

A Novel Approach to FRUC using Discriminant Saliency and Frame Segmentation

Natan Jacobson, Yen-Lin Lee, Vijay Mahadevan, *Student Member, IEEE*, Nuno Vasconcelos, Truong Q. Nguyen, *Fellow, IEEE*

Abstract—Motion-Compensated Frame Interpolation (MCFI) is a technique used extensively for increasing the temporal frequency of a video sequence. In order to obtain a high quality interpolation, the motion field between frames must be well-estimated. However, many current techniques for determining the motion are prone to errors in occlusion regions, as well as regions with repetitive structure. We propose an algorithm for improving both the objective and subjective quality of MCFI by refining the motion vector field. We first utilize a Discriminant Saliency classifier to determine which regions of the motion field are most important to a human observer. These regions are refined using a multi-stage motion vector refinement which promotes motion vector candidates based on their likelihood given a local neighborhood. For regions which fall below the saliency-threshold, a frame segmentation is used to locate regions of homogeneous color and texture via Normalized Cuts. Motion vectors are promoted such that each homogeneous region has a consistent motion. Experimental results demonstrate an improvement over previous Frame Rate Up-Conversion (FRUC) methods in both objective and subjective picture quality.

Index Terms—Frame Rate Up-Conversion (FRUC), Discriminant Saliency, Motion Segmentation, Motion Refinement, Motion Compensated Frame Interpolation (MCFI).

I. INTRODUCTION

FRUC is an area of significant research with many important applications. In mobile video, bandwidth restrictions make it infeasible to transmit at high frame rates. Instead the focus is on increasing spatial video quality while reducing the number of frames transmitted. FRUC is then used on the receiver end to recreate a smooth video. A typical example would be transmission at 15Hz with the FRUC engine performing up-conversion by a factor f two to 30Hz. Another important application is motion blur reduction for Liquid Crystal Display (LCD) televisions. This is necessary because of the sample-and-hold nature of LCD displays, which causes noticeable motion blur at low frame rates. Newer LCD displays on the market are capable of displaying at 120 to 240Hz thus significantly reducing the noticeable effect of motion blur. In order to take advantage of these high frame rates, FRUC is required to up-convert source material to the required rate.

This work is supported in part by the Center for Wireless Communications at UCSD and by Samsung Information Systems America (SISA).

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: {njacobso, ye vmahadev} @ucsd.edu, {nuno, nguyent} @ece.ucsd.edu).

Manuscript received —, 2009; revised March 19, 2010.

FRUC is composed of two portions: Motion Estimation (ME) and Motion Compensated Frame Interpolation (MCFI). A block-based ME algorithm operates by partitioning each frame into uniform blocks (generally 8x8 pixels) and determining the relative translation between each block in successive video frames. The result of the ME step is a motion field for the entire frame. Next, an intermediate frame is generated by the algorithm by interpolating along the motion field direction. Interpolation is performed bi-directionally to avoid any holes in the resultant frame. Given a motion vector (v_x, v_y) from the motion estimator, a block in the interpolated frame f_t is calculated as follows from the current frame f_{t+1} and reference frame f_{t-1} :

$$f_t(x, y) = 0.5f_{t-1}\left(x + \frac{v_x}{2}, y + \frac{v_y}{2}\right) + 0.5f_{t+1}\left(x - \frac{v_x}{2}, y - \frac{v_y}{2}\right) \quad (1)$$

Because FRUC is performed on a block basis, there are several issues which we aim to resolve. One limitation of a block-based method is that objects in the scene generally do not conform to block boundaries. Therefore, a single block may contain multiple objects with conflicting motion. Another limitation is that the motion vector which minimizes predicted block error may in fact not be the best choice. This can occur because of changes in luminance between frames or due to repetitive structures. Finally, FRUC can suffer from a ghosting artifact which is caused by large motions being assigned outside of object boundaries. These shortcomings are addressed in this work.

There are also difficulties experienced during the MCFI stage. One primary concern is motion aliasing due to low temporal sampling rates [1]. This occurs when the video frame rate falls below the Nyquist rate describing an object's trajectory. Because the trajectory is incorrect, temporal upconversion is incapable of recovering the true motion. However, these issues are negligible in MCFI for most video processing applications. Most FRUC methods assume that trackable objects in a scene have momentum over small time scales. As a result, object motions are close to linear between frames, and are safely interpolated as such.

We propose a novel method for FRUC aimed at improving both objective and subjective quality compared with previous methods. Saliency detection is employed in order to determine which regions of the scene are visually important to a human observer, thereby requiring very accurate motion vectors.

Conversely, motion-vector smoothness and consistency are enforced for non-salient regions using a fast frame segmentation.

The paper is organized as follows. In Section II, we present a review of previous research in FRUC and Motion Compensated Frame Interpolation (MCFI). A detailed overview of discriminant saliency is introduced in Section III and of frame segmentation in Section IV. The proposed algorithm is detailed in Section V along with a description of all parameters used. Objective and subjective experimental results for the proposed method are presented in Section VI. Finally, the paper is concluded in Section VII.

II. PREVIOUS WORK

Improvement of FRUC techniques has been the scope of many research projects over the past few decades. A very early contribution is the idea of 3D Recursive Search (3DRS) for ME [2], in which motion vectors are estimated based on spatial and temporal candidates from the same neighborhood. By considering candidates which have already been encountered, 3DRS is an efficient method which promotes a smooth motion field. Temporal Compensated ME with Simple Block-based Prediction (TC-SBP) is another method similar to 3DRS in that it exploits spatial and temporal candidates [3]. In this method, only three candidates are necessary for block prediction, while temporal update candidates aid in convergence of the global motion field. While these methods exploit spatial information for ME, frequency information is exploited by Phase Plane Correlation [4]. This method has the benefit of arbitrarily accurate motion vectors as well as good estimation of local motion. However, the assumptions behind PPC are only valid for blocks undergoing pure translation. More recent approaches such as [5] focus on improving FRUC for regions with high block error by merging neighboring regions with large error. Another approach is to iteratively refine the motion vector field while propagating motion vector candidates [6]. Gao et al. proposed the idea of Adaptive FRUC based on Motion Classification [7]. In this work, the scene is classified into global and local motion regions, and bidirectional or unidirectional ME is used based on the classification result.

This paper considers recent research into the field of saliency for determination of visually important regions. Saliency has previously been used for the task of video compression by Itti [8]. In this work, salient locations are determined for each frame of a video sequence and the frame is then blurred for regions sufficiently distant from the determined salient locations. This allows for the blurred regions to be compressed using fewer bits while the salient locations remain untouched. Research by Walther and Koch [9] models bottom-up saliency as a biologically plausible model of contrast between Gabor-filter orientation and color opponency. Gabor filtering is consistent with spatial filtering in the primary visual cortex, while color opponency is consistent with processing by retinal ganglion cells. In [10], the link between human recognition and bottom-up saliency is explored. It is determined that Human observers require very little spatial attention in order to recognize animals in images of natural scenes, supporting bottom-up saliency for detection of important regions in a scene. A bottom-up discriminant saliency

detector is proposed in [11], built upon a center-surround framework. This detector performs well for predicting human-eye fixation, and is also in agreement with related literature in psychophysics of the Human Visual System (HVS).

Segmentation has been used extensively for previous video processing applications. Among these, it is typical to use optical flow [12], [13] for motion estimation, rather than block-based methods. An early contribution to the field of motion segmentation is due to Thompson [14]. A combination of motion information and contrast is used in order to segment a scene into regions of consistent motion. This work makes use of a non-matching technique for motion estimation based on time-varying changes in intensity and image gradient. A contribution by Tian and Shah handles the problems of motion estimation and segmentation concurrently [15]. In this work, Markov Random Field (MRF) techniques are exploited in order to simultaneously compute the motion between adjacent frames and segment the scene into a collection of objects. Similar research is conducted in [16], where a Bayesian framework is introduced to model the motion field as the sum of a parametric motion model (based on the scene segmentation) and a residual motion vector. Khan and Shah propose a method for video segmentation using spatial location, color and motion for segmentation [17]. They show that the fusion of these features performs better than any single feature used independently. A separate approach is explored by Cremers and Soatto in [18]. Rather than segmenting each frame of a sequence into regions, this work aims to segment an entire sequence into disjoint phases of homogeneous motion, however demonstrated results depend heavily on the initialization of the segmentation parameters. Finally, a patent by Eastman Kodak [19] introduces a method for FRUC based on segmentation of the image into foreground and background regions. Separate motion data is used for each region and the occlusion regions are handled gracefully. However, this method assumes that the occlusions are manually marked by a Human technician rather than determined algorithmically.

III. DISCRIMINANT SALIENCY

Human observers typically focus their visual attention on small regions of the video frame that appear interesting. By subjecting only these attended regions to post-processing such as motion vector refinement, the quality of FRUC can be improved while keeping computational complexity manageable. The automatic selection of the regions of interest as perceived by the human visual system (HVS) has been well studied in the context of bottom-up saliency, and has been applied to improve video compression [8]. However, these techniques have been developed for static images and are not suitable for motion based region of interest identification. Therefore, in this work, we use the recently proposed discriminant center-surround model for motion saliency [20] to automatically identify salient moving objects.

A. Discriminant Center-Surround Motion Saliency

Discriminant center-surround saliency is a biologically plausible algorithm that has been shown to replicate the psychophysics of saliency mechanisms in the HVS. It can directly

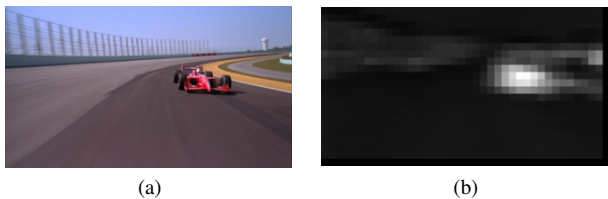


Fig. 1. Discriminant Saliency map using dynamic texture model, (a) input frame from “Speedway” sequence, (b) saliency map. Larger pixel intensity (closer to white) represents higher saliency value.

be applied to motion saliency simply by using appropriate motion models such as optical flow or dynamic textures [21]. In this work, a dynamic texture model is used to determine the motion-based feature response.

Dynamic texture data is obtained by determining an Autoregressive Moving Average (ARMA) model for a small piece of spatiotemporal data. This data is a three-dimensional volume with two spatial dimensions and one time dimension. The volume of data represents an observed sequence $\{y(t)\}$ seen as the output of a dynamic texture $\{I(t)\}$ with added noise $n(t)$. Using this notation, the dynamic texture coefficients can be determined using the following process:

$$\begin{cases} x(t) &= \sum_{i=1}^k A_i x(t-i) + Bv(t) \\ y(t) &= \phi(x(t)) + n(t) \end{cases} \quad (2)$$

where ϕ is a spatial filter, $I(t) = \phi(x(t))$, $v(t)$ is selected IID from an unknown distribution, and $n(t)$ is selected IID from a given distribution $p_n(\cdot)$.

Discriminant saliency is defined with respect to two classes of stimuli and a feature \mathbf{Y} : the class of visual stimuli in the center (with label $C = 1$), and class of visual stimuli in the background or surround (with label $C = 0$). The saliency of location l of the video, denoted $S(l)$, is the extent to which the feature \mathbf{Y} can discriminate between *center* and *surround* at l . This is quantified by the mutual information between features, \mathbf{Y} , and class label, C ,

$$S(l) = I_l(\mathbf{Y}; C) = \sum_{c=0}^1 \int p_{Y,C(l)}(\mathbf{y}, c) \log \frac{p_{Y,C(l)}(\mathbf{y}, c)}{p_Y(\mathbf{y})p_{C(l)}(c)} d\mathbf{y}. \quad (3)$$

A large $S(l)$ implies that center and surround have a large disparity of feature responses, i.e. large *local feature contrast* indicating that the location is salient. By selecting an appropriate feature \mathbf{Y} that encodes both spatial and temporal characteristics of the video (e.g. dynamic textures, optical flow) we can obtain regions that are spatiotemporally salient. Figure 1 shows the saliency map for the “Speedway” sequence obtained by using dynamic textures. The map shows that the regions predicted to have high saliency (e.g the car) are indeed the regions that appear visually salient to a human observer.

IV. SEGMENTATION

The goal of a segmentation algorithm is to partition each frame into distinct objects. Significant progress has been made on this research topic, although the problem itself is fundamentally ill-posed. For the scope of this paper, the segmentation algorithm presented in [22] is employed. This algorithm is based on Normalized Cuts [23] as well as Probability of

TABLE I
NORMALIZED-CUTS PARAMETERS FOR VARYING FRAME SIZE

Frame Size	n_{ev}	n_{spc}	n_{spf}
CIF (352x288)	50	100	200
HD720 (1280x720)	100	200	400
HD1080 (1920x1080)	200	400	800

Boundary (pB) [24] for detection of edges using color and texture information.

The method of [22] is performed using several steps. First, a contour detector is engaged in order to detect changes in brightness. The contour detector may take texture information into account, as is the case in pB. Next, a texture map is computed of the frame using textons [25]. A weight matrix W_{ij} is formed between each pair of pixels using these two cues to measure pixel similarity. Finally, N-Cuts is used to partition the image using information from the weight matrix. Further details can be explored in [22].

As is common in the literature, this segmentation scheme is used to oversegment the image. Each frame is segmented into a predetermined number of regions based on the image size. These settings are presented in Table I. The parameter n_{ev} specifies the number of eigenvectors which will be used for the Normalized Cuts algorithm. These are the eigenvectors with smallest eigenvalue and determine the initial segmentation of the frame. After Normalized Cuts, the image is further segmented into a coarse oversegmentation of n_{spc} regions by using K-means on the initial segmentation. The use of K-means is discussed in [26], and is chosen as it offers robust segmentation when initialized by N-cuts. This step is repeated once more to produce a fine oversegmentation with n_{spf} regions.

One concern with frame segmentation is the presence of motion blur in video sequences [1]. Motion blur is caused by large object motions occurring during the open shutter period of the video camera. Motion blur of sufficient magnitude may cause object boundaries to be falsely determined, thereby decreasing the performance of the proposed algorithm. In practice, this has not proven to be a concern for two reasons. First, any object with extreme motion blur will be travelling rapidly and will be difficult for the HVS to track. In addition, blurred pixels have similar luminance to the associated moving object, because the blur is caused by object regions moving along multiple pixels in the camera sensor. Therefore the proposed algorithm is capable of including proper blurred pixels into the region segmentation.

A. Merging Oversegmentation

With the frame oversegmented, the next step is to merge regions with similar characteristics. Regions with similar color and texture are merged on the assumption that they belong to the same object. This process is repeated until a small number of regions exist. The merge operation terminates when no two nodes can be located with a sufficiently small dissimilarity.

Color information is obtained directly in the RGB color space. It is important to use color rather than simply relying on

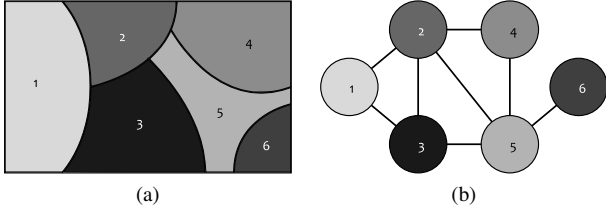


Fig. 2. Example graph for superpixel merge operation: (a) partition of region into six distinct superpixel regions, (b) superpixel neighbor graph $G = (V, E)$

luminance information, since a boundary between two objects may be isoluminant while varying between color channels.

In order to compute the texture measure, the variance of the AC coefficients of the Discrete Cosine Transform (DCT) of each 8×8 block is computed. In Eqs. 4- 6, the matrix $B_{p,q}$ contains the DCT coefficients for the input block $A_{i,j}$ with parameters $M = 8, N = 8$. The matrix is then vectorized, and the DC coefficient (the first entry) is removed because the mean of block $A_{i,j}$ says nothing about its texture. This results in a vector of length 63 denoted as \mathbf{a} . The variance of the resulting vector is given in Eq. 7 where $N = 63$.

$$B_{p,q} = \alpha_p \alpha_q \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} A_{i,j} \cos\left(\frac{\pi(2i+1)p}{2M}\right) \times \cos\left(\frac{\pi(2j+1)q}{2N}\right) \quad (4)$$

$$\alpha_p = \begin{cases} \sqrt{\frac{1}{M}} & p = 0 \\ \sqrt{\frac{2}{M}} & 1 \leq p \leq M-1 \end{cases} \quad (5)$$

$$\alpha_q = \begin{cases} \sqrt{\frac{1}{N}} & q = 0 \\ \sqrt{\frac{2}{N}} & 1 \leq q \leq N-1 \end{cases} \quad (6)$$

$$\text{var}(\mathbf{a}) = \frac{1}{N} \sum_i a_i^2 - \left(\frac{1}{N} \sum_i a_i\right)^2 \quad (7)$$

The superpixel merge procedure is posed as a problem over the graph $G = (V, E)$. Here, $\{v_1, \dots, v_n\} \in V$ is the set of all superpixel regions, and the edges $\{e_{i,j}\} \in E$ for $i, j \in [1, n]$ contain a dissimilarity measure between each pair of nodes. $E_{i,j} = 0$ if nodes $v_i, v_j \in V$ are non-adjacent. We use an indicator function $b_{i,j}$ to represent node adjacency. $b_{i,j} = 1$ if $v_i, v_j \in V$ are adjacent and $b_{i,j} = 0$ otherwise.

$$E_{i,j} = b_{i,j} \left(\lambda \max\{\mathbf{I}_i^{RGB} - \mathbf{I}_j^{RGB}\} + (1 - \lambda) |T_i - T_j| \right) \quad (8)$$

$$\mathbf{I}_i^{RGB} = \frac{1}{|\{v_i\}|} \left[\sum_{j \in v_i} R(j), \sum_{j \in v_i} G(j), \sum_{j \in v_i} B(j) \right]^T \quad (9)$$

where \mathbf{I}_i^{RGB} is the average intensity over the RGB color planes and T_i is the average texture measure for superpixel region v_i . The tuning parameter λ allows the user to emphasize either color or texture for the merging process. For all experiments conducted in this paper, the parameter is set to

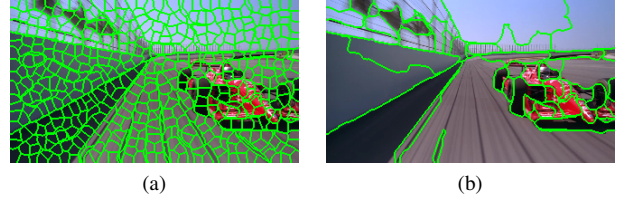


Fig. 3. Superpixel merge process: (a) oversegmentation of frame from “Speedway” sequence into $n = 200$ regions, (b) merge process after 175 iterations ($n = 25$ regions)

$\lambda = 0.5$. The merge procedure requires iteratively locating the pair of nodes $v_i, v_j \in V$ such that $E_{i,j}$ is minimized. These nodes are then merged, and the process continues. An example is demonstrated in Fig. 3 for the Speedway sequence.

V. PROPOSED ALGORITHM

The proposed FRUC architecture improves MV accuracy for salient regions while enforcing smoothness of the MV field for non-salient regions. In this way, both objective and subjective video quality will be increased. The proposed architecture is detailed in Algorithm 1.

Algorithm 1 Proposed MV Consistency and Refinement

input: frame data, oversegmented and merged region map R_1, \dots, R_n , saliency map S , saliency threshold τ

for region $R_i \in \{R_1, \dots, R_n\}$ **do**

if $\frac{1}{|j \in R_i|} \sum_{j \in R_i} S(j) < \tau$ **then**

 enforce region consistency for R_i as discussed in Section V-B

else

for all blocks contained in region R_i **do**

 perform MVR as described in Section V-C

end for

end if

end for

output: refined MV field

A. Saliency Map Generation

The saliency map is generated according to [20] with a dynamic texture model used for the feature \mathbf{Y} . A spatial window size of 8×8 pixels, and a temporal window size of 11 frames is employed for the spatiotemporal volume. The saliency map is normalized to have a maximum value of 1 pertaining to the most salient points, and a minimum value of 0 for non-salient points. The average saliency value for each region is calculated and compared with a threshold τ to determine whether region consistency or MVR is employed.

B. Region Consistency

The result of the frame oversegmentation and merging process is a segmentation with n distinct regions $\{R_1, \dots, R_n\}$ where $R_1 \cup \dots \cup R_n = I$. In order to promote *natural motion*, we restrict the candidate set of available motions to those which are statistically most likely. A MV histogram

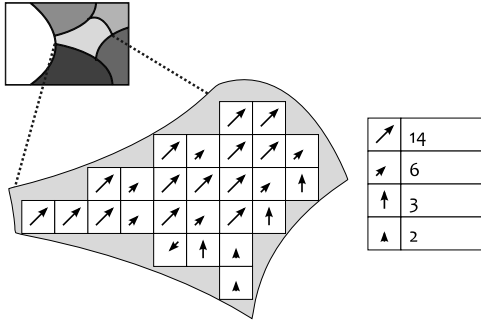


Fig. 4. Toy example for region consistency algorithm. The upper left portion demonstrates a frame which has been segmented into $n = 6$ regions. We create a MV histogram for region R_3 and select the $m = 4$ most commonly occurring motions for the candidate set $CS(R_3)$

is computed for each region R_i consisting of the motions assigned to all blocks $B \in R_i$. From this histogram, the m most commonly occurring motions are promoted as a candidate set. This process is demonstrated in Fig. 4. Selection of the parameter m is discussed in Section V-E. Denote the candidate set for region R_i as $CS(R_i) = \{\mathbf{mv}_1, \dots, \mathbf{mv}_m\}$.

For each candidate \mathbf{mv}_j in the candidate set, the Total Error $TE(\mathbf{mv}_j, R_i)$ is calculated over region R_i to determine which candidate best explains the total motion of the region. Denote the x and y-components of candidate \mathbf{mv}_j as v_{jx} and v_{jy} , respectively. For reference frame f_{t-1} and current frame f_t , TE is computed as:

$$TE(\mathbf{mv}_j, R_i) = \sum_{M \in R_i} \sum_{x, y \in M} |f_{t-1}(x + \frac{v_{jx}}{2}, y + \frac{v_{jy}}{2}) - f_t(x - \frac{v_{jx}}{2}, y - \frac{v_{jy}}{2})| \quad (10)$$

where M is a block contained in region R_i with upper-left pixel index (i, j) . Block ownership is determined by which region owns a majority of the block's pixels. Ties are broken arbitrarily. Penalties are applied to these candidates based on the total distortion produced by the candidate for the region R_i . In case of non-integer offsets ($\frac{v_{jx}}{2}, \frac{v_{jy}}{2} \notin \mathbb{Z}$), bilinear interpolation is used to determine TE . For candidate $\mathbf{mv}_j \in CS(R_i)$:

$$p(\mathbf{mv}_j) = \frac{TE(\mathbf{mv}_j, R_i)}{\sum_{k \neq j} TE(\mathbf{mv}_k, R_i)} \quad (11)$$

With the penalties determined over the candidate set, we are now able to promote MV consistency for each superpixel region. The Region Consistent MV (\mathbf{mv}_{rc}) for a block $B \in R_i$ is computed as:

$$\mathbf{mv}_{rc} = \min_{j: \mathbf{mv}_j \in CS(R_i)} \sum_{x, y \in M} |f_{t-1}(x + \frac{v_{jx}}{2}, y + \frac{v_{jy}}{2}) - f_t(x - \frac{v_{jx}}{2}, y - \frac{v_{jy}}{2})| p(\mathbf{mv}_j) \quad (12)$$

C. Motion Vector Refinement

For scene regions which exceed the saliency threshold τ , Motion Vector Refinement (MVR) is applied to increase the accuracy of the motion field. The refinement is computed without motion re-estimation in an approach similar to [27]. MVR

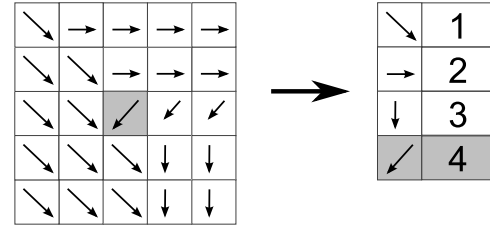


Fig. 5. Proposed candidate selection method for the center block (gray) with parameter $m = 3$. Here, the top three most commonly occurring motions in the neighborhood are considered as the first three motion vector candidates. The original motion vector for the center block is the fourth candidate.

is computed in three stages of decreasing local neighborhood, which is particularly important at object boundaries, where the MV field is difficult to determine. The method is based on the idea of *natural motion*, this is the assumption that, for any given area, there are a limited number of motions which need to be considered. The candidate selection process is demonstrated in Fig. 5. MVR is computed in multiple stages in order to improve the accuracy of the motion field around object boundaries. At each stage, the local neighborhood of consideration is decreased in order to consider more relevant MV candidates. In the first stage, enlarged block matching is considered with a 24×24 pixel measurement window for each 8×8 block. A MV histogram is created containing the original block motion and all spatial neighbors within a neighborhood of ± 2 blocks. These 25 MVs are analyzed, and the $m = 3$ most commonly occurring motions, as well as the original block motion, are promoted as a candidate set. As before, the candidate which produces the smallest error is chosen as the MV. For stage one, the error is calculated as:

$$SAD_1(v_x, v_y) = \sum_{x, y \in M_1} |f_{t-1}(x + \frac{v_x}{2}, y + \frac{v_y}{2}) - f_t(x - \frac{v_x}{2}, y - \frac{v_y}{2})| \quad (13)$$

using the Sum of Absolute Differences (SAD) error measure where M_1 is defined as in Eq. (14) for a 24×24 pixel enlarged measurement window with upper-left pixel located at (i, j) . The second stage proceeds in a similar fashion. The candidate set is increased to four motion histogram candidates and the original block motion. An 8×8 block is selected with no enlarged matching to improve the motion accuracy around object boundaries. The error for stage 2 is computed using block M_2 .

In the third stage, the resolution of the motion field is increased by a factor of two in each direction. Each block is partitioned into four 4×4 subblocks (quadrants), and refinement proceeds as in previous stages. The four subblocks are defined by $M_{3i}, i = 1, \dots, 4$

$$\begin{aligned} M_1 &= \{x, y : x \in [i - 8, i + 15], y \in [j - 8, j + 15]\} \\ M_2 &= \{x, y : x \in [i, i + 7], y \in [j, j + 7]\} \\ M_{31} &= \{x, y : x \in [i, i + 3], y \in [j, j + 3]\} \\ M_{32} &= \{x, y : x \in [i, i + 3], y \in [j + 4, j + 7]\} \\ M_{33} &= \{x, y : x \in [i + 4, i + 7], y \in [j, j + 3]\} \\ M_{34} &= \{x, y : x \in [i + 4, i + 7], y \in [j + 4, j + 7]\} \end{aligned} \quad (14)$$

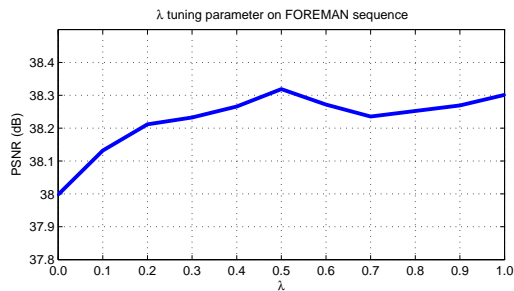


Fig. 6. PSNR of the proposed algorithm as a function of the parameter λ . performance is very nearly constant for $\lambda > 0.2$, with a peak at $\lambda = 0.5$.

D. Inconsistency between boundaries

It is common for the object boundaries determined by the segmentation algorithm to disagree with motion-based saliency boundaries. This is no cause for concern as the decision to perform MVR is based on the average saliency value of a region as determined by the segmentation algorithm. Therefore the processing method will be consistent within each region. This is the case because the segmentation boundaries are based on color and texture information, while saliency boundaries are based on differences in motion models. A broken color or texture edge would be disruptive to the HVS, and therefore is avoided by the proposed algorithm.

E. Parameter Selection

The proposed algorithm depends on several parameters which must be tuned for optimal performance. These include the saliency threshold τ , the candidate set size m , the balance between intensity and texture (λ in Eq. (8)), and the temporal window size for dynamic texture computation. The saliency threshold is heuristically set to 0.75. Recall from algorithm 1 that each region is processed based on the average saliency value. Therefore, any region with an average saliency value in the top 25% of the frame will be subjected to MVR. τ is fixed across all sequences. Next, m for region consistency is determined through testing. In Fig. 7 we measure the objective algorithmic performance as a function of the candidate set size. The **planes** sequence is used, with error averaged over all frames. Notice that PSNR achieves its maximum at $m = 2$. This is because with $m = 1$ candidate, the algorithm must blindly accept the majority vote. On the other hand, when $m > 3$, the regularizing effect of m disappears, thereby decreasing performance. SSIM decreases almost monotonically with increasing m . SSIM achieves its maximum at $m = 1$ because this forces large uniform motion fields. A good compromise on PSNR and SSIM performance is attained by selecting $m = 2$. Similar methods are employed to determine suitable parameters for the MVR candidate sets. We arrive at $m = 3$ for the first stage, and $m = 4$ for stages two and three. The parameter λ is chosen such that luminance and texture contribute equally to the region merging algorithm. This equal weighting was selected due to experimental results as demonstrated in Fig. 6. Here, the **foreman** sequence is interpolated with all parameters other than λ held fixed. The given PSNR results are an average

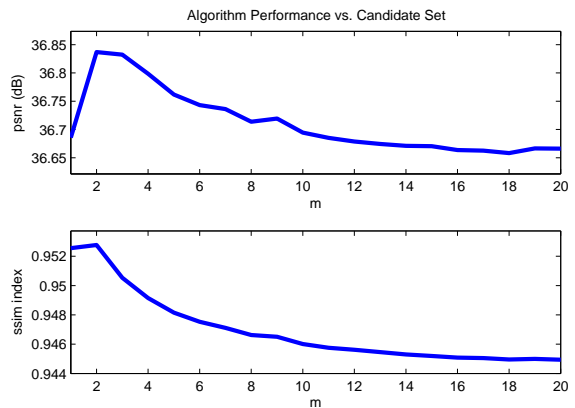


Fig. 7. Algorithm performance decays as the region consistency candidate set size (m) increases.

over the entire frame sequence. It is observed that performance is degraded when only texture information is used for the merging algorithm. However, for $\lambda > 0.2$, performance is nearly constant with the highest value attributed to $\lambda = 0.5$. It is concluded that color information is the most important for the region merging procedure, however the addition of texture information improves performance. From these results, we conclude that λ can be fixed to 0.5 for all experiments.

Finally, the dimensions of the spatiotemporal volume for dynamic texture computation are selected as discussed in [28]. In this work, the authors investigate the sensitivity of Discriminant Saliency with respect to the temporal window size τ and the spatial window size. It is concluded that performance remains nearly constant over a large gamut of spatiotemporal sizes. Because the range $\tau \in [5, 21]$ provides for roughly uniform performance, $\tau = 11$ is selected. While the authors believe that performance can be fine-tuned using an optimization over τ , it would not boost performance commensurate with the complexity of finding the optimal τ for each frame.

VI. EXPERIMENTAL SETUP

Objective results are calculated using the following experimental procedure. Each 24 frame per second (fps) video sequence is temporally reduced by a factor of two to 12fps. The 12fps sequence is then up-converted using MCFI via one of the FRUC algorithms discussed in this paper. The resulting interpolated frames are compared with the originals to determine the error.

A. Objective Results

The proposed algorithm is tested against several competing methods for FRUC. Among these are: Full Search (FS) with bidirectional MCFI [29], 3D Recursive Search (3DRS) [2], MSEA method with bidirectional MCFI [30] and a Multistage Motion Vector Processing method (MMVP) [31]. The metrics for comparison are Peak Signal to Noise Ratio (PSNR), which is calculated as the average Mean-Squared Error (MSE) of the predicted frame; and Structural Similarity Index (SSIM), which models error as perceived by a Human observer [32]. Eight sequences have been selected for comparison. Among

these are four CIF sequences (352x288) and four HD720p sequences (1280x720). The CIF sequences are: **coastguard**, **football**, **foreman** and **tennis**. These sequences are prevalent in the video processing literature. The HD sequences are: **dolphins**, **limit**, **planes** and **speedway**. All objective results for these sequences are tabulated in Table II. Additional results including video sequences, saliency information, and all error plots are located on our web site¹. In addition to objective results for full-frame comparison, performance results for the salient regions are included. We consider the top 25% of each saliency map as the mask for calculation of objective results in the salient region. This is consistent with our goal of improving the performance of FRUC most in visually salient regions.

First, the **football** sequence is examined. This is a difficult task for any FRUC engine, as there are many different motions, occlusions, and therefore numerous object boundaries. An example interpolated frame is demonstrated in Fig. 8 using the five FRUC methods. The most noticeable distortion occurs around the object boundary of player #41 in the middle of the frame. This is most evident in the 3DRS interpolation shown in Fig. 8(b). Here, significant blocking artifacts can be seen on the arms of player #41, as well as the leg of the player on the right side of the frame. Interpolation performance increases for the FS and MSEA methods in Figs. 8(c,d), which can be seen in the improved boundary of player #41. However, there are still errors in the leg of the player on the right of the frame. Because no constraints are imposed to promote consistent motion of objects, the previous methods all fail to properly assign motion in this region. The MMVP method in Fig. 8(e) combines block motions with high residuals, thus changing the appearance of the leg of the player on the right. The merging of motion vectors creates a consistent motion in this region, however the motion is too large. The result is a duplication of the leg appearing as a ghosted copy. Finally the proposed interpolation in Fig. 8(f) demonstrates consistent motion of the player on the right side of the frame. In addition, the saliency map determined for this frame sequence allows for motion vector refinement to player #41, resulting in further improvement over Full Search and MSEA. The error for **football** sequence is plotted as a function of frame number in Fig. 9(a,c). It becomes evident that the proposed method consistently improves interpolation quality for this scene. The average PSNR and SSIM values for this sequence are considerably higher than the four competing methods. The difference between the proposed method and the competing methods is plotted in Fig. 9(b,d) as a function of frame number. Here a positive value indicates that the proposed method outperforms the competing method. Notice that for all but a few frames, the PSNR and SSIM differences are positive when compared with 3DRS, FS, MSEA and MMVP.

The **foreman** sequence is examined in Fig. 10. In this sequence, 3DRS produces poor results. Notice distortion to the facial structure in Fig. 10(b). This is caused by the way in which 3DRS chooses motion vector candidates spatially and temporally. The facial motion is complex and proper

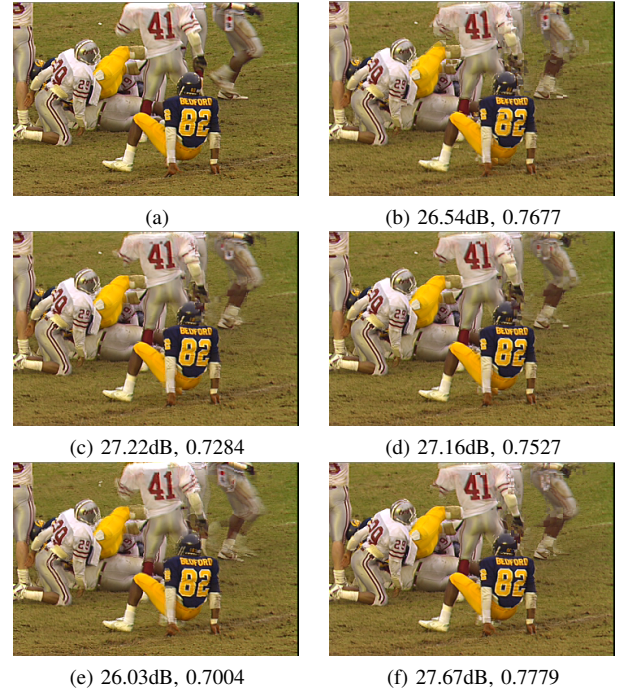


Fig. 8. Objective FRUC results for **football** sequence frame 74: (a) Original CIF frame, (b) 3DRS, (c) FS, (d) MSEA, (e) MMVP, (f) Proposed. PSNR, SSIM results shown for each frame.

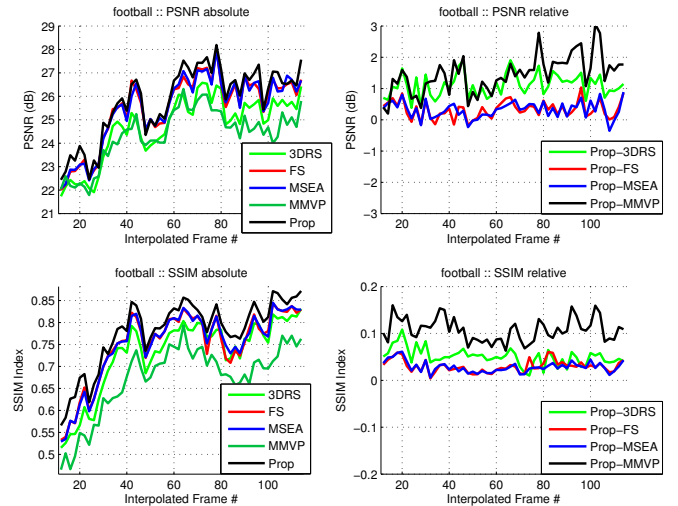


Fig. 9. Objective performance of Proposed FRUC algorithm for interpolated frames 12 through 114 of **football** sequence: (a) PSNR, (b) relative PSNR, (c) SSIM, (d) relative SSIM. Relative graphs compare the performance of methods 1-4 to that of the proposed method. A positive score denotes higher performance from the proposed method.

motion vectors may not exist in the above spatial and below temporal candidates. Fig. 10(c) demonstrates the advantage of FS over 3DRS. Here, the facial structure is mostly maintained, aside from an incorrect patch on the left side of the face. MSEA also results in noticeable distortion to the face in Fig. 10(d). The MMVP algorithm in Fig. 10(e) merges the face region and its immediate surround to a single motion. While this noticeably improves interpolation of the nose, it causes blurring at the edges of the face. The proposed algorithm

¹Additional results including video files may be found online at http://videoprocessing.ucsd.edu/~NatanHaim/TIP_2009/.

TABLE II
OBJECTIVE RESULTS FOR CIF AND HD720P TEST SEQUENCES. EACH CELL PROVIDES RESULTS IN PSNR (TOP ROW) AND SSIM (BOTTOM ROW)

Sequence	3DRS [2]	FS [29]	MSEA [30]	MMVP [31]	Proposed	3DRS	FS	MSEA	MMVP	Proposed	
CIF	<i>Entire frame</i>					<i>Salient frame region</i>					
	coastguard	34.4422 0.8973	36.9724 0.9444	37.0120 0.9448	36.0431 0.9401	37.5361 0.9505	32.4071 0.9833	33.8265 0.9866	33.8948 0.9868	33.7229 0.9858	34.8572 0.9893
	football	24.9455 0.7422	25.7013 0.7602	25.7035 0.7616	24.5524 0.6847	26.0087 0.7885	23.0517 0.9441	23.8816 0.9511	23.9127 0.9513	22.5563 0.9334	24.2448 0.9571
	foreman	37.6367 0.9413	38.5156 0.9499	38.5159 0.9502	34.6369 0.9416	38.4558 0.9530	36.1712 0.9869	37.5657 0.9904	37.6198 0.9905	36.4874 0.9883	37.6827 0.9911
	tennis	31.3513 0.8689	31.6365 0.8559	31.5762 0.8575	28.7834 0.7393	31.8027 0.8737	29.7027 0.9667	30.7389 0.9664	30.6444 0.9666	27.3591 0.9430	31.3106 0.9733
HD720p	<i>Entire frame</i>					<i>Salient frame region</i>					
	dolphins	34.0322 0.8585	35.1030 0.8790	35.0952 0.8814	35.1120 0.8835	34.9936 0.8832	30.6850 0.9417	31.8903 0.9504	31.8539 0.9511	31.6042 0.9497	31.9006 0.9537
	limit	39.3535 0.9151	39.2591 0.9156	39.2382 0.9150	39.4234 0.9159	39.5608 0.9209	37.5492 0.9855	38.2784 0.9866	38.5500 0.9871	38.2704 0.9866	38.6604 0.9876
	planes	34.2114 0.9258	36.3117 0.9517	36.2967 0.9510	36.3942 0.9469	36.8768 0.9516	36.6685 0.9940	37.1436 0.9950	37.2119 0.9950	37.1292 0.9944	38.2912 0.9952
	speedway	28.9685 0.8517	29.3508 0.8673	29.3658 0.8670	29.3960 0.8638	29.3729 0.8687	25.7847 0.9335	26.6485 0.9407	26.6092 0.9404	26.6632 0.9408	26.6846 0.9411



Fig. 10. Objective FRUC results for **foreman** sequence frame 78: (a) Original CIF frame, (b) 3DRS, (c) FS, (d) MSEA, (e) MMVP, (f) Proposed. PSNR, SSIM results shown for each frame.

addresses the shortcomings of competing methods by refining the motion field for the salient nose region while enforcing consistency in the background. This can be observed by in the interpolated frame in Fig. 10(f).

Objective results for the **tennis** sequence are demonstrated in Fig. 11. This is a complicated scene involving a moving ping-pong ball and paddle against a textured background. There are significant errors at the boundary of both the arm

and paddle caused by the 3DRS method. This can be seen in Fig. 11(b). Similar errors at the boundary of the paddle can be observed with methods 11(c,d). The MMVP method in Fig. 11(e) aims to resolve this problem, however is unable to do so properly without segmentation information. The result is a ghosted copy of the paddle. The advantage of the proposed method in this instance comes from the use of segmentation information. Because the paddle is determined to be a single object, the proposed method advances motion vectors which treat the paddle as a rigid object. The benefit of this feature can be observed in Fig. 11(f). In addition, the saliency detector ensures that motion vector candidates are refined for the region including the paddle and arm.

B. Subjective Results

In addition to objective results, it is crucial to determine the perceptual quality of the proposed algorithm. This is accomplished by performing double-blind subjective testing on a group of Human observers. Subjective results are obtained using the stimulus comparison non-categorical judgment method as described in [33]. A selected group of 20 observers were shown video clips which had been processed by the proposed method, in addition to 3DRS, FS and MSEA methods. In each instance, two video clips are shown side-by-side with each processed via a different method. The observer is presented with a rating scale on the range $[-3, 3]$, where a score of -3 corresponds with the left side appearing “much better” than the right side, and 3 corresponding with the right side “much better” than the left side. Any score between these two values is acceptable with 0 representing “no difference” between the two sequences. Findings are tabulated in Table III for the sequences: **football**, **planes**, **speedway** and **tennis** across all 20 observers. In this table, the mean (μ) and standard deviation (σ) are calculated for each sequence where a positive score on the mean corresponds to a perceptual improvement of the proposed method over the competing method. The rejection region (γ) is calculated using the Student’s T-Test, where a decision is made between the null hypothesis (the

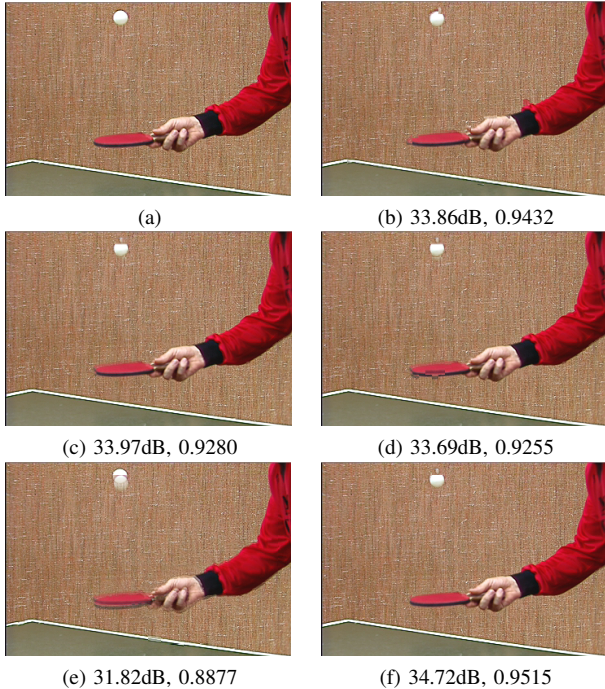


Fig. 11. Objective FRUC results for **tennis** sequence frame 20: (a) Original CIF frame, (b) 3DRS, (c) FS, (d) MSEA, (e) MMVP, (f) Proposed. PSNR, SSIM results shown for each frame.

TABLE III

SUBJECTIVE RESULTS FOR TWO CIF SEQUENCES AND TWO 720P SEQUENCES ACROSS 20 HUMAN OBSERVERS. A POSITIVE SCORE ON THE AVERAGE CORRESPONDS WITH A PERCEPTUAL IMPROVEMENT OF THE PROPOSED METHOD WHEN COMPARED WITH EACH COMPETING METHOD.

Sequence	Comp. Method	Std. Dev	Rej. Region	Average
Football	3DRS	0.50	0.19	2.34
	FULL	1.02	0.39	0.21
	MSEA	0.74	0.29	-0.15
Planes	3DRS	0.55	0.21	2.24
	FULL	1.26	0.49	1.11
	MSEA	0.77	0.30	1.48
Speedway	3DRS	0.30	0.12	2.81
	FULL	0.99	0.38	0.78
	MSEA	1.15	0.44	0.85
Tennis	3DRS	1.22	0.47	1.51
	FULL	0.51	0.20	0.21
	MSEA	0.86	0.33	0.26

proposed algorithm has no positive affect over the competing method) and the alternative hypothesis. Therefore, a mean score exceeding the calculated rejection region (τ) corresponds to a statistical improvement of the proposed method.

According to the subjective results, the proposed algorithm demonstrates a significant improvement over the competing methods for both HD sequences. However, no telling results are obtained for the CIF sequences. While the objective results are positive for the CIF sequences, the video size is too small for a significant perceptual improvement.

C. Computational Complexity

Complexity of the proposed algorithm is dependent on the methods used for saliency and segmentation calculation. For saliency, the method of [20] is used which can calculate

the saliency for a 720p frame in roughly 15 seconds. This assumes that the frame is downsampled by a factor of four to 320×180 . Segmentation of the same downsampled frame size is computed using an implementation of [34] which completes in 60 seconds. The remainder of the algorithm, including the region merging procedure, requires 10 seconds. In order to realize real-time performance, certain tradeoffs may be considered. For example, the saliency and segmentation maps may be computed every n frames and propagated using the refined/consistent motion vector field.

VII. CONCLUSION

Over the past decades, there has been significant research in methods to improve the performance of FRUC algorithms. This research has been fuelled by the high adoption rates of LCD televisions and the developing demand for mobile video. In this paper, we have proposed a novel method for FRUC which incorporates the ideas of frame segmentation and discriminant saliency. By exploiting these methods, we are able to increase the performance of interpolated image quality, especially in visually salient regions.

REFERENCES

- [1] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 4, pp. 531–545, April 2005.
- [2] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, "True-motion estimation with 3-d recursive search block matching," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 3, no. 5, pp. 368–379, 388, October 1993.
- [3] J. Wang, D. Wang, and W. Zhang, "Temporal compensated motion estimation with simple block-based prediction," *Broadcasting, IEEE Transactions on*, vol. 49, no. 3, pp. 241–248, September 2003.
- [4] M. Biswas and T. Nguyen, "A novel motion estimation algorithm using phase plane correlation for frame rate conversion," vol. 1, November 2002, pp. 492–496 vol.1.
- [5] A.-M. Huang and T. Nguyen, "Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation," *Image Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 740–752, April 2009.
- [6] S.-C. Tai, Y.-R. Chen, Z.-B. Huang, and C.-C. Wang, "A multi-pass true motion estimation scheme with motion vector propagation for frame rate up-conversion applications," *Display Technology, Journal of*, vol. 4, no. 2, pp. 188–197, June 2008.
- [7] X. Gao, Y. Yang, and B. Xiao, "Adaptive frame rate up-conversion based on motion classification," *Signal Process.*, vol. 88, no. 12, pp. 2979–2988, 2008.
- [8] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1304–1318, October 2004.
- [9] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, pp. 1395–1407, 2006.
- [10] W. Einhaeuser, T. N. Mundhenk, P. F. Baldi, C. Koch, and L. Itti, "A bottom-up model of spatial attention predicts human error patterns in rapid scene recognition," *Journal of Vision*, vol. 7, no. 10, pp. 1–13, July 2007.
- [11] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–6, October 2007.
- [12] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [13] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (darpa)," in *Proceedings of the 1981 DARPA Image Understanding Workshop*, April 1981, pp. 121–130.
- [14] W. B. Thompson, "Combining motion and contrast for segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 2, no. 6, pp. 543–549, November 1980.

- [15] T. Y. Tian and M. Shah, "Motion estimation and segmentation," *Machine Vision and Applications*, vol. 9, no. 1, pp. 32–42, January 1996.
- [16] M. Chang, A. Tekalp, and M. Sezan, "Simultaneous motion estimation and segmentation," *Image Processing, IEEE Transactions on*, vol. 6, no. 9, pp. 1326–1333, September 1997.
- [17] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, p. 746, 2001.
- [18] D. Cremers and S. Soatto, "Variational space-time motion segmentation," vol. 2, October 2003, pp. 886–893.
- [19] S. Fogel, "Segmentation-based method for motion-compensated frame interpolation;" U.S. Patent 6 008 865, December 28, 1999.
- [20] D. Gao, V. Mahadevan, and N. Vasconcelos, "On the plausibility of the discriminant center-surround hypothesis for visual saliency," *Journal of Vision*, vol. 8(7):13, no. 7, pp. 1–18, 2008.
- [21] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 439–446, 2001.
- [22] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vision*, vol. 43, no. 1, pp. 7–27, 2001.
- [23] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, August 2000.
- [24] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using brightness and texture," 2002.
- [25] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, March 1981.
- [26] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 551–556.
- [27] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion vector refinement for high-performance transcoding," *Multimedia, IEEE Transactions on*, vol. 1, no. 1, pp. 30–40, March 1999.
- [28] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 171–177, 2009.
- [29] M.-J. Chen, L.-G. Chen, and T.-D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 4, no. 5, pp. 504–509, October 1994.
- [30] X. Gao, C. Duanmu, and C. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *Image Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 501–504, March 2000.
- [31] A.-M. Huang and T. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *Image Processing, IEEE Transactions on*, vol. 17, no. 5, pp. 694–708, May 2008.
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. . Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [33] "Methodology for the subjective assessment of the quality of television pictures," *ITU-R Recommendation BT.500-11*, 2002.
- [34] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–326–II–333 Vol.2, June 2004.



Natan Jacobson (S'09) was born in Tucson, Arizona, in 1984. He received the B.S. degree in electrical engineering with a minor in mathematics from Arizona State University in May 2006, and the M.S. degree in electrical and computer engineering from the University of California San Diego in 2008, where he is currently pursuing the Ph. D. degree in image/video processing.

He has worked with NXP Semiconductor and Sharp Labs in the field of video processing. His current research interests include signal and image

processing, human vision and machine learning.



Yen-Lin Lee received the B.S. and M.S. degrees in Electrical and Control Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1999 and 2001, respectively. He received the Ph.D. degree in Electrical and Computer Engineering at University of California, San Diego, in 2009.

In 2001, he joined Silicon Integrated Systems Corporation, Hsinchu, Taiwan, where he was a digital circuit designer. He was involved in research and development on computer chipset and digital television solution. He was a technical section manager in 2005. He began his Ph.D. program since 2006. His research interests were in the field of architecture design and video processing for video compression system and motion compensated frame interpolation. After graduation he joined Mstar semiconductor, Incorporated, Hsinchu, Taiwan in 2009. He is involved in research and development on advanced mobile devices and digital television solutions.



Vijay Mahadevan received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, in 2002, and the M.S. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2003. He is currently working toward the Ph.D degree at University of California, San Diego. From 2004–2006, he was been with the Multimedia group at Qualcomm Inc, San Diego, CA. His interests are in computer vision, machine learning, signal and image processing and biomedical image analysis.



Nuno Vasconcelos received the licenciatura in electrical engineering and computer science from the Universidade do Porto, Portugal, in 1988, and the MS and PhD degrees from the Massachusetts Institute of Technology in 1993 and 2000, respectively. From 2000 to 2002, he was a member of the research staff at the Compaq Cambridge Research Laboratory, which in 2002 became the HP Cambridge Research Laboratory. In 2003, he joined the Electrical and Computer Engineering Department at the University of California, San Diego, where he

heads the Statistical Visual Computing Laboratory. He is the recipient of a US National Science Foundation CAREER award, a Hellman Fellowship, and has authored more than 75 peer-reviewed publications. His work spans various areas, including computer vision and image processing, machine learning, signal processing and compression, and multimedia systems. He is a senior member of the IEEE.



Truong Q. Nguyen (F'05) is currently a Professor at the ECE Dept., UCSD. His research interests are video processing algorithms and their efficient implementation. He is the coauthor (with Prof. Gilbert Strang) of a popular textbook, *Wavelets & Filter Banks*, Wellesley-Cambridge Press, 1997, and the author of several matlab-based toolboxes on image compression, electrocardiogram compression and filter bank design. He has over 300 publications.

Prof. Nguyen received the IEEE Transaction in Signal Processing Paper Award (Image and Multidimensional Processing area) for the paper he co-wrote with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks (1992). He received the NSF Career Award in 1995 and is currently the Series Editor (Digital Signal Processing) for Academic Press. He served as Associate Editor for the IEEE Transaction on Signal Processing 1994–96, for the Signal Processing Letters 2001–2003, for the IEEE Transaction on Circuits & Systems from 1996–97, 2001–2004, and for the IEEE Transaction on Image Processing from 2004–2005.