthetic heart-valve during a simulated lifetime test is another example. Figure 2-5
- A person moving across a parking lot is another event.
- A person throwing a karate punch is an event. Figure 2-6
- In another systems an event could be one cycle of a cam shaft, or the movement of a piston.

System monitoring now becomes the summation of event monitoring sub problems. Each event is periodic, semi-periodic, or randomly occurring. Events occur at different times and locations in the space and time of a system.

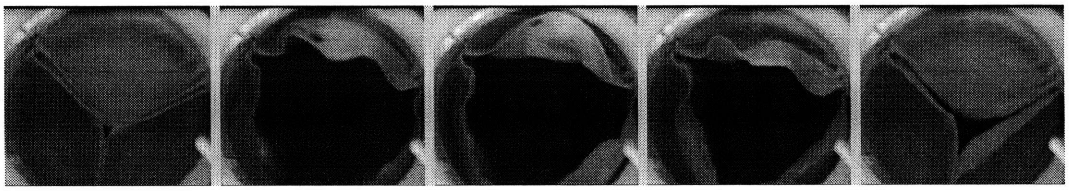


Figure 2-5: Video Event - Heart Valce Cycle

An event is defined as a finite sized physical object moving, or visually changing, through a finite space over a finite time. A video event is a video recording of this event. The video captures the appearance of the event as a function of space and time. The spatial extent and temporal duration of the video defines the bounds of the video event.

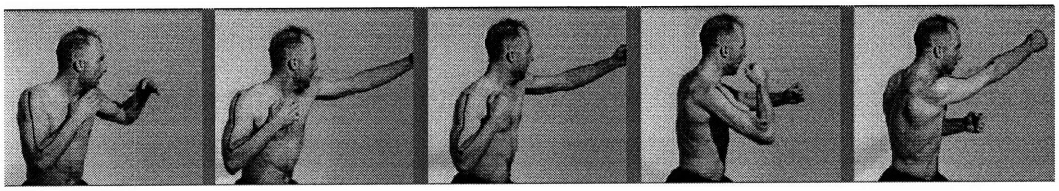


Figure 2-6: Video Event - Karate Punch

2.7 Relationship to Existing Work

Work in the areas of Video Classification and Video Event Detection is relevant to this work. The Video Classification problem is to determine to which set or sets of training samples a newly acquired (relatively short) Video Segment belongs. The Video Event Detection problem is to determine when and where a video event occurs inside of another video stream.

There are significant differences between the Video Classification and Video Event Problem as addressed in the literature and the requirements for system monitoring application for control and performance monitoring.

1. A system monitoring application requires determination of a near exact match to a single video event template, the Golden Master.

2. A system monitoring application requires a deterministic and detailed analysis of the template as found in the space and time of the test sequence.

This detailed deterministic comparison is motivated by the fact that we expect temporal and spatial repeatability and similarity in the video events.

Most work in the area of Video Classification purposely eliminates a connection to deterministic time in order to account for the (temporally) probabilistic sequencing of events. This is particularly important in applications that focus on human motion or gesture recognition where the template and test events are between very different actors. Much work represents time as a set of states; transitions from one state to the next are probabilistic events, [YOI92], [PGW02], [WB99] , [KSRC04], [KCC02], [KCC03], [KRCK02], [ZN04], [SK02]. Other work completely eliminates time and maps a video into a single representative image, [BD01], [NA94a], [BCND01] [CD00].

This destruction of time is necessary for applications that are concerned with human motion or human gesture recognition. If time were preserved as an important component then many contextually similar events would never be classified as related. In a surveillance application for example, we seek to detect and classify motions or appearances that are similar to some probabilistically defined template but are by necessity due to the variability of human motion, lightning and appearance - non-exact in the matching metric. These applications must also accommodate a level of spatial variability for the same reason.

2.8 Video Classification and Video Event Detection

Video classification, also referred to as recognition, is the classic machine-learning classification problem applied to video sequences. The main goal is to determine to which set or sets of training samples a newly acquired sample belongs. This idea is depicted in Figure 2-7. A newly acquired video of a girl kicking a soccer ball is classified as being most similar to video of a man kicking a rugby ball. For point of comparison to detection, classification applications generally start with a pre-segmented video sequence in time and space. That is, the videos are tightly "framed" spatially around the subject matter, and the video sequences begin and end with the recorded event.

Video event detection, Figure 2-8, is a more difficult problem. Instead of taking nicely segmented finite sequences as input, the problem now is to identify (segment) the combined spatial and temporal location, where a pattern of appearance or motion over time exists, that is similar to some prescription or is in some way unique. Video event detection is becoming an increasingly important application in machine vision largely driven by surveillance applications.

Historically, the majority of the work in detection has been concerned only with temporal segmentation. Temporal change detection has been used to detect pans,

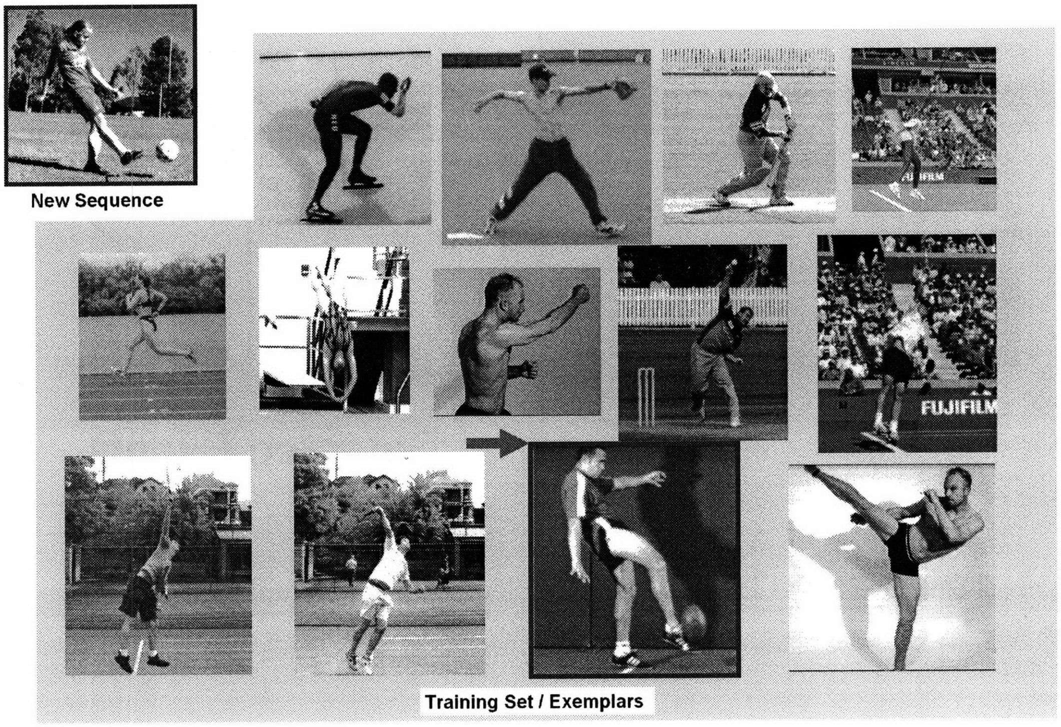


Figure 2-7: Video Classification Example. The main goal is to determine to which set or sets of training samples a newly acquired sample belongs.

fades, and scene changes that are a natural part of video streams associated with television, movies, sports video streams [MZ02], [NPC99], [NPZ03], [NPZC00]. Recent work has started to address the combined spatial and temporal detection problem [BGS$^+$05], [SI05], [KSH05]. Figure 2-8 depicts the idea behind video event detection.

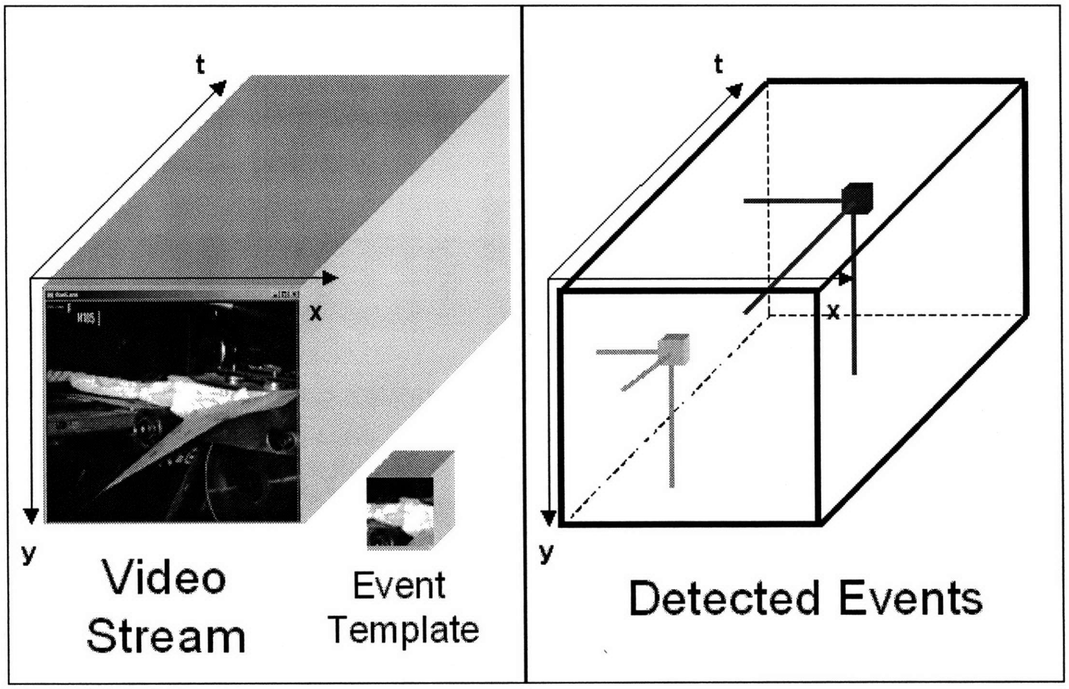


Figure 2-8: Video Event Detection - Found locations of a Template Video in the Test Video

The application areas for both classification and detection are very similar. The following applications have received the majority of the attention in literature.

- Surveillance and human gait. [YB99], [Bre97], [NA94b], [LB95], [NA94a], [Bla99], [SC02], [EBMM03], [LL03], [LL04], [ZN04], [BD01], [YOI92], [BCND01], [ZMI01], [BGS$^+$05], [SI05], [KSH05], [PN97], [CD00], [MG96].

- Human Machine Interfaces. [PGW02], [WB99], [MG98].

- Gesture recognition other than gait. [SK02], [AAYS05], [WPG00], [PGW02], [WB99], [MG98], [KSH05].

- As an aid in speech recognition. [YB99], [Bla99].

- Face recognition. [LGL01], [FCJ00], [FJK01], [KHS00].

- Database Query. [YFM03].

- General periodic motion. [DBR00], [PN97], [All91].

2.8.1 Appearance-Based or Model-Based

Similar techniques are used for both classification and detection applications. The general approaches to these problems, and many machine vision problems, are broadly categorized as model-based or appearance-based. Model-based approaches reduce the dimensionality of the raw pixel data by extracting "high-level" vectors of parameters. Appearance-based approaches use the "low-level" spatial-temporal distribution of pixels, dense optical flow, or other pixel centric values. Model based approaches are appropriate when a good model of the desired events exists a priori. Appearance-based approaches are a bit more flexible and can be applied to different activities, but require a bit more computation when determining similarity.

The prescribed template in a detection application or the example used in classification applications may be a composite determined from a set of training samples - this is the typical statistically robust technique that is used in most of the literature; it may be a single exemplar video sequence [SI05], [KSH05], [BGS$^+$05]; or it may be the specification of detecting or classifying an event that is different from the usual video, [PH04].

2.8.2 Appearance-Based Techniques

We focus on appearance-based approaches. Appearance-based approaches - those using a "low-level" spatial-temporal volume of pixels, dense optical flow, or other pixel centric quantities - are preferred for their lack of specificity. Appearance-based approaches tend not to be customized for one application. They are non-parametric and can thus handle the wide range of dynamic events that are conceivable in industrial and scientific applications. The approaches that we develop should seamlessly handle whichever dense volumetric data structures - pixels, optical flow, or various moments and combinations of each - that is appropriate for a given application.

2.8.3 Algorithms

Techniques that have been applied to non-industrial applications can, of course, be used or adapted to some industrial and scientific applications. Figure 2-9 is a high level guide that can serve as an aide in selecting the algorithmic primitive for a particular monitoring task.

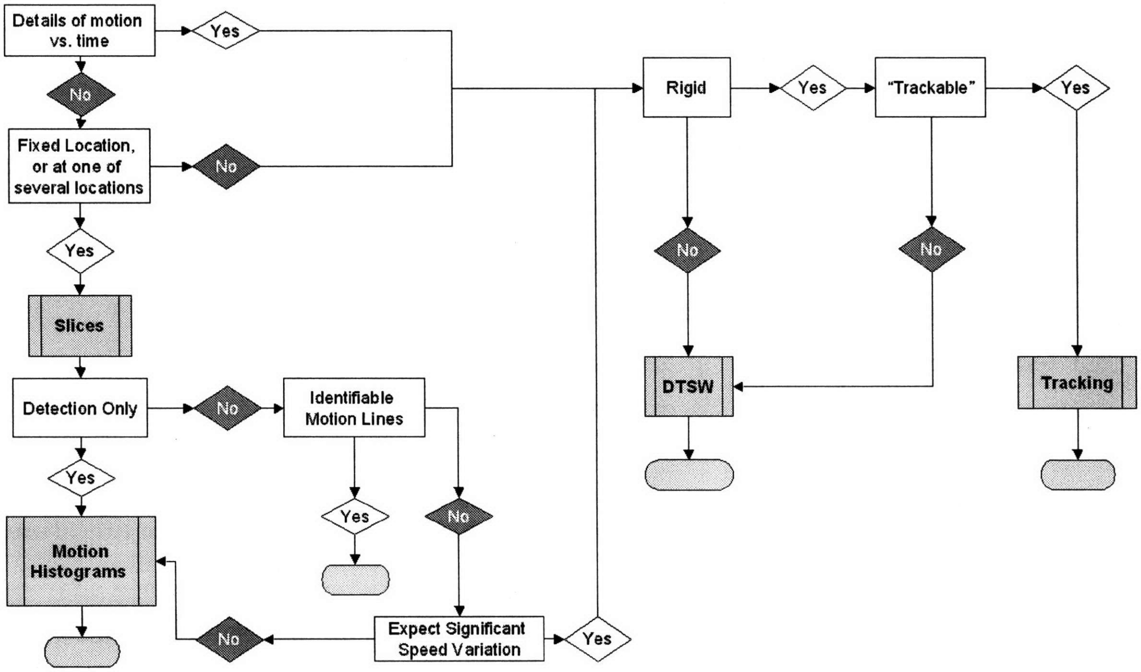

Details of motion vs. time
Yes
No
Fixed Location, or at one of several locations
No
Yes
Slices
Detection Only
No
Identifiable Motion Lines
Yes
No
Yes
Motion Histograms
No
Expect Significant Speed Variation
Yes
Rigid
Yes
"Trackable"
Yes
No
No
DTSW
Tracking

Figure 2-9: Selection of [video comparison] algorithmic primitives for industrial and scientific monitoring based on subject matter and analysis intent.

The flowchart of Figure 2-9 is a rough guide for selecting an algorithmic primitive for a monitoring task. It is not intended to be universally complete; it does not guide you to the solution for every monitoring problem. It does guide you through a series of questions that are generally appropriate. Your application's details may lead to a down an alternative path. Often, multiple techniques may be appropriate. And often, multiple techniques will be used simultaneously.

The questions that we ask when selecting an algorithm primitive for a monitoring task are:

- Do we need to know the details of motion versus time for the entire body of the subject matter?

- Is the subject matter consistently placed?

- Do we simply want to know that an event occurred? That is, are we simply interested in the temporal detection of an event and not interested in either the spatial location of the event or in a detailed temporal profile of the event.

- If the subject matter is consistently place, are there identifiable motion lines on a temporal slice out of the XYT volume?

- If the subject matter is consistently placed, do we expect significant variation in the temporal profile of the event that we want to detect or monitor?

- If we want to know details of motion versus time and space, is the subject matter rigid or amorphous and flexible?

- If we want to know details of motion versus time and space, does the subject matter have trackable features?

Temporal slices are appropriate if the subject matter is repeatedly placed and we are interested only in the motion at a few distinct locations. In this case, we monitor a process by examining the lines and curves that are created on a slice through the XYT video volume.

Motion Histograms are appropriate if we are interested in the approximate temporal occurrence of some event, and do not require detailed analysis of either the temporal or spatial profile. For every frame in the video sequence, a histogram of the motion in the frame is determined, often at multiple spatial scales. We detect the temporal occurrence of an event by comparing a desired histogram-versus-time profile to that obtained from the monitored scene. Further, such technique generally requires a static or a known background.

Feature tracking is appropriate if the subject matter is generally rigid and has features that are trackable.

DTSW, is appropriate if the subject matter is amorphous, flexible, and lacks distinct trackable features.

As Figure 2-9 suggests, there are limitations in the applicability of existing techniques to automated industrial and scientific applications. Applications that require detailed temporal and spatial deviation from a prescribed template are not appropriately addressed in the literature. The measurement of detailed deviation is further complicated if the subject materials lack distinct features in their appearance or in their motion and are therefore untrackable. (Condensation/factored sampling algorithms could be applicable in some scenarios, but they are computationally expensive.)

The need to monitor subject materials that are deformable, amorphous, or flexible [such as paper, metals, sprays, metal flows, biological materials] arises very frequently in industrial and scientific settings. Monitoring applications in this realm include paper production or handling, flexible product packaging, metal forming or flows, spray and droplet impact analysis, capping and sealing, welding flows, cell toxicity testing, and biological material testing.

2.9 Summary

The two most common uses of video in industrial and manufacturing environments are single image classification and high-speed motion-capture video. This work develops algorithms, techniques, and guidelines, to enable video comparison for system monitoring applications.

A dynamic system such as a a synthetic heart valve, a diaper packaging machine, or a karate instructor, is conceptually a collection of finite sized and finite duration "events". We capture the occurrence of these events with video. The similarity of the observed system is implicitly compared to another by explicitly comparing the video of the events of both systems.

Chapter 3

Concepts and Definitions

3.1 Introduction

This chapter defines background concepts and discusses the notation used in the remainder of this document. Throughout this document video is always referenced with a bold and upper-case letter, e.g $\mathbf{V}$. A frame is referenced with a non-bold upper-case letter, e.g frame p is written as $\mathbf{V_p}$. A spatial subregion of a frame is indicated with slanted bars $//$. A scalar distance is referenced with a lower-case d. A multidimensional array, e.g. a matrix or hypervolume, of distances is referenced with an upper-case D. We introduce local-distance hypervolumes, warp paths, and optimal warp distance. Two non-linear filters, the Local Minimum Filter and the Bi-causal Minimum Accumulation filter, are described.

3.2 Vector Sequences - A Video

A video sequence, $\mathbf{V}$, with P frames, where each frame is M_V pixels vertically (y-axis) by N_V pixels horizontally (x-axis), is a spatially arranged and temporally sequenced volume of information of dimension $N_V \times M_V \times P$.

Each frame, a spatially arranged vector at a single moment in time, is referenced by temporal index $\{p\}$ as V_p. Each voxel is referenced by its $\{n, m, p\}$ indices, $V_p(n, m)$. A voxel is frequently a scalar intensity level, but we will also consider voxels that are themselves vectors, e.g. optical flow.

The frames of $\mathbf{V}$ are given

$$\mathbf{V} = \{V_1, V_2, V_3, ..., V_p, ...V_P\}. \tag{3.1}$$

A full frame at time p (e.g. the pixels of frame p) is written as

$$V_p = \begin{pmatrix} V_p(1,1) & V_p(2,1) & ... & V_p(N_V,1) \\ V_p(1,2) & V_p(2,2) & ... & V_p(N_V,2) \\ ... & ... & ... & ... \\ V_p(1,M_V) & V_p(2,M_V) & ... & V_p(N_V,M_V) \end{pmatrix}.$$

3.3 Spatial Sub-regions

A spatial sub-region of a single frame or of a video sequence is denoted with slanted bars $//$. $/V_p(x+2, y+2)/_{3,4}$ indicates a sub-frame of frame V_p what is 3 columns (x-axis) by 4 rows (y-axis) in size, offset by two pixels in both directions from the upper-left origin. We will sometimes use the more compact notation $/V_p(2,2)/_{3,4}$.

If a function takes as input two images of equal size and returns a scalar value we allow the context to indicate the size of the sub frame. For example, $d(A, B)$ is a function that takes as input, two frames A and B of equal size; $d(U, /V_p(2,2)/)$ indicates passing a sub-region of frame V_p starting at pixel location $(2,2)$ that is the same dimension as frame U.

This concept is extended to video sequences. $/\mathbf{V}(2,2)/_{3,4}$ indicates a sub-video of video $\mathbf{V}$ where every frame starts at pixel location $(2,2)$ and is 3 columns by 4 rows in size.

3.4 Elemental Vector Distance

This distance between two similarly sized multi-dimensional vectors U and V is written:

$$d(U,V)$$

If U and V both are two dimensional and indexed by x and y we will sometimes write:

$$d(U, V) = d(U(x, y), V(x, y))$$

If U and V are two-dimensional, V if larger in size, then the distance between U and a region of V, the same size as U, located at x_k, y_k is given as

$$d(U, /V(x_k, y_k)/).$$

Note, the forward slashes used to indicate the same sized region.

3.4.1 Frame-to-Frame Elemental Distances

Consider two frames Q, of size $[N_Q \times M_Q]$, and C of size $[N_C \times M_C]$. Q is of smaller spatial size, $N_Q \leq N_C$, and $M_Q \leq M_C$. The matrix of elemental distances between every frame of Q at every (spatial) location in C is the *Frame-to-frame Elemental Distances.* "Elemental" refers to scalar valued distance between frame Q and a similarly sized region from C. If we do not allow edge-overlap then frame of Q can be positioned in $X(= N_C - N_Q + 1)$, different spatial locations, $Y(= M_C - M_Q + 1)$ different spatial locations with respect to C.

We define a matrix of scalar distances between a small frame, Q, and every similarly sized region of a larger frame, C, as the frame-to-frame elemental distance matrix. The elemental distance matrix is calculated through a series of linear filter operations plus element-to-element additions and multiplications.

We highlight two options for the elemental distance. Application specifics will dictate which of the following, or other, distance calculations is appropriate.

1. The squared Euclidean distance. The sum of the square of the differences between individual elements (pixels) in the template and in the image, is given as

$$D^E(x, y|Q, C)$$

$$= [Q - /C(x, y)/]^T[Q - /C(x, y)/]$$

$$= \sum_{m=0}^{M_Q-1} \sum_{n=0}^{N_Q-1} (\ Q(n, m) - C(x + n, y + m)\ )^2$$

$$= \sum_{m,n} (\ Q^2(n, m) + C^2(x + n, y + m) - 2Q(n, m)C(x + n, y + m)\ )$$

$$= \sum_{m,n} Q^2(n, m) + \sum_{m,n} C^2(x + n, y + m) - \sum_{m,n} 2Q(n, m)C(x + n, y + m) \tag{3.2}$$

The first part is the sum of square of the elements of Q. The second part is the sum of squares of the portion of C that overlap Q when in position (x, y). The third part is twice the negative correlation of Q and C. For a frame C that is locally uniform statistically, as the correlation between the template frame and a portion of the test frame increases the distance measure decreases. The first two parts make this measure sensitive to illumination.

This sensitivity to illumination may cause problems. This distance measure will be large in regions that do not correlate well, but it will also be large in regions of high intensity.

2. In some applications, illumination insensitivity may be important, this motivates the use of a distance measure based on normalized correlation. The maximum value of the normalized correlation is 1; 1 minus the normalized correlation between the template frame and the test frame is an appropriate distance measure.

$$D^N(x,y|Q,C)$$

$$= 1 - \frac{\sum_{m=0}^{M_Q-1}\sum_{n=0}^{N_Q-1} Q(n,m)C(x+n,y+m)}{\sqrt{\sum_{m=0}^{M_Q-1}\sum_{n=0}^{N_Q-1} Q^2(n,m)}\sqrt{\sum_{m=0}^{M_Q-1}\sum_{n=0}^{N_Q-1} C^2(x+n,y+m)}} \tag{3.3}$$

$$= 1 - \frac{\sum_{m,n} Q(n,m)C(x+n,y+m)}{\sqrt{\sum_{m,n} Q^2(n,m)}\sqrt{\sum_{m,n} C^2(x+n,y+m)}}$$

This distance measure is insensitive to illumination scaling in the two images. That is, it will give the same distance value for two regions that are different by a scale factor. This makes this distance metric suitable for application where a uniform change in illumination is expected. It may be inappropriate at times due to this very scale invariance; two regions in an image that are different by a scale factor but are contextually very different regions are seen as similar.

Higher order distances measure, those that operate on a combination of pixel intensity and optical flow, combined moments of either, and many other distance measure are possible. Application specifics will guide the selection of the appropriate distance measure. One such alternative is a modification with a positive definite weight matrix, M, that reflects the relative importance of each spatial element (pixel) of Q and C. For the Euclidean distance this is written:

$$D^E(x,y|Q,C) = [Q - /C(x,y)/]^T M[Q - /C(x,y)/].$$

We may use any of the following compact representations. Context will reinforce meaning.

$$D(x,y|Q(x,y),C(x,y)) = D(x,y|Q,C) = D(x,y) = D(Q,C)$$

For some x_k, y_k, the value of $D(x_k, y_k|Q(x,y), C(x,y))$ is a scalar value:

$$D(x_k, y_k) = d(Q, /C(x + x_k, y + y_k)/).$$

3.5 Elemental Distances Hypervolume

Consider two video sequences $\mathbf{Q}$, of size $[N_Q \times M_Q \times J]$, and $\mathbf{C}$, of size $[N_C \times M_C \times I]$. $\mathbf{Q}$ is of smaller spatial size, $N_Q \leq N_C$, and $M_Q \leq M_C$. The hypervolume of elemental distances between every frame of $\mathbf{Q}$ at every (time and space) location in $\mathbf{C}$ is defined as the *Elemental Distances Hypervolume.* "Elemental" refers to scalar valued distance between a single frame of $\mathbf{Q}$ and a similarly sized region from $\mathbf{C}$.

If we do not allow edge-overlap then a frame of $\mathbf{Q}$ can be positioned in $X(= N_C - N_Q + 1)$, different spatial locations, $Y(= M_C - M_Q + 1)$ different spatial locations, and I different temporal locations as shown in Figure 4-2.

The hypervolume of elemental distances is defined

$$D_L(\mathbf{Q}, \mathbf{C}) = D_L(i, j, x, y, |\mathbf{Q}, \mathbf{C}) = d(Q_j, /C_i(x, y)/) \tag{3.4}$$

3.6 Warp Path

The ordered quadruplet indicating position k in a four-dimensional hypervolume, $[I \times J \times X \times Y]$, is given as

$$\mathbf{w_k} = \{i_k, j_k, x_k, y_k\}. \tag{3.5}$$

A "path" in such a hypervolume is defined as a sequence of ordered indexes. The path from (i_1, j_1, x_1, y_1) to (i_K, j_K, x_K, y_K) is given as

$$\mathbf{W_K} = \{w_1, w_2, w_3, ..., w_j, ...w_K\}. \tag{3.6}$$

A "scaled path" through the four dimensional hypervolume would be a path such that at a particular location, k, on the path the ratio of $\frac{i_k}{j_k}$ would be the same (ignoring integer rounding) as the ratio $\frac{I}{J}$. In other words, the temporal portion of a "scaled path" from times $(i = 0, j = 0)$ to times $(i = I, j = J)$ is a straight line. A "warp path" is a path such that the ratio $\frac{i_k}{j_k}$ is not necessarily equal to $\frac{I}{J}$. A "warp path" allows temporal distortion other than linear scaling.

3.7 Time Terminated Warp Path

For our application I and J are the two temporal dimensions of the Elemental Distances Hypervolume and X and Y are the spatial dimensions. A time terminated warp path in the four-dimensional hypervolume, of dimension $[I \times J \times X \times Y]$, is defined as a sequence of ordered indexes from $(i = 1, j = 1, x_1, y_1)$ to $(i = I, j = J, x_K, y_K)$.

3.8 Bi-Causal Time Terminated Warp Path

For our application I and J are two temporal dimensions of the Elemental Distances Hypervolume and X and Y are spatial dimensions. A time terminated warp path in the four-dimensional hypervolume , $[I \times J \times X \times Y]$, is defined as a sequence of ordered indexes from $(i = 1, j = 1, x_1, y_1)$ to $(i = I, j = J, x_K, y_K)$. A Bi-causal Warp path is a path for which $i_k \leq i_{k+1}$ and $j_k \leq j_{k+1}$ for all k.

3.9 Warp Distance

The warp distance between two vector sequences Q and C along warp path W is written:

$$\begin{aligned} d_W(\mathbf{Q}, \mathbf{C} | w(k), k = 1..K) &= d_{W_{K},a,N(a)}(\mathbf{Q}, \mathbf{C}) && (3.7) \\ &= \frac{\sum_{k=1}^{K} D_L(w(k) | \mathbf{Q}, \mathbf{C}) a(k)}{N(W)} && (3.8) \end{aligned}$$

$$= \frac{\sum_{k=1}^{K} D_L(i_k, j_k, x_k, y_k|\mathbf{Q}, \mathbf{C})a(k)}{N(W)} \tag{3.9}$$

$$= \frac{\sum_{k=1}^{K} d(Q_{j_k}, /C_{i_k}(x_k, y_k)/)a(k)}{N(W)} \tag{3.10}$$

Where $a(k)$ is a scaling parameter which is a function of path position, and $N(W)$ is a path dependent normalization factor.

The un-normalized warp distance, d_W^0, between two vector sequences $\mathbf{Q}$ and $\mathbf{C}$ along warp path W, is written:

$$\begin{aligned}
d_W^0(\mathbf{Q}, \mathbf{C}|w(k), k = 1..K) &= d_{W_K,a}^0(\mathbf{Q}, \mathbf{C}) && (3.11)\\
&= \sum_{k=1}^{K} D_L(w(k)|\mathbf{Q}, \mathbf{C})a(k) && (3.12)\\
&= \sum_{k=1}^{K} D_L(i_k, j_k, x_k, y_k|\mathbf{Q}, \mathbf{C})a(k) && (3.13)\\
&= \sum_{k=1}^{K} d(Q_{j_k}, /C_{i_k}(x_k, y_k)/)a(k) && (3.14)\\
&= d_{W_{K-1},a}(\mathbf{Q}, \mathbf{C}) + d(Q_{j_K}, /C_{i_K}(x_K, y_K)/)a(K) && (3.15)
\end{aligned}$$

The un-normalized warp distance between two video sequences $\mathbf{Q}$ and $\mathbf{C}$ is a path dependent accumulation of elemental vector distances. The sequence of elemental distances W, defines a "warp path" through the elemental distance hypervolume D_L. It is recursively defined as the un-normalized warp distance along path $w(k = 1..K-1)$ plus the elemental distance value indexed by $w(K)$, multiplied by $a(K)$.

3.10 Optimal Warp Distance

For the finite set of Bi-Causal Warp Paths the optimal warp path between two video sequences $\mathbf{Q}$ and $\mathbf{C}$ is written:

$$
\begin{aligned}
d^*_W(\mathbf{Q},\mathbf{C}) &= \min_W d_W(\mathbf{Q},\mathbf{C}) \\
&\quad \text{s.t. } W \text{ Satisfies path constraints.}
\end{aligned}
$$

The optimal path, W^* is then given

$$
\begin{aligned}
W^*(\mathbf{Q},\mathbf{C}) &= \arg\min_W d_W(\mathbf{Q},\mathbf{C}) && (3.16) \\
&\quad \text{s.t. } W \text{ Satisfies path constraints.} && (3.17)
\end{aligned}
$$

The optimal warp path distance, $d^*_W(Q,C)$ is found by minimizing over all possible paths. It is the minimum cost warp path distance that aligns **Q** and **C** from temporal beginning to temporal end according to some constraints in allowable paths.

$\mathbf{Q}^*$ and $/\mathbf{C}^*/$ are the optimally aligned **Q** and **C** .

$$
\begin{aligned}
\mathbf{Q}^* &= \{Q_{j_1^*}, Q_{j_2^*}, ..., Q_{j_K^*}\} && (3.18) \\
/\mathbf{C}^*/ &= \{/C_{i_1^*}(x_1^*, y_1^*)/, /C_{i_2^*}(x_2^*, y_2^*)/, ..., /C_{i_K^*}(x_K^*, y_K^*)/\} && (3.19)
\end{aligned}
$$

3.11 Optimal Warp Distance Hypervolume

The Optimal Warp Distance Hypervolume, D_O, is a hypervolume of optimal warp distance values, indexed by (i,j,x,y).

Each indexed value, (i_k, j_k, x_k, y_k), is the warp distance optimally "aligning" $Q_1 \leftrightarrow /C_1/$ through $Q_{j_k} \leftrightarrow C_{i_k}(x_k, y_k)$. Equivalently it is the optimal warp path distance from an indexed location $(1,1,x,y)$ to indexed location (i_k, j_k, x_k, y_k) $\forall$ $i_k \in 1..I$, $j_k \in 1..I$, $x_k \in 1..(N_C - N_Q + 1)$, $y_k \in 1..(N_C - N_Q + 1)$ through the elemental distances hypervolume $D_L(\mathbf{Q},\mathbf{C})$.

$$D_O(i,j,x,y) = d^*_W(Q_{1..j}, C_{1..i} \mid /C_i/ = /C_i(x,y)/ \ ) \tag{3.20}$$

The plane of minimum values on an $\{x,y\}$ plane of the hypervolume for fixed i_k, j_k is written $D^*_O(i = i_k, j = j_k, x, y | Q_{1..j_k}, C_{1..i_k})$.

3.12 Local Minimum Filter

The Local Minimum Filter selects, from its neighborhood of support, the minimum of the signal value multiplied by the coefficients of the filter. The filter is not implemented in place; storage for both the input signal and output signal are required.

The Local Minimum Filter, $MF(x,y)$, operation given signal s and filter f of size $M_f \times N_f$ is written:

$$\begin{aligned} MF(x,y|s(x,y),f(x,y)) &= s(x,y) \bullet f(x,y) \\ &= \min(s(x+n,y+m)\ f(n,m))\ \forall\ n \in \{0..N_f-1\}, m \in \{0..M_f-1\} \end{aligned}$$

This operation is illustrated in Figure 3-1.

3.13 Bicausal Minimum Accumulation Filter

The Bicausal Minimum Accumulation Filter replaces a signal value, with the signal value plus the minimum of the current and preceding signal values multiplied by the coefficients of the filter. The filter is implemented in place. It is called Bicausal because it is a casual filter along two directions.

The Bicausal Minimum Accumulation Filter, $MA(x,y)$, operation given signal s and filter a of size $M_a \times N_a$ is written:

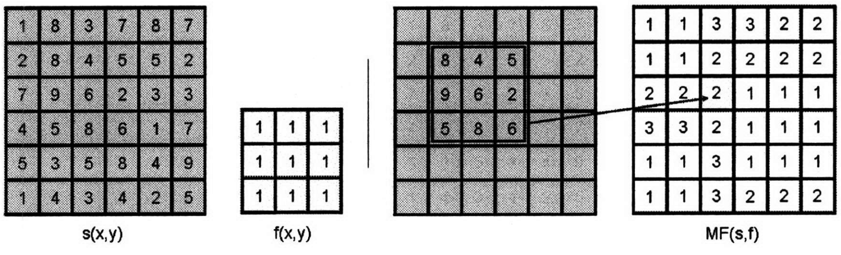


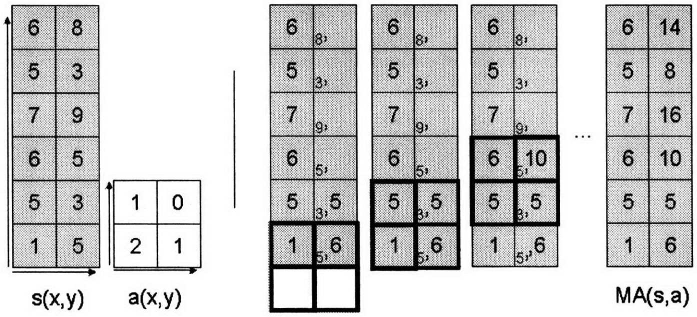
Figure 3-1: Depiction of a Local Minimum Filter. On the left half of the figure the signal frame, $s(x, y)$, and filter kernel $f(x, y)$ are depicted. On the right half of the figure the filter kernel is shown at location $x = 2, y = 3$ in the signal frame. Every term of f is 1, therefore we are only concerned with the minimum value of s under the support of f. The minimum value of the signal "under" the filter kernel at its current location is 2, this is pointed to in the local minimum filter output labeled $MF(s, f)$.

$$\begin{aligned} MA(x, y|s(x, y), a(x, y)) &= s(x, y) \diamond a(x, y) \\ &= s(x, y) + \min(s(x - N_a + n + 1, y - M_a + m + 1)\ a(n, m)) \\ &\ \forall\ n \in \{0..N_a - 1\}, m \in \{0..M_a - 1\} \end{aligned}$$

This operation if illustrated in Figure 3-2

3.14 Summary

We introduced concepts and discussed the notation used in the remainder of this document. A video is always referenced with a bold and upper-case letter, e.g $\mathbf{V}$. A frame is referenced with a non-bold upper-case letter, e.g frame i is written as $\mathbf{V_i}$. A spatial subregion of a frame is indicated with slanted bars $//$. A scalar distance is referenced with a lower-case d. A multidimensional array, e.g. a matrix or hypervolume, of distances is referenced with an upper-case D. The Local Minimum Filter, the Bi-causal Minimum Accumulation filter, distance hypervolumes, warp paths, and optimal warp distance are concepts that are important for the development of DTSW.



Bi-Causal Minimum Accumulation Filter

Figure 3-2: Depiction of a Bicausal Minimum Accumulation Filter. On the left half of the figure the signal frame, $s(x, y)$, and filter kernel $s(x, y)$ are depicted. On the right half of the figure the filter kernel is shown at consecutive locations. **Location in the signal frame : $\mathbf{x = 1, y = 0}$**. The new filtered value is equal to the current value 5 plus the minimum among the individual terms of the signal multiplied by the kernel, i.e. minimum of $[1 \times 1]$. The result is a (minimum accumulation) filtered value of 6. **Location in the signal frame : $\mathbf{x = 1, y = 1}$**. The new filtered value is equal to the current value 3 plus the minimum among the individual terms of the (already filtered) signal multiplied by the kernel, i.e. minimum of $[1 \times 5, 2 \times 1, 1 \times 6]$. The result is a filtered value of 5. **Location in the signal frame : $\mathbf{x = 1, y = 2}$**. The new filtered value is equal to the current value 5 plus the minimum among the individual terms of the (already filtered) signal multiplied by the kernel, i.e. minimum of $[1 \times 6, 2 \times 5, 1 \times 5]$. The result is a filtered value of 10. The result for $x = 1, y = 0..5$ is labeled $MA(s, a)$.

Chapter 4

Dynamic Time and Space Warping (DTSW)

4.1 Introduction

Dynamic Time and Space Warping (DTSW) compares two videos - a template, **Q**, and a test, **C**. DTSW determines and reports a detailed comparison in time and space between a template and a test video. In this chapter we develop the DTSW algorithm. We discuss the constraints and objective that form the DTSW problem statement. We discuss its algorithmic structure and highlight implementation efficiencies.

4.2 Preliminaries

Dynamic Time and Space Warping compares two videos - a template, **Q**, and a test, **C**, by finding the optimal path of the template video through the time and space of the test video. The test video is of wider spatial extent and is temporally longer or shorter than the template video. The optimal path obeys local continuity rules that control how far any one frame in the template is shifted in space and time in the volume of the test video. Dynamic Time and Space warping finds the optimal, lowest cost, path through a hypervolume of elemental distances. Figure 4-1 shows the input and output of the algorithm.

- Operates on spatially arranged frames of data such as pixel intensity, optical flow, edges, or combinations of such information.
- Input
  - Template sequence **Q**.
  - Test sequence **C**.
  - Application specific transformation functions, such as filters, optical flow calculations, etc.
  - Local Distance Function.
- Ouput
  - The DTSW distance.
  - The optimal space time alignment of Q through C.
  - A warped **Q**.
  - A warped **C**.

DTSW is an extension of Dynamic Time Warping (DTW). DTW is used to compare scalar or vector time-sequences. It allows time warping in the comparison algorithm. DTW has been used for application in speech recognition [PLVS04], [KLMB87], [VRCB88], [MRR80], [HCLR83], database query [cFKLR05], [KR04], [KP99], [KP01], [CKHP02], and chemical batch processing [KMT98]. The unique aspects of this extension are the consideration of multiple dimensions and the efficiencies that arise when specifically applied to video, which allows much of the required low level distance calculation to be performed with a series of linear filters, non-linear filters, and simple arithmetic operations.

DTSW provides a detailed comparison in time and space between a template and a test video. It is a view-based algorithm; it uses a spatial-temporal representation of information, such as pixel intensity, optical flow, or other low level parameters. We use DTSW to compare videos events of amorphous or flexible objects lacking

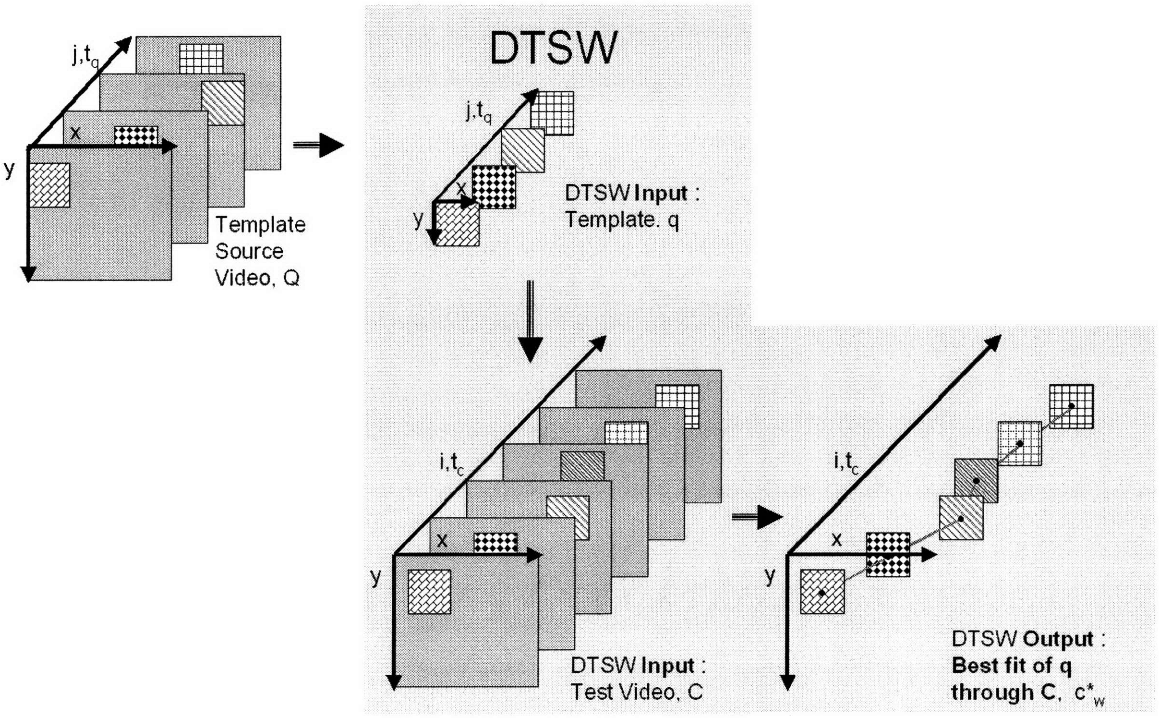


Figure 4-1: DTSW. Dynamic Time and Space Warping compares two videos - a template, **Q**, and a test, **C**, by finding the optimal path of the template video through the time and space of the test video. The optimal path obeys local continuity rules that control how far any one frame in the template is shifted in space and time in the volume of the test video. Dynamic Time and Space warping finds the optimal, lowest cost, path through a hypervolume of elemental distances.

distinct features. Applying DTSW to two videos' volumetric-data we find an optimal space-shifting and time-warping alignment between two videos with a measure of local similarity between each aligned frame. This alignment problem is stated as an optimization problem solvable via a dynamic programming algorithm. In this way, the DTSW algorithm can be implemented as a linear filter bank followed by a dynamic programming algorithm. A test video is passed through a linear filter bank whose filter kernels are the frames of the template video. The filter bank output is a four-dimensional hypervolume which is then optimally traversed via a dynamic programming algorithm.

We further show that the structure of the DTSW algorithm is such that it can be implemented via the iterative application of several linear filters and two non-linear filters and that the structure of the algorithm is parallelizable. We develop a compact, principal component representation of the known or query video, this allows us to reduce the size of the linear filter-bank and facilities efficient calculation of the data-structure (the four-dimensional hypervolume) to which we iteratively apply the two non-linear filters.

In this chapter, we develop the DTSW algorithm, and the DTSW distance. We develop efficient implementations of DTSW. We explain various relaxation of the DTSW algorithm in order to reveal its underlying structure.

4.3 DTSW Definition

The Dynamic Time and Space Warping (DTSW) algorithm finds the best path through the 4 dimensional space of "frame element to "frame element" distances such that the total distance between the template sequence and some sub-sequence (spatial) of the test sequence is minimized and the warp path satisfies continuity, bi-causality, and spatial drift constraints.

DTSW takes as input a template video, $\mathbf{Q}$, and a test video, $\mathbf{C}$. It outputs a video $\mathbf{C}^*$; $\mathbf{C}^*$ is the best fit (while undergoing warping, and shifting) of $\mathbf{Q}$ in $\mathbf{C}$. $\mathbf{Q}_\mathbf{S}$ is the video that is the source of the template video $\mathbf{Q}$, the Golden Master Video. $\mathbf{Q}$ is of the same temporal length as $\mathbf{Q_S}$. A frame of $\mathbf{Q}$ is at most the same spatial size of a frame of $\mathbf{Q_S}$, but will, in general, be smaller. The frames of $\mathbf{Q}$ follow some spatial path through $\mathbf{Q_S}$. $\mathbf{Q_S}$ and $\mathbf{Q}$ are both J frames in length.

As part of the DTSW algorithm we determine the similarity between a frame of the template and a similarly sized region of the test video. If we follow a rule that says we can only calculate the similarity between like-sized regions, then a frame of the template can be positioned in $X(= N_C - N_q + 1)$ different spatial locations horizontally, $Y(= M_C - M_q + 1)$ different spatial locations vertically, and I different temporal locations as shown in Figure 4-2.

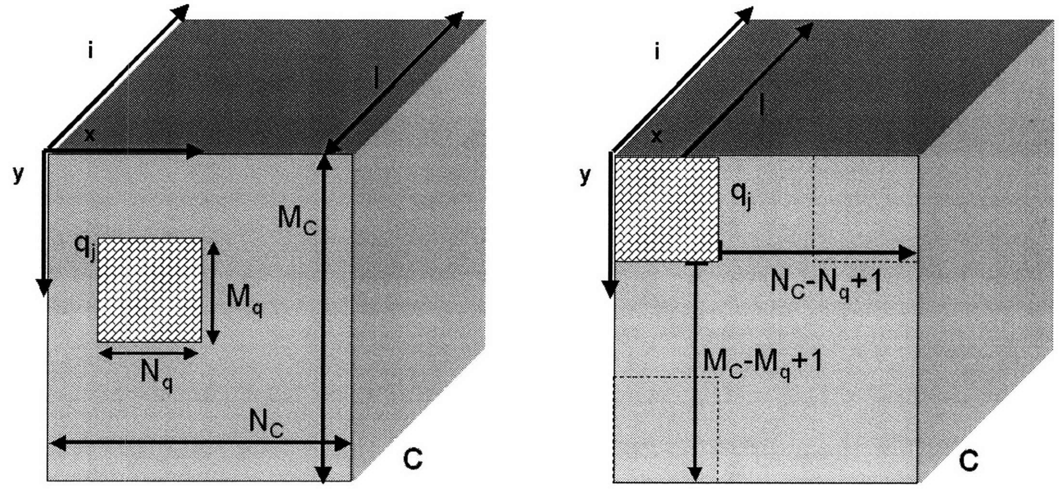


Figure 4-2: The test video $\mathbf{C}$ and a single frame, Q_j, of the template video. Depiction of dimensions and the possible locations were a frame of the template can be located in the test video.

The first step in the DTSW algorithm is to calculate the elemental similarity between every frame of the template at every location in the test video and place the results in an *Elemental Distances HyperVolume*, $D_d = |I \times J \times X \times Y|$. "Elemental" similarity refers to a similarity measure between a single frame of $\mathbf{Q}$ and a similarly sized region from $\mathbf{C}$. *In practice this complete determination of "elemental" similarity is not performed*; however, for ease of explanation we consider the fully populated Elemental Distance Hypervolume.

Next we want to find the minimum cost casually directed path through D_L from the beginning of time, $(i = 0, j = 0)$, to the simultaneous end of time $(i = I, j = J)$, for both videos according to local continuity or transition rules in both the time and spatial dimensions. We consider a set of indices, $\{W\}$, through the $[I \times J \times X \times Y]$ hypervolume of elemental distances. The path from $(i = 1, j = 1, x_1, y_1)$ to $(i = I, j = J, x_K, y_K)$ is given as

$$\mathbf{W} = \{w_1, w_2, w_3, ..., w_j, ...w_K\}. \tag{4.1}$$

The ordered quadruplet indicating a position in the hypervolume is given as

$$\mathbf{w_k} = \{j_k, i_k, x_k, y_k\}. \tag{4.2}$$

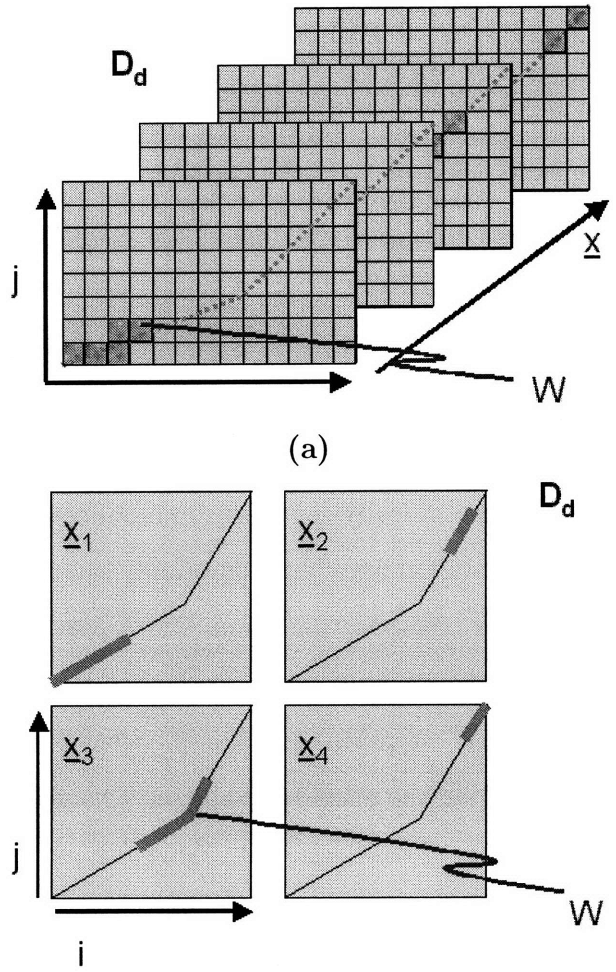
The length of the path, K, is bounded by the maximum length of either $\mathbf{C}$ or $\mathbf{q}$ and by the sum of their lengths, $\max(I, J) \leq K \leq (I + J)$.

The Elemental Distance Hypervolume and minimum cost path are shown in Figure 4-3a. Note that the multiple spatial dimensions of D_L have been collapsed to a single axis in order to allow a three-dimensional representation. Each frame of $\mathbf{V}$ is matched to at least one frame in $\mathbf{q}$ and vice-versa; no frames are skipped. The minimum cost path through D_d matches each frame, q_{j_k}, to some location, $\{x_k, y_k\}$, in frame C_{i_k} . The warp path is shown in bold. It is displayed on the temporal projection, on the IJ plane, of the warp path for every $\{x, y\}$ location.

A planar representation of the Elemental Distance Hypervolume is show in Figure 4-3b. A single IJ plane of the hypervolume is shown for every $\{x, y\}$ location.

4.4 DTSW Constraints

The DTSW constraints were explained in Chapter 1. The constraints are again summarized here.



(a)

(b)

Figure 4-3: Elemental Distances Hypervolume with Representative Warp Path

- Spatial continuity. Figure 1-8. We expect that frames of the template are time sequential and that the position change from frame to frame is bounded.

- Spatial drift. Figure 1-9. We locally match a temporal and spatial instance in one video with a temporal and spatial instance in another. We do not match to a single instance in time at more than one location. If it is necessary to "replicate" a frame of the test video (to match to two or more frames of the template video) then the matching must occur at the same spatial location in the one frame of the test video for all frames of the template.

- Temporal continuity. Figure 1-10. Though we do expect temporal variation between two videos we limit the local extent of that variation.

- Bi-Temporal causality. Figure 1-11. We consider events that are progressing in time. Local time scales may vary; we don't expect moments in either video to be out of time-sequential order.

- End points. Figure 1-12. This constraint is easily relaxed and move in order to accommodate imprecise temporal clipping and facilitate scanning.

We enforce a causal (monotonically increasing) alignment in time, motivated by the notion that we are interested in matching temporally sequenced (forward advancing) events. Casual matching ensures a finite number of possible paths between any two locations in D_d. Acasual matching could be allowed if we also constrained the length of a path, but generally casual matching is physically motivated. We enforce casual matching by constraining the set of allowable predecessors to a location in D_L. The allowable predecessors are those such that, along the path through D_d, time is monotonically increasing $\frac{di}{dj} \geq 0$ and $\frac{dj}{di} \geq 0$.

Additional global and local path constraints such as allowable temporal slope, deviation from a prescribed path, staying within a global temporal bound, are all locally constrained by the set of allowable predecessors to a location in D_L.

We also assume that we match the first frame of the template to some location in the first frame of the test video and the last frame of the template to some location in the last frame of the test video, $w(1) = [1, 1, x_1, y_1]$, $w(K) = [J, I, x_K, y_K]$. This constraint will be relaxed in practice to allow deviation at the beginning and end of the sequence. But as mentioned in the introduction we often have temporal sequences that have been temporally segmented, or clipped, by outside control signals.

4.5 DTSW Distance

The DTSW warp distance for some path aligning $\mathbf{Q}$ and $\mathbf{C}$ subject to the previously described constraints is $d_{dtsw}(\mathbf{Q}, \mathbf{C})$. The optimal distance d^*_{dtsw} is found by minimizing over all possible paths, W.

$$\begin{aligned} d^*_{dtsw}(\mathbf{Q},\mathbf{C}) &= \min_W d_{dtsw}(\mathbf{Q},\mathbf{C}) \\ \text{s.t.} \quad & W \text{ satisfies continuity} \\ & W \text{ satisfies bi-causality} \\ & W \text{ avoids spatial drift} \end{aligned} \tag{4.3}$$

The optimal DTSW path, W^* is then given

$$\begin{aligned} W^*(\mathbf{Q},\mathbf{C}) &= \arg\min_W d_{dtsw}(\mathbf{Q},\mathbf{C}) \\ \text{s.t.} \quad & W \text{ satisfies continuity} \\ & W \text{ satisfies bi-causality} \\ & W \text{ avoids spatial drift} \end{aligned} \tag{4.4}$$

The DTSW distance for a feasible path is given

$$\begin{align} d_{dtsw}(\mathbf{Q},\mathbf{C}|w(k),k=1..K) &= d_{W_K,a,N(a)}(\mathbf{Q},\mathbf{C}) \tag{4.5} \\ &= \frac{\sum_{k=1}^{K} D_L(w(k)|\mathbf{Q},\mathbf{C})a(k)}{N(W)} \tag{4.6} \\ &= \frac{\sum_{k=1}^{K} D_L(i_k,j_k,x_k,y_k|\mathbf{Q},\mathbf{C})a(k)}{N(W)} \tag{4.7} \\ &= \frac{\sum_{k=1}^{K} d(Q_{j_k},/C_{i_k}(x_k,y_k)/)a(k)}{N(W)} \tag{4.8} \\ & \tag{4.9} \end{align}$$

$d_{dtsw}(\mathbf{Q},\mathbf{C})$ is the sum of all local frame distances along the path between pairs of frame regions weighted by $a(k)$ and normalized by $N(W)$. $d_{dtsw}(Q_{j_k},/C_{i_k}(x_k,y_k)/)$ is the scalar distance between frame Q_{j_k} and an equally sized subframe of C_{i_k} located at (x_k,y_k), $C_{i_k}(x_k,y_k)$. Local distances are weighted by $a(k)$ depending on the local transition to the k path point from the $k-1$ path point. $a(k)$ is used to implement the continuity, bi-causality, and spatial drift constraints. $N(W)$ is a normalization factor, the form of which will depend on the weighting function $a(k)$. It is used to calculate a warp path distance that is independent of the path's length.

4.6 Dynamic Programming Statement of Algorithm

With constraint appropriate definitions for a(k), the problem can be written as

$$d^*_{tdsw}(\mathbf{Q},\mathbf{C}) = \min_W \frac{\sum_{k=1}^K D_L(w(k)|\mathbf{Q},\mathbf{C})a(k)}{N(W)} \tag{4.10}$$

It is possible to solve such optimization problems. Dynamic programming can be used if the normalization factor is independent of optimal path. The problem can then be stated as as recursive solution to a local optimization problem. Normalization is important for unbiased comparison of different length trajectories, either when comparing $\mathbf{q}$ to $\mathbf{C_1}$ and to $\mathbf{C_2}$ or when evaluating different possible endpoints when comparing $\mathbf{q}$ to $\mathbf{C_1}$.

We may simply ignore the normalization while determining the minimum path and then normalize the results. The problem can be written as

$$d^*_{dtsw}(\mathbf{Q},\mathbf{C}) = \frac{1}{N(a)} \min_W \sum_{k=1}^K D_L(w(k)|\mathbf{Q},\mathbf{C})a(k). \tag{4.11}$$

This is equivalent to defining the following weighting function and normalization

$$a(k) = \begin{cases} [i_k - i_{k-1}] + [j_k - j_{k-1}] \\ \qquad \forall\ (0 \le i_k - i_{k-1} \le 1), (0 \le j_k - j_{k-1} \le 1) \\ \\ \infty \qquad \text{otherwise} \end{cases} \tag{4.12}$$

$$N(a) = \sum_{k=1}^K a(k) \tag{4.13}$$

$$= \sum_{k=1}^{K}([i_k - i_{k-1}] + [j_k - j_{k-1}]) \tag{4.14}$$

$$= i_K + j_K = I + J \tag{4.15}$$

Other continuity constraints are easily defined by defining alternative weighting functions.

The path-independent shortest-path problem that we will solve is

$$d^*_{dtsw0}(\mathbf{Q}, \mathbf{C}) = \min_W \sum_{k=1}^{K} D_L(w(k)|\mathbf{Q}, \mathbf{C})a(k). \tag{4.16}$$

where, again, we have made explicit the recursive nature of the calculation.

The normalized DTSW distance is then simply found by scaling by the sum of the length of **Q** and **C**

$$d^*_{dtsw}(\mathbf{Q}, \mathbf{C}) = \frac{1}{I+J} d^*_{dtsw0}(\mathbf{Q}, \mathbf{C}). \tag{4.17}$$

The solution of Equation 4.17 may be solved via dynamic programming. We let W^* be the optimal global path in the $IJXY$ hypervolume of elemental distances. There are two rules that define the optimal local policy for finding this path.

1. If W^* goes through an (i, j, x, y) point, then the optimal path to the (i, j, x, y) point is part of W^*.

2. The optimal path to the (i, j, x, y) point depends only on the previous points.

These two statements define a recursive dynamic programming relationship.

$$D^*_0(i, j, x, y) = D_L(i, j, x, y) + \min \begin{cases} D^*_0(& i-1, & j, & x - \Delta x, & y - \Delta y) \\ D^*_0(& i-1, & j-1, & x - \Delta x, & y - \Delta y) \\ D^*_0(& i, & j-1, & x, & y) \end{cases} \tag{4.18}$$

D_L is the elemental distances hypervolume and D^*_0 is the optimal DTSW distance hypervolume. From the spatial continuity constraint it follows that $|\Delta x| \leq b_x$ and $|\Delta y| \leq b_y$ limit the evolution of the path along a local spatial region.

4.7 Elemental Distance Hypervolume

For the moment we ignore all potential efficiencies of implementation. Internal to the DTSW algorithm is the calculation of the [elemental] distance between every frame of the template sequence and every spatial and temporal location where that frame could be found in the test sequence. In practice it is not necessary to fully populate the elemental distance hypervolume.

The elemental distances hypervolume, D_L, is calculated by passing each frame of the test video, $\mathbf{C}$, through a series of linear filter banks. The selected elemental distance measure, e.g Euclidean, one-minus-normalized correlation, etc... dictates the exact form of the filterbank. However, most low-level distance functions will include a correlation term.

The spatial correlation between each frame of the template video with each frame of the test video is implemented as a linear filter where each frame of the template video is a filter. This implementation is depicted in Figure 4-4.

The minimum DTSW distance is then found according to the above recursive relationship that takes us from the lower left corner of the D_L to the upper right corner of the D_L, as illustrated in Figure 4-5. We may choose to store the optimal DTSW distance hypervolume, which then allows us to find the minimum cost path.

4.8 Simple Numerical Example of DTSW

Consider the template and test video depicted in Figure 4-6. The template video temporal length, J, is 4 frames; each frame is 2 pixels wide by 1 pixel in height. The test video temporal length, I, is 6 frames; each frame is 5 pixels in width and 1 pixel in height. The first frame of the template video is $<\ 1\ 4\ >$. The first frame of the test video is $<\ 0\ 1\ 5\ 6\ 1\ >$.

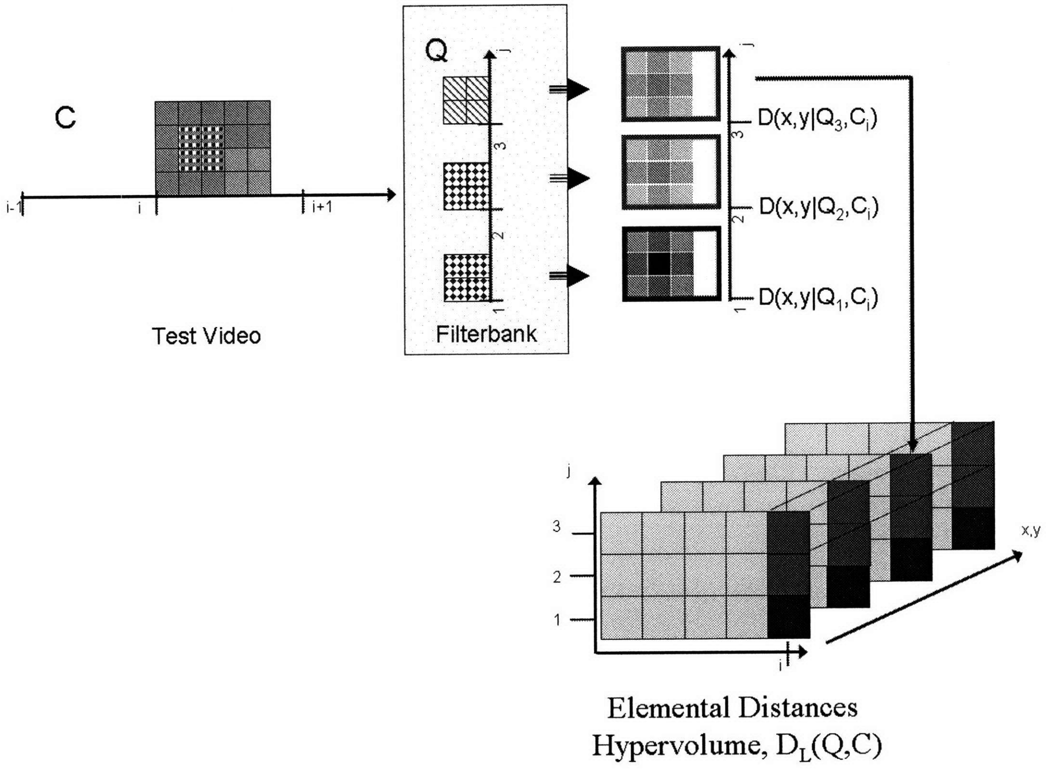


Figure 4-4: Filter Bank Implementation - Each Template Frame as a Filter

Find the minimum cost path to the upper-right edge, ● ,from the lower-left, ◣ ,of the Elemental Distance Hypervolume.

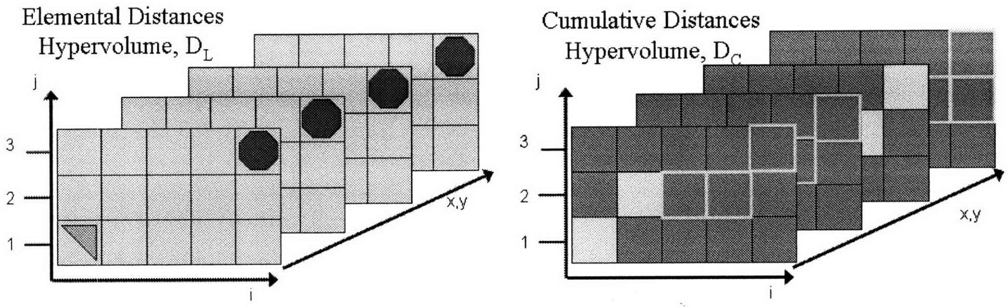


Figure 4-5: DTSW - Dynamic Programming Intuition. The optimal path to an $\{i, j, x, y\}$ point depends only on the previous grid points. If the optimal path, W^*, goes through an $\{i, j, x, y\}$ point then the optimal path to the $\{i, j, x, y\}$ point is part of the optimal path. This is a recursive relationship by which we define the cumulative distance to every position.

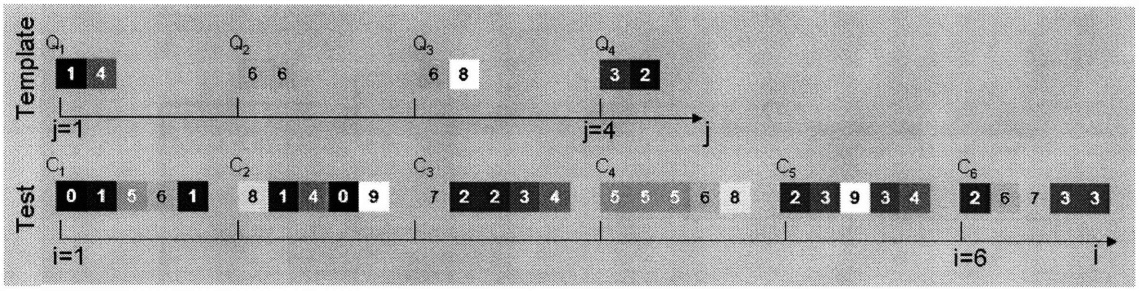
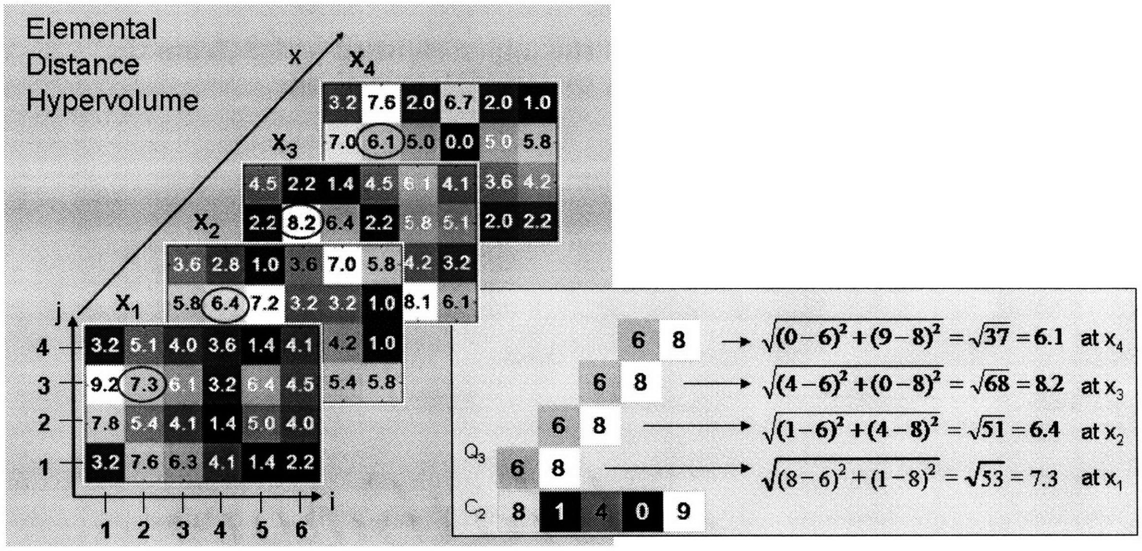


Figure 4-6: Template and Test Videos for a Simple Example of DTSW. The template video temporal length, J, is 4 frames; each frame is 2 pixels wide by 1 pixel in height. The test video temporal length, I, is 6 frames; each frame is 5 pixels in width and 1 pixel in height. The first frame of the template video is $<\ 1\ 4\ >$. The first frame of the test video is $<\ 0\ 1\ 5\ 6\ 1\ >$.

A template frame may be found at one of 4 possible horizontal locations and one of 1 possible vertical location in a frame of the test sequence. The elemental distances hypervolume, D_L, and the cumulative distance hypervolume, D_c are both $J \times I \times X_1 \times X_2 = 4 \times 6 \times 4 \times 1$ in size.

Figure 4-7 depicts the four $I \times J$ planes of D_L. The elemental distance between frame Q_3 and the four locations where it can be placed in frame C_2 are circled.



$\rightarrow \sqrt{(0-6)^2+(9-8)^2} = \sqrt{37} = 6.1$ at x_4

$\longrightarrow \sqrt{(4-6)^2+(0-8)^2} = \sqrt{68} = 8.2$ at x_3

$\longrightarrow \sqrt{(1-6)^2+(4-8)^2} = \sqrt{51} = 6.4$ at x_2

$\longrightarrow \sqrt{(8-6)^2+(1-8)^2} = \sqrt{53} = 7.3$ at x_1

Figure 4-7: Elemental Distances Hypervolume for a Simple Example of DTSW. The elemental distance between frame Q_3 and the four locations where it can be placed in frame C_2 are circled.

We generate D_C, and find the optimal warp path according to the local continuity rules.

- The lower left corner of D_c is initialized with the lower left corner of D_L. $D_c(i = 0, j = 0, \bar{x}) = D_L(i = 0, j = 0, \bar{x})$.

- When calculating the cumulative distance to any point we consider a neighborhood around the current $(i, j, \bar{x})$ location. According to our continuity rules, a path transition in space (of the test sequence) must accompany a transition in time (in the test sequence). Therefore, the values $D_c(i = 0, j > 0, \bar{x})$ in the first column of an $I \times J$ plane of D_c are simply the sum of the cumulative distance below, $D_c(i, j-1, \bar{x})$, plus the corresponding elemental distance $D_L(i = 0, j, \bar{x})$.

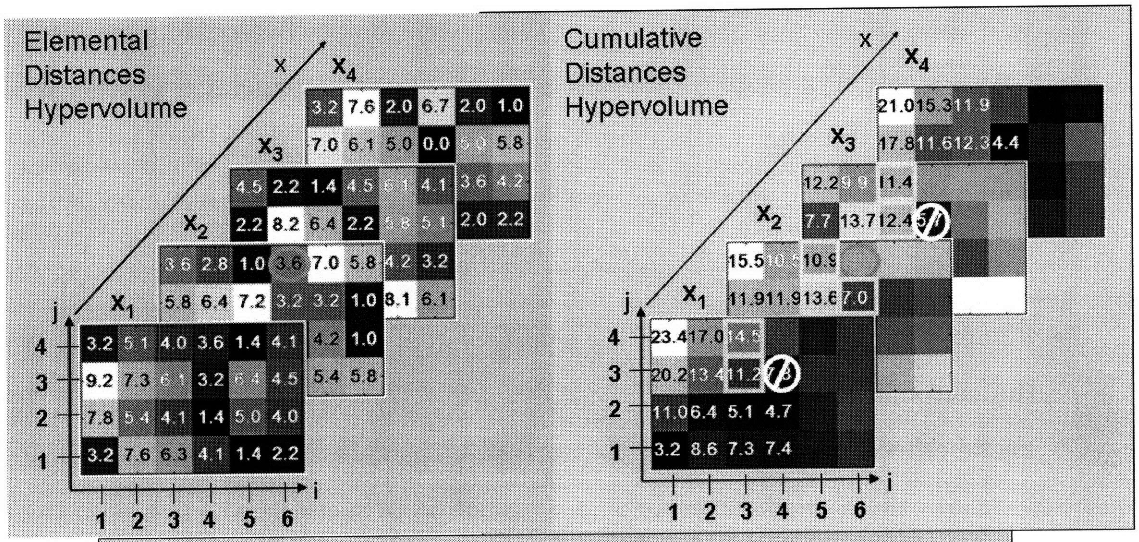
- Figure 4-8. The remainder of the hypervolume, D_c,is populated according to the continuity rules. For $D_c(i = 4, j = 4, x = 2)$ we indicate with boxes the neighborhood locations in D_c that are valid predecessors; we contra-indicate locations that would correspond to a transition in space (of the test sequence) without a transition in time (in the test sequence).

- Figure 4-9. The cumulative distances hypervolume is shown as a volume, and flattened so that every IJ plane is visible. The minimum cost warp distance is 7.4.

- Figure 4-10. The minimum cost warp path through D_c is indicated.

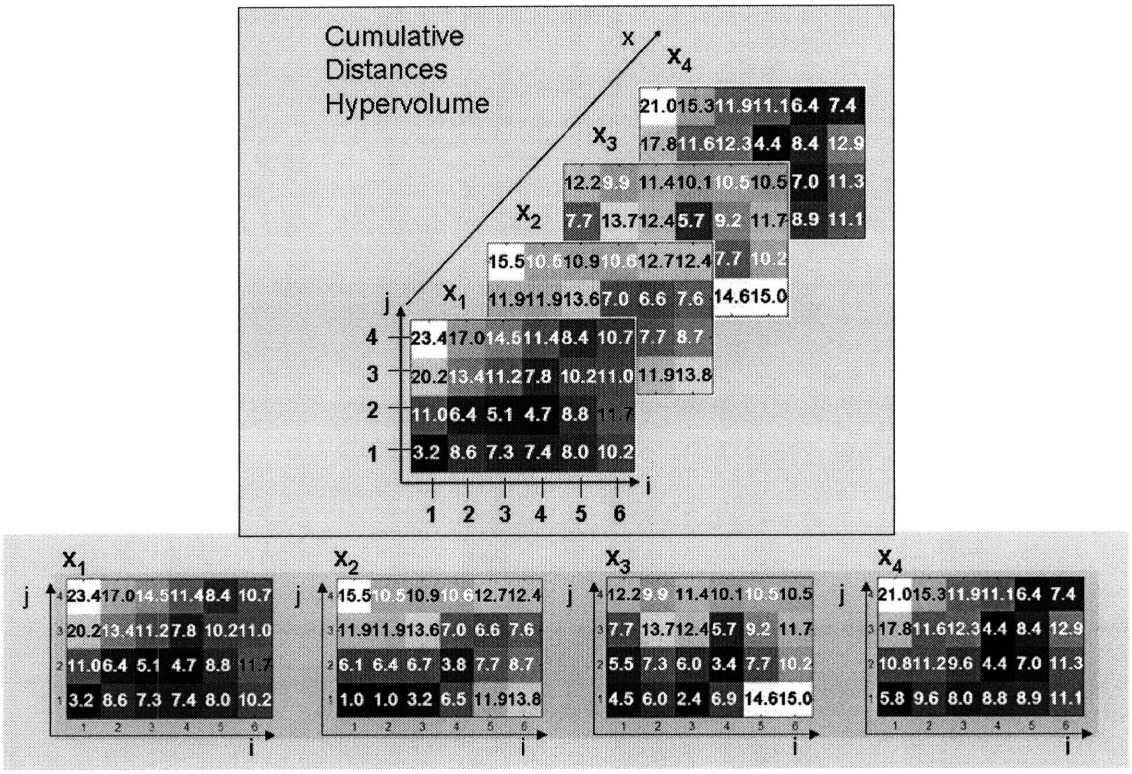
- Figure 4-11. The corresponding warped template, Q^*_W and the found template, C^*_W are shown.

- Figure 4-12. The temporal component of the optimal path, the position component of the optimal path, and the local and cumulative distances along the optimal path are shown.



$$D_c(i=4, j=4, x=2) = 3.6 + \min\begin{cases} 1\times(14.5, 10.9, 11.4) \\ 1\times(11.2, 13.6, 12.4) = 10.6 \\ 1\times(7.0) \end{cases}$$

Figure 4-8: Calculating the Cumulative Distances Hypervolume for a Simple Example of DTSW. D_c,is populated according to the continuity rules. For $D_c(i = 4, j = 4, x = 2)$ we indicate with boxes the neighborhood locations in D_c that are valid predecessors; we contra-indicate locations that would correspond to a transition in space (of the test sequence) without a transition in time (in the test sequence).



x_1

j \ i	1	2	3	4	5	6
4	23.4	17.0	14.5	11.4	8.4	10.7
3	20.2	13.4	11.2	7.8	10.2	11.0
2	11.0	6.4	5.1	4.7	8.8	11.7
1	3.2	8.6	7.3	7.4	8.0	10.2

x_2

j \ i	1	2	3	4	5	6
4	15.5	10.5	10.9	10.6	12.7	12.4
3	11.9	11.9	13.6	7.0	6.6	7.6
2	6.1	6.4	6.7	3.8	7.7	8.7
1	1.0	1.0	3.2	6.5	11.9	13.8

x_3

j \ i	1	2	3	4	5	6
4	12.2	9.9	11.4	10.1	10.5	10.5
3	7.7	13.7	12.4	5.7	9.2	11.7
2	5.5	7.3	6.0	3.4	7.7	10.2
1	4.5	6.0	2.4	6.9	14.6	15.0

x_4

j \ i	1	2	3	4	5	6
4	21.0	15.3	11.9	11.1	6.4	7.4
3	17.8	11.6	12.3	4.4	8.4	12.9
2	10.8	11.2	9.6	4.4	7.0	11.3
1	5.8	9.6	8.0	8.8	8.9	11.1

Figure 4-9: Cumulative Distances Hypervolume for a Simple Example of DTSW.

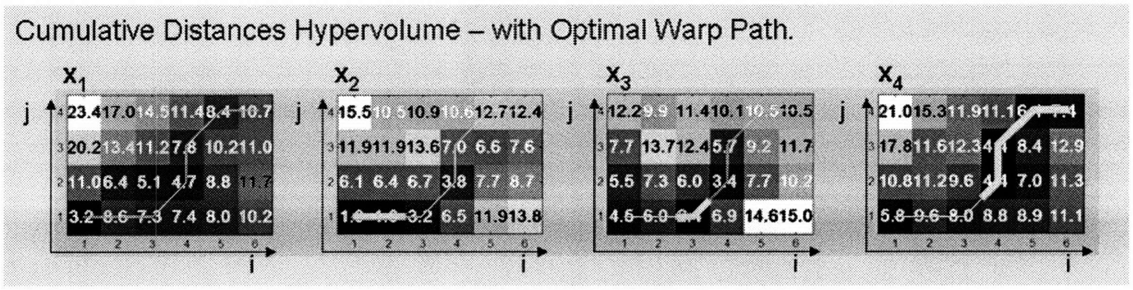
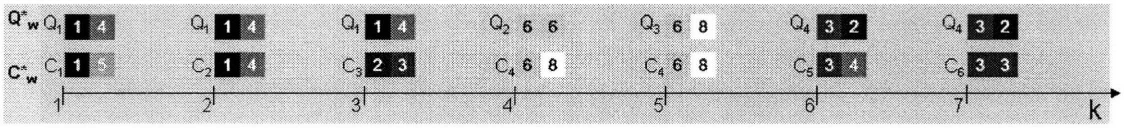


Figure 4-10: Cumulative Distances Hypervolume for a Simple Example of DTSW with Optimal Warp Path.



k	1	2	3	4	5	6	7
Q^*_w	Q_1 1 4	Q_1 1 4	Q_1 1 4	Q_2 6 6	Q_3 6 8	Q_4 3 2	Q_4 3 2
C^*_w	C_1 1 5	C_2 1 4	C_3 2 3	C_4 6 8	C_4 6 8	C_5 3 4	C_6 3 3

Figure 4-11: The Optimally Warped and Found Template.

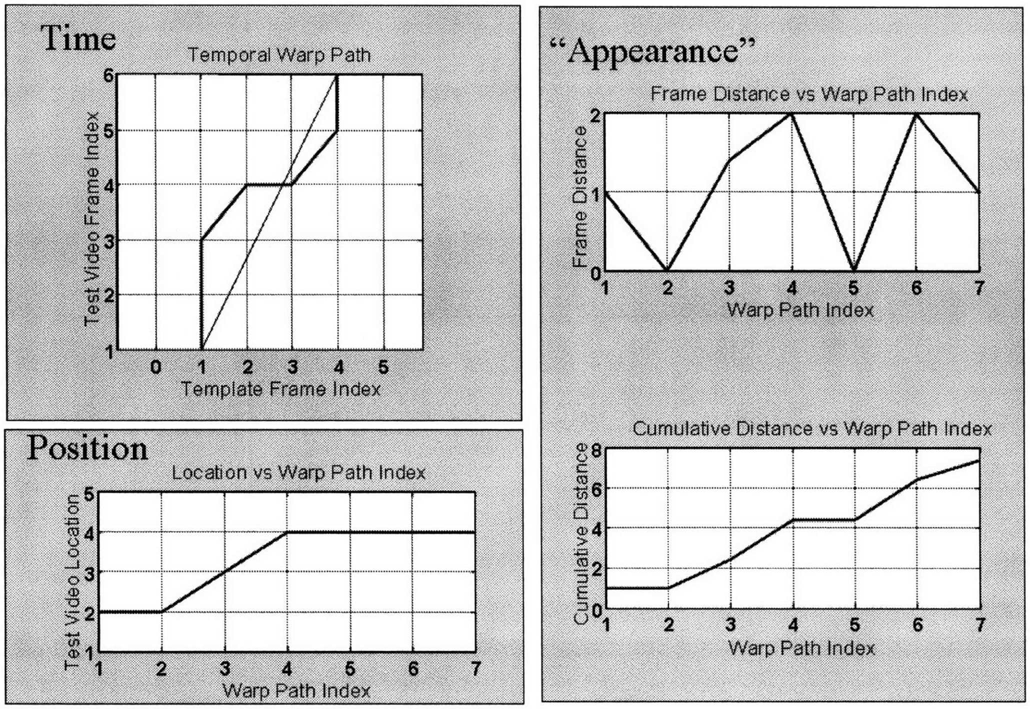


Figure 4-12: DTSW Algorithm Output. The temporal component of the optimal path, the position component of the optimal path, and the local and cumulative distances along the optimal path are shown.

4.9 Implementation

We evaluate the DTSW algorithm on several real world examples in Chapters 5 and 6. Figures 6-3 and 5-3 show the warped output sequences for the two examples that we discussed previously, the heart-valve, and the karate punch. We now explore the structure of the DTSW algorithm and discuss how DTSW is efficiently implemented.

4.9.1 Filterbanks for Determining Elemental Distances Hypervolume

As we discussed, the elemental distances hypervolume, D_L, is efficiently calculated through a series of linear filter operations plus simple element to element additions and multiplications. The filter kernels of one of the filterbanks that produces D_L are the frames of the template video.

4.9.2 Reduced Dimensionality Filterbanks for Approximating the Elemental Distances Hypervolume

For smoothly varying template videos we observe and generally expect that many of the frames of the sequence, that are close to one another in time, will be very similar. This suggests that we attempt to represent the template sequence with a reduced set of "canonical" frames that can be linearly combined with reconstruction coefficients to closely approximate the original. We first find an exact but alternative representation of the template sequence. We then approximate the original sequence by using only those "canonical" frames that capture the majority of the variation in the original sequence. The canonical frames are then used as filters.

Consider a range of (vectorized) template frames, $[Q_j]$, to be a set of observations vectors, $\{y_j = [Q_j], j = 1..R\}$ in an $N_Q \times M_Q$ dimensional space. We compute an orthonormal coordinate system B that is optimally aligned with the variation in the observations [frames]. The columns of B are the orthonormal basis vectors arranged from most dominant axis to least. We project each template frame into this new coordinate system, $c_j = B^T y_j$. That is, we determine the scalar projection coefficient of a template frame along each axis of the coordinate system defined by B.

We can express, with no loss of information, each observation [frame] as a linear combination of the basis vectors.

$$
\begin{aligned}
y_j &= BB^T y_j && (4.19)\\
y_j &= B(B^T y_j) && (4.20)\\
y_j &= Bc_j && (4.21)\\
y_j &= \sum_{i=1}^{J} b_i c_j(1) && (4.22)\\
y_j &= b_1 c_j(1) + b_2 c_j(2) + \ldots + b_R c_j(R) && (4.23)
\end{aligned}
$$

We approximate each frame and therefore a range of frames in template sequence by truncating the summation at $j = P, P \leq R$.

$$
\begin{aligned}
y_1 &\approx b_1 c_1(1) + b_2 c_1(2) + \ldots + b_P c_1(P)\\
y_2 &\approx b_1 c_2(1) + b_2 c_2(2) + \ldots + b_P c_2(P)\\
&\ \ \cdot\\
&\ \ \cdot\\
y_R &\approx b_1 c_R(1) + b_2 c_R(2) + \ldots + b_P c_R(P)
\end{aligned}
\tag{4.24}
$$

Now, instead of filtering the test sequence with every frame of the template sequence we filter it with the truncated set of basis frames. The outputs from these basis filters are then linearly combined using the projection coefficients to approximately determine the terms in the elemental distance hypervolume. This approximate filter-bank implementation is depicted in Figure 4-13.

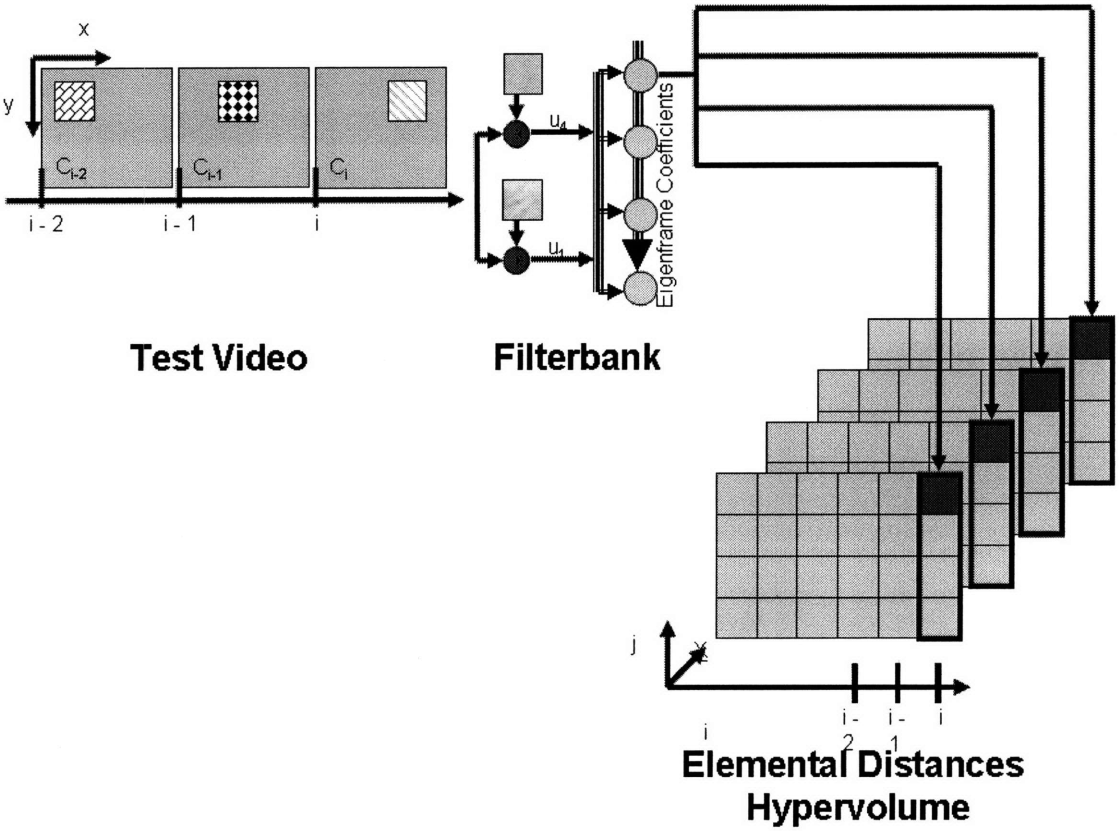


Figure 4-13: Filter Bank Implementation - Eigenframes as Filter. Instead of filtering the test sequence with every frame of the template sequence we filter it with the truncated set of basis frames. The outputs from these basis filters are then linearly combined using the projection coefficients to approximately determine the terms in the elemental distance hypervolume.

4.9.3 Principal Component Analysis (PCA) for Eigenframe Filter Determination

The intent is to determine a reduced set of canonical frames, the principal components or basis vectors, that can be used to approximately "represent" the original set of frames. The principle components are the eigenvectors of the covariance matrix of the frames of the template sequence.

Consider the set of the frames $\mathbf{q} = \{q_1, q_2, q_3, ..., q_j, ...q_R\}$, a range of frames from the template sequence, as a set of observation vectors. PCA determines an orthonormal basis set of the covariance of these observation vectors. The orthonormal bases are the principal axes or eigenvectors of the covariance matrix of the observation vectors. The first principal component captures the majority of the variation in the observations, and the second principal component captures the second largest amount of variance, etc. We call these principal components the Eigenframes, to be consistent with the naming conventions that are frequently used in the literature.

The maximum number of the principal components is identically the number of samples in the original set of observations, R. The original frames can be identically recovered as a linear combination of all R Eigenframes. We approximate the original observation set, $\mathbf{Q}$, with a minimal but sufficient number of the best Eigenframes. The hope is that the number of Eigenframes required to capture 90%, for example, of the variance in the original observation set is far less than J. We expect this to be the case for the majority of the template sequences encountered in practice, even those that evolve rapidly over time.

The reduced set of Eigenframes that capture the majority of the variance in the observation set will be the filters in a more efficient filter-bank. The size of the Eigenframe FB should, in general, be much smaller than the original full frame sequence filterbank; the amount of computation is decreased accordingly. The outputs of the Eigenframe filter bank are combined to approximately determine what the output would have been had the filter bank consisted of every frame in the template sequence.

Calculating the Eigenframes

Each frame from a range of frames from the sequence $\mathbf{Q}$ is an observation in a high dimensional space, an $N_Q \times M_Q$ space, if we are considering frames of pixel intensities, higher if we consider frames of optical flow values or various moments of either. The average frame is given

$$\bar{Q} = \frac{1}{R}\sum_{j=1}^{R} Q_j \tag{4.25}$$

The mean removed frames are given

$$\hat{Q}_j = Q_j - \bar{Q} \quad j = 1..R \tag{4.26}$$

We now determine the principal components of these zero mean observations. We seek a set of R orthonormal vectors, $\mathbf{u_j}$, and associated eigenvalues λ_j which are aligned to the energy distribution of the observations. The vectors $\mathbf{u_j}$ and scalars λ_j are the eigenvectors and eigenvalues of the covariance matrix of the zero mean observations. The observations are arranged as $1 \times NM$ dimensional vectors. This is functionally noted with the square brackets [ ].

$$\begin{aligned} C &= covariance(\hat{\mathbf{q}}) && (4.27)\\ C &= \frac{1}{R}\sum_{j=1}^{R}[\hat{q}_j][\hat{q}_j]^T && (4.28)\\ C &= AA^T && (4.29) \end{aligned}$$

.

The dimension of C is normally quite large. If, for example, each frame of the template sequence is 100 pixels by 100 pixels, then C will be a 10000×10000 matrix. This is too large to find the eigenvectors numerically. The first R eigenvalues of this matrix are useful; the remainder are in the null space of the covariance matrix and will be zero.

In order to calculate the eigenvector and eigenvalues of C efficiently we rely on the fact that the eigenvalues of a matrix A^TA and first J eigenvalues of AA^T are identical, and that if e is an eigenvector of A^TA then Ae is an eigenvector of AA^T. In order to demonstrate these statements consider the following.

Let e_i be an eigenvector of A^TA, whose eigenvalue is λ_i.

$$\begin{aligned}(A^TA)e_i &= \lambda_i e_i && (4.30)\\ A(A^TA)e_i &= A(\lambda_i e_i) && (4.31)\\ (AA^T)(Ae_i) &= \lambda_i(Ae_i) && (4.32)\end{aligned}$$

Therefore if e_i is an eigenvector of A^TA, then Ae_i is an eigenvector of AA^T with the same eigenvalue.

The J eigenvectors form the complete basis. We renumber the eigenvalues from largest, 1, to smallest, R. We correspondingly order the eigenvectors as column vectors in a matrix from largest to smallest eigenvalue. The magnitude of the eigenvalues indicate the relative amount of energy along the corresponding eigenvector. We select the amount of the variance that we want to retain in the approximation and then determine the number of eigenvectors to retain by computing the smallest integer P such that :

$$\text{Percentage of Retained Variance} \leq \frac{\sum_i^P \lambda_i}{\sum_i^R \lambda_i} * 100. \qquad (4.33)$$

The basis vectors $b_j, j = 1..P$ of Equation 4.24 are the ordered eigenvectors.

4.9.4 Eigenframe Filters - Temporal Clustering

The simplest set of frames to consider for PCA decomposition is the entire template sequence. However, doing that ignores the natural temporal clustering that is likely present in the sequence. We fully expect two frames that are close in time to be similar. We expect that template frames far removed in time to be relatively dissimilar.

The PCA approximation minimizes the error when averaged over all observations. Approximating the template sequence using every frame as an observation minimizes the spatial error of the approximation *averaged over all observations.* Some frames will be better approximated than others. Given the expected natural temporal evolution of the video sequence that means that some temporal ranges of frames will be better approximated than other temporal ranges.

It seems best to uniformly distribute approximation errors throughout all frames of the template sequence so that no one temporal range of the template is exactly approximated at the expense of another range's poor approximation.

A reasonable approach is to identify temporal ranges of frames that are similar using a technique like spectral clustering (k-means). We then approximate a temporal range of the frames using PCA, capturing the same percentage of the variance in each temporal cluster. In this way, we equalize the approximation error over all frames and use as few Eigenframes as possible to approximate the template sequence.

Also note that independent of considering time in the determination of the Eigenframes, it is often advantageous to *whiten* each observation [frame] by dividing by its norm after removing the mean frame. This equalizes the single observation's variance to 1. The reconstruction must therefore undo this equalization - unless of course we choose to transform all frames to be zero mean unit variance, this choice will be application specific.

4.9.5 Non-linear Filters for Recursing through the Elemental Distances Hypervolume

The DTSW algorithm is written in pseudo code in Algorithm 4.9.1. This implementation is illustrated in Figure 4-14.

We simplify the statement of the DTSW algorithm by recognizing that the inner most loops, *Find Minimum Previous Sections 1 and 2*, over x_p and y_p for a given i_p and j_p can be "simultaneously" found for all x and y by applying a Non Linear Minimum Filter to the $(i_p, j_p, x \in (0..X-1), y \in (0..Y-1))$ plane[1]of the D_C hypervolume. The filter kernel, H, in this situation is all 1 and is of size $(dx*2+1) \times (dy*2+1)$. This is done on the "preceding edge" at $i-1$ for all j. We store the minimum filtered cumulative distance.

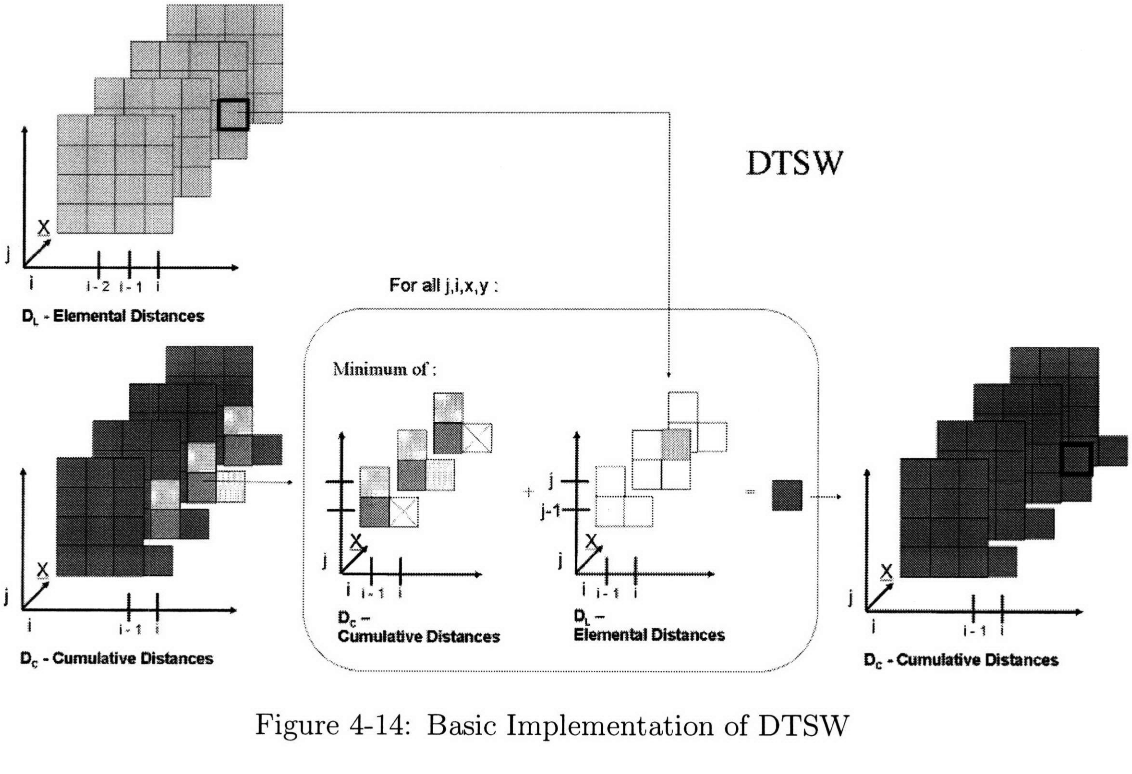


Figure 4-14: Basic Implementation of DTSW

[1] We will use the shorthand notation $\{x\}$ to mean $x \in (0..X-1)$. For example $(i_p, j_p, x \in (0..X-1), y \in (0..Y-1))$ will be written $(i_p, j_p, \{x\}, \{y\})$.

```
Algorithm 4.9.1: DTSW($Q, C$)

$D_C \leftarrow zeros(I, J, X, Y)$
for $i \leftarrow 0$ to $I - 1$
  do
    $Cframe = Csequence(j);$
    for $j \leftarrow 0$ to $J - 1$
      do
        $Qframe = Qsequence(j);$
        for $x \leftarrow 0$ to $X - 1$
          do
            for $y \leftarrow 0$ to $Y - 1$
              do
                $d_{min} = \text{inf};$
                comment: Find Minimum Previous, Section 1
                do
                  $(i_p, j_p) \leftarrow (i - 1, j)$
                  for $x_p \leftarrow (x - dx)$ to $(x + dx)$
                    do
                      for $y_p \leftarrow (y - dy)$ to $(y + dy)$
                        do
                          $d_{min} \leftarrow$
                          $min(d_{min}, D_C(i_p, j_p, x_p, x_p);$

                comment: Find Minimum Previous, Section 2
                do
                  $(i_p, j_p) \leftarrow (i - 1, j - 1)$
                  for $x_p \leftarrow (x - dx)$ to $(x + dx)$
                    do
                      for $y_p \leftarrow (y - dy)$ to $(y + dy)$
                        do
                          $d_{min} \leftarrow$
                          $min(d_{min}, 2 * D_C(i_p, j_p, x_p, x_p);$

                comment: Find Minimum Previous, Section 3
                do
                  $(i_p, j_p) \leftarrow (i, j - 1)$
                  $x_p \leftarrow x$
                  $y_p \leftarrow y$
                  $d_{min} = min(d_{min}, D_C(i_p, j_p, x_p, x_p);$
                $Csmall = /Cframe(x, y)/;$
                $d_e = d(Qframe, Csmall);$

                comment: Optional, other distance penalites.
                $d_{other} = function(i, j, x, y);$

                $D_C(i, j, x, y) = d_e + d_{other} + d_{min};$
return $(D_C(I, J, X, Y))$
comment: Optional:  Return entire cumulative distance hypervolume
```

We re-order the *for* loops around Section 3 and the summation of the elemental distance with the minimum found predecessor. We recognize that the leading edge of the D_L can be found by applying a Minimum Accumulation Filter on every ij plane of the "preceding edge" of the minimum filtered cumulative distance hypervolume appended with the "leading edge" of the elemental distance hypervolume. The major steps of the DTSW algorithm are now explained as follows.

For a new test frame, C_i:

- Run a Minimum Filter on every xy plane at the preceding edge of D_C to generate of D_{CMF} (i.e. for $(i-1), j \in (0..J-1)$.

- Calculate the elemental distances volume between the video Q and the frame C_i. This is the leading edge of D_L.

- Append this volume to the preceding edge of D_{CMF}.

- Run a Bi-Casual Minimum Accumulation Filter on this appended D_{CMF} for every ij plane (i.e. for all x,y)

The Minimum Filter filter kernel, H, is all 1 and is of size $(dx*2+1) \times (dy*2+1)$. The Bi-Casual Minimum Accumulation Filter filter kernel, A, is $A(0,0) = 2, A(0,1) = 1, A(1,0) = 1, A(1,1) = 0$.

The DTSW algorithm written in this way is outlined in pseudo code in Algorithm 4.9.2. The implementation is illustrated in Figure 4-15.

```
Algorithm 4.9.2: DTSW($Q, C$)

  $D_C \leftarrow zeros(I, J, X, Y)$
  for $i \leftarrow 0$ to $I - 1$
    do
      $C_i = Csequence(i)$;
      $D_L(i, \{j\}, \{x\}, \{y\}) = LinearFilterBank(\mathbf{Q}, C_i)$
      for $j \leftarrow 0$ to $J - 1$
        do
          $D_{CMF}(i-1, j, \{x\}, \{y\}) = MF(D_C(i-1, j, \{x\}, \{y\}), H)$;
      for $x \leftarrow 0$ to $X - 1$
        do
          for $y \leftarrow 0$ to $Y - 1$
            do
              $D_C(i, \{j\}, x, y) = MA([D_{CMF}(i-1, \{j\}, x, y)\ \ D_L(i, \{j\}, x, y)], A)$
  return $(D_C(I, J, X, Y))$
  comment: Optional:  Return entire cumulative distance hypervolume
```

The DTSW algorithm structured in this way can be implemented using parallel processing. Each Minimum Filtering operation on every xy plane of the preceding edge of D_C is performed independently, as is the Minimum Accumulation Filter on every ij plane of D_{CMF} appended with the leading edge of D_L. We can use one processor for every xy plane and one processor for every ij plane. Also, the generation of D_L can be performed in parallel; use one processor for every linear-filter in the filterbank. In monitoring applications when fast decisions are necessary, the system hardware is easily scaled because of this parallel structure. The computational complexity is further discussed in Chapter 8.

The simple example that we used previously is used to illustrate the algorithm in this form in Figure 4-16.

4.10 Discussion

Various extensions and modifications can be made to the basic DTSW algorithm. We commonly constrain temporal warps path to stay within a tolerance of proportional

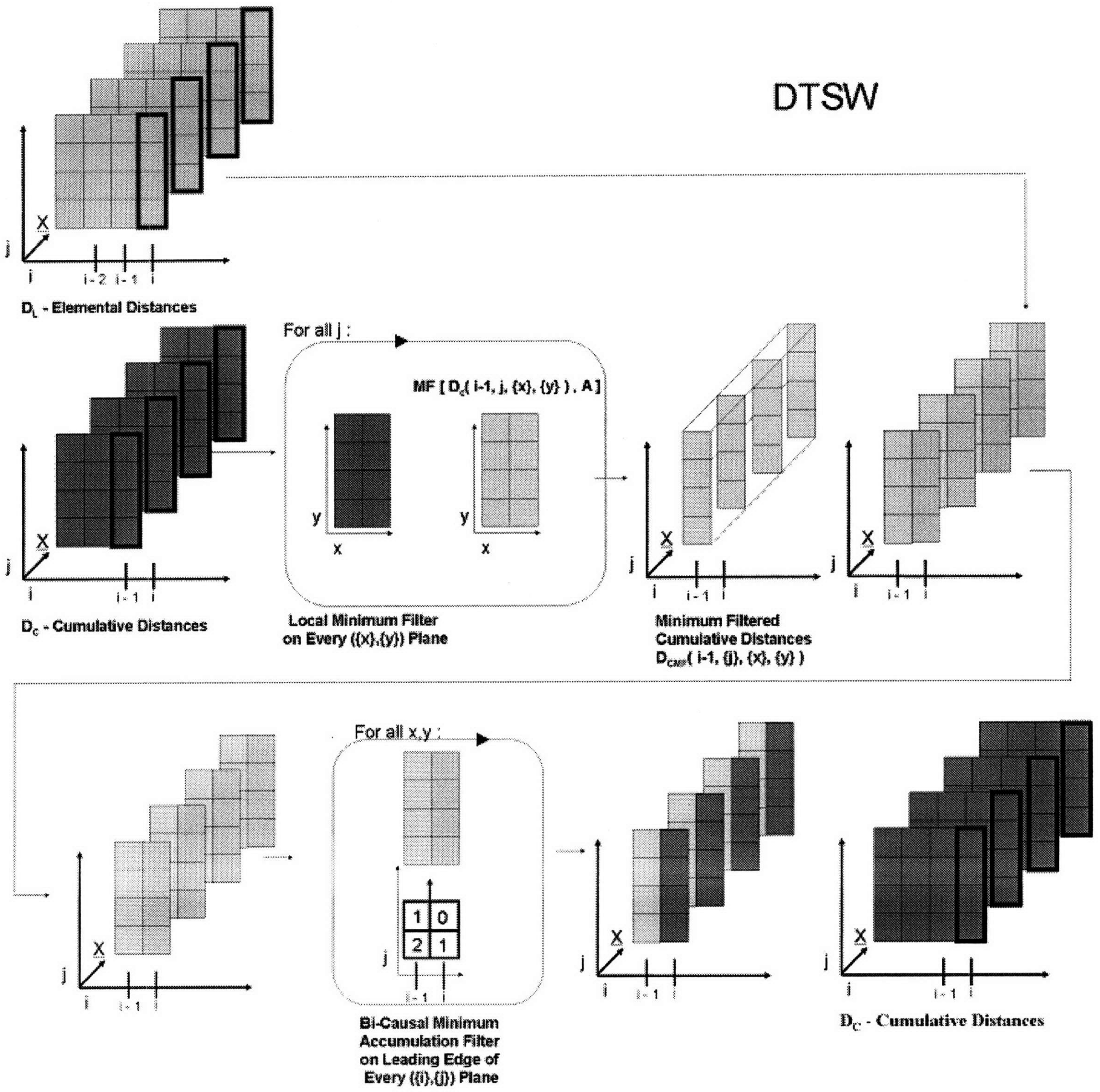
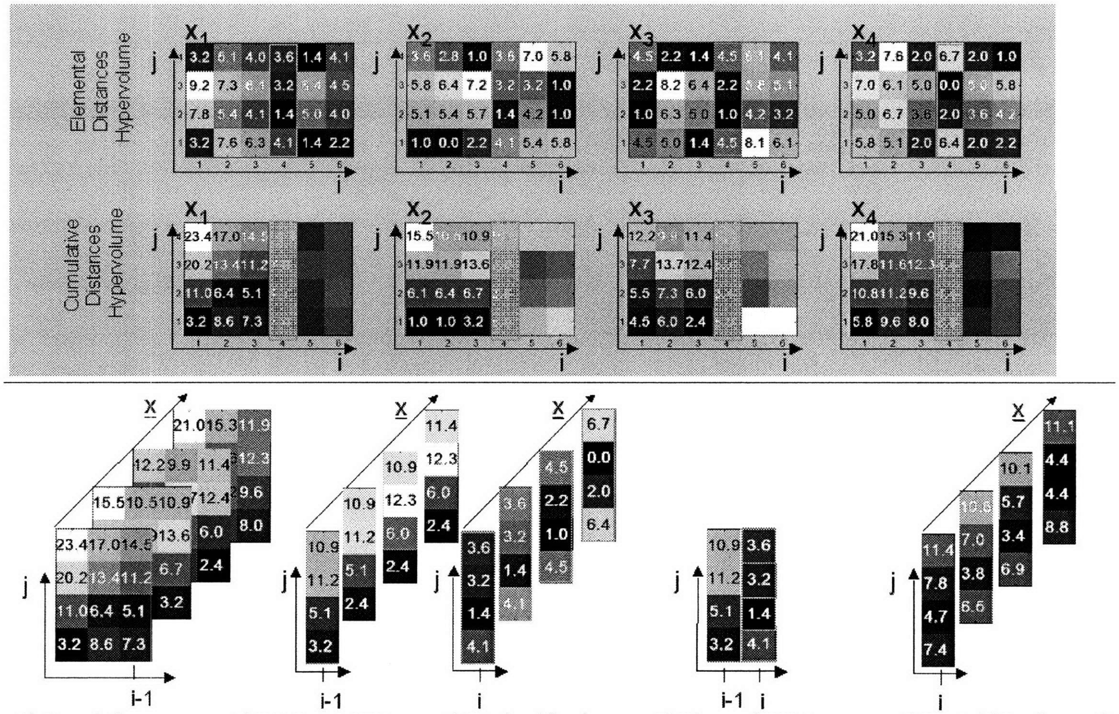


Figure 4-15: Parallel Implementation of DTSW


Elemental Distances Hypervolume
Cumulative Distances Hypervolume
x1
x2
x3
x4
j
i
x
i-1

0) Cumulative Distances Hyptervolume

1) **Minimum Filter** on Spatial Planes

2) Stack with sub-volume from **Elemental Distances**

3) Bi-causal **Minimum Accumulation Filter** on **Temporal Planes**

4) Next "ith" volume of the Cumulative Distances Hypervolume

Figure 4-16: Simple Example Demonstrating Parallel Implementation of DTSW

scaling. We also adjust the way that the elemental distances are accumulated at the start of the algorithm in order to better accommodate error in the pre-DTSW temporal segmentation.

4.10.1 Constraints

The spatial continuity constraint is implemented by the filter kernel used for Minimum Filter operation on every xy plane. By modifying the filter kernel we make some spatial transitions more or less favorable and control the spatial smoothness of the template video alignment, or fit, to the test. The filter kernel used for the Minimum Accumulation Filter implements temporal continuity. Again, by modifying the kernel we control the preferred and allowable temporal transitions. Temporal causality can be relaxed by modifying the way that the Minimum Accumulation Filter is applied.

For industrial application, deviation beyond the expected or normal is flagged as an error; once we have deviated beyond an extreme it is not necessary to continue. Often we know that the template and test are very similar. In these situations, we do not have to fully populate either D_L or D_C. We expect temporal alignments that are close to diagonal and spatial alignments that are close to the locations from where the template was extracted from its source. These two things allow us to dramatically reduce computation. We would not, for example, consider matching a frame at the beginning of one sequence to a frame at the end of the other sequence. Also, we can modify the algorithm to follow a nominal temporal and spatial path and to align the two videos in a neighborhood about that path.

It might be important to prevent long "flat" paths, where a frame of either video is fit to a wide temporal range of frame in the other video. This is useful if the template or test has a temporal range of frames that are nearly identical. This would happen if the subject matter were stationary for a few moments of its complete event cycle.

We have neglected to consider that the first and last frames of the two videos might not be the best place to start and terminate the video alignment. There may be some temporal segmentation "error" at either the beginning or the end of the test video, if for example the external sensors that provide the temporal clipping are a little off. Examining multiple last frame alignments is straightforward and almost automatic. The cumulate distance hypervolume already contains the DTSW distance and alignments for videos that are temporally shorter. Instead of looking only at the upper-right edge of D_C, $D_C(I, J, \{x\}, \{y\})$, we examine the top face [2]of D_C for test videos that are one to several frames shorter $D_C(i < I, J, \{x\}, \{y\})$. The DTSW distance to all of these points may be compared after normalization - i.e. Divide by sum of the temporal length of the template, J, and test video, $i < I$.

Several options exist to handle variation at the beginning of the video. The first option is to simply allow the DTSW algorithm to run its course. Frequently what happens is that the algorithm will match the first frame (or several) of the template to all of the "extra" frames at the beginning of the test video. The accumulated distance and potentially erroneous matches at the beginning of the video may force some deviation away from the correct path at its actual onset - once the true temporal start of the event is reached. This can be accounted for by subtly modifying the DTSW algorithm. We can handle temporal variation at beginning by reducing or eliminating the accumulation of distance for moves along $j = 0$, for some range of $i, i = 0..i_b$. After we determine D_C, we find the warp path from the temporal end to the temporal beginning; we stop as soon as the path gets to any point where $j = 0, i \leq i_b$.

4.10.2 Small Temporal Variation

Some additional intuition for the structure of the DTSW algorithm can be obtained if we assume for a moment that the temporal lengths of the template and test are identical and that we don't allow any temporal variation between the two sequences. This simplification might be appropriate if the temporal axis of the template and the test vary only slightly.

This relaxation may be used as the first approximation before full DTSW or it may be employed for the detection problem of Chapter 7. If temporal deviation is

[2] We do expect that the template is well clipped, as the template is a user selected video, but if not, we can also examine $D_C(I, j < J, \{x\}, \{y\})$.

necessary, we can employ a multiple stage strategy. First, approximate an alignment ignoring temporal warping, and then pass a segmented video block surrounding the detected location to the DTSW algorithm for detailed refinement. In detection applications we are generally interested in an approximate temporal and spatial localization of an event, not its detailed spatial and temporal evolution; we may be able to ignore temporal variation. Time warping is not necessary or desired for applications where we indeed want to limit detection of events to those that are temporally similar.

If the time axis between template and the test are the same then the DTSW distance can be approximated as the Fixed Time Distance, D_{FT}. In this example we use the Euclidean distance for the low-level frame-distances. The Fixed Time Distance, D_{FT}, is almost the three-dimensional Euclidean distance. The difference is that the summation over time is interrupted each addition by a non-linear minimum filter applied to the time accumulated xy plane.

$$\begin{aligned} &D_{FT}(x,y,t|Q,C) = \\ &\quad \textstyle\sum_{k=0}^{T-1} MF_{xy} \left[\sum_{m=0}^{M_Q-1} \sum_{n=0}^{N_Q-1} (\ Q(m,n,k) - C(y+m, x+n, t+k)\ )^2 \right] \end{aligned} \tag{4.34}$$

If we also ignored spatial variation of an event, then the problem is trivial. By ignoring both temporal and spatial warping, the problem of video comparison is nothing other than a three-dimensional normalized correlation of the template video stream with the test video stream or a straight Euclidean distance.

4.11 Summary

Dynamic Time and Space Warping compares two videos - a template, **Q**, and a test, **C**, by finding the optimal path of the template video through the time and space of the test video. Dynamic Time and Space warping finds the optimal, lowest cost path through a hypervolume of elemental distances subject to several constraints.

The elemental distance hypervolume is efficiently found by first representing the template video as a series of Eigenframes and then filtering the test video with the Eigenframes. The DTSW distance (a cumulative distance) is found via an iterative application of a Local Minimum Filter and a Bi-causal Minimum Accumulation Filter on the elemental distances hypervolume. The cumulative distances hypervolume is examined to determine the optimal warp path aligning the two videos. The structure of the algorithm is such that it can be easily performed with parallel processing. The algorithm can be relaxed to accommodate variation in the temporal segmentation of the original videos, or it can be modified if we do not expect or want to allow temporal warping.

Chapter 5

Evaluation, Non-industrial Applications

5.1 Introduction

There are interesting applications of DTSW to "systems" that are neither industrial nor purely scientific in nature. An appearance-based comparison of videos will allows us to either supplement or replace application-specific model-based analyses.

We believe that DTSW is an enabling technology for improved sports training and physical therapy. As an example, we use a karate instructor attempting to perfect and teach a move. A video, the template, of the ideal move is captured and stored. A video of the same move is later captured. We simultaneously align the two videos and quantify their local similarity. Armed with detailed similarity to the "perfect" move, the instructor can guide a student.

We present an application for DTSW that can automate a (human) labor intensive process of motion-capture video and human-in-the-loop analysis. Horse-gait analysis is important to the business of thoroughbred racing. We use DTSW to compare low-resolution videos of horses. The horses are different colors, have different jockeys, features of interest such of feet and shoulders are occluded and of insufficient resolution to be tracked.

5.2 Human Biomechanics, Physical therapy

A professional karate instructor trains to eliminate variability from the various moves that he performs in the practice of karate. The instructor is recorded performing a punching move. As a source for the template, we use a sequence where he feels that he was as close to perfect as possible in the execution of the move. A few frames of this video are depicted in Figure 5-3. A subregion of the instructor is localized in every frame that includes his torso, head, and arms when they are close to his body. We use this reduced spatial region as the template, **Q**, for the DTSW algorithm. Other templates are possible that would expose more subtle variations of the move, such as focusing just on the motion of an arm or the torso alone. The selection of the head, torso, arm, indicate that we are interested in examining the aggregate move.

Subsequently the instructor uses videos of himself and his students performing the same move. The video clips alone give him some understanding for the when and where a performed move deviates from the ideal. A comparison of the ideal template video with the subsequent video clips using the DTSW algorithm provides a level of detail that allows him to perfect both his and his students moves.

This has obvious application in baseball pitching, golf, diving, ice skating, and other sports where it is important to learn an ideal motion and then learn to perform that motion repeatedly.

The first few Eigenframes (filters for the filterbank implementation) are shown in Figure 5-1.

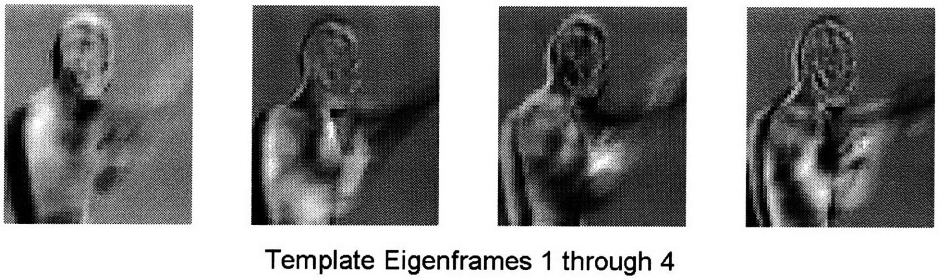


Figure 5-1: Eigenframes for the Karate Punch Template Video

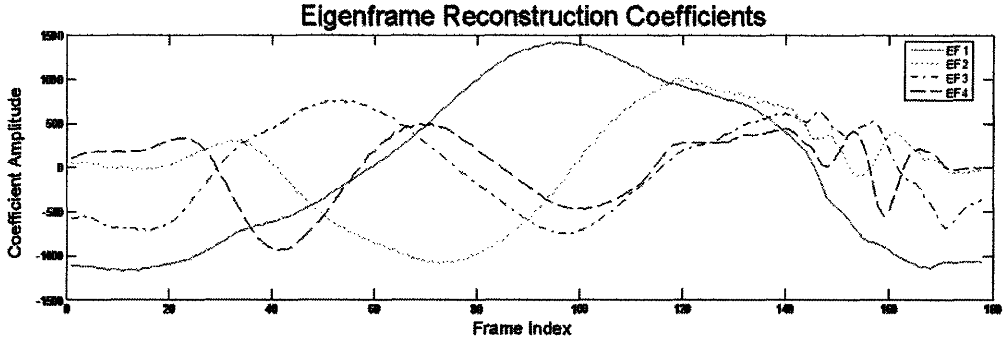


Figure 5-2: Eigenframes Reconstruction Coefficients for the Karate Punch Template Video

5.2.1 DTSW Algorithm

The template and test videos are visually similar but are of different duration, and the event transpires at different spatial locations in the video. Figure 5-3 shows a few frames of the input and output sequences.

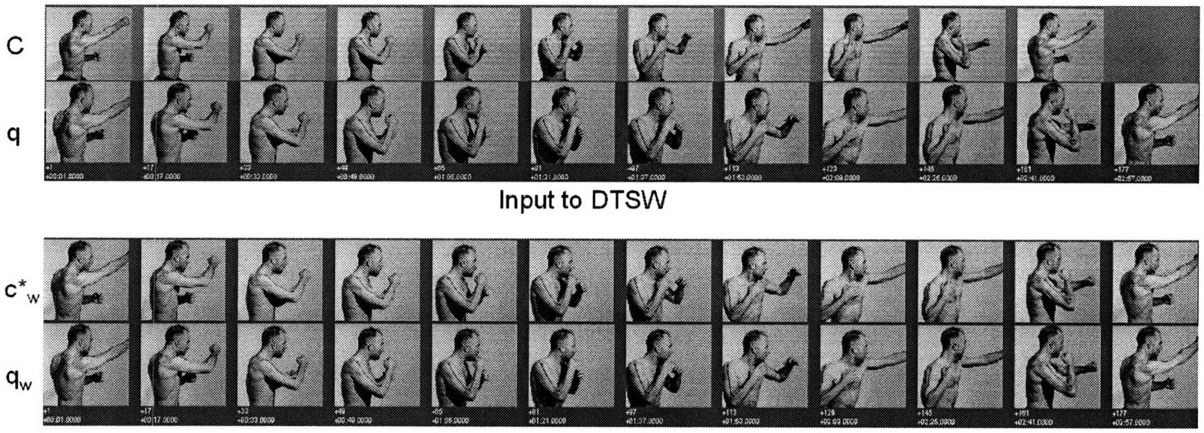
Figure 5-4 shows a flattened representation of a few planes from the Elemental Distance Hypervolume along with the optimal warp path. Figure 5-5 shows a flattened representation of a few planes from the Cumulative Distance Hypervolume along with the optimal warp path. The solid black line is the temporal projection of the warp path onto each spatial plane. The yellow dots indicate if the spatial component of the warp path is in the indicated spatial plane.

Figure 5-6 shows the temporal and spatial localization as a function of warp path.

Figure 5-7 shows the spatial and temporal difference (error in some contexts) between the template and test versus time of the template video.

Figure 5-8 shows the cumulative and average differences (error in some contexts) between the template and test versus time of the template video.

Figure 5-9 shows the cumulative and elemental squared distances along the warp path. A Euclidean distance between pixels normalized by the size of the template frame was used for this test. The elemental distance can therefore be interpreted as the average squared difference between a pixel from the template and a pixel from



Input to DTSW

Output from DTSW

Figure 5-3: DTSW Input and Output Sequences for the Karate Punch Example. Test, $\mathbf{C}$, and template, $\mathbf{Q}$, input video sequences are shown in the top of the figure. $\mathbf{C}$ is a video acquired why practicing the punching move. $\mathbf{Q}$ is a video of the perfect punch. • DTSW temporally warps and spatially shifts $\mathbf{C}$ and $\mathbf{Q}$ creating output sequences that have minimum temporal and spatial variation - maximum similarity - throughout the sequence. The bottom two strips show the videos - $\mathbf{C}^*_\mathbf{w}$ and $\mathbf{Q}_\mathbf{w}$ - output from the DTSW algorithm. The frames of $\mathbf{C}$ are spatially larger than the frames of $\mathbf{Q}$. The frames of $\mathbf{C}^*_\mathbf{w}$ are the same size as the frames of $\mathbf{Q}$ or $\mathbf{Q}_\mathbf{w}$. Each frame of $\mathbf{C}^*_\mathbf{w}$ has been extracted from a frame of $\mathbf{C}$ to locally match to a frame of $\mathbf{Q}$. The output videos have a common time axis. The common time axis is, generally, different from either of the time axes of the input videos. Output videos temporally aligned to either input video can be obtained by projecting the warped time axis onto either input time axis. • As defined by the constraints of DTSW, the temporal evolution of the output videos along their common time axis is maximally similar. Also, as defined by the constraints of DTSW, the spatial arrangement of information in each aligned frame-pair of the output sequences is maximally similar. This can be seen by studying the video strips at the bottom of the figure. This overall similarity is best understood when one views the two input videos played side-by-side followed by the two output video played side-by-side. The output videos are visually dramatic and very appealing to the intelligent viewer. The warp-path or overall similarities, both temporal and spatial, are used for automated process-monitoring.

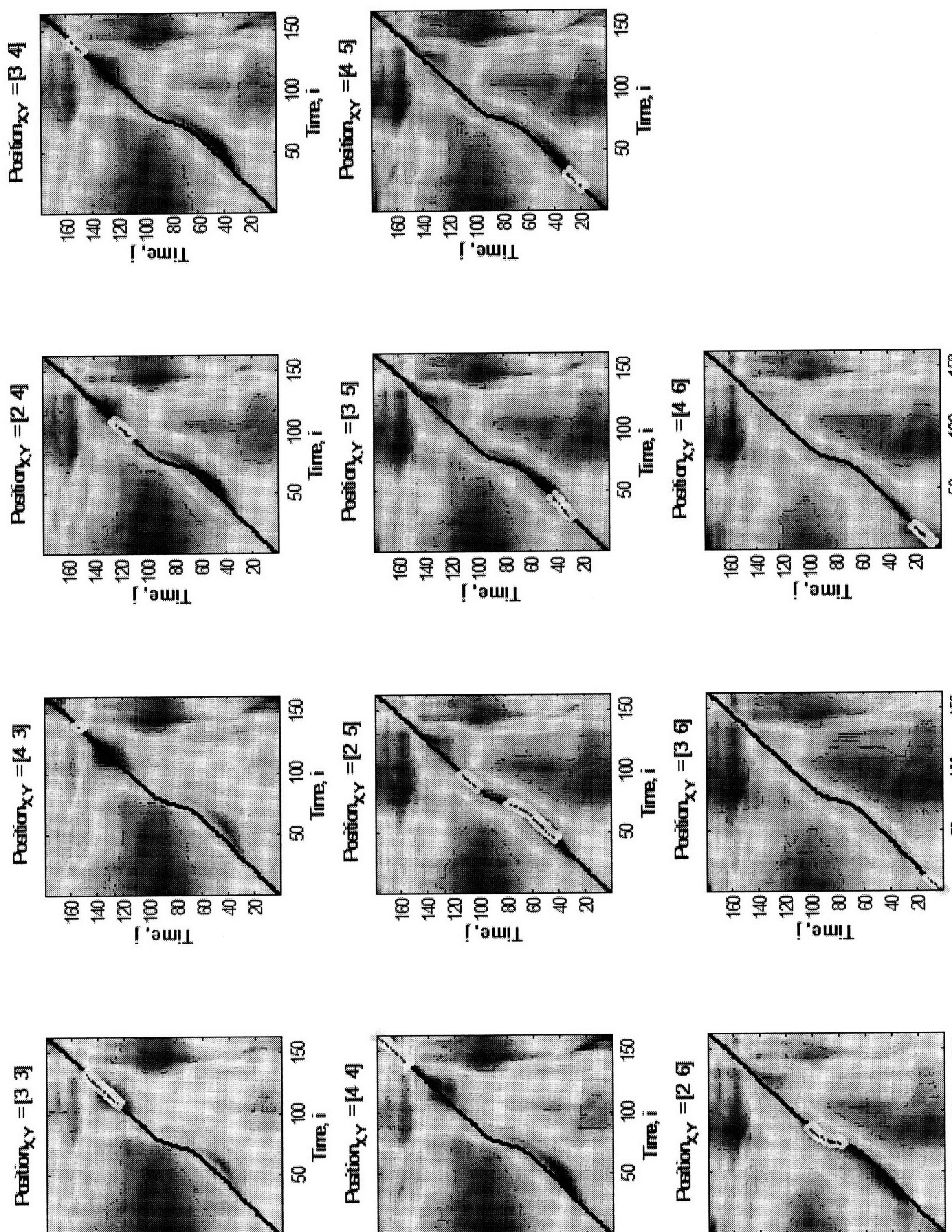


Figure 5-4: Relevant Spatial Planes of the Elemental Distance Hypervolume, D_L, with Optimal Warp Path. DTSW results for the ideal karate punch video, **C**, and an unknown karate punch video, **Q**.

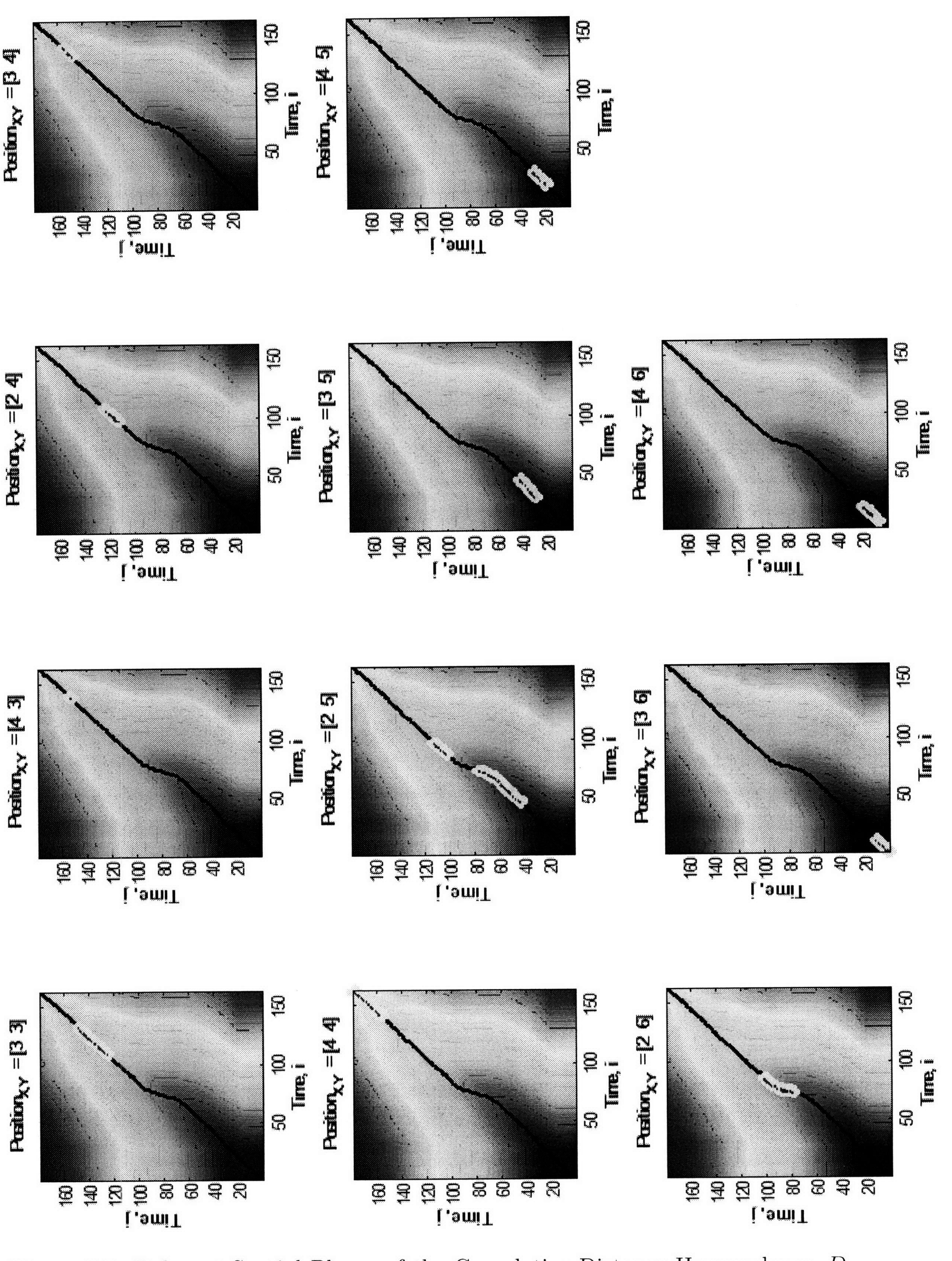
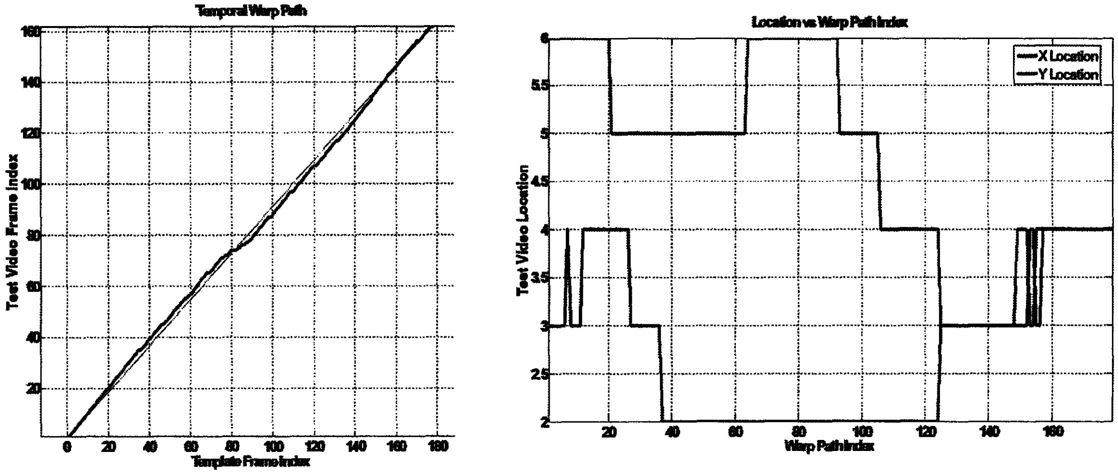


Figure 5-5: Relevant Spatial Planes of the Cumulative Distance Hypervolume, D_C, with Optimal Warp Path. DTSW results for the ideal karate punch video, **C**, and an unknown karate punch video, **Q**.



(a)

(b)

Figure 5-6: Spatial and Temporal Localization for Karate Punch Example

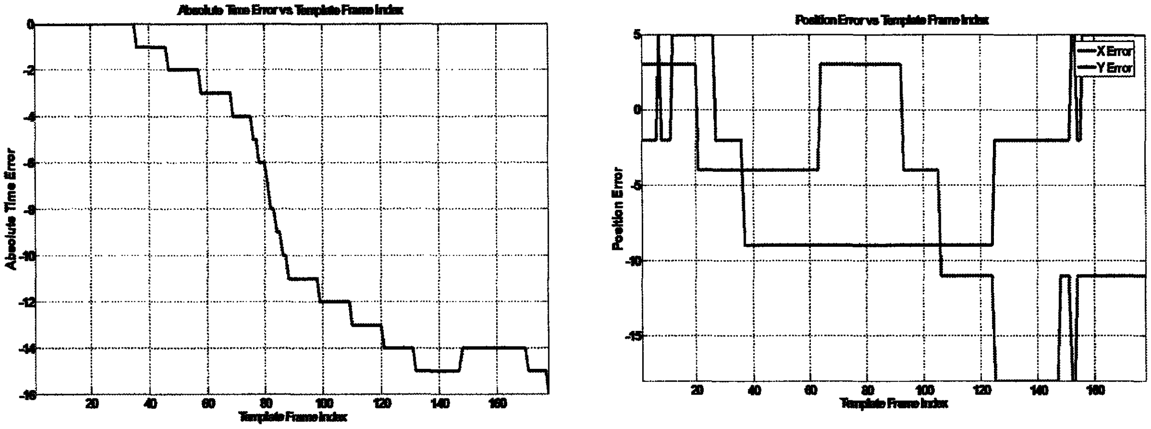
the test frame for the given location on the warp path.

The output videos, $\mathbf{q}_\mathbf{w}$ and $\mathbf{c}^*_\mathbf{w}$, are synchronized in time and in space. The warp path reveals pronounced temporal acceleration between the test and template about half-way through the punching move. This detailed information can now be used as part of a training regimen.

5.3 Horse Gait Comparison

Horse racing was one of the first applications of [high-speed] motion photography. A bet was placed between two very rich men regarding a conjecture that a horse at some point during its racing stride had all of its feet off of the ground and was at that point airborne.

Horse racing remains a very big market segment for the motion capture video. Owners and trainers are interested in the motion of a horse as it runs; they make decisions about horses based on trade secret metrics derived from an analysis of the horses' motion. Owners hire horse analysis service companies. These service companies take video equipment and a legion of people to a training track. Hundreds of videos of horses running around the track are recorded during the day. During the



(a)

(b)

Figure 5-7: Spatial and Temporal Error for Karate Punch Example

evening, the videos of the horses are manually analyzed. The next day the analysis information is used by owners to decide if a horse is a future winner. Owners use many theories, all of them proprietary trade secrets, in their analysis of horses' kinematics and dynamics. The price for a predicted winner ranges from tens of thousands of dollars to millions of dollars.

We have explored producing a system to automatically analyze the motion of the horses in conjunction with different horse analysis service companies. Details such as the positions of the hoofs, the tilt of the neck, and the height of the hindquarters with respect to the neck are some of the parameters of interest. The problem is that the videos are often of insufficient resolution, and much of the time the body features that are of interest are under mud (the hooves) or obstructed by the jockey or other horses.

We think that the DTSW algorithm is one component of a possible solution tactic for an automatic horse analysis system. A template horse stride would be recorded or otherwise produced. This template would then used to extract full strides in other videos. The localized strides can then be analyzed via the fit that they have with template stride.

The first few Eigenframes (filters for the filterbank implementation) are shown in Figure 5-10.

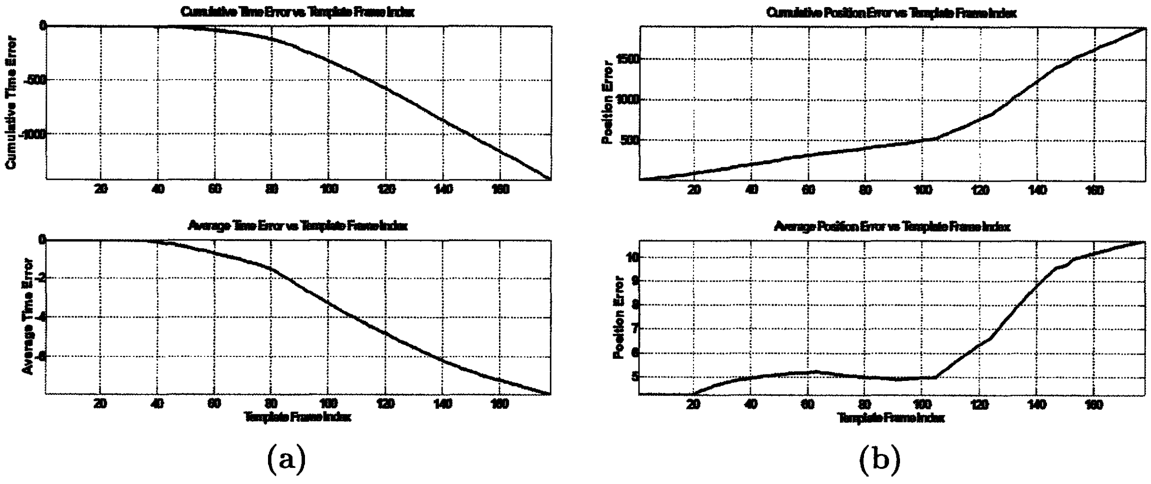


Figure 5-8: Cumulative and Average Error (Spatial and Temporal) for Karate Punch Example

We show the output of the DTSW algorithm comparing a "winning" racehorse to a second place finisher.

5.3.1 DTSW Algorithm

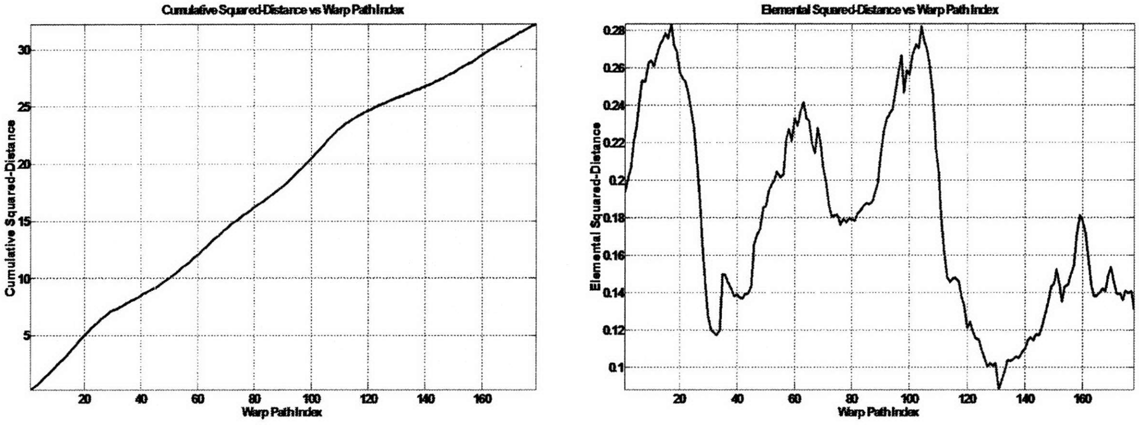
The template and test videos are visually similarly but clearly of different duration, and the event transpires at different spatial locations in the video. Figure 5-12 shows a few frames of the input and output sequences.

Figure 5-13 shows the temporal and spatial localization as a function of warp path.

Figure 5-14 shows the spatial and temporal difference (error in some contexts) between the template and test versus time of the template video.

Figure 5-15 shows the cumulative and average differences (error in some contexts) between the template and test versus time of the template video.

Figure 5-16 shows the cumulative and elemental squared distances along the warp path. A Euclidean distance between pixels normalized by the size of the template frame was used for this test. The elemental distance can therefore be interpreted as the average squared difference between a pixel from the template and a pixel from the test frame for the given location on the warp path.



(a) (b)

Figure 5-9: Cumulative and Elemental Squared Distances Along the Optimal Warp Path for Karate Punch Example

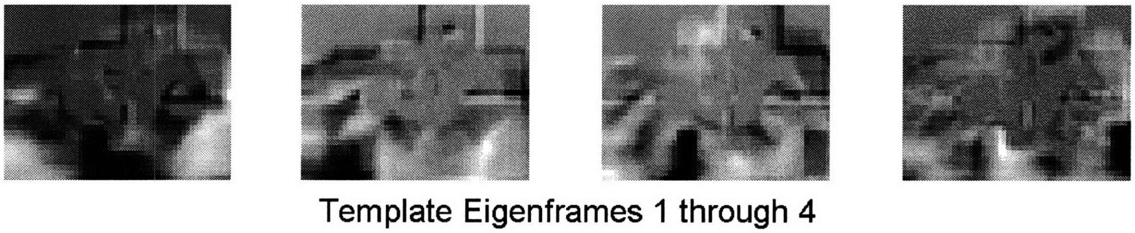


Figure 5-10: Eigenframes for the Horse Template Video

The output videos, $\mathbf{q_w}$ and $\mathbf{c}^*_{\mathbf{w}}$, are synchronized in time and in space. The warp path reveals pronounced temporal differences between the winning horse and the second place horse. This detailed information can now be used for evaluation or for training.

5.4 Summary

DTSW can be used to make appearance-based comparison of videos. An appearance-based comparison of videos allows us to either supplement or replace application-specific model-based analyses in fields such as physical therapy, sports training, and other motion analysis applications. We explored applications in karate training and horse gait analysis as interesting applications of DTSW.

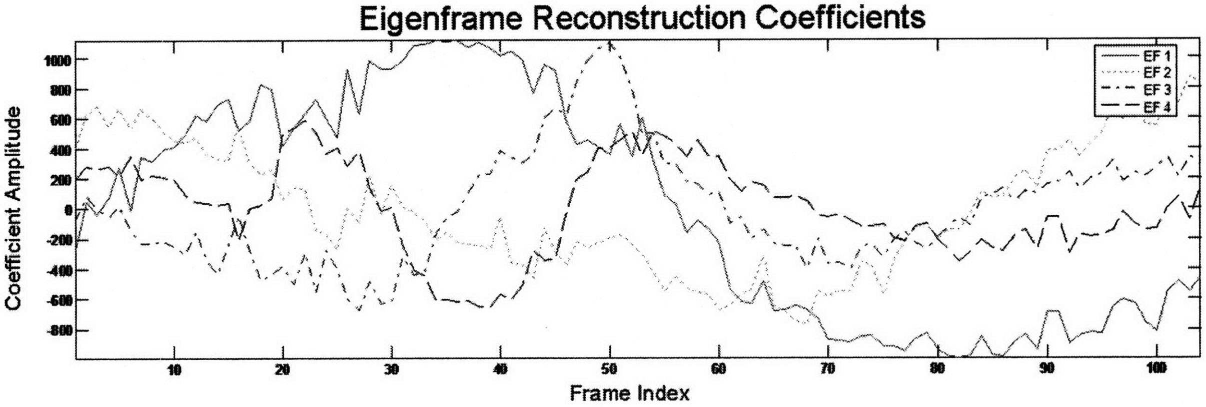


Figure 5-11: Eigenframes Reconstruction Coefficients for the Horse Template Video

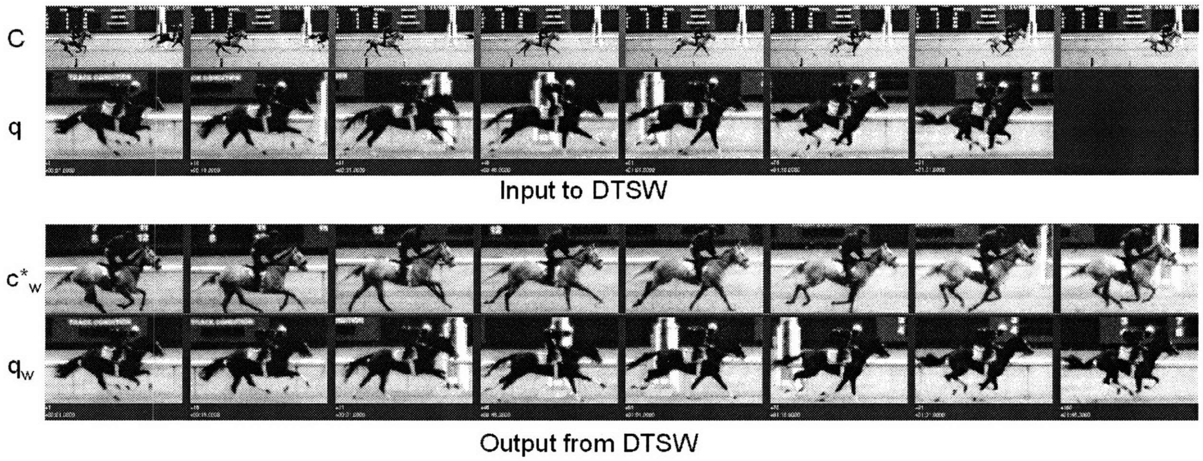
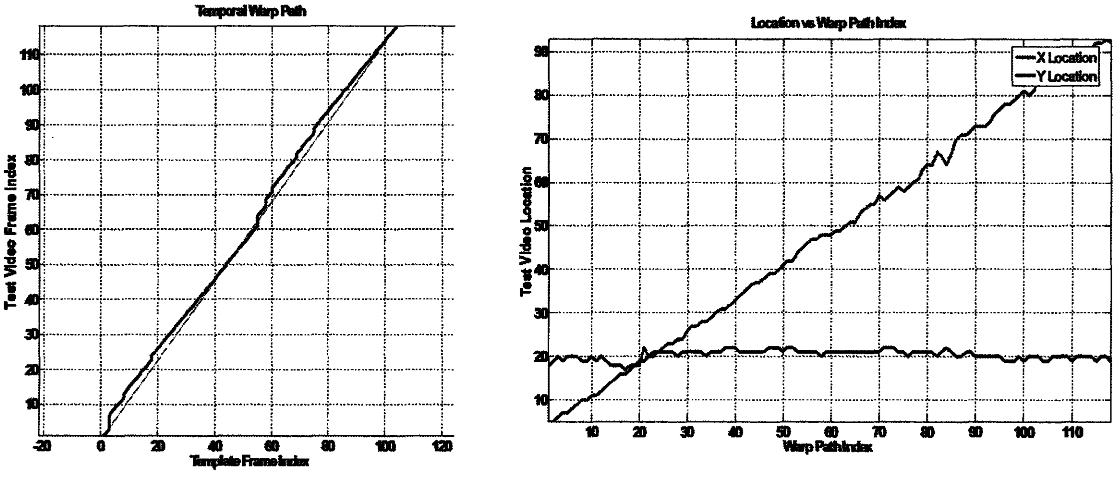
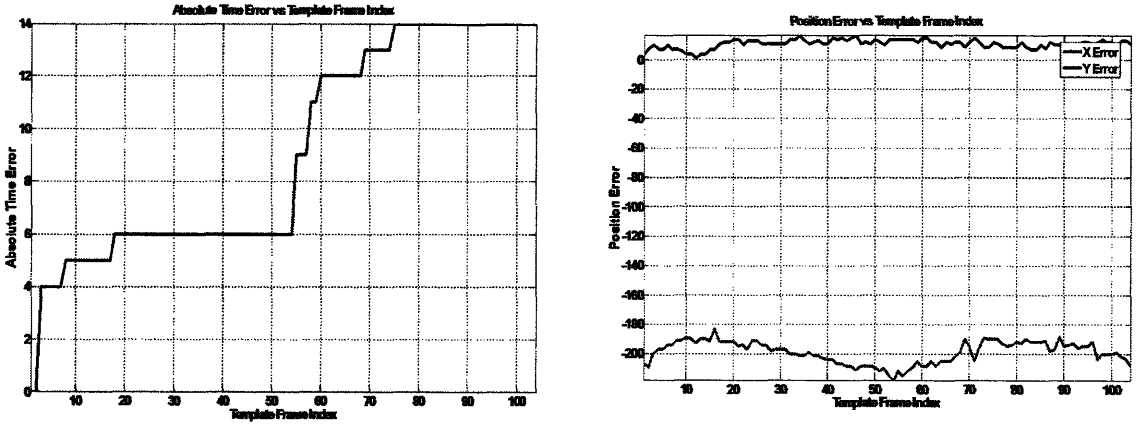


Figure 5-12: DTSW Input and Output Sequences for the Horse Gait Example. Test, **C**, and template, **Q**, input video sequences are shown in the top of the figure. **C** is a video of one full stride of a slow horse. **Q** is a video of one full stride of a fast horse. • DTSW temporally warps and spatially shifts **C** and **Q** creating output sequences that have minimum temporal and spatial variation - maximum similarity - throughout the sequence. The bottom two strips show the videos - $\mathbf{C}^*_{\mathbf{w}}$ and $\mathbf{Q}_{\mathbf{w}}$ - output from the DTSW algorithm. The frames of **C** are spatially larger than the frames of **Q**. The frames of $\mathbf{C}^*_{\mathbf{w}}$ are the same size as the frames of **Q** or $\mathbf{Q}_{\mathbf{w}}$. Each frame of $\mathbf{C}^*_{\mathbf{w}}$ has been extracted from a frame of **C** to locally match to a frame of **Q**. The output videos have a common time axis. The common time axis is, generally, different from either of the time axes of the input videos. Output videos temporally aligned to either input video can be obtained by projecting the warped time axis onto either input time axis. • As defined by the constraints of DTSW, the temporal evolution of the output videos along their common time axis is maximally similar. Also, as defined by the constraints of DTSW, the spatial arrangement of information in each aligned frame-pair of the output sequences is maximally similar. This can be seen by studying the video strips at the bottom of the figure. This overall similarity is best understood when one views the two input videos played side-by-side followed by the two output video played side-by-side. The output videos are visually dramatic and very appealing to the intelligent viewer. The warp-path or overall similarities, both temporal and spatial, are used for automated process-monitoring.



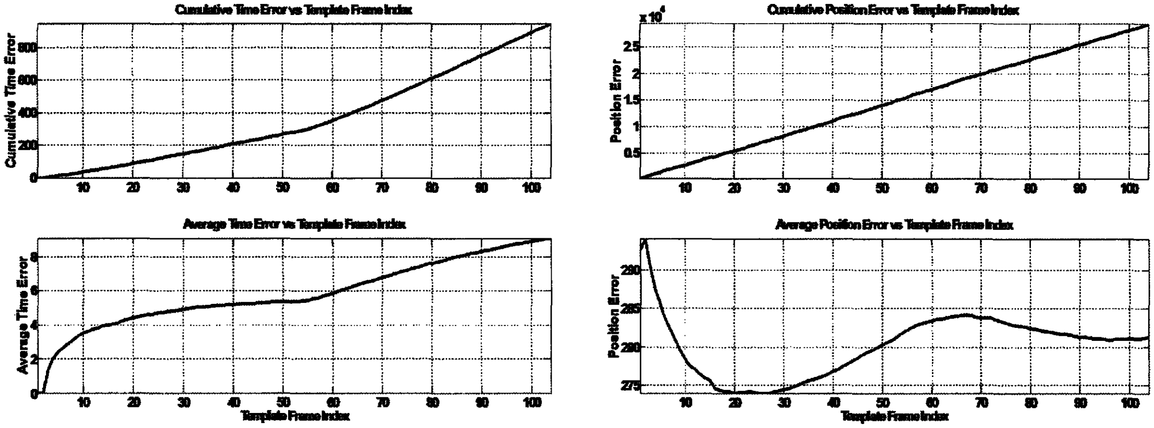
(a) (b)

Figure 5-13: Spatial and Temporal Localization for Horse Stride Example



(a) (b)

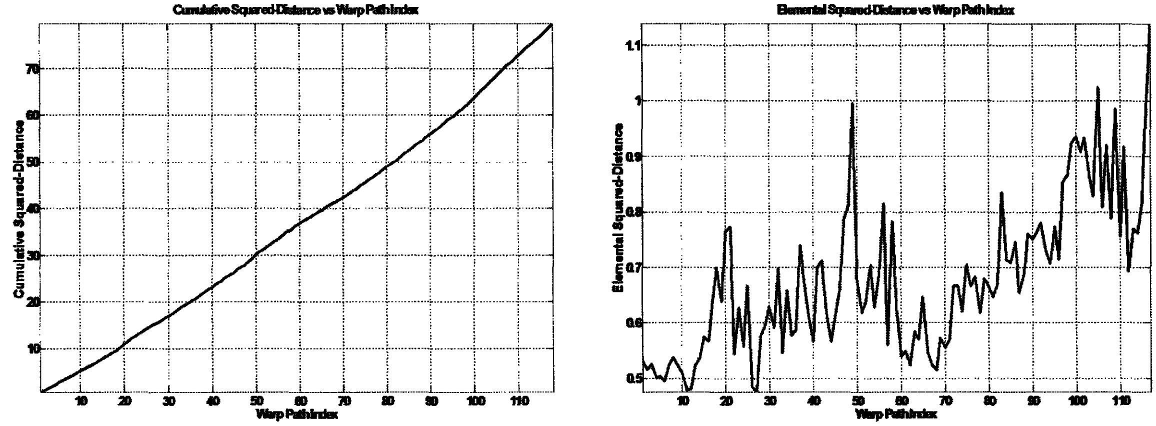
Figure 5-14: Spatial and Temporal Error for Horse Stride Example



(a)

(b)

Figure 5-15: Cumulative and Average Error (Spatial and Temporal) for Horse Stride Example



(a)

(b)

Figure 5-16: Cumulative and Elemental Squared Distances Along the Optimal Warp Path for Horse Stride Example

Chapter 6

Evaluation, Automated Monitoring

6.1 Introduction

The DTSW algorithm can be used for automated monitoring of industrial machinery, manufacturing processes, and scientific experiments. This chapter discusses the DTSW algorithm as used in various monitoring scenarios. We discuss failure detection - both post event and during event failure detection. We discuss lifetime testing and process monitoring. We explore example applications to get a feeling for how the DTSW machinery is used in practice.

6.2 Preliminaries

The DTSW algorithm can be used for automated monitoring in manufacturing and scientific experiments. The targeted subject materials are objects that are amorphous or that lack distinct or trackable features. This chapter discusses the ways that the DTSW algorithm can be used in various monitoring scenarios. The DTSW algorithm is used to detect failures either after a completed event or during an event.

The various outputs from the DTSW algorithm include :

1. The warp path that optimally locates the template in space and time in the test video.

This includes :

   (a) The spatial location of the template in the test video along the warp path.

   (b) The temporal warping of the template in the test video along the warp path.

2. The similarity (alternatively, the difference or distance) between the template and test videos as a function of position along the optimal warp path.

   This includes :

   (a) The total cumulative similarity. A single number that represents the overall similarity of the template video to the found template in the test video.

   (b) The rate of similarity accumulation along the warp path.

3. It is also possible, though more memory intensive, to find and store alternative paths that are close to the optimal path. In the limit, the algorithm can output the minimum cost path to every location in the elemental distance hypervolume.

We've seen how we can use the warp path to produce visually appealing, temporally synchronous, and spatially aligned videos. Automated monitoring applications do not require that we produce videos for viewing. Instead we use the outputs from the DTSW algorithm to make decisions about the recorded event.

Monitoring needs dictate how this information is used. In some situations it will be sufficient to make a pass / fail decision solely on the total cumulative similarity. In more complex situations, a detailed analysis of the warp path, the cumulative distance along the warp path, and the temporal and spatial distortions along the path are necessary to determine similarity, timing, and positional relationships between the template and the test videos.

6.3 Post-Event Failure Detection

Post-event detection is useful when it is sufficient to make a decision about the monitored event in its entirety after the event has completed.

Post event failure detection (or complete event comparison) is most analogous to single image template matching. A complete video - the test video - of the monitored event is acquired. Then the video template is found in (compared to) the test video. The DTSW algorithm finds the optimal warp path locating the template in the test video. Statistics on the various outputs from the DTSW algorithm are used to make decisions.

When applied to events that are high speed but have a low repetition rate, the computing power required to implement the DTSW algorithm can be reduced to make the monitoring system less expensive.

6.4 During Event Failure Detection

When decisions must be made before a monitored event has completed, the determination of failure moves into the internals of the DTSW algorithm. The failure detection mechanisms are as varied as they were for post-event failure-detection. Now, however, we must monitor several (potentially numerous) candidate warp paths (branches) through the elemental distance hypervolume. Failure detection decisions are made based on the cumulative distance, spatial shift, or temporal shift along the warp path that has the highest probability of being part of the optimal warp path. Or, if the cumulative distance has exceeded a prescribed threshold on all potential warp paths, we do not need to determine which path is the most likely.

Typically, however, instead of probabilistically assigning likelihood to the various warp paths, it is sufficient to monitor the warp path that is the least spatially and temporal warped. A threshold is set on the cumulative distance along this, the least distorted (or most central) path. This is not a bad compromise. We expect high similarity (in all of space, time, and "appearance") between the template and the test. If the least distorted path does not turn out to be part of the optimal path, then that again is interpreted as a failure, detected by exceeding the threshold on the cumulative distance along the path.

During-event failure detection generally requires more memory and computation power than post-event failure detection. We track the several (potentially numerous) candidate warp paths (branches) through the elemental distance hypervolume. During event failure detection is necessarily "real-time"; the DTSW algorithm must run close to the frame rate used to record the monitored event.

6.5 Lifetime Testing

The following example demonstrates an application of post event monitoring.

6.5.1 Introduction to Heart Valve Application

A candidate replacement heart valve for the human heart has been developed. The valve is made of tissue from the heart of a pig held in place by a polymer ring.

As part of the valve development, in order to demonstrate its viability as a replacement valve for humans, it must undergo accelerated lifetime failure tests. The heart valve is placed in a cross section of pipe. A clear fluid is flowed back and forth across the valve, by cyclically varying the pressure across the valve. This simulates the oscillating pressure found in a beating heart. The valve will open and close in response to the fluid flow. A high-speed camera, recording at 1000 fps, is setup to look down the pipe through a clear enclosure, focused on the valve. The system is set to run unattended for a long period of time. We want to monitor the deterioration of the valve over time.

Deterioration for this application is defined as the change in temporal profile of the opening and closing of the valve. A single representative number of this deterioration is determined by integrating temporal error between the template and the test sequences over one complete cycle. Other measures could be defined.

DTSW is a perfect candidate for this application. We must identify and spatially localize a deformed and flexible object, undergoing significant changes in appearance, through a sequence while simultaneously aligning the two events' temporal axes. Detailed quantification of the temporal occurrence of the event is required.

Signals from the pressure cycling equipment, provide a temporal signal that temporally segments the continuous video stream into single cycle clips.

We acquire a single cycle clip of the valve after it has first been put into operation. This video is the video from which we extract the temporal template. The entire ring moves with respect to the camera, in a plane perpendicular to the fluid flow direction; this is motion that is of no importance to the test. We are mainly interested in the valve flaps. We want to monitor the appearance and motion versus time of the tissue inside the synthetic ring. We extract a spatial subregion composed mainly of the valve flap tissue from every frame of the source video. This spatially reduced sequence is the template video, $\mathbf{Q}$ for this application.

Several frames from the template sequence are shown in Figure 6-1

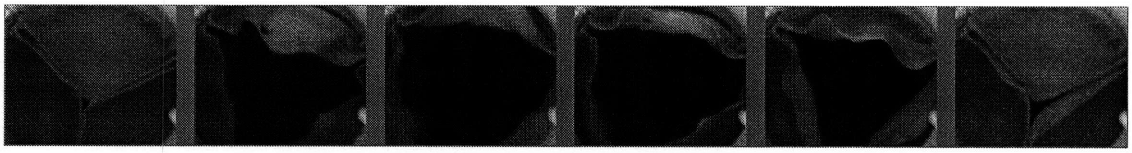


Figure 6-1: Heart Valve Template Video

The first few Eigenframes (filters for the filterbank implementation) are shown in Figure 6-2.

We show the output of the DTSW on two sequences over the duration of the accelerated lifetime test. One sequence is acquired very near the beginning of the lifetime test; the other sequence is after many thousands of simulated heartbeats. The results from the two sequences are shown together.

6.5.2 Two Events

We examine two test videos, a video of the heart valve towards the beginning of the lifetime test, a "new" valve, and a video of the heart valve cycle towards the end of the lifetime test - an "old" valve. The template video is a different cycle of the heart valve at the very beginning of the lifetime test - it is a cycle that an intelligent operator has deemed ideal.

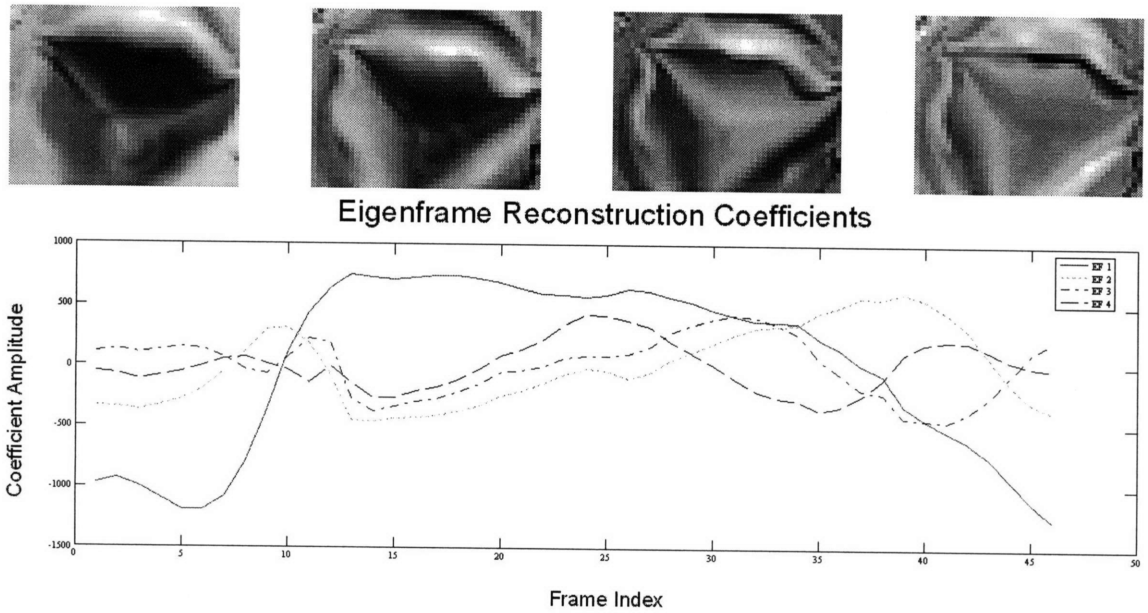


Figure 6-2: Eigenframes and Reconstruction Coefficients for Heart Valve Example

The template and "new" test videos are nearly identical. The template and "old" test videos are visually very different, and there is a visible change in the temporal characteristics of the two events. Figure 6-3 shows a few frames of the input and DTSW output sequences for the older sequence.

Figure 6-4 shows a flattened representation of a few planes from the Elemental Distance Hypervolume and optimal warp path for the older sequence. Figure 6-5 shows a flattened representation of a few planes from the Cumulative Distance Hypervolume along with the optimal warp path for the older sequence. The solid black line is the temporal projection of the warp path onto each spatial plane. The yellow dots indicate if the spatial component of the warp path is in the indicated spatial plane.

Figure 6-6 shows the spatial and temporal localization as a function of warp path.

Figure 6-7 shows the spatial and temporal difference between the template and test versus time of the template.

Figure 6-8 shows the cumulative and elemental distances along the warp path. A

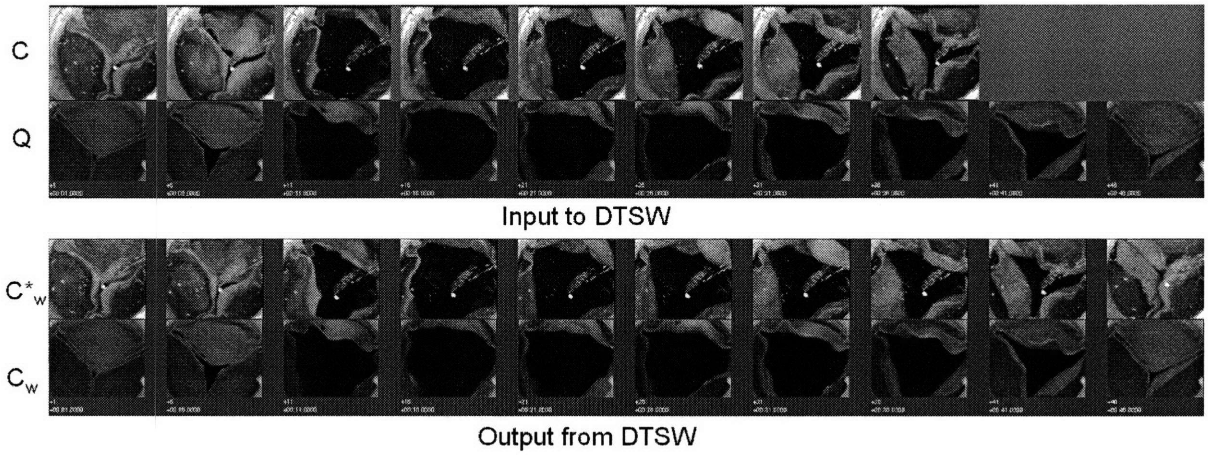


Figure 6-3: DTSW Input and Output Sequences for the Heart Valve Example. Test, $\mathbf{C}$, and template, $\mathbf{Q}$, input video sequences are shown in the top of the figure. $\mathbf{C}$ is a video of a heart valve cycle towards the end of the lifetime test - an "old" valve. $\mathbf{Q}$ is a video of the heart valve at the very beginning of the lifetime test - a "new" valve. • DTSW temporally warps and spatially shifts $\mathbf{C}$ and $\mathbf{Q}$ creating output sequences that have minimum temporal and spatial variation - maximum similarity - throughout the sequence. The bottom two strips show the videos - $\mathbf{C}^*_\mathbf{w}$ and $\mathbf{Q}_\mathbf{w}$ - output from the DTSW algorithm. The frames of $\mathbf{C}$ are spatially larger than the frames of $\mathbf{Q}$. The frames of $\mathbf{C}^*_\mathbf{w}$ are the same size as the frames of $\mathbf{Q}$ or $\mathbf{Q}_\mathbf{w}$. Each frame of $\mathbf{C}^*_\mathbf{w}$ has been extracted from a frame of $\mathbf{C}$ to locally match to a frame of $\mathbf{Q}$. The output videos have a common time axis. The common time axis is, generally, different from either of the time axes of the input videos. Output videos temporally aligned to either input video can be obtained by projecting the warped time axis onto either input time axis. • As defined by the constraints of DTSW, the temporal evolution of the output videos along their common time axis is maximally similar. Also, as defined by the constraints of DTSW, the spatial arrangement of information in each aligned frame-pair of the output sequences is maximally similar. This can be seen by studying the video strips at the bottom of the figure. This overall similarity is best understood when one views the two input videos played side-by-side followed by the two output video played side-by-side. The output videos are visually dramatic and very appealing to the intelligent viewer. The warp-path or overall similarities, both temporal and spatial, are used for automated process-monitoring.

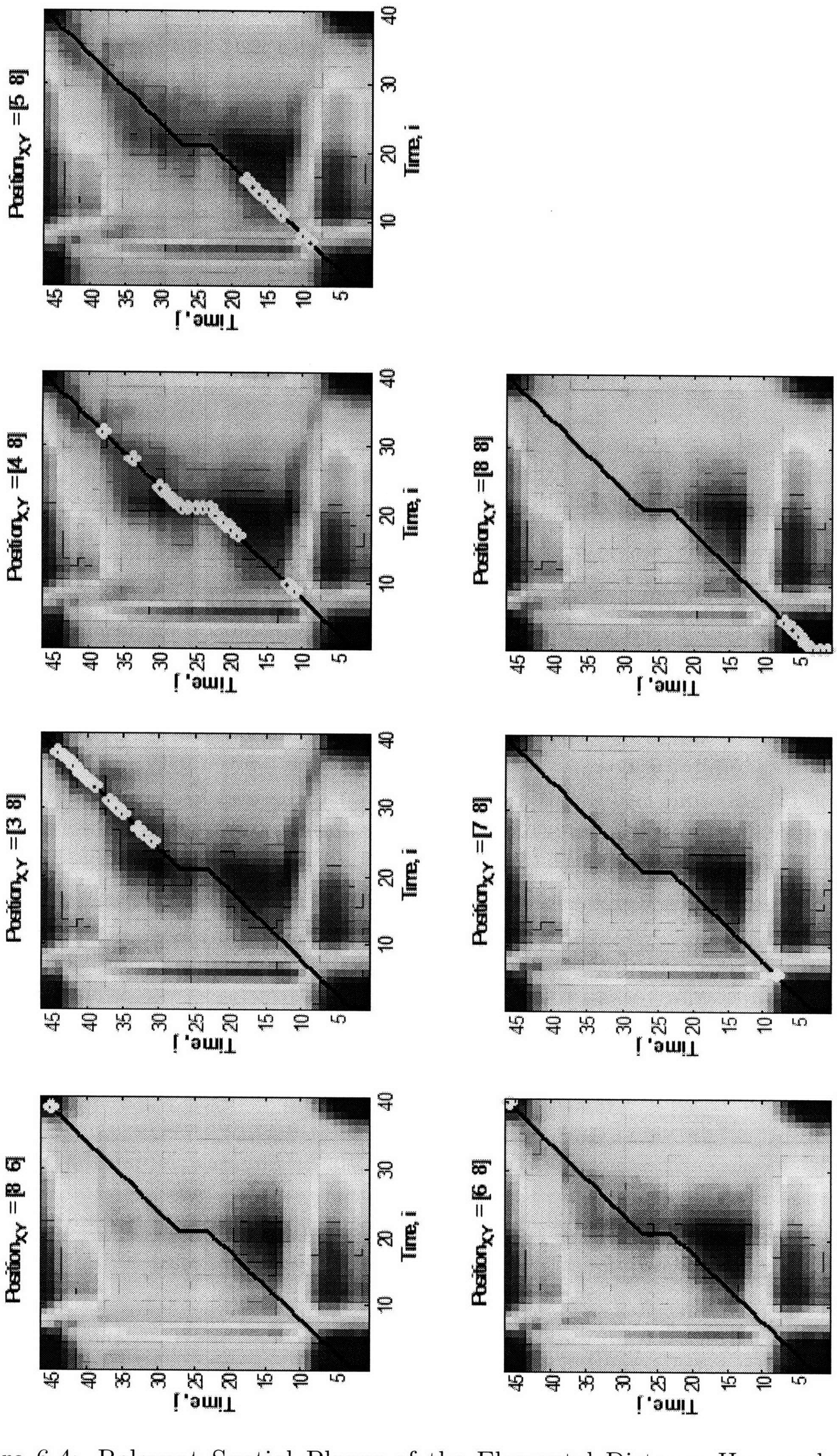


Figure 6-4: Relevant Spatial Planes of the Elemental Distance Hypervolume, D_L, with Optimal Warp Path. DTSW results for an old heart valve video, **C**, and a new heart valve video, **Q**.

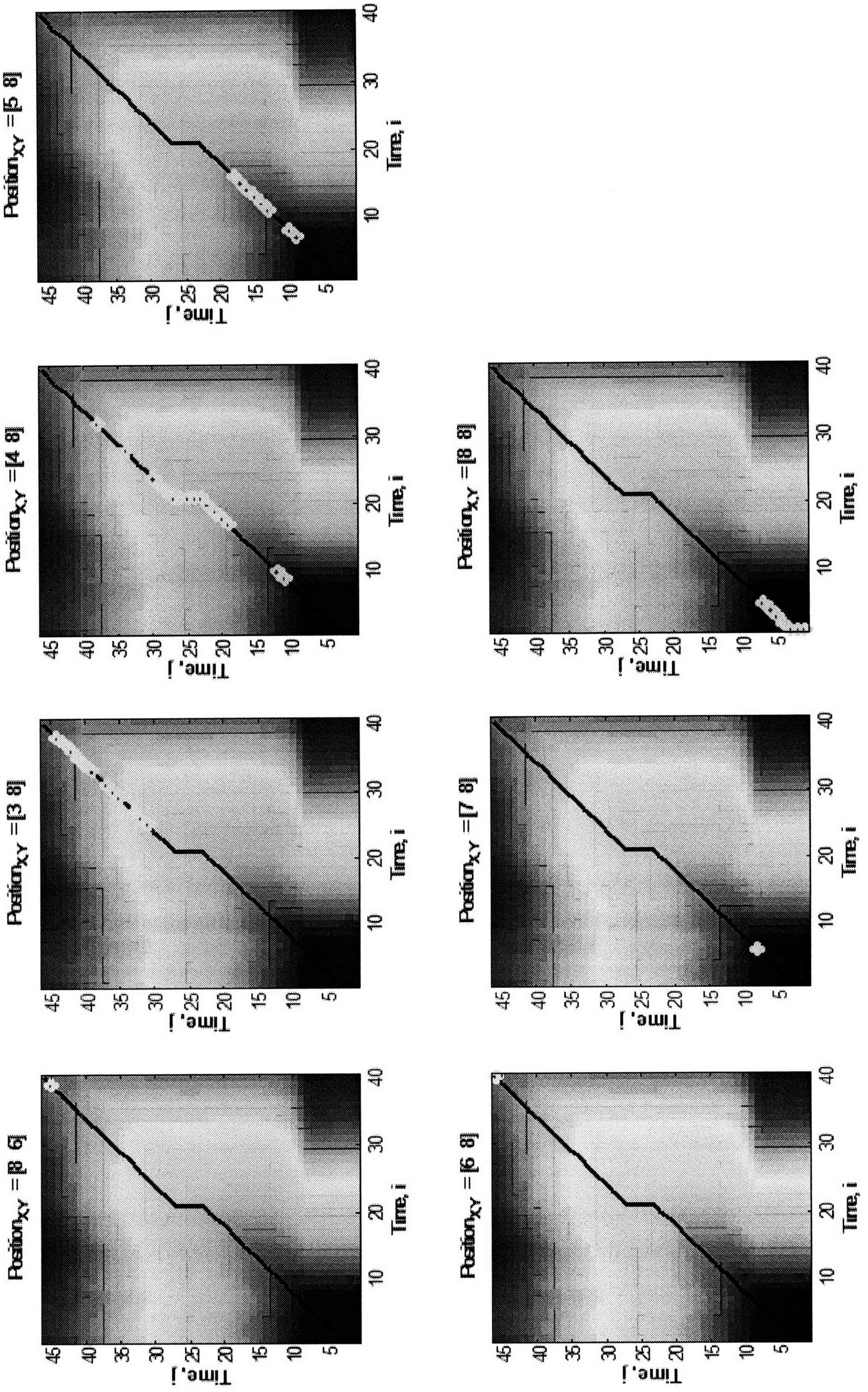


Figure 6-5: Relevant Spatial Planes of the Cumulative Distance Hypervolume, D_C, with Optimal Warp Path. DTSW results for an old heart valve video, **C**, and a new heart valve video, **Q**.

Euclidean distance between pixels normalized by the size of the template frame was used for this test. The elemental distance can therefore be interpreted as the average pixel difference between a pixel from the aligned template frame and test frame.

Detailed event timing along the warp path is the most relevant information in this application. The "new" valve is indeed opening and closing, just like the template, as is evident from the constant temporal slope between the time axes from each as seen in Figure 6-6(a). The temporal shift at the the beginning and end of the "new" valve event is due to imprecise pre-temporal segmentation. In Figure 6-6(a) or 6-7(a) we see a temporal delay, relative to the template, in the middle of the "old" heart valve event. This is physically due to material wear leading to reduced stiffness ; the valve's response to changes in fluid flow is altered.

The event level summary for timing, position, and appearance is presented in Table 6.1. Though not used in a feedback control system, this data shows that we could monitor and detect "failure" of the heart valve by monitoring the event level time statistics. The valve's rigid-body position changes are from induced motions of the valve support structure caused by operation of the valve and fluid flow. We are not concerned with the variations in valve position in this test.

Figure 6-9 presents the L_2 norm event time error for two contiguous periods of the valve lifetime test. We have eliminated a large section of time between the new and aged valves. If we intended to autonomously monitor and detect wear; we would first set a threshold on the expected performance of the valve during early operation and then detect when that threshold is exceeded after hundreds of event cycles.

6.5.3 Summary

Test, **C**, and template, **Q**, input video sequences are shown in the top of Figure 6-3. **C** is a video of a heart valve cycle towards the end of the lifetime test - an "old" valve. **Q** is a video of the heart valve at the very beginning of the lifetime test - a "new" valve.

DTSW temporally warps and spatially shifts **C** and **Q** creating output sequences

	Event Statistic		
		New	Old
Position			
	Mean Error (Average)	10.60	16.64
	RMS Error (L2 Norm)	10.76	17.68
	Max Error	13.04	24.17
	Std. Errror	1.82	5.98
Time			
	Mean Error (Average)	-1.87	-3.80
	RMS Error (L2 Norm)	1.94	4.31
	Mean Abs. Error (L1 Norm)	1.91	3.80
	Max Error	1.00	0.00
	Min Error	-2.00	-6.00
	Std. Errror	0.54	2.03
	Sum Diff	1.00	-6.00
	Sum Abs Diff	5.00	6.00
Relative Time			
	Mean Error (Average)	-2.33	-0.85
	RMS Error (L2 Norm)	2.40	1.32
	Mean Abs. Error (L1 Norm)	2.33	1.08
	Max Error	0.00	1.15
	Min Error	-2.91	-2.85
	Std. Errror	0.60	1.01
	Sum Diff (should be close to zero)	0.02	-0.15
	Sum Abs Diff	5.81	11.85
Appearance			
	DTSW Distance	17.89	45.40
	Mean Error (Average Frame (Pixel) Distance)	0.37	0.99
	Max Error	0.51	1.26
	Std. Errror	0.08	0.13
Path			
	Path Length	49	46
	Relative Path Length	1.07	1.00
	Length Q	46	46
	Length C	47	40

Table 6.1: Summary of DTSW Event Statistics for a New and Old Heartvalue Event (Single "Beat")

that have minimum temporal and spatial variation - maximum similarity - throughout the sequence. The bottom two strips show the videos - $\mathbf{C}_{\mathbf{w}}^{*}$ and $\mathbf{Q}_{\mathbf{w}}$ - output from the DTSW algorithm. The frames of $\mathbf{C}$ are spatially larger than the frames of $\mathbf{Q}$. The frames of $\mathbf{C}_{\mathbf{w}}^{*}$ are the same size as the frames of $\mathbf{Q}$ or $\mathbf{Q}_{\mathbf{w}}$. Each frame of $\mathbf{C}_{\mathbf{w}}^{*}$ has been extracted from a frame of $\mathbf{C}$ to locally match to a frame of $\mathbf{Q}$. The output videos have a common time axis. The common time axis is, generally, different from either of the time axes of the input videos. Output videos temporally aligned to either input video can be obtained by projecting the warped time axis onto either input time axis.

As defined by the constraints of DTSW, the temporal evolution of the output videos along their common time axis is maximally similar. Also, as defined by the constraints of DTSW, the spatial arrangement of information in each aligned frame-pair of the output sequences is maximally similar. Again, this can be seen by studying the video strips at the bottom of the Figure 6-3. This overall similarity is best understood when one views the two input videos played side-by-side followed by the two output video played side-by-side. The output videos are visually dramatic and very appealing to the intelligent viewer. The warp-path or overall similarities, both temporal and spatial, are used for automated process-monitoring.

6.6 Process Monitoring

The following example demonstrates an application of post event monitoring.

6.6.1 Introduction to Diaper Packaging Application

A well-known paper products manufacturer packages a myriad of paper and fabric-like products. During part of the diaper packing process, diapers are fed along a conveyor, wrapped, and placed into a box.

At times, due to slight variation in the way that the paper has folded or that glue has been applied to various parts of the paper or fabric, a product will get caught on the machinery. A single product that has "glitched" will set into motion a complete failure (jam) of the packaging process. This failure can happen in a matter of seconds or fractions of a second depending on the type of paper product. Such a failure leads to significant downtime as machinery must be cleared of paper that is wedged into places it was never meant to go. (Think of the hassle associated with clearing the paper from a printer or photocopier and increase this by two orders of magnitude.)

The goal is to monitor the paper products as they are glued, moved along conveyer belts, and forced through folding machines. As soon as motion or appearance occurs that deviates from the expected, the desire is to shut the machinery down immediately. Then the single guilty culprit can be removed from the line instead of the hundreds of diapers that get backed up if the glitch is not detected in time.

We demonstrate process monitoring on the entrance to the wrapping section of the diaper packaging line. As a diaper moves along a conveyor belt, it is squeezed closed, and fed into a machine that places a thin plastic wrapper around the diaper. Faults or glitches for this application are defined as any significant temporal, spatial, or appearance deviation from the expected temporal template.

DTSW is a good candidate for this application. The task is to identify and spatially localize a deformed and flexible object - undergoing significant changes in appearance - through a sequence, and compare it to the standard template in space, time, and appearance.

Control and detection signals along the conveyor system, provide a temporal signal that temporally segments the continuous video stream into single cycle clips.

We extract a single cycle clip of the diaper flow during normal operation. This video is the source video from which we extract the temporal template. We separately extract templates for the leading edge, trailing edge, and entire diaper. For each template, we extract a spatial sub-region from every frame of the source video. Each template video, $\mathbf{Q}$, is used in a separate instance of the DTSW algorithm.

Figure 6-10 shows a few frames of the input and output sequences monitoring for the leading edge template.

The first few Eigenframes (filters for the filterbank implementation) are shown in Figure 6-11.

We have a video from the wrapping machinery that is numerous cycles before and during a complete jam and shutdown of the machinery. Visually inspecting the cycles before the jam, we note the following details:

- **Event n-1** - This diaper is the last completely correct diaper wrapping cycle before the events leading to jam and shutdown.
- **Event n** - At first glance, this diaper appears to speed up a little compared to the norm. Upon closer examination, it appears as if the far-end (away from the camera) of the diaper gets stuck or caught on the machinery, rotates about the snag, and accelerates at the near-end (closest to the camera).
- **Event n+1** - This diaper gets too close to its predecessor and bunches a little at the front.
- **Event n+2** - This diaper appears to travel smoothly (position versus time) and is unencumbered by the preceding events. The front of the diaper, however, is a little close to the back of the diaper that preceded it but they do not touch.
- **Event n+3** - This diaper appears to travel smoothly at the beginning but then abruptly slows as it enters the wrapping section. It is blocked by an internal machinery obstruction.
- **Event n+4** - The diaper is further blocked; the machinery is completely jammed at this point.

The interpretation of the sequence of events is that sequence **n** upsets the timing of the conveyor and wrapping system and continues to stick to the internals of the equipment. That diaper, as is enters the machinery, initiates a fault that is catastrophic 4 cycles later. The manufacturer would like to detect **Event n** and halt the machinery immediately. We demonstrate the feasibility of early detection, detecting the first subtle glitch, before complete failure.

The six sequences discussed above are shown in Figure 6-12. A small arrow has been drawn on the diaper of **Event n-1** both to indicate direction of travel (left-to-right) and to make it easier to understand the progression of the diapers through the machinery.

6.6.2 Sequences

The template and test videos are nearly identical for this example. Very close examination of Figure 6-12 will reveal subtle differences among the several test videos after **Event n**. DTSW is good at highlighting these subtle differences. Figure 6-13 shows the spatial and temporal localization as a function of warp path. Figure 6-14 shows the spatial and temporal difference between the template and test versus time of the template.

Figure 6-15 shows the cumulative and elemental distances along the warp path. A Euclidean distance between pixels normalized by the size of the template frame was used for this test. The elemental distance can therefore be interpreted as the average pixel difference between a pixel from the aligned template frame and test frame.

Event time, location, and appearance are all relevant in this application. Faults or glitches for this application are defined as any significant temporal, spatial, or appearance deviation from the expected temporal template. The event-level error statistics for timing, position, and appearance are presented in Table 6.2.

Several event statistics prior to, including, and after the first glitch at **Event n** are plotted in Figure 6-16. In order to autonomously monitor and detect faults, we first set a threshold based on the expected performance of the diapers during normal operation and then detect when that threshold is exceeded. We see that it would be easy to select thresholds that detect a problem starting at **Event n** and going through **Event n+4**.

6.6.3 Summary

Test, **C**, and template, **Q**, input video sequences are shown in the top of Figure 6-10. **C** is a video of one cycle from the diaper packaging line. **Q** is a video of the ideal packaging cycle.

DTSW temporally warps and spatially shifts **C** and **Q** creating output sequences that have minimum temporal and spatial variation - maximum similarity - throughout the sequence. The bottom two strips show the videos - $\mathbf{C}^*_\mathbf{w}$ and $\mathbf{Q}_\mathbf{w}$ - output from the

	Event Statistic						
		n-1	n	n+1	n+2	n+3	n+4
Position							
	Mean Error (Average)	1.50	3.39	18.83	11.56	3.15	16.74
	RMS Error (L2 Norm)	1.64	4.45	30.59	20.36	4.52	25.58
	Max Error	5.40	10.20	67.74	48.00	8.25	56.04
	Std. Errror	1.30	2.88	24.11	16.76	3.24	19.34
Time							
	Mean Error (Average)	0.00	-1.32	0.36	0.00	0.00	-2.52
	RMS Error (L2 Norm)	0.00	1.84	0.60	0.00	0.00	2.72
	Mean Abs. Error (L1 Norm)	0.00	1.32	0.36	0.00	0.00	2.52
	Max Error	0.00	0.00	1.00	0.00	0.00	0.00
	Min Error	0.00	-3.00	0.00	0.00	0.00	-3.00
	Std. Errror	0.00	1.29	0.48	0.00	0.00	1.02
	Sum Diff	0.00	0.00	0.00	0.00	0.00	0.00
	Sum Abs Diff	0.00	6.00	2.00	0.00	0.00	6.00
Relative Time							
	Mean Error (Average)	0.00	-1.32	0.36	0.00	0.00	-2.52
	RMS Error (L2 Norm)	0.00	1.84	0.60	0.00	0.00	2.72
	Mean Abs. Error (L1 Norm)	0.00	1.32	0.36	0.00	0.00	2.52
	Max Error	0.00	0.00	1.00	0.00	0.00	0.00
	Min Error	0.00	-3.00	0.00	0.00	0.00	-3.00
	Std. Errror	0.00	1.29	0.48	0.00	0.00	1.02
	Sum Diff (should be close to zero)	0.00	0.00	0.00	0.00	0.00	0.00
	Sum Abs Diff	0.00	6.00	2.00	0.00	0.00	6.00
Appearance							
	DTSW Distance	5.02	12.13	17.84	21.22	12.74	27.22
	Mean Error (Average Frame (Pixel) Distance)	0.25	0.43	0.68	0.83	0.51	0.97
	Max Error	0.40	0.80	1.05	1.39	0.60	1.63
	Std. Errror	0.08	0.14	0.23	0.35	0.06	0.32
Path							
	Path Length	25	28	26	25	25	28
	Relative Path Length	1.00	1.12	1.04	1.00	1.00	1.12
	Length Q	25	25	25	25	25	25
	Length C	25	25	25	25	25	25

Table 6.2: Summary of Diaper Wrapping Event Statistics Prior to Jam. Event time, location, and appearance are all relevant in this application. Faults or glitches for this application are defined as any significant temporal, spatial, or appearance deviation from the expected temporal template. The event-level error statistics for timing, position, and appearance are presented in the table.

DTSW algorithm. The frames of $\mathbf{C}$ are spatially larger than the frames of $\mathbf{Q}$. The frames of $\mathbf{C}^*_\mathbf{w}$ are the same size as the frames of $\mathbf{Q}$ or $\mathbf{Q}_\mathbf{w}$. Each frame of $\mathbf{C}^*_\mathbf{w}$ has been extracted from a frame of $\mathbf{C}$ to locally match to a frame of $\mathbf{Q}$. The output videos have a common time axis. The common time axis is, generally, different from either of the time axes of the input videos. Output videos temporally aligned to either input video can be obtained by projecting the warped time axis onto either input time axis.

As defined by the constraints of DTSW, the temporal evolution of the output videos along their common time axis is maximally similar. Also, as defined by the constraints of DTSW, the spatial arrangement of information in each aligned frame-pair of the output sequences is maximally similar. Again, this can be seen by studying the video strips at the bottom of the Figure 6-10. This overall similarity is best understood when one views the two input videos played side-by-side followed by the two output video played side-by-side. The output videos are visually dramatic and very appealing to the intelligent viewer. The warp-path or overall similarities, both temporal and spatial, are used for automated process-monitoring.

6.7 Summary

The DTSW algorithm can be used for automated monitoring in manufacturing and scientific experiments. The targeted subject materials may be amorphous or lacking distinct or trackable features. The DTSW algorithm is used to detect failures either post a completed event or during an event. The DTSW algorithm can be used in various monitoring scenarios such as lifetime testing and failure detection.

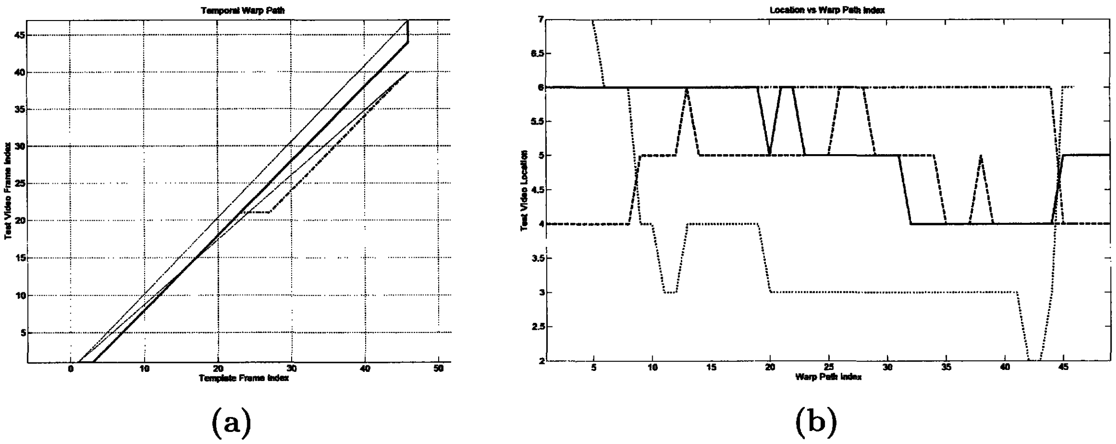
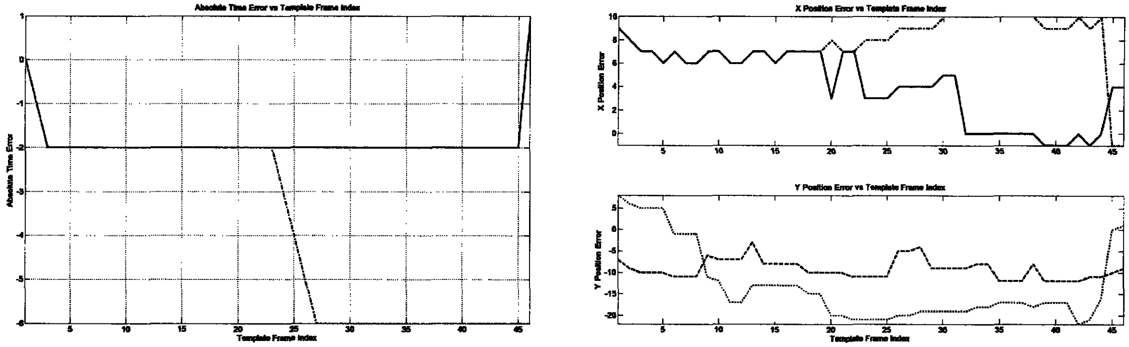
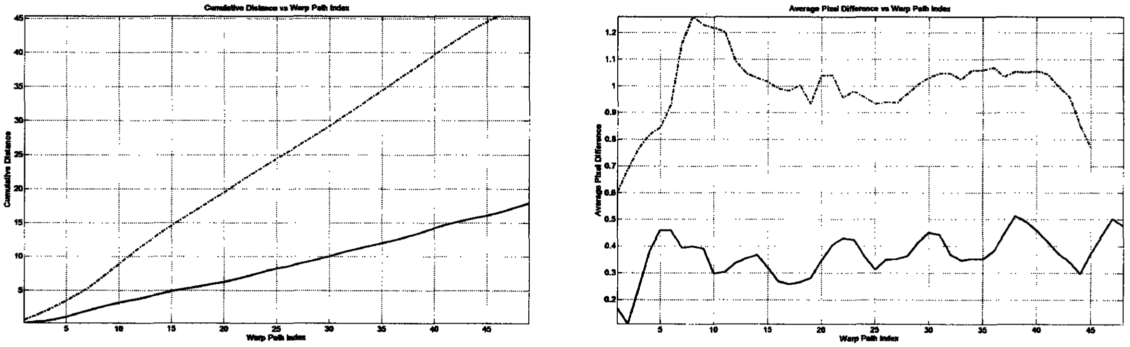


Figure 6-6: Spatial and Temporal Localization of Synthetic Heart Valve at Beginning and End of Lifetime Test. • Time. The temporal projection of the DTSW warp-path for the template and "new" valve sequence is a solid-bold line. The temporal projection of the DTSW warp-path for the template and "old" valve sequence is a dashed-dotted red line. The linearly-scaled time-axes are also shown. They are the thin black lines connecting the temporal beginning and end of each sequence. • Detailed event timing along the warp path is the most relevant information in this application. The "new" valve is indeed opening and closing just like the template, as is evident from the constant temporal slope between the time axes as seen in (a). The temporal shift at the beginning and end of the "new" valve event is due to imprecise video clip segmentation. We see a temporal delay, relative to the template, in the middle of the "old" heart valve event. This is due to wear changing the material stiffness; the valve's response to changes in flow is altered. • Position. The x-positions and y-positions for the location of each template frame matched to (found in) a frame of the "new" valve sequence are drawn in a solid-blue and a dashed-blue line respectively. The x-positions and y-positions for the location of each template frame matched to (found in) a frame of the "old" valve sequence are a dashed-dotted-red and a dot-red line respectively. • The position information is not relevant to this test. We are not concerned with the variations in valve position that are induced motions of the valve support structure caused by the operation of the valve and the fluid flow.



(a) (b)

Figure 6-7: Spatial and Temporal Error of Synthetic Heart Valve at Beginning and End of Lifetime Test. • Time. The temporal error along the DTSW warp-path between the time axes of the template and "new" valve sequence is a solid-bold line. The temporal error along the DTSW warp-path between the time axes of the template and "old" valve sequence is a dashed-dotted red line. • Position. The x-error and y-error between the location of each template frame found in a frame of the "new" valve sequence and the location of the template frames as originally extracted from its source video are drawn in a solid-blue and a dashed-blue line respectively. The x-errors and y-errors for the "old" valve sequence are a dashed-dotted-red and a dot-red line respectively.



(a) (b)

Figure 6-8: Cumulative and Elemental Distances Along the Optimal Warp Path for Synthetic Heart Valve at Beginning and End of Lifetime Test. The cumulative and elemental distances along the DTSW warp-path for the "new" valve sequence are drawn with a solid-bold line. The cumulative and elemental distances along the DTSW warp-path for the "old" valve sequence are drawn with a dashed-dotted red line. The Euclidean elemental distance normalized by the spatial size of the template frame was used for this test. The elemental distance is therefore interpreted as the average pixel difference between the frame-pairs of the DTSW output sequences.

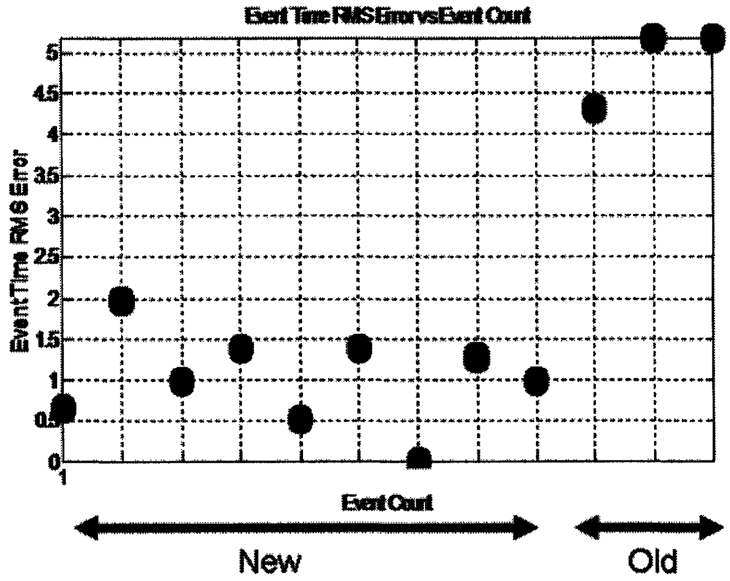


Figure 6-9: Heart Valve Event-Level Time Error During Lifetime Test. The data points plot the temporal (frame) RMS error between one complete cycle of the test valve opening-and-closing and the template video. We have eliminated a large section of time between the new and aged valves. The left portion of the graph with time error under 2 frames (RMS error over one cycle) shows that there is small variation between each heart valve cycle and the template. The right three data point with time error over 4 frames (RMS error over one cycle) shows the increased temporal variation as the valve ages. If we intended to autonomously monitor and detect wear; we would first set a threshold on the expected performance of the valve during early operation and then detect when that threshold is exceeded. For this example a threshold of 3 frames RMS would seem to be appropriate.

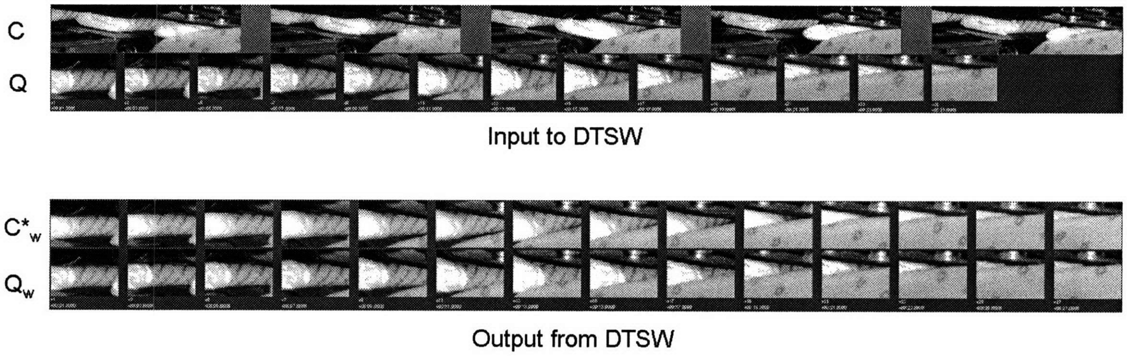


Figure 6-10: DTSW Input and Output Sequences for the Diaper Packaging Example. Test, **C**, and template, **Q**, input video sequences are shown in the top of figure. **C** is a video of one cycle from the diaper packaging line. **Q** is a video of the ideal packaging cycle. • DTSW temporally warps and spatially shifts **C** and **Q** creating output sequences that have minimum temporal and spatial variation - maximum similarity - throughout the sequence. The bottom two strips show the videos - $\mathbf{C}^*_{\mathbf{w}}$ and $\mathbf{Q}_{\mathbf{w}}$ - output from the DTSW algorithm. The frames of **C** are spatially larger than the frames of **Q**. The frames of $\mathbf{C}^*_{\mathbf{w}}$ are the same size as the frames of **Q** or $\mathbf{Q}_{\mathbf{w}}$. Each frame of $\mathbf{C}^*_{\mathbf{w}}$ has been extracted from a frame of **C** to locally match to a frame of **Q**. The output videos have a common time axis. The common time axis is, generally, different from either of the time axes of the input videos. Output videos temporally aligned to either input video can be obtained by projecting the warped time axis onto either input time axis. • As defined by the constraints of DTSW, the temporal evolution of the output videos along their common time axis is maximally similar. Also, as defined by the constraints of DTSW, the spatial arrangement of information in each aligned frame-pair of the output sequences is maximally similar. Again, this can be seen by studying the video strips at the bottom of the figure. This overall similarity is best understood when one views the two input videos played side-by-side followed by the two output video played side-by-side. The output videos are visually dramatic and very appealing to the intelligent viewer. The warp-path or overall similarities, both temporal and spatial, are used for automated process-monitoring.

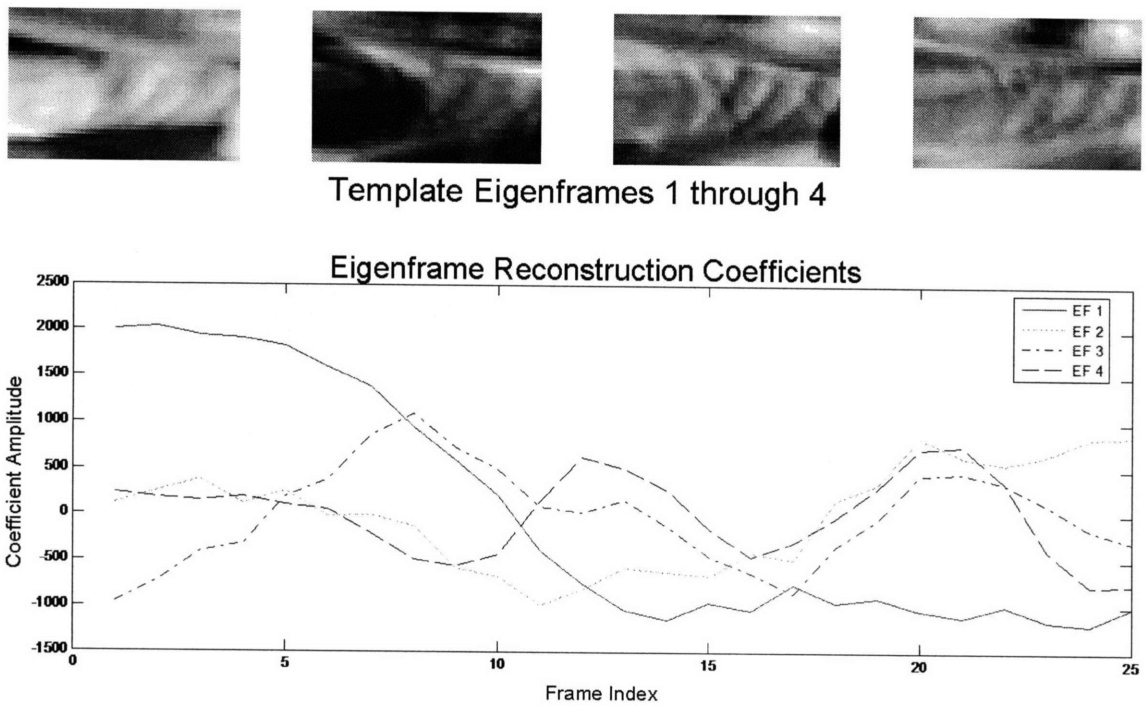


Figure 6-11: Eigenframes and Reconstruction Coefficients for Diaper Packaging Monitoring Example

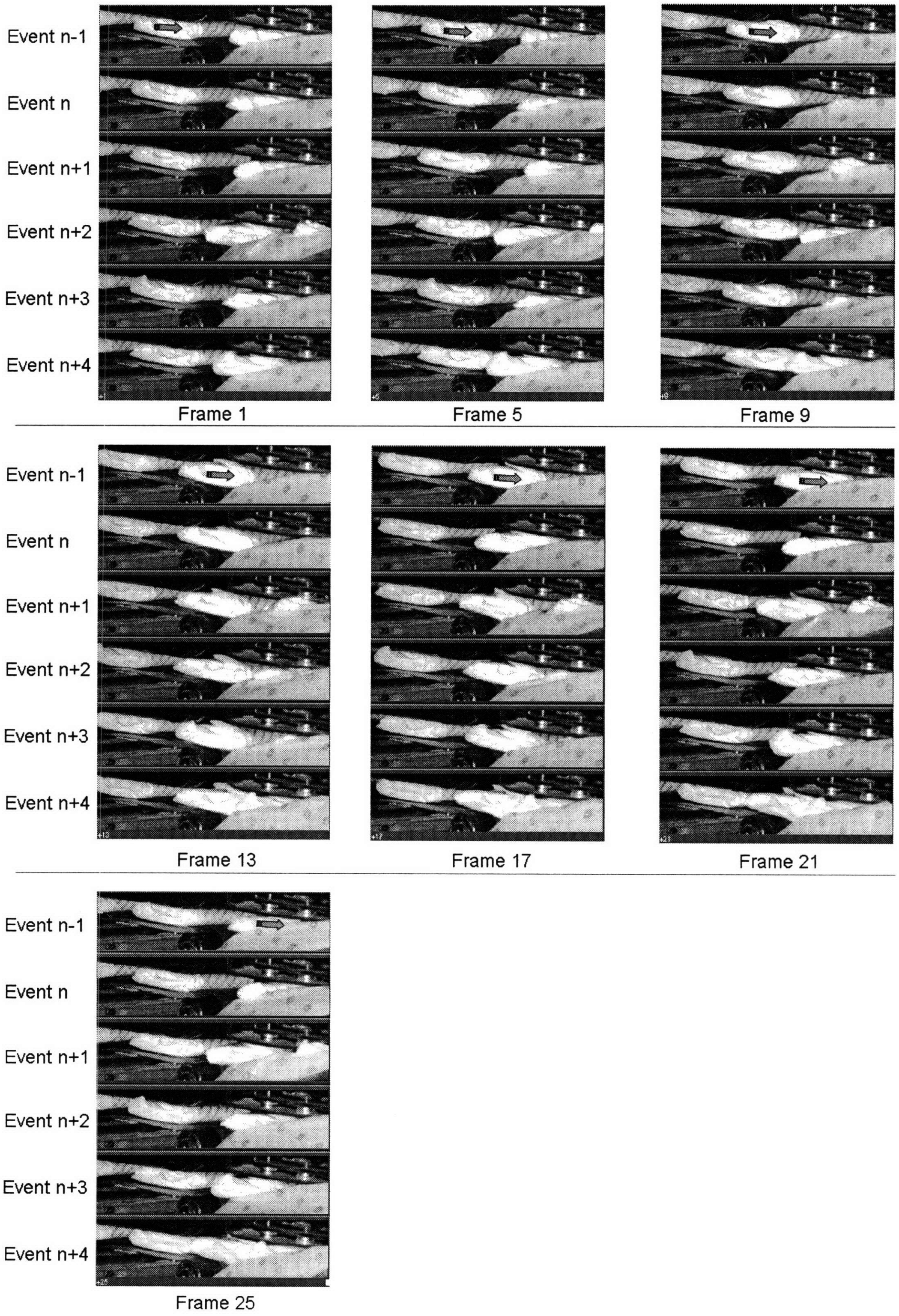
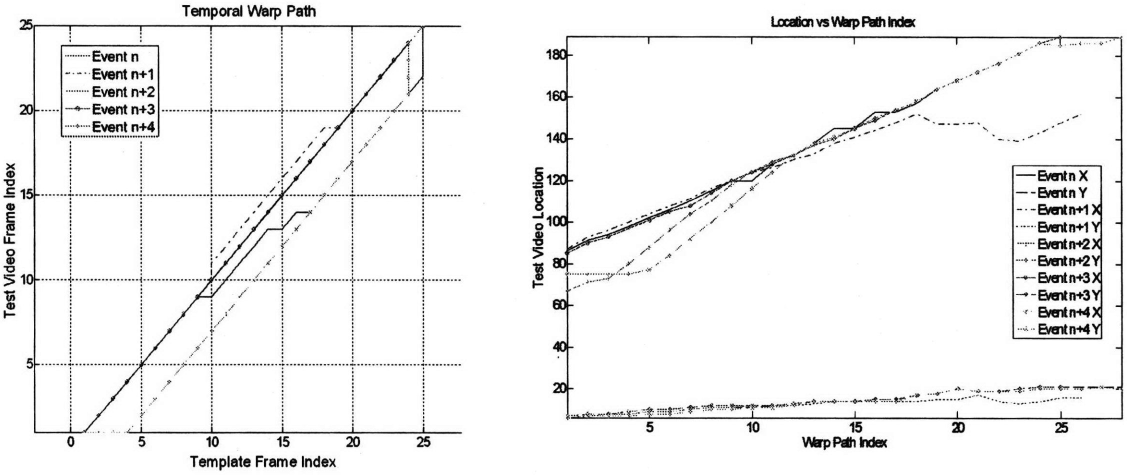
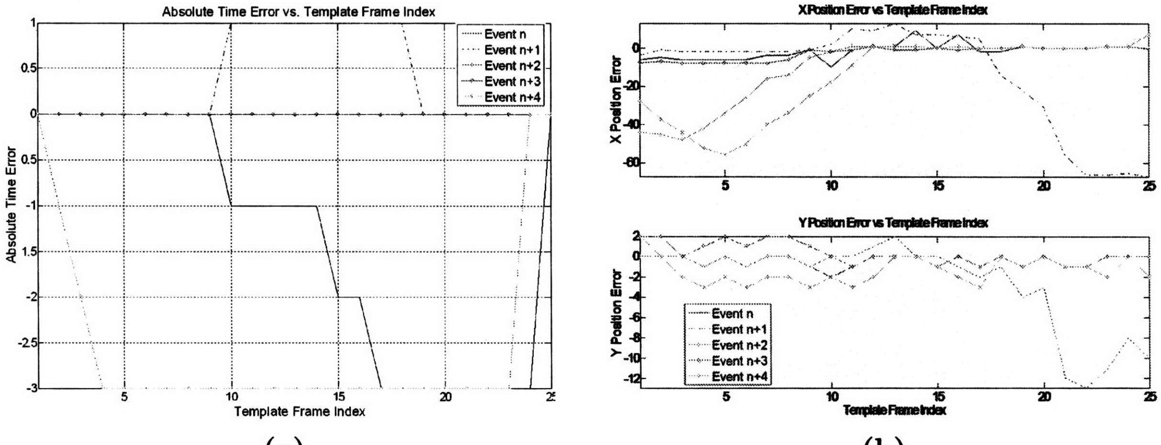


Figure 6-12: Six Events of Diapers Entering Wrapping Machinery



(a) (b)

Figure 6-13: Spatial and Temporal Localization of Last Five Diaper Events Prior to Complete Diaper Jam



(a) (b)

Figure 6-14: Spatial and Temporal Error of Last Five Diaper Events Prior to Complete Diaper Jam

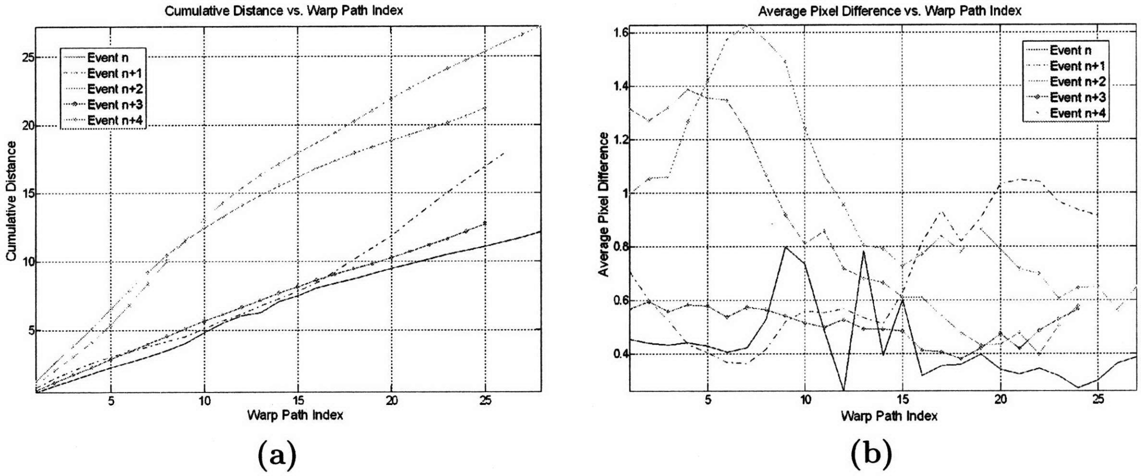


Figure 6-15: Cumulative and Elemental Distances Along the Optimal Warp Path of Last Five Diaper Events Prior to Complete Diaper Jam

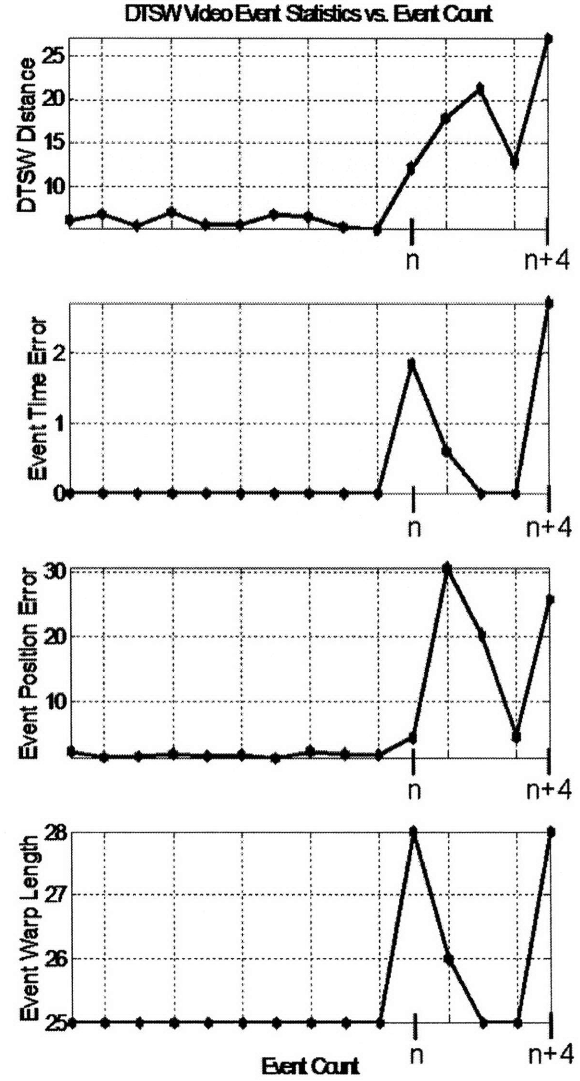


Figure 6-16: Diaper Wrapping Event-Level Statistics: DTSW Distance, L_2 Time Error, L_2 Position Error, Event Warp Length. Full-event (accumulated error over the entire DTSW path) statistics prior to, including, and after the first glitch at **Event n** are plotted in the figure. In order to autonomously monitor and detect faults, we first set a threshold during normal operation and then detect when that threshold is exceeded. We see that it would be easy to select thresholds that detect a problem starting at **Event n** and through **Event n+4**.

Chapter 7

Evaluation, Video Event Detection and Classification

7.1 Introduction

In industrial and scientific application there is often the luxury of planning and setup in order to create imagery that is amenable to analysis. Our experience has been that in most industrial and scientific monitoring applications there are external sensors or timing signals that are used to temporally segment a video stream. Furthermore, cameras are configured to "place" only one template similar event in the field of view of the camera. The test video is temporally clipped and framed into a single event video for test.

DTSW is used to find the singular best path through a test video by matching each template frame to a test frame at some time and some spatial location. The path information (time, location, "appearance") is then analyzed to make decisions about the quality of the event in the test video. DTSW finds the lowest cost path to every spatial alignment of the last frame of the template from the first frame of the test video. DTSW can be used to identify more than one template similar event occurring in the same time duration of a test video. A detailed examination of the Cumulative Distance Hypervolume for a given test video will expose other low cost paths that are similar in appearance and temporally segmented similar to the template.

Video event detection is an important and active area of research for non-industrial applications such as surveillance and human gesture detection. In some applications, such as surveillance, there are no external signals available to temporally clip a continuous stream into short segments. Detection is necessary for those industrial applications that do not have external signals that can be used to temporally segment the video stream into single event clips.

Given an unknown test video sequence, the detection problem is to determine whether the template sequence matches any of the space-time volumes in the video sequence. We compare the template to all sub-volumes. Low dissimilarity is an indication that the sub-video can be classified as "like" the template and indicates a "detection". Generally we do not need the detailed temporal and spatial similarity throughout the volume of the template; instead we just want the single moment in time and space about which the template sequence is located.

In this chapter we discuss how to use the algorithmic machinery of the DTSW algorithm for the video event detection and classification problems. These applications' requirements are different than the requirements for event comparison that we have considered. We are not concerned with the detailed determination of the time and space path that a template takes through the video **C**; instead we are interested in, 'Did the event occur?' , 'Approximately when and where?'; 'Are there multiple events?'. We no longer consider a finite length video, for which we determine a singular warp path; instead we detect the occurrence of multiple events in an infinite video sequence. We then use the detection minimums to discriminate among different templates and demonstrate applicability to classification applications.

7.2 Discussion

We consider a continuous video stream, **S**, and a video event, **Q**, of smaller spatial and temporal extent. We detect the occurrence of the template **Q** in **S**, and report its temporal and spatial location.

DTSW operates on the full spatial extent if the video; next we scan the DTSW machinery over time. A spatial plane of DTSW distances from D_C, e.g. $D_C(I, J, \{x\}, \{y\})$, is stacked into a detection volume, V_D, for each segment. A local minimum in V_D, smaller than some threshold, is flagged as detection. There will be regions of low distance at adjacent locations and times around each local minimum. In this way we compare overlapping and consecutive segments of the continuous video stream to the template sequence. We use the local minimums in V_D to extract temporal segments that are largely similar to the template.

We select a temporal range of frames from **S** larger in number than the template frame length. The increased temporal length is sufficient to accommodate the expected, or permitted, variation between the template and events in the continuous stream. We run DTSW, or a relaxed version of DTSW, in order to compare the template **Q** to the extracted segment **C**. We reduce the computational complexity by only calculating the portion of D_C that covers the expected, or allowed, temporal and spatial deviation from the template. We do not store the entire D_C hypervolume as we do not need to know the detailed path along which the template was found. We only need the DTSW distances found in the last temporal volume of the D_C hypervolume.

In the discussion on last-frame alignments we saw how the DTSW algorithm simultaneously finds the DTSW distances for the test video and for the same video shortened by a few frames by examining $D_C(i < I, J, \{x\}, \{y\})$ after normalizing by the sum of the temporal length of the template, J, and test video, $i < I$. Also we saw how to accommodate variation in the first-frame alignment by not accumulating distance for moves along $j = 0$, for some range of $i, i = 0..i_b$. We use both of these extensions in the detection algorithm.

Figure 7-1 outlines the detection algorithm.

1. Extract a temporal segment from the continuous video stream, **S**, and call that the test video for the DTSW algorithm.

2. Run DTSW (or a modified version of DTSW) in order to compare the template and selected segment.

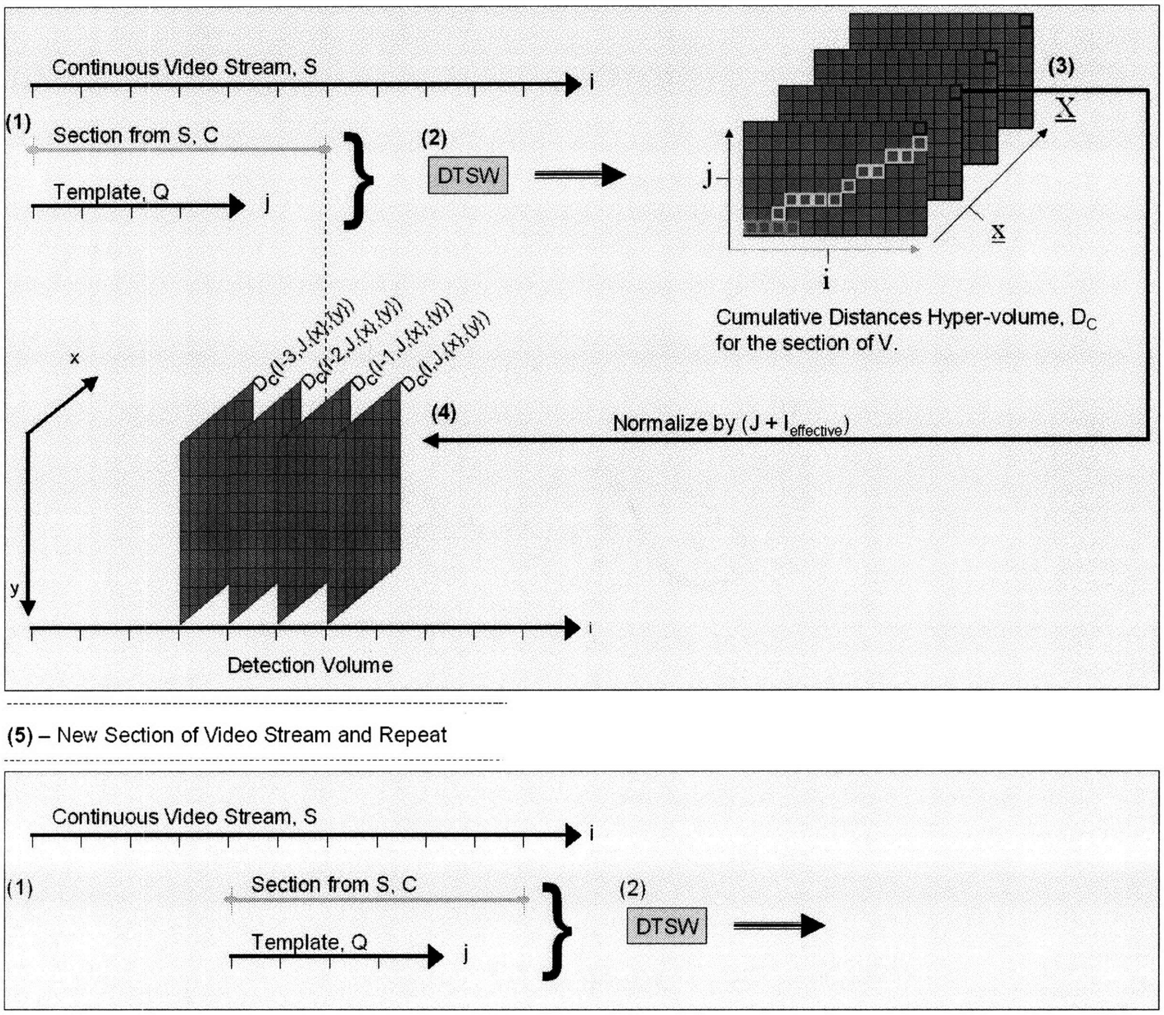


Figure 7-1: Using DTSW for Detection

The complexity of the event will dictate if full DTSW is necessary. If we are interested in detecting events that have moderate to significant temporal variation, then using full DTSW might make sense. If we are only interested in detecting events that have roughly the same temporal duration then the complexity of the problem is decreased.

3. Extract the plane of DTSW distances to every spatial location at $j = J, i = I$ and normalize by (J+I), i.e. $D_C(I, J, \{x\}, \{y\})$. Do the same for $D_C(I - 1, J, \{x\}, \{y\})$, etc.

4. Stack the extracted plane of normalized DTSW distances into a Detection Volume.

5. Select a new section of the continuous video stream and repeat.

This idea is not as computationally expensive as it may seem. The Eigenframes are used to filter the test stream into several Eigenstreams. Then the elemental distance hypervolume D_L for any segment is simply a series of multiplicative and additive combinations of these filtered streams (plus, if we are considering Euclidean distance, the local norm of the test frames and norm of template frames). Application specifics will dictate how much of the variance of the template must be captured by the set of selected Eigenframes. The number of Eigenframes is generally far fewer than the number of frames in the template video. Furthermore, the number of Eigenframes necessary to implement a successful detection application is fewer than the number that is necessary to provide a detailed analysis of the temporal and spatial evolution.

The optimal warp path and cumulative distance hypervolume found for $DTSW(Q, S(i-I, I))$ can be used to guide the DTSW algorithm for $DTSW(Q, S(i - I + 1, I + 1))$.

If the template is J frames long, we'd consider the I frames $I > J$, e.g. $I = J \times 1.1$, frames from the test video stream. Considering a test video longer than the template ensures that we can identify either a longer or a shorter sequence - as DTSW had be modified to handle alternative test video end frames. It is not necessary to run the DTSW algorithm at every time step of the test video stream, but to restart the algorithm every x frames, e.g. $1 <= x <= 1.1 \times J$.

7.2.1 Results

No background segmentation is used in any of these examples. The volumetric data or "pixels" used were various moments of Lukas-Kanade optical flow, (u, v). We experimented with volumetric data moments of v, $v^2 - u^2$, uv, and $v - u$ all with similar results.

Beach

Figure 7-2 shows the results using the DTSW detection algorithm to **find a person walking in a beach video**. The test video, $187 \times 369 \times 77$, is from the web page associated with Shechtman and Irani, 2005 [SI05]. The template video, $86 \times 50 \times 13$, of a clothed person walking in front of a brick wall, is from the web page associated with the work by Blank et al. 2005 [BGS+05]. The template was 13 frames long; 2 Eigenframes capture 50% of the variance. Notice the quasi-periodicity of the detection volume temporal minimums; we capture the irregular gait pattern present in the video, which we assume to be due to the variability of walking on sand. Scale was selected by comparing the output from the DTSW detection algorithm for the template, the template up-sampled by 2 and 4, and the template down-sampled by 2 and 4. The correct scale was selected; which is down-sampled by 2. The scale of the template person is still about 10% different than the scale of the person in the beach video.

Bird

Figure 7-3 shows the results of using the DTSW detection algorithm to find the **wing beats of a bird flying in a wind tunnel**. The template, $328 \times 456 \times 23$, is one beat of the wings extracted from the same video used as test, $760 \times 976 \times 357$. 10 Eigenframes capture 80% of the variance. The temporal minimum of the detection volume is plotted. The center frames of the detected [video] volumes are displayed along with the corresponding portion of the detection volume as an image mask. The deep minimum with zero distance at frame 136 is the template. The reduced amplitude peaks later in the sequence are due to the presence of a large directional bias in the optical flow and the slight rotation of the bird. The template optical flow is not determined from a body-centered sequence; instead it is first calculated in the original sequence when the bird was generally moving down the field of view. Toward the temporal end of the test sequence the bird is angled and predominantly moving from left-to-right. The camera is not normal to the plane of the bird, there is about a 10% scale change for the bird from the beginning to the end of the sequence.

Ballet

Figure 7-4 shows the results of applying the DTSW detection algorithm on a **ballet video**. The test and template videos are from the web page associated with Blank et al. 2005 [BGS$^+$05]. The video is highly compressed 192 × 144 × 750 with a frame rate of 15 fps, moving camera, and changing zoom. The template was 81 × 81 × 10. 4 Eigenframes capture 80% of the variance. There are two dancers one male and one female. The template is the male dancer performing a "cabriole" pan - beating feet together at an angle in the air. We correctly detected all instances of the template, male and female. We had no false positives. We detected the action of the male dancer at frame 276 that is mostly occluded by the female dancer; this event was ignored by the techniques of [BGS$^+$05]. The deep minimum with zero distance at frame 304 is the template. Two events, at frames 468 and 574, would have been detected as false positives for a different threshold. The event at frame 724 is also a relatively shallow local minimum; we, correctly, do not detect it. In [BGS$^+$05] the action centered at this frame is incorrectly labeled as the *"cabriole" pa* and then incorrectly detected; the dancer jumps into the air but does not beat his feet together.

Diving

Figure 7-5 shows the results of applying the DTSW detection algorithm to detect **diving into the pool** in a broadcast video of a swimming competition. The test, 241 × 369 × 1209, and template videos, 57 × 127 × 24, are from the web page associated with Shechtman and Irani, 2005 [SI05]. 6 Eigenframes capture 80% of the variance.

A short section of the much longer sequence is shown in the figure. The displayed section shows one miss, one false positive, and five correct detections. There is one other (temporal) portion of the entire video that has swimmers diving into the pool. Most of the video is the camera following the swimmers the length of the pool and back showing a variety of above and underwater shots, water splashing, and flip-turns. We correctly detect all other template instances; we had no other false positives through the remainder of the video. Like the work in [SI05], we have the same false positive at frame 1068 due to a similar motion pattern in the water. We missed detection at frame 1056. Unlike the work in [SI05], we detect the diver partially occluded by the logo in frame 1128; we also detect the four other divers. DTSW, in conjunction with very noisy optical flow, achieves very good detection results. Increasing the variance to 90% might have reduced the errors.

For all of these examples, the detection threshold was selected to be between 0.65 and 0.70 on a scale from 0 to 1. The results suggest that we are insensitive to scale variation on the order of 15% to 25%.

7.2.2 Detection and Classification

We used a video database from Blank et al, 2005 [BGS+05] in order to systematically investigate detection accuracy, applicability to classification, and classification accuracy. Nine actors perform five actions following a frontal parallel trajectory. The actions are "walking", "skipping", "galloping-sideways" (facing camera), "running", and "jumping-forward-on-two-legs". For the walk, run, and skip actions, one actor performs the actions twice, for a total of 48 sequences. Representative sequences are shown in Figure 7-6. In all sequences there are multiple cycles of the action. From each sequence, we extracted a body-centered sub-sequence that contains one cycle of the action, e.g. one jump or one stride. This creates a set of 48 template videos.

The volumetric data or "pixels" we used were various moment of Lukas-Kanade optical flow, (u, v). We experimented with volumetric data representations v, $v^2 - u^2$, uv, and $v - u$. The template sequences were on average 12 frames long: minimum

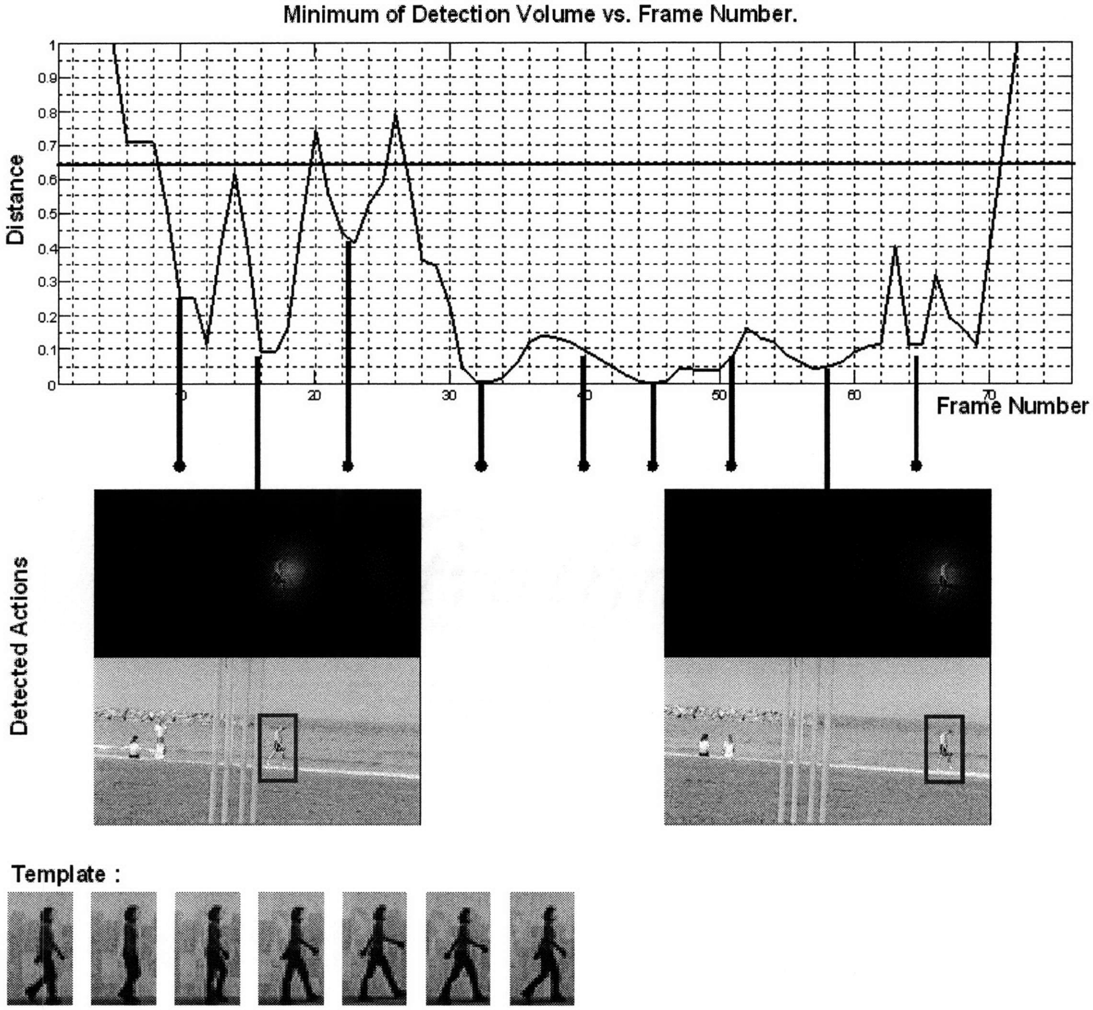


Figure 7-2: Detection of human gait cycle of person walking on a beach.

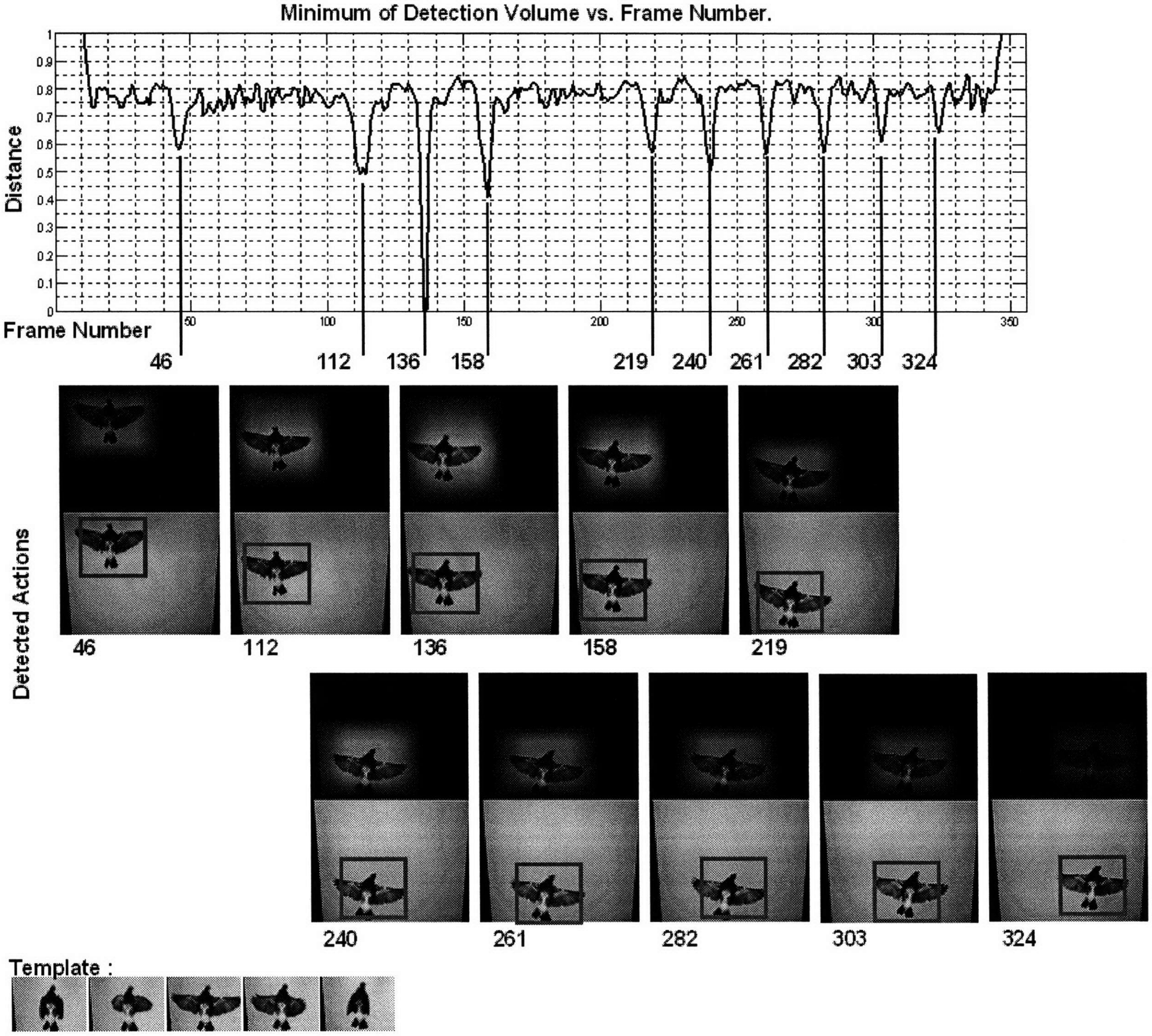
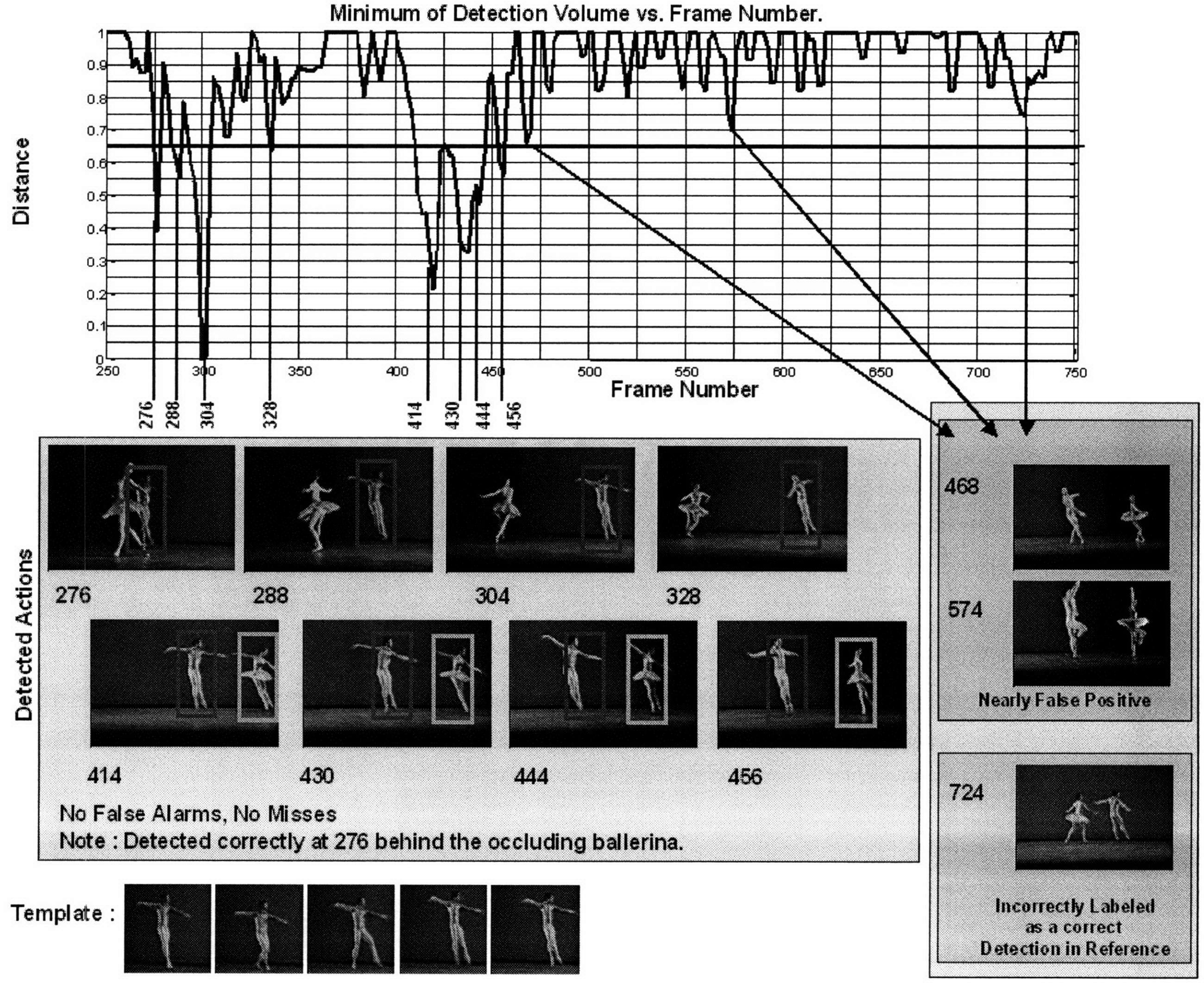


Figure 7-3: Detection of wing beats for a bird flying in a wind tunnel.



No False Alarms, No Misses

Note : Detected correctly at 276 behind the occluding ballerina.

Nearly False Positive

Incorrectly Labeled as a correct Detection in Reference

Figure 7-4: Detection of "cabriole" pa during a ballet.

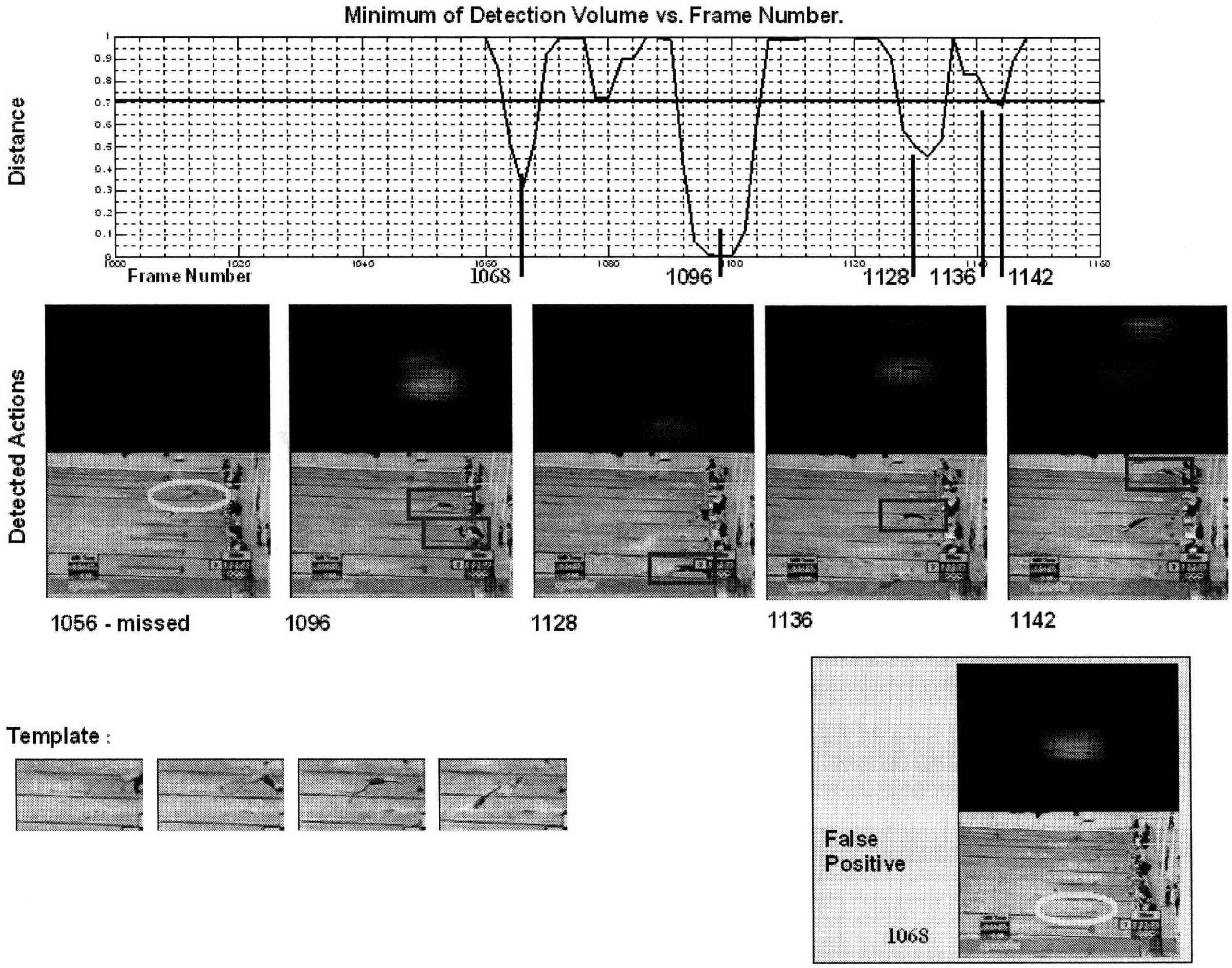


Figure 7-5: Detection of diving into the pool during a swimming competition.

	EF Count		
Variance %	Min	Mean	Max
50	1	1.56	2
60	1	2.08	3
70	2	2.54	4
80	2	3.23	4
90	3	4.98	7

Table 7.1: Number of Eigenframe Versus Retained Variance for Template Videos of Various "Walking" Styles. Volumetric Data Type, $(v-u)$. Down-sampled by a factor of 4.

of 7, maximum of 16. In general, results are presented for a variance threshold of 80%; all frames of the template were simultaneously considered for the purpose of determining the Eigenframes. The template sequences were on average represented by 3.23 Eigenframes: minimum of 2, maximum of 4. Each template was originally 52 by 84 pixels. Each test sequence was originally 180 by 144 pixels. In general, results are presented for sequences that were down-sampled by a factor of 4. We explored sensitivity to captured variance and down-sampling. Table 7.1 summarizes the number of Eigenframes required to represent the template as a function of variance for volumetric data representation $v - u$.

We used 1-Normalized Correlation for the low-level distance function. We ran the DTSW detection algorithm for each of the 48 templates on each of the 48 original sequences. We analyzed the local minimums of the detection volume. The location of each local minimum indicates the time and space where the template is detected. The depth of the minimum indicates how well the template matched the indicated volume.

Detection

For each full sequence, we run the DTSW detection algorithm for each template of the same activity. We leave-out the template originally extracted from the test sequence. We find the local minimums in the detection volume. We record the temporal and spatial locations of the minimums. In this way we classify every location [time and space] in a full sequence as detected or not-detected, i.e. "best local match to the

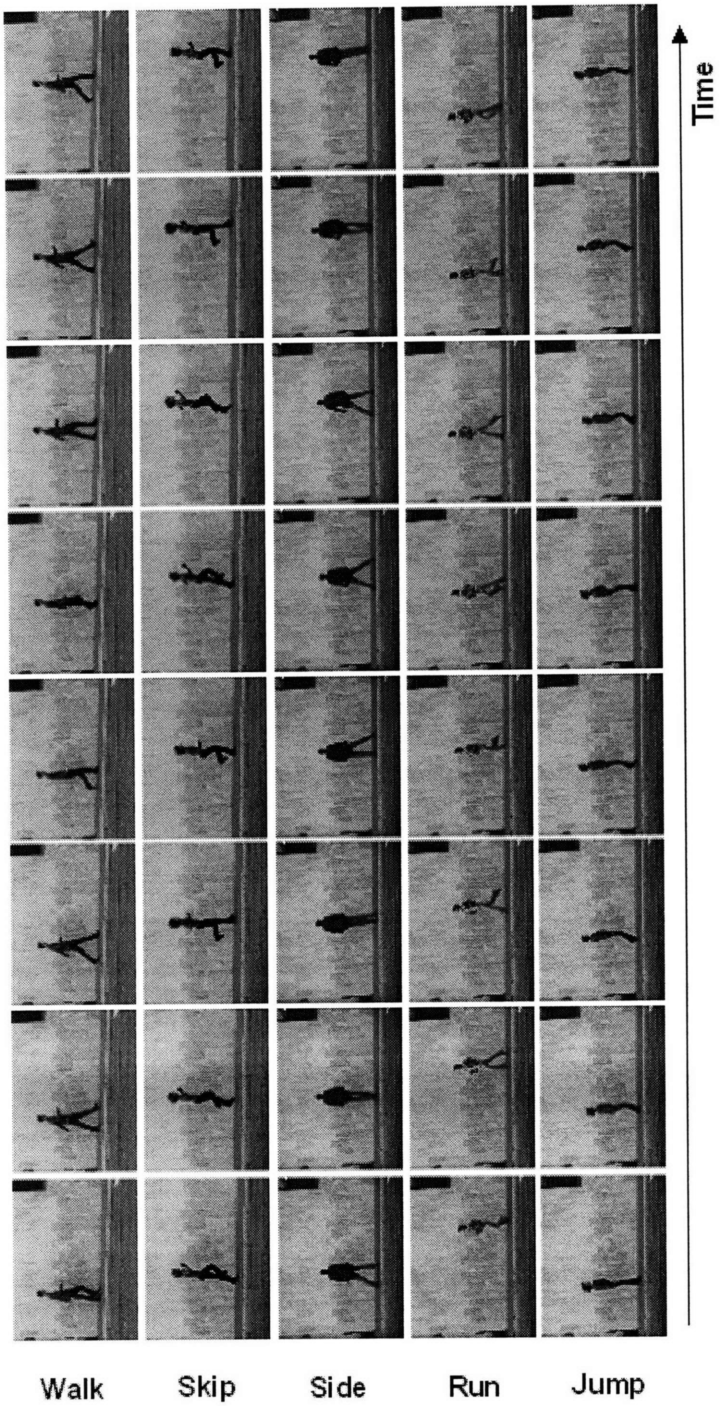


Figure 7-6: Samples from the Video Database used for Classification Experiments

template" or "not the best local match to the template". We determine the rate at which a location in a test sequence is flagged as detected by all templates of the same activity. In order to quantify detection performance, we compare the time and location origin of the template that was extracted from each full sequence to the times and locations detected by all other templates of the same activity. Essentially this is a leave-one-out classification experiment; we determine the rates that locations [time and space] are classified as being the location of the template (detected) or not (not-detected). We compare the local minimums that are closest to the location from where a sequence's template was extracted.

Each full sequence contains several cycles of each activity. Figure 7-7 shows the local minimum of the detection volume versus time for the walking activity for one full "walking" sequence and all "walking" templates. The plot in bold [and red] is for the template that was extracted from the full sequence. This is the combination that is the "truth"; it is the combination that is left out when determining detection rates. The plus signs (+) on the other plots indicate the local minimum that is nearest the local minimum associated with the extracted template.

Tables 7.2 and 7.3 summarize the detection rates, or accuracy, for these 5 activities relative to the known location for each activity. Frame-to-frame motion of the body is on average 6 pixels. The temporal variation in the selection of a cycle of the activity for the templates is about 2 pixels. We report the temporal detection rate and the spatial detection rates. We report the detection rates as a function of the absolute distance between the true template location and the detected location.

For example, the walking activity (the template is from one heal leaving the ground to the next) is detected within 1 frames and 1 pixel at a rate of 99% and 93% respectively. The detection rates were not significantly different across the several volumetric data types. We explored the sensitivity to variance and down sampling. Tables 7.4 and 7.5 summarize the average detection rates for all activities as a function of variance retained in the templates. Tables 7.6 and 7.7 summarize the average detection rates for all activities as a function of spatial down-sampling.

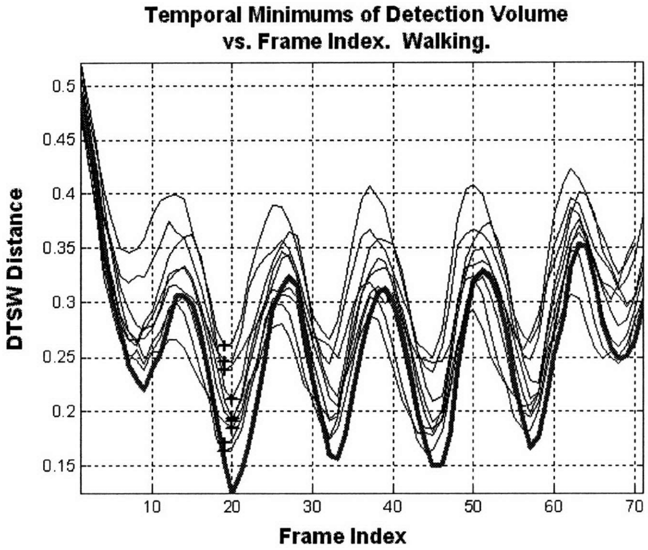


Figure 7-7: Temporal Minimum of Detection Volume vs. Time for the Walking Activity. Each plot is for a different template of the same activity. The plot in bold [and red] is for the template that was extracted from the given test sequence.

	Temporal Detection Offset						
Activity	0	±1	±2	±3	±4	±5	$\pm \geq 6$
Walk	0.58	0.99	0.99	1.00	1.00	1.00	1.00
Skip	0.41	0.86	0.96	0.98	0.98	0.98	1.00
Side	0.42	0.79	0.92	0.99	1.00	1.00	1.00
Run	0.27	0.68	0.87	0.93	0.94	0.96	1.00
Jump	0.28	0.58	0.85	0.94	1.00	1.00	1.00
Average	0.39	0.78	0.92	0.97	0.98	0.99	1.00

Table 7.2: Temporal Detection Rates for DTSW Detection of Various "Walking" Styles. Volumetric Data Type, $(v - u)$. Variance retained in Eigenframes, 80%. Down-sampled by a factor of 4. We use walking to describe this table. Walking cycles are detected correctly 58% of the time with zero time error. Walking cycles are detected correctly 99% of the within 1 frame of their actual temporal occurrence (temporal detection offset).

	Spatial Detection Offset						
Activity	0	±1	±2	±3	±4	±5	$\pm \geq 6$
Walk	0.46	0.93	1.00	1.00	1.00	1.00	1.00
Skip	0.28	0.72	0.91	0.98	1.00	1.00	1.00
Side	0.43	0.92	0.97	0.99	0.99	0.99	1.00
Run	0.36	0.76	0.93	0.97	0.97	0.97	1.00
Jump	0.22	0.78	0.96	1.00	1.00	1.00	1.00
Average	0.35	0.82	0.95	0.99	0.99	0.99	1.00

Table 7.3: Spatial Detection Rates for DTSW Detection of Various "Walking" Styles. Volumetric Data Type, $(v - u)$. Variance retained in Eigenframes, 80%. Down-sampled by a factor of 4. We use walking to describe this table. Walking cycles are detected correctly 46% of the time with zero spatial error. A walking cycles are detected correctly 93% of the within 1 pixel of actual spatial occurrence (spatial detection offset).

		Temporal Detection Offset						
Variance %		0	±1	±2	±3	±4	±5	$\pm \geq 6$
50	Avg. Detection Rate	0.32	0.66	0.79	0.87	0.89	0.92	1.00
60		0.35	0.74	0.87	0.93	0.95	0.97	1.00
70		0.37	0.76	0.90	0.96	0.98	0.99	1.00
80		0.39	0.78	0.92	0.97	0.98	0.99	1.00
90		0.41	0.79	0.94	0.97	0.99	0.99	1.00

Table 7.4: Average Temporal Detection Rates for all Activities as a Function of Variance Retained in the Templates. Volumetric Data Type, $(v - u)$. Down-sampled by a factor of 4.

		Spatial Detection Offset						
Variance %		0	±1	±2	±3	±4	±5	$\pm \geq 6$
50	Avg. Detection Rate	0.26	0.72	0.87	0.93	0.96	0.98	1.00
60		0.32	0.74	0.91	0.97	0.99	0.99	1.00
70		0.32	0.79	0.94	0.98	0.99	0.99	1.00
80		0.35	0.82	0.95	0.99	0.99	0.99	1.00
90		0.33	0.79	0.94	0.98	0.99	1.00	1.00

Table 7.5: Average Spatial Detection Rates for all Activities as a Function of Variance Retained in the Templates. Volumetric Data Type, $(v-u)$. Down-sampled by a factor of 4.

		Temporal Detection Offset						
DS		0	±1	±2	±3	±4	±5	$\pm \geq 6$
none	Avg. Detection Rate	0.27	0.67	0.85	0.94	0.96	0.98	1.00
2		0.32	0.73	0.94	0.97	0.99	0.99	1.00
4		0.39	0.78	0.92	0.97	0.98	0.99	1.00
8		0.14	0.37	0.55	0.63	0.69	0.78	1.00
12		0.13	0.34	0.47	0.57	0.68	0.80	1.00

Table 7.6: Average Temporal Detection Rates for all Activities as a Function of Spatial Down-Sampling (DS). Volumetric Data Type, $(v - u)$. Variance retained in Eigenframes, 80%.

		Spatial Detection Offset						
DS		0.00	±1	±2	±3	±4	±5	$\pm \geq 6$
none	Avg. Detection Rate	0.07	0.19	0.34	0.44	0.56	0.69	1.00
2		0.21	0.52	0.73	0.86	0.91	0.94	1.00
4		0.35	0.82	0.95	0.99	0.99	0.99	1.00
8		0.28	0.67	0.82	0.90	0.95	1.00	1.00
12		0.23	0.70	0.90	0.97	1.00	1.00	1.00

Table 7.7: Average Spatial Detection Rates for all Activities as a Function of Spatial Down-Sampling (DS). Volumetric Data Type, $(v - u)$. Variance retained in Eigenframes, 80%.

Classfication

We used the global minimum of the detection volume for classification. The global minimum is the smallest DTSW distance between the template and every sub-sequence in the test video. We classify every test video using a k-nearest-neighbors procedure based on the similarity to each template. In the procedure, we don't use the template (i.e. leave one-out) that was extracted from the test video. The leave-one-out error rate estimator is an (almost) unbiased estimator of the true error rate of the classifier. For all 48 sequences, we classify it on all but the one template that was extracted from the sequence. This is repeated 48 times. The recognition rate is the ratio of the number of correctly classified test samples out of the total 48.

Tables 7.8 through 7.11 show the confusion[1]matrices for these 5 actions for various volumetric data types. The entries in the matrix indicate what fractions of the labeled sequences were classified as belonging to a particular class. The columns are indicated with the test video labels; the rows with the template labels. The trace of the

Pixel :	$v - u$				
	Walk	Skip	Side	Run	Jump
Walk	1	0	0	0	0
Skip	0	0.6	0	0.2	0.2
Side	0	0	1	0	0
Run	0	0	0	1	0
Jump	0	0	0	0	1
	Trace :		4.60 (92%)		
	Secondary Trace :		4.90 (98%)		

Table 7.8: Confusion matrix for DTSW distance based classification of various "walking" styles. Volumetric Data Type, $(v - u)$. Variance retained in Eigenframes, 80%. Down-sampled by a factor of 4.

confusion matrix is a measure of the accuracy of the classifier. For this experiment, 5 is the perfect classification, whereas 1 would indicate uniformly random classification. We also indicate the trace of a "secondary" confusion matrix where we count a correct classification if either the first or the second best classification is the same as the label. The best classification rate of 92% was found for volumetric data $v - u$; either the first *or second* nearest class is found to be correct at a rate of 98%.

The results do vary for the several moments of optical flow that we investigated. The variation, a trace of 4.6(92%) for $v - u$; 4.19 (84%) for $v^2 - u^2$; 4.06(81%) for v; and 3.54(71%) for uv, indicates some sensitivity to the volumetric data type. An interesting combination of this work and that of Shechtman and Irani, 2005 [SI05], will be to use their rank increase measure as the volumetric data values; this is left for future work. Also, we have limited ourselves to scalar volumetric data values, future work will explore the performances-computation tradeoff if we consider vector valued volumetric data. Table 7.12 summarizes the average classification rates for all activities as a function of variance retained in the templates. Table 7.13 summarizes the average classification rates for all activities as a function of spatial down-sampling.

[1] A confusion matrix shows the level of agreement (or disagreement) between classifications. A confusion matrix is a visualization tool typically used in supervised learning or classification applications. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. High values along the diagonal indicate good classification results. Off diagonal terms indicate incorrect classifications. The trace of the matrix can be used to summarize an overall classification rate. Assume that the data is scaled such that a 1 on the diagonal indicates perfect classification. Then, a confusion matrix for a five-class application would have a trace of 5 for perfect classification.

Pixel :	$v^2 - u^2$				
	Walk	Skip	Side	Run	Jump
Walk	1	0	0	0	0
Skip	0.1	0.3	0	0.4	0.2
Side	0	0	0.89	0.11	0
Run	0	0	0	1	0
Jump	0	0	0	0	1
	Trace :	4.19 (84%)			
	Secondary Trace :	4.70 (94%)			

Table 7.9: Confusion matrix for DTSW distance based classification of various "walking" styles. Volumetric Data Type, $(v^2 - u^2)$. Variance retained in Eigenframes, 80%. Down-sampled by a factor of 4.

Pixel :	v				
	Walk	Skip	Side	Run	Jump
Walk	0.8	0	0	0.2	0
Skip	0	0.7	0	0.3	0
Side	0	0	0.67	0.11	0.22
Run	0	0	0	1	0
Jump	0	0	0	0.11	0.89
	Trace :	4.06 (81%)			
	Secondary Trace :	4.69 (94%)			

Table 7.10: Confusion matrix for DTSW distance based classification of various "walking" styles. Volumetric Data Type, (v). Variance retained in Eigenframes, 80%. Down-sampled by a factor of 4.

Pixel :	uv				
	Walk	Skip	Side	Run	Jump
Walk	0.6	0	0	0.4	0
Skip	0	0.6	0	0.4	0
Side	0	0	0.67	0.33	0
Run	0	0	0	1	0
Jump	0	0.11	0	0.22	0.67
	Trace :	3.54 (71%)			
	Secondary Trace :	4.69 (94%)			

Table 7.11: Confusion matrix for DTSW distance based classification of various "walking" styles. Volumetric Data Type, (uv). Variance retained in Eigenframes, 80%. Down-sampled by a factor of 4.

	Avg. Classification Rate		EF Count		
Variance %	Primary	Secondary	Min	Mean	Max
50	0.44	0.74	1	1.56	2
60	0.75	0.92	1	2.08	3
70	0.83	0.96	2	2.54	4
80	0.92	0.98	2	3.23	4
90	0.90	1.00	3	4.98	7

Table 7.12: Average Classification Rates for all Activities as a Function of Variance Retained in the Templates. Volumetric Data Type, $(v - u)$. Down-sampled by a factor of 4.

	Avg. Classification Rate		EF Count		
DS	Primary	Secondary	Min	Mean	Max
none	0.92	0.98	4	6.77	9
2	0.92	0.98	4	5.81	7
4	0.92	0.98	2	3.23	4
8	0.29	0.50	1	1.69	3
12	0.22	0.41	1	1.60	3

Table 7.13: Average Classification Rate for all Activities as a Function of Spatial Down-Sampling. Volumetric Data Type, $(v - u)$. Variance retained in Eigenframes, 80%.

7.3 Summary

The DTSW has promise in detection and classification applications. DTSW finds the minimum cost path aligning the template video to the test video ending at every spatial location of the last frame of the test sequence. DTSW can be used to simultaneously identify more than one template similar event occurring in the same time duration of a test video. We scan the DTSW machinery through an infinite temporal length video in order to detect video events that are similar to the template. This scanning process generates a detection volume; local minimums in the detection volume potentially correspond to a temporal-spatial match between the template and the corresponding portion of the test video. Various schemes are used to improve the efficiency of this scanning process. Local minimums below a threshold are classified as detected regions of the test video. The local minimum values were used to classify actions amongst similar templates.

Chapter 8

Performance, Computational Complexity, and Quality Templates

8.1 Introduction

The DTSW algorithm may be applied to video detection, video classification, and other video comparison or alignment problems. It is uniquely suited to system monitoring applications because of its ability to provide a detailed time and space comparison between two videos even when the subject matter is amorphous, flexible, and lacking of distinct features or specific volumetric structure.

In this chapter we explore the performance characteristics of DTSW, its computation complexity, and we benchmark DTSW when applied to the detection problems. We discuss how to evaluate and select a "quality" template; this is very important in industrial monitoring applications.

8.2 Factors which effect Performance

The minimum cost path is a four-dimensional valley through the Cumulative Distance Hypervolume. If there is a single valley, no local minimum offshoots, and the valley walls are steep, then the path is more "stable" than if the valley is wide, flat-bottomed, and has multiple low cost offshoots.

Numerous factors and settings effect the performance and therefore the evaluation of the DTSW Algorithm. The effect that many of these factors have on the DTSW algorithm can be explored in the context of how to alter the shape-and-structure of the Elemental and Cumulative Distance Hypervolumes.

The factors which effect performance include :

- Temporal Size :

  Length of Test Video, **C**.

  Length of the Template Video, **Q**.

  Relative length between the two.

- Smoothness of template position over time.

- Spatial Size :

  Spatial Size of a frame of **Q**.

  Spatial Size of a frame of **C**.

  The relative size between the two.

- The temporal evolution of the template **Q** one constant image vs. time, or significant variation from frame to frame.

- Appearance :

  Statistics (similarity) of background compared to the object.

  Degradation of the test video compared to the original source of the template.

  Scale changes or aspect ratio effects.

- Algorithm Settings :

  Elemental Distance Function, e.g. Euclidean, 1-Normalized Correlation.

  Low-Level Volumetric Data Used, e.g. pixel value, optical flow, other.

  Percentage of Variance preserved in the Eigenframes.

Spatial Continuity Limit.

Temporal Continuity Limit.

8.2.1 Simulation - Generation of Synthetic Videos

In order to evaluate the DTSW algorithm, we performed experiments on randomly generated videos.

The tests were performed as follows:

- Create a template video Q_t (of various sizes and varying levels of temporal evolution).
- Randomly stretch and compress the time axis of the template. Store the randomly generated time modification sequence, $j_t(i_t)$ of length I.
- Create a spatially-larger frame sequence of temporal length I that has similar frame statistics as the template. This sequence will eventually become the test video, $\mathbf{C_t}$, for the DTSW algorithm.
- On each large video frame, randomly locate in space, a small template frame according to the time modification sequence. Store the randomly selected spatial placements, $x_t(i_t)$ and $y_t(i_t)$.
- Add random noise to each frame of the resulting video, this is C_t, the test video.
- Use DTSW to find the template $\mathbf{Q_t}$ in the $\mathbf{C_t}$ video. Compare the time and space locations output by the algorithm with those stored when warping, placing, and distorting $\mathbf{Q_t}$ to get $\mathbf{C_t}$.

We randomly generate the first frame of the template sequence. We assume a gray-scale 8 bit sensor; pixel values range from 0 to 255. Each pixel is selected from a uniform discrete distribution ranging from 0 to 255. We generate the remaining frame of the template sequence starting with this first frame. We randomly select P_E percent of the pixels in frame j and change them by up to 25% of full scale to create frame $j + 1$.

We then degrade the template sequence by P_D percent. We randomly select R percent of the pixels in each frame and modify them by up to 25% of full scale. We generate the "background" of the unknown sequence in the same way, with the same appearance statistics. We generate a random spatial and temporal path. We place the degraded template sequence into the background according to this path. The template is visually indistinguishable from the background. This is obviously an extreme situation and is not expected in practice.

We run DTSW using the original un-degraded signal as template. We find the spatial-temporal warping that aligns the template in the unknown. The unknown contains a degraded version of the template. We compare the path DTSW finds with the path that we used to place the degraded template into the background. In the following experiments the template sequence is 100 frames long. The test sequence is randomly selected to be between 80 and 120 frames long. The template frame size is $N \times N$ and the test sequence frame size is $M \times M$. We use $M = 4N$.

8.2.2 Simulation - Evaluation of DTSW Event Statistics on Synthetic Videos

We vary the template size, N; the temporal evolution percentage, P_E; and the degradation percentage, P_D. For each N, P_E, P_D combination we perform the above experiment 100 times. We determine the average and maximum event statistics for all 100 trials. The results are presented in Tables 8.1 through 8.4.

For fixed N and fixed P_D the temporal and spatial errors decrease (reading across each table) as the template temporal evolution percentage, P_E, increases. This is expected. The more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger temporal evolution percentage makes adjacent frames more unique.

For fixed P_E and fixed P_D the temporal and spatial errors decrease (reading down each table) as the template size increases. This is expected as well. Again, the more unique the frames of the template, the more likely that degraded versions of these

	Frame Evol. P_E	0.1	0.25	0.5
	Degr. P_D	0.1	0.1	0.1
Size, N	**Event Stats**			
5 × 5	DTSW Dist.	167.71	168.45	159.41
11 × 11	DTSW Dist.	93.83	111.43	109.55
25 × 25	DTSW Dist.	43.68	46.15	45.39
5 × 5	RMS Pos. Error	1.15	0.75	0.18
11 × 11	RMS Pos. Error	0.90	0.74	1.23
25 × 25	RMS Pos. Error	0.47	0.43	0.56
5 × 5	RMS Time Error	0.39	0.20	0.12
11 × 11	RMS Time Error	0.18	0.63	0.25
25 × 25	RMS Time Error	0.29	0.20	0.19

Table 8.1: DTSW Event Statistics for 100 Synthetic Video Trials versus Template Evolution Percentage, P_E and Template Frame Size, $N \times N$. Degradation Percentage, $P_D = 10\%$. **Reading across the table:** For fixed N and fixed P_D the temporal and spatial errors decrease as the template temporal evolution percentage, P_E, increases. This is expected. The more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger temporal evolution percentage makes adjacent frames more unique. **Reading down the table:** For fixed P_E and fixed P_D the temporal and spatial errors decrease as the template size increases. This is expected as well. Again, the more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger frame size tends to make a frame more unique. **Reading from one table to the next:** For fixed P_E and fixed N the temporal and spatial errors generally remain constant. The ability of DTSW to robustly find the correct alignment paths is demonstrated by the fact that for moderate frame sizes the temporal and spatial event errors hold constant even for large degradation percentages. As we expect the best templates are those that are both spatially and temporally unique.

	Frame Evol. P_E	0.1	0.25	0.5
	Degr. P_D	0.25	0.25	0.25
Size, N	Event Stats			
5 x 5	DTSW Dist.	286.37	299.40	290.46
11 x 11	DTSW Dist.	141.54	133.88	144.67
25 x 25	DTSW Dist.	67.72	62.52	68.74
5 x 5	RMS Pos. Error	1.55	1.30	1.11
11 x 11	RMS Pos. Error	0.73	0.56	0.92
25 x 25	RMS Pos. Error	0.46	0.35	0.52
5 x 5	RMS Time Error	0.41	0.31	0.18
11 x 11	RMS Time Error	0.44	0.14	0.20
25 x 25	RMS Time Error	0.66	0.20	0.22

Table 8.2: DTSW Event Statistics for 100 Synthetic Video Trials versus Template Evolution Percentage, P_E and Template Frame Size, $N \times N$. Degradation Percentage, $P_D = 25\%$. **Reading across the table:** For fixed N and fixed P_D the temporal and spatial errors decrease as the template temporal evolution percentage, P_E, increases. This is expected. The more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger temporal evolution percentage makes adjacent frames more unique. **Reading down the table:** For fixed P_E and fixed P_D the temporal and spatial errors decrease as the template size increases. This is expected as well. Again, the more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger frame size tends to make a frame more unique. **Reading from one table to the next:** For fixed P_E and fixed N the temporal and spatial errors generally remain constant. The ability of DTSW to robustly find the correct alignment paths is demonstrated by the fact that for moderate frame sizes the temporal and spatial event errors hold constant even for large degradation percentages. As we expect the best templates are those that are both spatially and temporally unique.

	Frame Evol. $\mathbf{P_E}$	0.1	0.25	0.5
	Degr. $\mathbf{P_D}$	0.5	0.5	0.5
Size, N	**Event Stats**			
5 × 5	DTSW Dist.	396.31	400.18	401.21
11 × 11	DTSW Dist.	194.76	198.19	197.86
25 × 25	DTSW Dist.	87.12	88.52	89.15
5 × 5	RMS Pos. Error	1.19	1.24	1.26
11 × 11	RMS Pos. Error	1.04	0.94	1.02
25 × 25	RMS Pos. Error	0.57	0.44	0.53
5 × 5	RMS Time Error	0.98	0.35	0.20
11 × 11	RMS Time Error	0.66	0.44	0.19
25 × 25	RMS Time Error	0.26	0.23	0.15

Table 8.3: DTSW Event Statistics for 100 Synthetic Video Trials versus Template Evolution Percentage, P_E and Template Frame Size, $N \times N$. Degradation Percentage, $P_D = 50\%$. **Reading across the table:** For fixed N and fixed P_D the temporal and spatial errors decrease as the template temporal evolution percentage, P_E, increases. This is expected. The more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger temporal evolution percentage makes adjacent frames more unique. **Reading down the table:** For fixed P_E and fixed P_D the temporal and spatial errors decrease as the template size increases. This is expected as well. Again, the more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger frame size tends to make a frame more unique. **Reading from one table to the next:** For fixed P_E and fixed N the temporal and spatial errors generally remain constant. The ability of DTSW to robustly find the correct alignment paths is demonstrated by the fact that for moderate frame sizes the temporal and spatial event errors hold constant even for large degradation percentages. As we expect the best templates are those that are both spatially and temporally unique.

	Frame Evol. P_E	0.1	0.25	0.5
	Degr. P_D	1	1	1
Size, N	**Event Stats**			
5 x 5	DTSW Dist.	550.92	576.77	560.40
11 x 11	DTSW Dist.	262.28	263.60	270.70
25 x 25	DTSW Dist.	121.30	120.28	123.60
5 x 5	RMS Pos. Error	1.55	1.54	0.67
11 x 11	RMS Pos. Error	1.13	0.86	1.06
25 x 25	RMS Pos. Error	0.53	0.58	0.36
5 x 5	RMS Time Error	1.54	0.67	0.32
11 x 11	RMS Time Error	1.09	0.48	0.39
25 x 25	RMS Time Error	1.30	0.20	0.48

Table 8.4: DTSW Event Statistics for 100 Synthetic Video Trials versus Template Evolution Percentage, P_E and Template Frame Size, $N \times N$. Degradation Percentage, $P_D = 100\%$. **Reading across the table:** For fixed N and fixed P_D the temporal and spatial errors decrease as the template temporal evolution percentage, P_E, increases. This is expected. The more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger temporal evolution percentage makes adjacent frames more unique. **Reading down the table:** For fixed P_E and fixed P_D the temporal and spatial errors decrease as the template size increases. This is expected as well. Again, the more unique the frames of the template, the more likely that degraded versions of these frames will be correctly located in space and time. A larger frame size tends to make a frame more unique. **Reading from one table to the next:** For fixed P_E and fixed N the temporal and spatial errors generally remain constant. The ability of DTSW to robustly find the correct alignment paths is demonstrated by the fact that for moderate frame sizes the temporal and spatial event errors hold constant even for large degradation percentages. As we expect the best templates are those that are both spatially and temporally unique.

frames will be correctly located in space and time. A larger frame size tends to make a frame more unique.

For fixed P_E and fixed N the temporal and spatial errors generally remain constant (reading from one table to the next). The ability of DTSW to robustly find the correct alignment paths is demonstrated by the fact that for moderate frame sizes the temporal and spatial event errors hold constant even for large degradation percentages. As we expect the best templates are those that are both spatially and temporally unique.

8.3 Computational Complexity

For this discussion we assume that $\mathbf{C}$ is I frames long and each frame is $M \times M$ and that $\mathbf{Q}$ is J frames long each is $N \times N$. The spatial continuity bounds are plus and minus $S/2$ pixel in both the x and y directions. The temporal continuity bounds are minus 0 or 1 frame in both the time of Q and the time of C.

The computational cost of correlating C_i and Q_j using the forward and inverse Fast Fourier Transform (FFT) is $12M^2 \log_2(M)$ real multiplications $18M^2 \log_2(M)$ real additions[1]. The computational cost is M^2N^2 multiplications and additions if implemented through a direct method. This ignores the factor of 2 gain associated with using real values, and the fact that we only want the central, valid, portion of the correlation. If $M >> N$ then the direct method is faster than the transform method. We've also ignored the cost of both calculating the local norms of Q_j and C_i and normalizing. We simply reference the computational cost of the correlation implemented with the FFT.

The computational cost of a Minimum Filter for a signal that is $L \times L$ and an

[1] A typical bit reversed radix-2 FFT of a complex $M \times M$-point signal requires $\frac{M^2}{2} \log_2(M^2)$ complex multiplications and $M^2 \log_2(M^2)$ complex additions. A complex multiplication requires four real multiplications and two real additions. A complex addition requires two real additions. And cost of an inverse FFT (IFFT) is the same as the forward FFT. The correlation computation requires two $M \times M$-point FFT, M^2 multiplies, and one $M \times M$-point IFFT resulting in $12M^2 \log_2(M) + M^2$ real multiplies and $18M^2 \log_2(M)$ real additions.

When the input is real the $M \times M$-point FFT can calculated with a complex $\frac{M \times M}{2}$ FFT plus $\frac{M \times M}{2}$ extra complex multiplies and $\frac{M \times M}{2}$ complex additions.

all 1's filter kernel that is $S \times S$ is S^2L^2 real compares; compares are as expensive as additions. The computational cost of a Bi-Causal Minimum Accumulation Filter for a signal that is $L \times P$ and an all 1's filter kernel that is $P \times P$ is P^3L real compares plus L additions.

Table 8.5 summarizes the computational cost for the non-optimized DTSW algorithm.

For one C_i frame-step of DTSW, the J correlations can be performed simultaneously, then the J MFs can be performed simultaneously, finally followed by the $(M-N-1)^2$ MAs also simultaneously. We could use up to a maximum of either J or $(M-N-1)^2$ processors for straight forward parallelization. The computational cost for computing DTSW in a detection application, i.e. shifting one time-step of a continuous test video per DTSW evaluation, is reduced because we calculate all but the last frame of the elemental distances hypervolume, D_L, from the previous scan location.

Consider a template, $\mathbf{Q}$, that is $25 \times 25 \times 80$ and a test video, $\mathbf{C}$ that is $200 \times 200 \times 100$, $M = 200, N = 25, I = 100, J = 80$. We allow up to plus or minus 2 pixels of motion in either direction from frame to frame, $S = 5$. We allow local time distortion of 0 or 1 frame in both the time of $\mathbf{Q}$ and the time of $\mathbf{C}$, $P = 2$. A moderate digital signal processor (DSP) ,e.g. TMS320DM67x family from Texas Instruments, or general purpose processors, e.g. a Pentium M, can readily sustain 1200 Million Multiply Accumulate Cycles per Second (MMACS). Digital signal processing platforms that contain 10 such DSPs are commercially available.

Table 8.6 summarizes the computational cost, time, and rates for this scenario. The results are consistent with the results that we have seen on a Pentium M 2.13 GHz Dell Latitude D810 Laptop.

8.3.1 Algorithm Optimization and Efficiencies

Representing the template as a series of Eigenframes provides a significant cost reduction without hurting the temporal and spatial path finding. Assume that E Eigenframes capture 80% of the variance in $\mathbf{Q}$. The cost of the J correlations becomes,

		Cost	
Calculation		**Real Multiplies**	**Real Additions or Compares**
Basic Operations			
	Correlation	$12M^2log_2(M)$	$18M^2\log_2(M)$
	MF. Each spatial plane is $(M-N+1)\times(M-N+1)$. The MF kernel is $S \times S$.		$(M-N+1)^2 \cdot S^2$
	MA The temporal edge is $P \times J$. The MA kernel is $P \times P$		$J(P^3 \text{ compares} + 1 \text{ add})$
DTSW step for 1 New Frame of C			
	J Correlations	J Correlations	J Correlations
	For every j, $j \in \{0..(J-1)\}$. MF the spatial planes.		J MFs
	For every $\{x, y\}$, $x, y \in \{0..(M-N)\}$. MA leading temporal edges.		$(M-N+1)^2$ MAs
DTSW - All I frames of **C**			
	I DTSW Steps	IJ Correlations	$I \cdot (J$ Correlations $+ J$ MFs $+(M-N+1)^2$ MAs)
DTSW - All I frames of **C** if scanning.			
	I DTSW Steps	J Correlations	J Correlations $+ I \cdot J$ MFs $+I \cdot (M-N+1)^2$ MAs)

Table 8.5: Non-optimized Computational Cost for DTSW

		Cost		1 Proc.		10 Proc.	
Calculation		Multiply	Add / Compare	Time (sec)	Rate (Hz)	Time (sec)	Rate (Hz)
Basic Operations							
	Correlation	3.67E+06	5.50E+06				
	MF		7.74E+05				
	MA		7.20E+02				
DTSW step for 1 New Frame of **C**							
	J Correlations	2.94E+08	4.40E+08				
	J MF		6.20E+07				
	$(M - N + 1)^2$ MA		2.23E+07				
	Total	2.94E+08	5.25E+08	0.34	2.93	0.03	29.34
DTSW - All I frames of **C**							
	I DTSW Steps	2.94E+10	5.25E+10	34.09	0.03	3.41	0.29
DTSW - All I frames of **C** if scanning							
	I DTSW Steps	2.94E+08	8.87E+09	3.82	0.26	0.38	2.62

Table 8.6: Example Non-optimized Computational Cost for DTSW.$M = 200, N = 25, I = 100, J = 80, S = 5, P = 2$. Processor 1200 MMACS. For these settings one pass of the DTSW algorithm requires 2.94E+10 multiplies and 5.25E+10 additions (and/or compares). With one processor it will take about 34 seconds to compare the test video, **C**, to the template **Q**. With 10 processors the DTSW algorithm is easily run in parallel and would take 3.4 seconds. Notice that if we scan the DTSW machinery, as in a detection application, we experience an order of magnitude decrease in computation cost per DTSW iteration. This is due to the temporal overlap associated with extracting consecutive overlapping segments from a video stream.

	Variance Preserved :	100	80	70	60	50
	Number of Filters	178	**10**	6	4	3
Appearance						
	DTSW Distance	0.00	56.48	69.11	80.45	86.08
Position						
	RMS Error	0.00	0.00	0.19	0.53	0.97
	Max Error	0.00	0.00	**9.84**	9.84	9.84
	Std. Errror	0.00	0.00	0.86	1.47	2.01
Time						
	RMS Error	0.00	0.17	0.11	**1.37**	1.63
	Max Error	0.00	1.00	1.00	**6**	6.00
	Min Error	0.00	0.00	0.00	**-4**	-4.00
	Std. Errror	0.00	0.17	0.11	1.37	1.63
Path						
	Path Length	178	179	179	189	189
	Relative Path Length	1.00	1.01	1.01	1.06	1.06
	Length Q	178	178	178	178	178
	Length C	178	178	178	178	178

Table 8.7: Eigenframe count, DTSW distance, and errors for the Karate example as a function of template variance reduction. Position and time errors increase only slightly when using the first 10 Eigenframes instead of all 178 original frames. If we use fewer Eigenframes the position and time errors start to increase; the onset of significant error increase as the number of Eigenframes decrease are in bold.

E correlations $+ JM^2E$ multiplications $+ JM^2(E-1)$ additions. We have not seen worse than a 50% reduction from J to E filters, i.e $E \leq 0.5J$, for 80% variance, in any of the applications that we have explored. Generally, however, the reduction is far more significant, i.e $E \leq 0.1J$. Table 8.7 summarizes the DTSW accuracy for the Karate example as a function of variance reduction when re-finding the template in its source. Table 8.8 summarizes the DTSW accuracy for the Karate example as a function of variance reduction when re-finding the template in its source.

An alternative or further efficiency arises from the fact that we can generally narrow the temporal path to lay within a range along the temporal diagonal path, i.e. uniform scaling of time axes between template and test. The DTSW algorithm is easily modified to principally follow the temporal diagonal and limit temporal matching globally, to within plus or minus 10%, for example, of the template time duration. This is computationally equivalent to using a template that is $J' = 0.2J$.

Table 8.9 summarizes the computational cost for the example assuming that $E =$

	Variance Preserved :	100	98	95	90	80	60	50
	Number of Filters	46	20	13	8	4	2	1
Appearance								
	DTSW Distance	0.00	13.68	15.07	17.32	20.84	24.91	27.73
Position								
	RMS Error	0.00	0.08	0.08	0.25	0.47	0.82	0.90
	Max Error	0.00	0.92	0.92	1.25	2.56	2.56	3.09
	Std. Errror	0.00	0.55	0.55	0.67	0.81	0.90	0.93
Time								
	RMS Error	0.00	0.00	0.00	**0.26**	0.68	0.97	2.38
	Max Error	0.00	0.00	0.00	1.00	1.00	1.00	3.00
	Min Error	0.00	0.00	0.00	0.00	0.00	-1.00	-1.00
	Std. Errror	0.00	0.00	0.00	0.25	0.50	0.85	1.71
Path								
	Path Length	46	46	46	47	48	49	52
	Relative Path Length	1.00	1.00	1.00	1.02	1.04	1.07	1.13
	Length Q	46	46	46	46	46	46	46
	Length C	46	46	46	46	46	46	46

Table 8.8: Eigenframe count, DTSW distance, and errors for the Heart-Valve example as a function of template variance reduction. Position and time errors increase only slightly when using the first 13 Eigenframes instead of all 46 original frames. If we use fewer Eigenframes the position and time errors start to increase; the onset of significant error increase as the number of Eigenframes decrease are in bold.

10. Table 8.10 summarizes the computational cost for the example assuming that $E = 10$ and $J' = 0.2J$. The TMS320DM64x line of DSPs from Texas Instruments can achieve 8000 MMACS. Table 8.11 summarizes the computational cost, time, and rates for the example using both the Eigenframe and temporal diagonal path options for this faster processor. This is an example of the computational power that will be used for industrial applications.

There are further efficiencies that may be appropriate. The suitability of these modifications is made on an application specific basis. It is easy to extend the DTSW machinery to principally follow a prescribed temporal and spatial path and to find the best path deviated from the prescription. A conceptually related scheme is to narrow the DTSW temporal and spatial search range after we have matched the temporal beginning of the template if we have found a dominant path that is significantly lower cost than any other path in D_C. Down sampling and other mutli-scale efficiencies are also possible; these extensions will be discussed in future work. For a detection application the IJ MFs and $I(M-N+1)^2$ MAs have to be repeated every time step in order to regenerate D_C. However, preliminary work suggests that there is an efficient mechanism to update the D_C when scanning the DTSW machinery[2].

8.4 Quality Templates

We want to aid the individual who is using DTSW in the selection of a quality template. Knowing the "quality" of a template will allow the engineer to prepare the subject matter differently, to adjust lighting, or to adjust acquisition or processing parameters.

The notion of a quality template can be asked in several ways. How stable is a warp path given local changes in the template or the test video? Will the template be found along an alternative warp path if there were small changes to the template or test video? Are there alternative (or bifurcated) warp paths that would be selected given a small change to the template or test video. How steep is the cumulative distance valley perpendicular to the warp path?

We explore some of these questions with the heart valve example. We ran DTSW on the heart valve template and the video from which it was originally extracted. We examined the Cumulative Distance Hypervolume, D_C , to quantify the quality of the template video. We calculated the partial derivative of D_C along the optimal warp path with respect to the temporal dimension, j, and with respect to the spatial dimension, x.

The temporal beginning and end of the template sequence when the valve flaps are fully closed is not as spatially unique as the remainder of the video. The template at the temporal ends is a relatively uniform gray field extracted from a larger relatively uniform gray field. The template is not unique in these regions. In the center of the video when the valve is open, the uniquely shaped and colored (dark and light

[2] Remember, the computational cost for computing DTSW in a detection application, i.e. shifting one time-step of a continuous test video per DTSW evaluation, is already reduced because we calculate all but the last frame of the elemental distances hypervolume, D_L, from the previous scan location.

		Cost		1 Proc.		10 Proc.	
Calculation		Multiply	Add / Compare	Time (sec)	Rate (Hz)	Time (sec)	Rate (Hz)
Basic Operations							
	Correlation	3.67E+06	5.50E+06				
	MF		7.74E+05				
	MA		7.20E+02				
DTSW step for 1 New Frame of **C**							
	E Correlations	3.67E+07	5.50E+07				
	J MF		6.20E+07				
	$(M-N+1)^2$ MA		2.23E+07				
	Total	3.67E+07	1.39E+08	0.07	13.64	0.01	136.38
DTSW - All I frames of **C**							
	I DTSW Steps	3.67E+09	1.39E+10	7.33	0.14	0.73	1.36
DTSW - All I frames of **C** if scanning							
	I DTSW Steps	3.67E+07	8.48E+09	3.55	0.28	0.35	2.82

Table 8.9: Example Computational Cost for DTSW.$M = 200, N = 25, I = 100, J = 80, S = 5, P = 2$. Processor 1200 MMACS. 10 Eigenframes. For these settings one pass of the DTSW algorithm requires 3.67E+09 multiplies and 1.39E+10 additions (compared to 2.94E+10 multiplies and 5.25E+10 additions if not using an Eigenframe representation). With one processor it will take about 7.3 seconds (compared to 34 seconds if not using an Eigenframe representation) to analyze the test video, **C**, against the template **Q**. With 10 processors the DTSW algorithm is easily run in parallel and would take 0.73 seconds. Notice that if we scan the DTSW machinery, as in a detection application, we experience a decrease in computation cost per DTSW iteration. The net improvement when scanning isn't as significant in this case since when we shift one time step we still have to calculate an entire new plane of the elemental and cumulative distances hypervolumes.

		Cost		1 Proc.		10 Proc.	
Calculation		Multiply	Add / Compare	Time (sec)	Rate (Hz)	Time (sec)	Rate (Hz)
Basic Operations							
	Correlation	3.67E+06	5.50E+06				
	MF		7.74E+05				
	MA		1.44E+02				
DTSW step for 1 New Frame of **C**							
	E Correlations	3.67E+07	5.50E+07				
	J MF		1.24E+07				
	$(M - N + 1)^2$ MA		4.46E+06				
	Total	3.67E+07	7.19E+07	0.045	22.104	0.005	221.041
DTSW - All I frames of **C**							
	I DTSW Steps	3.67E+09	7.19E+09	4.524	0.221	0.452	2.210
DTSW - All I frames of **C** if scanning							
	I DTSW Steps	3.67E+07	1.74E+09	0.740	1.351	0.074	13.507

Table 8.10: Example Computational Cost for DTSW.$M = 200, N = 25, I = 100, J = 80, S = 5, P = 2$. Processor 1200 MMACS. 10 Eigenframes. Temporal Diagonal, $J' = 0.2J = 16$. The main thing to note here when compared to the case when using 10 Eigenframes and NO Temporal Diagonal is the significant improvement in scanning cost. We no longer calculate an entire new plane of the elemental and cumulative distances hypervolumes. Instead we only calculate a portion of each new plane that is along the temporal diagonal.

		Cost		1 Proc.		10 Proc.	
Calculation		Multiply	Add / Compare	Time (sec)	Rate (Hz)	Time (sec)	Rate (Hz)
Basic Operations							
	Correlation	3.67E+06	5.50E+06				
	MF		7.74E+05				
	MA		1.44E+02				
DTSW step for 1 New Frame of **C**							
	E Correlations	3.67E+07	5.50E+07				
	J MF		1.24E+07				
	$(M - N + 1)^2$ MA		4.46E+06				
	Total	3.67E+07	7.19E+07	0.01	147.36	0.00	1473.61
DTSW - All I frames of **C**							
	I DTSW Steps	3.67E+09	7.19E+09	0.68	1.47	0.07	14.74
DTSW - All I frames of **C** if scanning							
	I DTSW Steps	3.67E+07	1.74E+09	0.11	9.00	0.01	90.05

Table 8.11: Example Computational Cost for DTSW.$M = 200, N = 25, I = 100, J = 80, S = 5, P = 2$. Processor 8000 MMACS. 10 Eigenframes. Temporal Diagonal, $J' = 0.2J = 16$. The main thing to note here is that with state of the art processors the DTSW algorithm completes in about 0.07 seconds (0.01 seconds if used in a scanning or detection). This is fast enough for real-time video comparison applications in manufacturing and process control.

regions) template frames are spatially unique compared to their surroundings. These results are shown in Figure 8-1. If possible, we would want to consider changing the lighting to induce repeatable shadows or add markings to the valve flaps.

The temporal shape of D_C varies along the warp path. The minimum cost valley is roughly flat, not locally unique, in the temporal j dimension in the range of frames 15 through 20. If we examine those frames, we see that the template frames are changing very little from one frame to the next. These results are shown in Figure 8-2. We may want to ignore or reduce the accumulation of distance in this temporal region when running DTSW on later sequences. Adding markings or considering a combination of appearance and motion will not improve the lack of temporal uniqueness in this region.

8.5 Summary

Numerous factors affect the performance of the DTSW algorithm. The sensitivity to many of these factors can be explored in the context of the effect that they have on the depth of the minimum cost valley through the cumulative distance hypervolume. Similarly, the quality of the template is evaluated in the context of the depths of the minimum cost valley versus position along the path.

The computational complexity and parallel structure of the DTSW make it useful for monitoring and detection applications. Modern processors and hardware are sufficient to run comparison applications at a rate in the range of 15 Hz, and detection applications at a rate in the range of 100 Hz. There rates are sufficient for the high-speed low-repetition rate events that are often found in industrial applications.

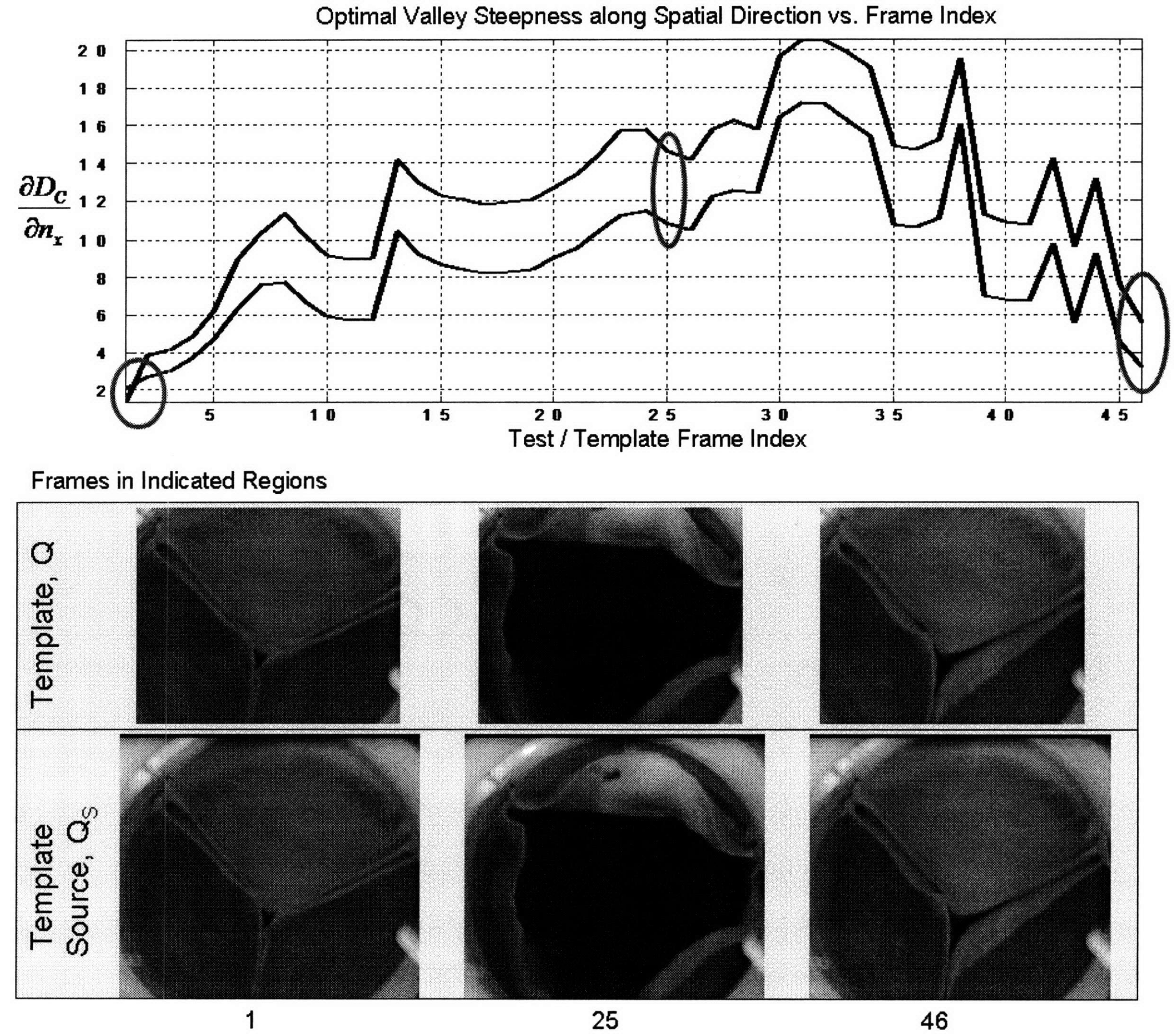


Figure 8-1: Local Shape of Cumulative Distance Hypervolume for the Heart Valve Example. The partial derivative of D_C along the x spatial direction, $\frac{\partial D_C}{\partial x}$, is small at the temporal beginning and end of the sequence. This means that the minimum cost valley is locally flat and less unique than elsewhere in the sequence. The temporal beginning and end of the template sequence when the valve flaps are fully closed is not as spatially unique as the remainder of the video. The template at the temporal ends is a relatively uniform gray field extracted from a larger relatively uniform gray field. The template is not unique in these regions. In the center of the video when the valve is open, the uniquely shaped and colored (dark and light regions) template frames are spatially unique compared to their surroundings. If possible, we would want to consider changing the lighting to induce repeatable shadows or add markings to the valve flaps.

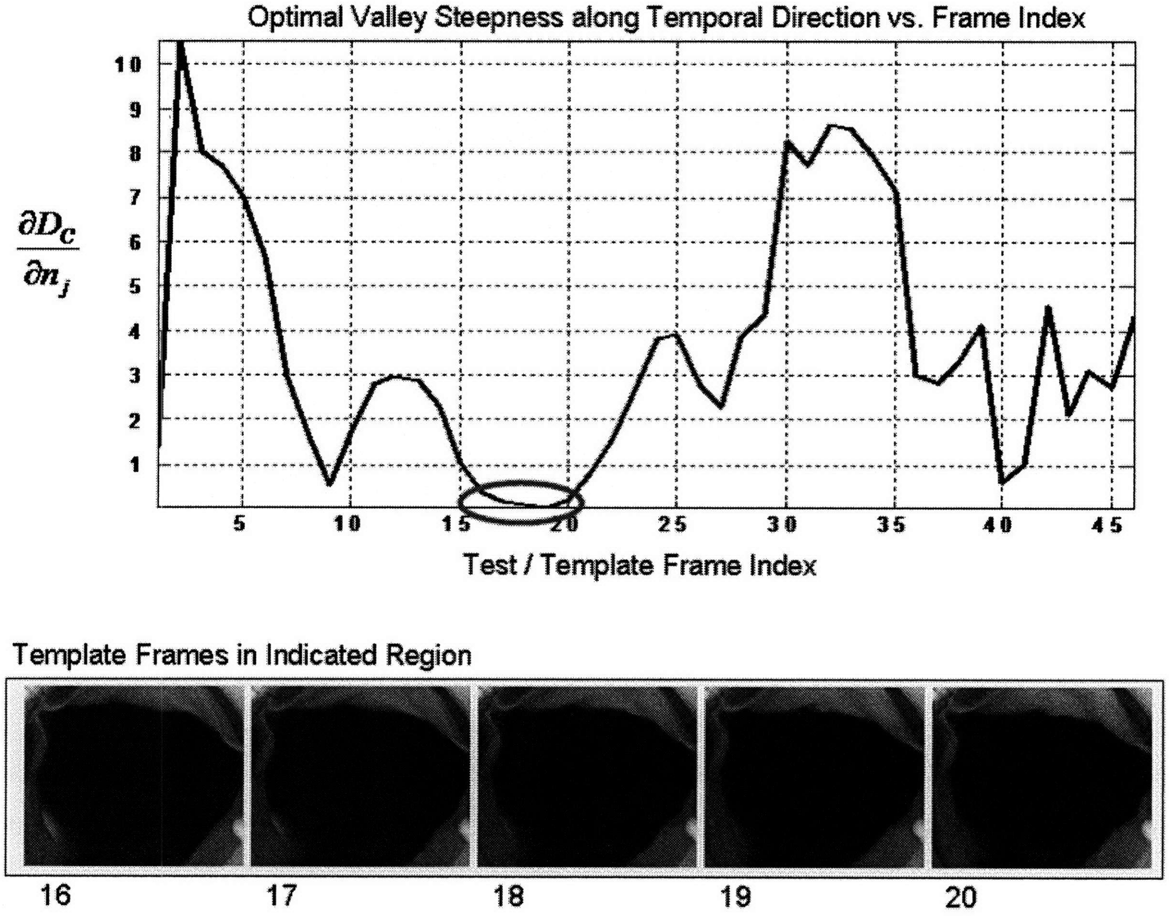


Figure 8-2: Local Shape of Cumulative Distance Hypervolume for the Heart Valve Example. The partial derivative of D_C along the j temporal direction, $\frac{\partial D_C}{\partial j}$, is small in the range of frames 15 through 20. This means that the minimum cost valley is locally flat and less unique than elsewhere in the sequence. If we examine those frames, we see that the template frames are changing very little from one frame to the next. We may want to ignore or reduce the accumulation of distance in this temporal region when running DTSW on later sequences. Adding markings or considering a combination of appearance and motion will not improve the lack of temporal uniqueness in this region.

Chapter 9

Summary

9.1 Contributions

The DTSW algorithm is an important part of a video monitoring system. It allows for a detailed and simultaneous temporal and spatial comparison between a template video and a test video. DTSW can be used to address the subject matter and monitoring needs that cannot be addressed via existing algorithms. It deterministically compares a single template video to a new a new test video. It does this by finding the minimum cost path locating the template through the test video.

DTSW can be used to monitor the implicit similarity between an ideal or exemplar system and a system under test. A system is a collection of events that have some temporal and spatial relationships. We monitor the significant events that are unique and representative of the particular system. Various output metrics from the DTSW algorithm are used to the input to a decision mechanism that makes higher level decisions regarding temporal similarity, spatial similarity, or similarity of appearance. We've demonstrated the monitoring capabilities on several real-world industrial examples.

We have shown how to efficiently extend the DTSW machinery to the video detection problem. We've shown that we are able to detect video templates that are fluid, amorphous, and difficult to segment from the video background. Even when using noisy optical flow, the nonlinear time space filtering and accumulation mechanism of

DTSW performs well.

A compact representation of the template video, i.e query, is uniquely applied in the DTSW algorithm. DTSW is efficiently implemented using a bank of template Eigenframes as filters, followed by a non-linear Local Minimum Filter and a Local Minimum Accumulation Filter. It has a structure that is easily implemented on parallel processors. Practical considerations provide further increases in efficiency. The processing rates required for "real-time" monitoring in industrial, manufacturing, and scientific environments are achievable with modern processors.

We've shown that the DTSW distance can be used for video classification on a simple human walking database with a high detection rate.

The specific contributions of this work include:

- A new technique, Dynamic Time and Space Warping (DTSW) is developed.

  DTSW enables detailed simultaneous comparison in time and space between a query video and a test video. It is a view-based algorithm which we interpret to mean as using a temporal-spatial representation of information, such as pixel intensity, optical flow, or other low level parameters for representation of all videos - template and test videos. It is used to compare, with a detailed analysis of, videos of amorphous or flexible objects lacking distinct features.

  It deterministically compares a template video to a test video. It does this by finding the minimum cost path of the template through the test video.

- Compact representation of the Template Video sequences, using PCA, uniquely applied in the DTSW algorithm.

- A filter based implementation of DTSW with a nice parallel structure.

- A novel technique for video event detection based on DTSW or a relaxed version of DTSW.

- A framework for [Automatic] System Monitoring from Detection through Detailed Analysis.

- Exploration of how to use video in automated industrial monitoring and control.

- Metrics based on the structure of the cumulative distance hypervolume, to aid the non-video engineer in the selection of a quality template video.

- The main advantages of DTSW compared to existing appearance-based techniques is that it can be used to determine detailed similarities between two videos as a function of space and time; is structured such that is can be implemented using parallel processing to increase the rate of operation; uses motion, pixels, or any other volumetric data that is application appropriate; and is appropriate for subject matter that is amorphous and flexible. Further, it can be used for video comparison or alignment, video event detection, and video classification applications. DTSW is computationally expensive, but practical considerations - and its parallel structure - allow it to be implemented at reasonable rates, on the order of 100 Hz, for video sizes that are industrially relevant.

9.2 Future work

Throughout this work we limited ourselves to scalar valued volumetric data like pixel intensities, or various moments of the optical flow. An extension to higher-order "pixels" is straightforward and should serve to improve the DTSW accuracy. Future work will examine gains made with vector valued "pixels" as a function of increased computational expense. We will also explore other scalar valued pixels such as the rank increase measure [SI05].

Preliminary work has suggested that there is a mechanism to efficiently update the cumulative distance hypervolume, D_C, when scanning the DTSW machinery in a detection application. This would serve to further speed-up detection applications of DTSW.

For some applications it is more appropriate to create a composite template from numerous training samples. This is accomplished by first aligning all training samples to a common time and spatial axis. We then use all training videos to determine the Eigenframe filters and Eigenframe reconstruction coefficients.

We've calculated the elemental distances on a full-frame to full-frame basis. In the future we will consider calculating the elemental distances based on cubes, i.e. several frames at a time, of the template. This will alter the FB mechanism for calculating the D_L. The filter kernel will be 3D cubes instead of 2D frames. Further, we will consider breaking a template video into smaller sequences, and running DTSW on the sub-videos, and then combining in a multi-scale fashion. Such strategies would allow us to independently apply spatial and temporal weights to portions of a template video when determining both local and cumulative similarity.

The structure of D_C can be used to guide an industrial user to modify the template video by adding marks to the subject matter or by adjusting lighting. Alternatively, we can make algorithmic adjustments based on the local structure of D_C. We can emphasize or de-emphasize portions, spatial and temporal, of the template video guided by a sensitivity analysis of the D_C for the template re-found in its source video. This idea has been briefly explored; we will consider it further in future work.

Bibliography

[AAYS05] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff. Simultaneous localization and recognition of dynamic hand gestures. *IEEE Motion Workshop*, 2005.

[All91] Mark Charles Allmen. Image sequence description using spatiotemporal flow curves: toward motion-based recognition. *PhD thesis, University of Wisconsin, Madison*, 1991.

[BCND01] Chiraz BenAbdelkader, Ross Cutler, Harsh Nanda, and L. S. Davis. Eigengait: Motion-based recognition of people using image self-similarity. *Proc. Intl Conf. on Audio and Video-based Person Authentication (AVBPA)*, 2001.

[BD01] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2001.

[BGS$^+$05] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005.

[BJ94] Michael J. Black and Allan Jepson. Estimating multiple independent motions in segmented images using parametric models with local deformations. *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, 1994.

[BJ98] M. J. Black and A. D. Jepson. Recognizing temporal trajectories using the condensation algorithm. *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, 1998.

[Bla99] Michael J. Black. Explaining optical flow events with parameterized spatio-temporal models. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, 1999.

[Bre97] Christoph Bregler. Learning and recognizing human dynamics in video sequences. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997.

[BYJF97] Michael J. Black, Yaser Yacoob, Allan D. Jepson, and David J. Fleet. Learning parameterized models of image motion. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on,*, 1997.

[CD00] Ross Cutler and Larry S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[cFKLR05] Ada Wai chee Fu, Eamonn Keogh, Leo Yung Hang Lau, and Chotirat Ann Ratanamahatana. Scaling and time warping in time series querying. *VLDB*, 2005.

[CKHP02] S. Chu, E. Keogh, D. Hart, and M. Pazzani. Iterative deepening dynamic time warping. *Second International Conference on Data Mining*, 2002.

[DBR00] James Davis, Aaron Bobick, and Whitman Richards. Categorical representation and recognition of oscillatory motion patterns. *Proceedings. IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[EBMM03] Alexai A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003.

[FBJ98] David J. Fleet, Michael J. Black, and Allan D. Jepson. Motion feature detection using steerable flow fields. *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, 1998.

[FCJ00] R. S. Feris, T. E. Campos, and R. M. Cesar Jr. Detection and tracking of facial features in video sequences. *Lecture Notes on Artificial Intelligence, Proceedings of MICAI-2000*, 2000.

[FJK01] Rogerio S. Feris, Roberto M. Cesar Jr, and Volker Kruger. Efficient real-time face tracking in wavelet subspace. *IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, 2001.

[HCLR83] Hans D. Hohne, Cecil Coker, Stephen E. Levinson, and Lawrence R. Rabiner. On temporal,alignment of sentences of natural and synthetic speech. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 1983.

[HSJ98] H. Haussecker, H. Spies, and B. Jhne. Tensor-based image sequence processing techniques for the study of dynamical processes. *Proceedings International Symposium On Real-time Imaging and Dynamic Analysis*, 1998.

[IB98] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998.

[JHS$^+$98] B. Jhne, H. Haussecker, H. Scharr, H. Spies, D. Schmundt, and U. Schurr. Study of dynamical processes with tensor-based spatiotemporal image processing techniques. *European Conference on Computer Vision (ECCV'98), Freiburg, Germany*, 1998.

[JM92] David Jones and Jitendra Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 1992.

[KCC02] Amit Kale, Naresh Cuntoor, and Rama Chellappa. A framework for activity based human recognition. *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2002.

[KCC03] Amit Kale, Amit K Roy Chowdhury, and Rama Chellappa. Towards a view invariant gait recognition algorithm. *Proceedings of the IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, 2003.

[KHS00] Volker Kruger, Alexander Happe, and Gerald Sommer. Affine real-time face tracking using gabor wavelet networks. *15th International Conference on Pattern Recognition (ICPR'00)*, 2000.

[KLMB87] R.A Kavaler, M. Lowy, H. Murveit, and R.W. Brodersen. A dynamic-time-warp integrated circuit for a 1000-word speech recognition system. *Solid-State Circuits, IEEE Journal of*, 1987.

[KMT98] Athanassios Kassidas, John F MacGregor, and Paul A Taylor. Synchronization of batch trajectories using dynamic time warping. *American Institute of Chemical Engineers. AIChE Journal.*, 1998.

[KP99] E. Keogh and M. Pazzani. Scaling up dynamic time warping to massive datasets. *Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databas*, 1999.

[KP01] E. Keogh and M. Pazzani. Dynamic time warping with higher order features. derivative dynamic time warping. *First SIAM International Conference on Data Mining (SDM'2001),*, 2001.

[KPZG04] E. Keogh, T. Palpanas, V. Zordan, and D. Gunopulos. Indexing large human-motion databases. *International Conference on Very Large Databases (VLDB)*, 2004.

[KR04] E. Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems (KAIS)*, 2004.

[KRCK02] Amit Kale, A.N. Rajagopalan, Naresh Cuntoor, and Volker Kruger. Gait based recognition of humans using continuous hmms. *Proceedings of the International Conference on Face and Gesture Recognition*, 2002.

[KSH05] Yan Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005.

[KSRC04] A. Kale, A. Sundaresan, A. RoyChowdhury, and R. Chellappa. Gait-based human identification from a monocular video sequence. *Handbook of Pattern Recognition and Computer Vision (Eds. C.H.Cheng and P.S.P.Wang) 3rd Ed*, 2004.

[LB95] Jim Little and Jeffrey Boyd. Describing motion for recognition. *Proceedings of Sympomsium on Computer Vision*, 1995.

[LGL01] Yongmin Li, Shaogang Gong, and Heather Liddell. Recognising trajectories of facial identities using kernel discriminate analysis. *Proc. British Machine Vision Conference*, 2001.

[LL03] Ivan Laptev and Tony Lindeberg. Space-time interest points. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003.

[LL04] Ivan Laptev and Tony Lindeberg. Velocity adaptation of space-time interest points. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004.

[Mes03a] Rudolf Mester. A new view at differential and tensor-based motion estimation schemes. *Proceedings Pattern Recognition 2003*, 2003.

[Mes03b] Rudolf Mester. On the mathematical structure of direction and motion estimation. *Research Note Visual Sensorics and Information Processing*, 2003.

[MG96] Stephen McKenna and Shaogang Gong. Tracking faces. *2nd International Conference on Automatic Face and Gesture Recognition*, 1996.

[MG98] Stephen McKenna and Shaogang Gong. Gesture recognition for visually mediated interaction using probabilistic event trajectories. *BMVC*, 1998.

[MRR80] Cory Myers, Lawrence R. Rabiner, and Aaron E. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 1980.

[MZ02] Yu-Fei Ma and Hong-Jiang Zhang. Motion texture: A new motion based video representation. *16th International Conference on Pattern Recognition (ICPR'02)*, 2002.

[NA94a] Sourabh A. Niyogi and Edward H. Adelson. Analyzing and recognizing walking figures in xyt. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94*, 1994.

[NA94b] Sourabh A. Niyogi and Edward H. Adelson. Analyzing gait with spatiotemporal surfaces. *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, 1994.

[NPC99] Chong-Wah Ngo, Ting-Chuen Pong, and Roland T. Chin. Detection of gradual transitions through temporal slice analysis. *Computer Vision and Pattern Recognition, 1999*, 1999.

[NPZ03] Chong-Wah Ngo, Ting-Chuen Pong, and Hong-Jiang Zhang. Motion analysis and segmentation through spatio-temporal slices processing. *IEEE Transactions on Image Processing*, 2003.

[NPZC00] Chong-Wah Ngo, Ting-Chuen Pong, Hong-Jiang Zhang, and Roland T. Chin. Motion characterization by temporal slices analysis. *Computer Vision and Pattern Recognition (CVPR'00)*, 2000.

[Pen91] Shou-Ling Peng. Temporal slice analysis of image sequences. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91.*, 1991.

[PGW02] Alexandra Psarrou, Shaogang Gong, and Michael Walter. Recognition of human gestues and behaviour based on motion trajectories. *Image and Vision Computing*, 2002.

[PH04] F. Porikli and T. Haga. Event detection by eigenvector decomposition using object and frame features. *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, 2004.

[PLVS04] S. Phadke, R. Limaye, S. Verma, and K. Subramanian. On design and implementation of an embedded automatic speech recognition system. *VLSI Design, 2004. Proceedings. 17th International Conference on*, 2004.

[PM89] Shou-Ling Peng and Gerard Medioni. Interpretation of image sequences by spatio-temporal analysis. *USC Computer Vision*, 1989.

[PN92] Ramprasad Polana and Randal C. Nelson. Recognition of motion from temporal texture. *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, 1992.

[PN97] Ramprasad Polana and Randal Nelson. Detection and recognition of periodic, nonrigid motion. *International Journal Computer Vision*, 1997.

[RJ92] A. Ravishankar Rao and Ramesh C. Jain. Computerized flow field analysis: Oriented texture fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[RK04] C. A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. *Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, 2004.

[RS89] A. Ravishankar Rao and Brian G. Schunck. Computing oriented texture fields. *Proceedings CVPR'89, San Diego, CA*, 1989.

[SC02] Josephine Sullivan and Stefan Carlsson. Recognizing and tracking human action. *Lecture Notes in Computer Science, Computer Vision - ECCV 2002 : 7th European Conference on Computer Vision*, 2002.

[SI05] Eli Shechtman and Michal Irani. Space-time behavior based correlation. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005.

[SK02] Y. Sato and T. Kobayashi. Extension of hidden markov models to deal with multiple candidates of observations and its application to mobile-robot-oriented gesture recognition. *Pattern Recognition, 2002. Proceedings. 16th International Conference on ,*, 2002.

[ST04] Peter Sand and Seth Teller. Video matching. *ACM Transactions on Graphics (TOG)*, 2004.

[Sub89] Muralidhara Subbarao. Interpretation of image flow: A spatio-temporal approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989.

[VRCB88] E. Vidal, H.M. Rulot, F. Casacuberta, and J.-M. Benedi. On the use of a metric-space search algorithm (aesa) for fast dtw-based recognition of isolated words. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 1988.

[WB99] Andrew D. Wilson and Aaron F. Bobick. Realtime online adaptive gesture recognition. *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*, 1999.

[WPG00] Michael Walter, Alexandra Psarrou, and Shaogang Gong. An incremental approach towards automatic model acquisition for human gesture recognition. *Human Motion, 2000. Proceedings. Workshop on,*, 2000.

[WS03] Lior Wolf and Amnon Shashua. Kernel principal angles for classification machines with applications to image sequence interpretation. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.

[YB99] Yaser Yacoob and Michael J. Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 1999.

[YC95] Yi-Sheng Yao and Rama Chellappa. Tracking a dynamic set of feature points. *Image Processing, IEEE Transactions on*, 1995.

[YD97] Yaser Yacoob and Larry S. Davis. Temporal multi-scale models for flow and acceleration. *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[YD98] Yaser Yacoob and Larry S. Davis. Learned temporal models of image motion. *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[YD00] Yaser Yacoob and Larry S. Davis. Learned models for estimation of rigid and articulated human motion from stationary or moving camera. *International Journal of Computer Vision*, 2000.

[YFM03] Hong-Jiang Zhang Yu-Fei Ma. Motion pattern based video classification and retrieval. *EURASIP Journal of Applied Signal Processing*, 2003.

[YOI92] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992.

[ZMI01] Lihi Zelnik-Manor and Michal Irani. Event-based video analysis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[ZN04] Tao Zhao and R. Nevatia. Tracking multiple humans in complex situations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2004.