Video Snapshots: Creating High-Quality Images from Video Clips

Kalyan Sunkavalli, *Student Member, IEEE,* Neel Joshi, *Member, IEEE,* Sing Bing Kang, *Fellow, IEEE,* Michael F. Cohen, *Senior Member, IEEE,* and Hanspeter Pfister, *Senior Member, IEEE*

Abstract

We describe a unified framework for generating a single high-quality still image ("snapshot") from a short video clip. Our system allows the user to specify the desired operations for creating the output image, such as super-resolution, noise and blur reduction, and selection of best focus. It also provides a visual summary of activity in the video by incorporating saliency-based objectives in the snapshot formation process. We show examples on a number of different video clips to illustrate the utility and flexibility of our system.

Index Terms

image fusion, image enhancement, super-resolution, sharpening, deblurring, saliency, photomontage

- *K. Sunkavalli and H. Pfister are with the School of Engineering and Applied Sciences, Harvard University, 33 Oxford St., Cambridge, MA 02138. E-mail: {kalyans,pfister}@seas.harvard.edu*
- *N. Joshi, S.B. Kang, and M. F. Cohen are with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: {neel,sbkang,Michael.Cohen}@microsoft.com*

Video Snapshots: Creating High-Quality Images from Video Clips

1 Introduction

Often the most important photographic moments are unexpected and difficult to predict—the proud grandfather wanting to capture his grandson's first home run or a delighted mother trying to catch that perfect smile from her daughter. The typical capture scenario is for the photographer to stay ready, finger on the trigger, trying to time the shutter release perfectly. Unfortunately, these important moments are often missed, leaving a photographer frustrated with a photograph taken just a bit too early or a touch too late. In other cases, there is no one right instant; the moment can only be captured in a still image by combining multiple instances in time.

In these situations, a good alternative is is to take a video to capture the whole action. This is an increasingly available option as practically all cameras and phones today have a video mode. The video provides a temporally dense sampling of the action that ensures not only that the right moment is never missed, but that it can be revisited later on.

Unfortunately, using a video camera in lieu of a still camera comes at a cost. Even high-end video cameras today have a much lower resolution and higher noise levels than still cameras. And since the best camera is the one that you have with you, it is increasingly likely that these short videos are shot on cellphones, smartphones, or iPods with low-quality cameras. Moreover, video clips on these portable devices are compressed aggressively. As a result, a single video frame has a much lower quality than a corresponding photograph shot with a still camera, making it not very satisfying to use directly.

In this work, we consider the problem of creating a single high-quality still image—a *snapshot*—from a video clip. The snapshots we produce have higher resolution, lower noise, and less blur than the original video frames. Our system models scene motion and saliency, and can produce either a snapshot of a single moment in time where scene motion is suppressed (Fig. 1c), or a snapshot that summarizes the motion of salient objects and actions (Fig. 1d).

We assume the input to our system is a short video clip and a user-specified reference frame. We request a user-specified reference frame because picking the most important moment in a video is a subjective activity that depends on the goals, intentions, and preferences of the user. Our algorithm first aligns neighboring frames in the video to the reference frame, and then combines these frames using a Bayesian multi-image enhancement formulation to perform super-resolution, denoising, sharpening, and/or motion summarization.

Previous work either uses *all* of the aligned frames equally to generate a restored image, or selects a single frame for each pixel to create a composition (such as digital photomontage [1]). In contrast, our algorithm combines each image and pixel contribution differently using a set of *importance-based weights*. Our primary contribution is a novel importance-based framework that bridges the gap between traditional multi-image super-resolution and multi-image compositing. It can create images where stationary, non-salient parts of a scene are enhanced by combining data from multiple frames, while the salient, moving objects are enhanced using support from a single frame. Furthermore, by computing per-pixel, per-frame weights, we incorporate aspects of *lucky imaging*, where poor-quality frames in the video are not weighted as heavily when computing the resulting snapshot [2].

2 Related Work

Image enhancement techniques such as super-resolution and denoising have a long history in image processing and computer vision. Also, recent work on image fusion has looked at the problem of using user-defined preferences to fuse a collection of images into a single photomontage. Our work is related to both these problems, and in this section we briefly review these areas.

2.1 Image enhancement

Since the early work of Tsai and Huang [3], image super-resolution has been studied extensively. Park et al. [4] present a comprehensive survey of a number of methods in super-resolution. Super-resolution is inherently an ill-posed problem, and early work focused on using multiple low-resolution frames with aliasing to create a high-resolution image. The image formation process is modeled as a warping and subsampling of the high-resolution image, and these techniques explicitly invert this process to solve for a higher-resolution image that is consistent with the warped and blurred low-resolution observations [5]. Often, the parameters of the warping and subsampling are assumed to be known; this requirement can be removed by marginalizing over these parameters in a Bayesian framework [6], [7]. However, these techniques depend on the aliasing in the low-resolution frames, and because cameras often band-limit the high-frequencies to minimize aliasing, there is a theoretical limit [8], [9] on the amount of resolution

(a) Example video frames (b) Bicubic upsampling (c) Video snapshot with motion suppression (d) Video snapshot with motion summarization

Fig. 1. Comparisons of frame enhancement. (a) Four frames from a short clip showing a man jumping from a cliff. Each of these frames has low resolution, high noise and compression, and captures the man at only one time instant. (b) Bicubic upsampling one particular frame of interest. Note the high frequency texture on the rocks on the left and the trees on the right are lost, and there are blocking artifacts in the water. Our framework leverages the multiple frames in the video to produce a super-resolved, denoised snapshot. We can do this while suppressing the motion of the jumping man (c) to freeze the motion in time, or while summarizing the motion (d) to capture the activity in a single image. Note that in both these results the rocks and trees are sharper, and the blocking artifacts in the water have been removed.

enhancement (approximately an upsampling factor of 2) that these methods can provide.

More recent work has generalized super-resolution to scenes with arbitrary motion by using non-local means methods [10] or by using high-quality optical flow methods to estimate per-pixel motion [11]. Parallel to the work on multi-image super-resolution, researchers have also looked at the problem of super-resolving a single image. This problem is less constrained than multi-image super-resolution, and this is often dealt with by making use of dictionaries of images patches [12], [13], or sparse priors [14]. Another way to constrain this problem is to use the fact that image patches often recur (possibly at different scales and orientations), and recent work has used this to spatially super-resolve images [15], and spatio-temporally upsample videos [16].

Our work leverages the information in all the frames of the video clip to create a super-resolved video snapshot. Similar to classic multi-frame super-resolution [5], we estimate the snapshot by modeling the warping and subsampling, and explicitly inverting them. However, unlike most work on super-resolution where all the pixels in the video clip are treated in the same way, we introduce the notion of importance-based weights that encode the influence each pixel has on the final snapshot. This allows us to perform a number of other operations in the multi-image super-resolution framework.

Like super-resolution, image denoising is a well studied problem in image processing, and we refer the reader to Chatterjee and Milanfar [17] for a survey of recent work. Early work in image denoising made use of the sparsity of coefficients when transformed into the wavelet domain [18], [19]; here large wavelet coefficients were assumed to correspond to image structure and were retained, while small coefficients were removed. Edge-preserving filters [20], [21] have also been used to smooth noise out while retaining image structure. Priors based on natural image statistics have been incorporated in image denoising [22]. More recently, researchers have looked at making use of image sparsity in the spatial domain for image denoising. This has led to a class of algorithms where an image is modeled as consisting of a small set of patches. The K-SVD algorithm [23] learns an over-complete dictionary for image patches that can then be used for denoising [24]. In non-local means methods [25], patches across the image are aggregated, using weights based on their similarity, to smooth noise out. While all these techniques were proposed for single images, they have been used subsequently for video clips. Many video denoising techniques use motion estimation to align spatial neighborhoods. Once aligned, these frames can be merged using weights

based on spatio-temporal bilateral filter weights [26] or denoised using a temporal extension of non-local means techniques [27].

Like other video denoising techniques, we combine multiple frames to denoise video clips and create a video snapshot. However, we use a combination of weights based on sharpness, saliency, motion accuracy, etc. that allows us to incorporate a number of other effects into the denoised snapshot.

2.2 Image fusion

Agarwala et al. [1] propose a system that combines multiple images to create a single *photomontage*. In their system, users define objectives – locally by using strokes, or globally by specifying attributes to be used – that are used to decide which image each pixel in the photomontage is copied from. Similarly, "Salient Stills" [28] create a single image by fusing multiple images using different global criteria. While our goal is similar to this class of techniques, our work differs from them in its ability to automatically combine image-enhancement as well as photomontage-style image fusion in the same unifying framework.

3 Importance-based Image Enhancement

Given multiple video frames and one user-selected reference frame, our goal is to generate a clean, enhanced version of the reference frame. We adopt an image formation model that maps the restored image to the original frames that are deemed "degraded". This image formation model is popular in multi-image restoration techniques such as super-resolution (e.g., [5]). The restoration process uses multiple degraded observations to invert this image formation model and estimate the high-quality input. Our framework introduces importance-based weights into this inversion process. While our framework can be easily applied to any linear image formation model, we will discuss it here in the context of multi-image super-resolution.

Given a set of N video frames $L_k, k = 1, 2, \cdots N$ of resolution $h \times w$, multi-image super-resolution seeks to combine the frames to obtain a single high-resolution $sh \times sw$ image H. The standard super-resolution problem [29] assumes a generative image formation model given by:

$$L_k = D_s(P(T_k H)) + \eta, \tag{1}$$

where T encodes the camera motion, P denotes the camera's anti-aliasing filter, D_s is a decimation by factor s, and η is the observation noise.

D_s, P, and T are all linear operators and can be combined into a single operation $M_k(\cdot) = D_s(P(T_k(\cdot)))$. Under the assumption of zero-mean Gaussian noise, i.e., $\eta \sim N(0, \sigma_\eta^2)$, this reduces to solving for H by

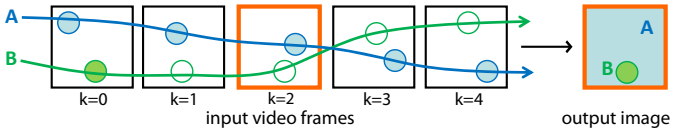


Fig. 2. Manipulating the weights in Eqn. 4 allows us to handle multi-image enhancement operations while preserving salient objects. The weights for blue patches A in all the frames are equal (i.e., $W^0 = \cdots = W^4$), and the output patch A is a linear combination of all the input patches A^k as in Eqn. 2. The weights for the green patches B are non-zero only in frame 0 (i.e., $W^0 = 1, W^1 = \cdots = W^4 = 0$), and the output patch B is copied as is from it.

minimizing the following energy function:

$$E_d = \sum_{k=1}^{N} ||(L_k - M_k H)||^2 / \sigma_\eta^2. \tag{2}$$

While multi-image super-resolution is better conditioned than single-image super-resolution, errors in alignment, saturation, noise, etc. can make solving Eqn. 2 ill-posed. This is often handled by regularizing the solutions with a prior. By using a sparse prior on the distribution of image gradients that is based on natural image statistics [30], the total energy to minimize has the form:

$$E_t = \sum_{k=1}^{N} ||L_k - M_k H||^2 / \sigma_\eta^2 + \lambda(\nabla H)^{0.8}. \tag{3}$$

Eqn. 3 represents the standard multi-image super-resolution problem. The high-resolution image H can be solved for using iterative re-weighted least squares (IRLS) [31].

In this formulation, every output pixel $H(x_h, y_h)$ is a linear combination of *all* the aligned input pixels $L_k(x_l, y_l), k = \{1, 2, \cdots N\}$. In many scenarios this is ideal; for example, the noise in the low-resolution frames is most suppressed when all frames are combined. However, in some cases, some frames (or some regions of frames) are inherently more important than others (e.g., a smiling face or a moving object), and it is usually desirable to preserve them in the final result.

This idea is the basis of image fusion algorithms such as digital photomontage [1], where every output pixel $H(x_h, y_h)$ is set to exactly one of the corresponding input pixels $L_k(x_l, y_l)$. The choice of which pixel is picked is decided by user-specified objectives. In contrast to multi-image enhancement, this approach preserves important regions, but at the cost of retaining the resolution and noise of the input frames.

Our goal is to combine aspects of these two approaches – multi-image super-resolution and image fusion – in a single framework that combines multiple low-importance pixels while preserving important pixels as is. To bridge this gap we introduce the notion of

importance-based weights into the restoration equation:

$$E_t = \sum_{k=1}^{N} ||W_k \{L_k - M_k H\}||^2 / \sigma_\eta^2 + \lambda (\nabla H)^{0.8}. \quad (4)$$

$W_k(x, y)$ encodes the importance of each (low-resolution) input pixel $L_k(x_l, y_l)$, and decides how they are combined to produce the (high-resolution) output pixels $H(x_h, y_h)$ that they are aligned with. The incorporation of these weights allows us to generalize Eqn. 3 in many different ways. For instance, by using equal weights, i.e., $W_k(x_l, y_l) = 1 \forall k$, Eqn. 4 reduces to the original multi-image super-resolution problem of Eqn. 3. On the other hand, using sparse weights, i.e., $W_k(x_l, y_l) \in \{1, 0\}, \sum_k W_k(x_l, y_l) = 1$, Eqn. 4 reduces to the digital photomontage framework. More importantly, since the weights are defined per-pixel, we can combine both of these scenarios in the same image, as illustrated in Fig. 2. By setting the weights appropriately, some parts of the output image can be enhanced by combining multiple frames, while the others can be preserved from an individual frame.

While the importance-based enhancement of videos has been discussed in terms of super-resolution in Eqn. 4, it can be easily generalized beyond this operation. Many imaging operations, including filtering, denoising, deblurring, stitching, and compositing can be expressed as a linear processing of the input video pixels, and for the appropriate choice of operator M_k, have the same form as Eqn. 4.

4 Creating Video Snapshots

Based on these ideas we now discuss how to create snapshots from a video clip. We assume that the camera motion in the video is well-approximated by an affine tranform. Given an input video clip and the user-specified reference frame, we detect interest points [32] in the video frames, and estimate an affine motion model using RANSAC [33]. We assign the weights for each frame based on three different spatial features – motion confidence, local sharpness, and temporal saliency – and time. Finally, we combine the different importance weights, and use them to solve Eqn. 4 for the output snapshot.

4.1 Motion confidence

Motion estimation is a challenging problem, and even state-of-the-art algorithms make errors while handling general scenes with arbitrary camera motion. To ensure that these errors do not lead to artifacts in the snapshots, we use weights based on the re-projection error of the estimated motion. To make this motion confidence measure robust to noise and compression artifacts, we first blur the frames using a low-pass Gaussian filter with $\sigma = 1.0$ to create the smoothed frames L'_{ref} and L'_k. We then warp the filtered reference frame L'_{ref} to the k^{th} frame using the estimated motion T_k^{-1} and assign the motion confidence as:

$$W_k^m = N(T_k^{-1}(L'_{ref}) - L'_k; 0, \sigma_m^2), \quad (5)$$

where $\sigma_m = 0.01$. Filtering the images ensures that the differences between pixels of the blurred images correspond to the spatially-weighted differences between neighborhoods of pixels in the original images.

4.2 Local sharpness

Motion blur (due to camera or scene motion) and defocus blur (due to an out-of-focus camera) often degrade the quality of a video. While creating a snapshot, we avoid pixels that are blurred by using the local sharpness measured at every pixel as weights. Our local sharpness measure estimates the high-frequency content in the neighborhood of a pixel, and is computed as a difference of Gaussians of each input frame:

$$W_k^{ls} = |L_k - G_\sigma \otimes L_k|, \quad (6)$$

where G_σ is Gaussian filter with standard deviation 3.

4.3 Temporal saliency

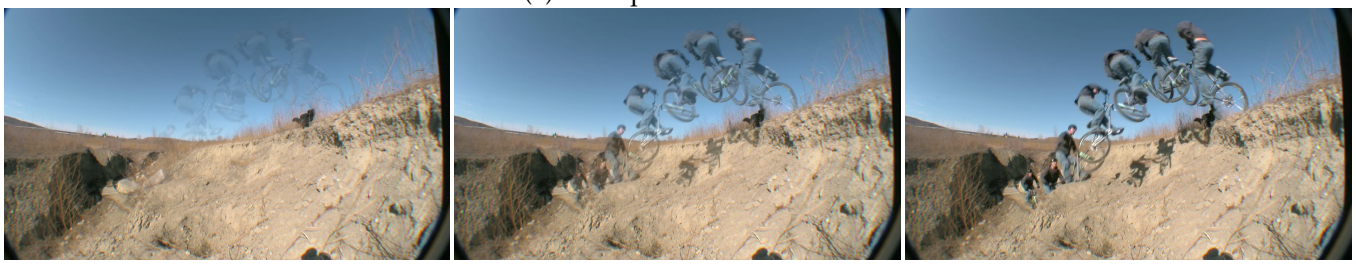
To preserve object motion in the video, we use a temporal saliency measure that detects and preserves salient regions in the scene. Many measures have been proposed for both spatial [34] and spatio-temporal saliency [35]. We use a simpler variation of the "flicker conspicuity" measure used by Itti and Baldi [35]. Our method estimates temporal saliency as the deviation of the video frames from an estimated background model. We first align all the video frames and median filter them to remove moving objects and create a background model for the video. We assign saliency weights to the input pixels based on how much they deviate from this background model. To ensure that this measure detects moving objects while staying robust to noise, compression artifacts, and small frame misalignments, we first blur the median image and the video frames using a low-pass Gaussian filter (with standard deviation set to 2.0) to create the smoothed frames L'_k and L'_{median}. The saliency weights are then set as:

$$W_k^{sal} = 1 - N(T_k^{-1}(L'_{median}) - L'_k; 0, \sigma_{sal}^2), \quad (7)$$

where $\sigma_{sal} = 0.03$. Note that because we use deviations from the median image to detect salient objects, all stationary (and even very slow-moving objects) will not register as being salient, and will be retained as part of the background in the final snapshot.

While saliency can be used to capture moving people and objects, and summarize actions in snapshots, sometimes a user might want to create snapshots where the moving parts have been *removed*, i.e., a "clean-plate" image. For example, while filming a building, the pedestrians photographed walking back and forth in front of it are often undesirable elements that the photographer

(a) Example video frames



(b) $r_{saliency} = 0$ (c) $r_{saliency} = 1$ (d) $r_{saliency} = 6$

Fig. 3. Exponentiating the feature weights makes them sparse, resulting in some pixels in the output snapshot being reconstructed from very few frames. This is illustrated on this video (35 frames, 960×540 resolution) of a bicyclist (a). The saliency measure picks out the moving bicyclist. (b) When the exponent for the saliency measure is 0, the weights are uniform, all the frames are combined, and the bicyclist is blurred out. (d) As the exponent is increased to 6, the saliency weights become sparse, and the bicyclist is reconstructed from single frames. The non-salient regions of the image are not affected by this, and continue to be estimated from all the frames. Credit: Vimeo user *markusarulius*.

might want to remove. To be able to do this in our framework, we use the notion of anti-saliency which is defined as:

$$W_k^{isal} = 1 - W^{sal}. \tag{8}$$

This formulation gives higher weights to stationary parts of the scene while removing transient objects.

4.4 Time

Artists and scientists often use tools such as shear, blur, and action lines [36] to create the perception of movement in static images. We manipulate the saliency weights estimated from Eqn. 7 using time to create perceptual cues about the motion of the salient objects in the snapshot. In particular, we use three different weighting schemes:

1) *Sampling*. Saliency weights are retained at periodic frames and set to 0 at all other frames, i.e., $W_k^{samp} = W_k^{sal}\delta(k - ik_0)$. In video clips where the object motion is very small, this makes sure that the snapshot is not cluttered.
2) *Linear Ramp*. Saliency weights are scaled linearly from the first frame to the last, i.e., $W_k^{ramp} = kW_k^{sal}$. Gradually accentuating the salient object over time creates cues for the direction of motion.
3) *Overlaying*. When regions identified as salient in different frames overlap spatially, only the latter of the regions is retained and all the others are removed, i.e.,

$$W_k^{over}(x,y) = 0, \; if \; W_l^{sal}(x,y) \; > \; \beta \quad \forall l = \{k+1, \cdots, N\} \tag{9}$$

This creates the impression of motion in the direction of time. Alternatively, we can reverse this to create the impression of motion against time by setting the weights as:

$$W_k^{rev-over}(x,y) = 0, \; if \; W_l^{sal}(x,y) \; > \; \beta \quad \forall \, l = \{1, \cdots, k-1\} \tag{10}$$

4.5 Combining feature weights

To combine the weights computed on each feature, we normalize them to the $[0, 1]$ range, scale and exponentiate them, and finally sum them:

$$W_k' = \sum_f \alpha_f (W_k^f)^{r_f} + \epsilon, \tag{11}$$

where ϵ is a small number (set to 0.001) that ensures that every input pixel is given a non-zero weight. By varying the exponent r_f in Eqn. 11, we can smoothly transition between uniform ($r_f = 0$) and sparse weights ($r_f \to \infty$). This allows us to unify multi-image enhancement and photomontage in a single framework. The effect of manipulating this exponent is illustrated in Fig. 3. The salient regions of each frame all have high weights, while all other regions have uniformly low weights. When the saliency weights are raised to exponent zero, all the frames are combined to denoise the video; however, this blurs the salient regions out. As the exponent is increased, the difference in the weights of the salient and non-salient regions is accentuated until they are copied directly from the input video into the output snapshot. Meanwhile, regions of the video that are never salient and have uniformly low weights (ϵ in Eqn. 11) continue to be reconstructed by combining multiple frames. In practice, we found that $r_f = 6$ worked well for our examples.

Enhancements / Videos	Super-resolution	Noise reduction	Sharpening	Motion suppression	Salient object summary	Temporal effects
jump (Fig. 1)	X	X	X	X	X	
ditchjump (Fig. 3)				X	X	X
dunks (Fig. 4)	X	X	X	X	X	
mural (Fig. 5)	X	X	X			
focus (Fig. 6)	X	X	X			
calendar (Fig. 7)	X		X			
foliage (Fig. 7)	X		X	X		
basketball (Fig. 8)	X	X	X	X	X	
dive (Fig. 9)	X	X	X	X	X	
bounce (Fig. 11)				X	X	X
walk (Fig. 12)	X	X	X	X	X	

TABLE 1
A summary of the enhancements we apply to our input videos.

4.6 Normalizing weights

To ensure that the error at each output snapshot pixel is weighted equally in the total energy, we normalize the weights. This is done by first warping the weights by the motion estimated on the video frames, normalizing them, and then unwarping them:

$$W_k = (T_k)^{-1} \left\{ T_k(W'_k) / \sum_{k=1}^{N} T_k(W'_k) \right\}. \qquad (12)$$

4.7 Image Prior

In traditional image enhancement, every pixel in the output image is a linear combination of approximately the same number of input image pixels. As a result, in most cases, the prior used in Eqn. 3 is spatially constant. However, in our case, the application of the spatially-varying weights changes the support of each output pixel. To take this into account, we use a spatially-varying image prior. We identify the number of input pixels that are aligned with, and contribute to the reconstruction of each output snapshot pixel; in practice, we test for this by thresholding the weights W_k by $0.1/K$, i.e., 10% of the value that a uniform weight would take. We scale the prior term by the inverse of the number of input pixels that contribute to each snapshot pixel. Incorporating this spatially varying prior into our framework leads to a graceful transition between very little regularization at pixels with large data support, and more regularization at pixels with small or no data support.

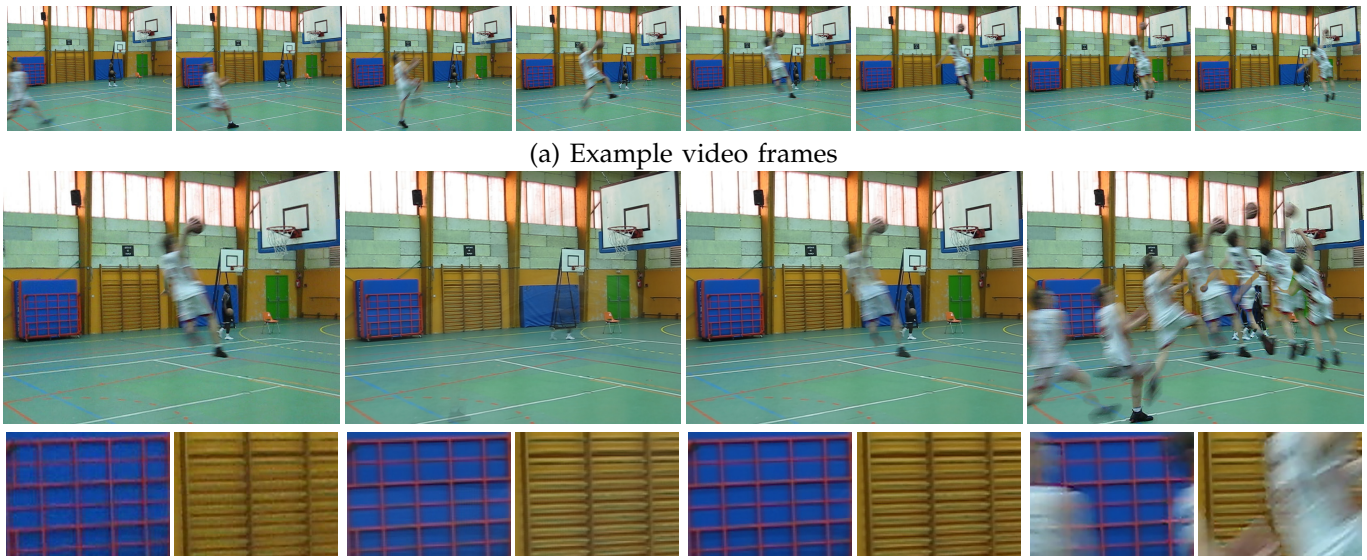
5 Results

We now present the results of enhancing a number of short video clips using our framework. These clips and our results are also available in full-resolution as companion material to this submission. All these videos clips were either captured with low-quality video cameras or downloaded from the video sharing website Vimeo (http://www.vimeo.com). They range in length from 11 frames to 31 frames and have a combination of low-resolution, high camera noise, and compression artifacts. The enhancements and effects we apply to each of them are summarized in Table 1.

We assume that the motion in the video clip is well modeled by an affine camera model. For each video clip, we estimate the inter-frame motion by fitting an affine model to interest points. The motion and the video frames are then used estimate the importance weights. With the exception of Figs. 3, 7, and 11, all results are produced using a super-resolution factor of 2. The anti-aliasing point spread function (P in Eqn. 1) is set to a Gaussian filter with $\sigma = 1.2$ and the noise level (σ_η in Eqn. 4) is automatically estimated from the reference frame using the method of Liu et al. [37]. Finally, we put the weights and the estimated motion together to set up the energy function of Eqn. 4. We solve for the output video snapshot by minimizing this energy function using conjugate gradients. We perform 5 iterations of IRLS for every result and each IRLS iteration uses 10 iterations of conjugate gradients. The time taken to compute a snapshot is almost completely dominated by the time spent in minimizing Eqn. 4; this depends approximately linearly on the resolution of the output snapshot and the number of input frames being used. Our unoptimized C++ solver takes anywhere from 6 minutes on our smallest example (Fig. 4) to 15 minutes on our largest example (Fig. 8) on an i7 2.67 GHz PC.

The quality of results from super-resolution closely depends on the accuracy of the motion estimation. This is especially true of video with complex camera motion and moving objects in the scene. By using weights based on motion confidence we ensure that only pixels where the motion estimates are reliable are used. Because they are computed with respect to the reference frame, motion confidence weights also help in suppressing moving objects in the video, while moving objects in the reference frame are preserved in their position. The results of using motion confidence in our framework are illustrated in Figs. 1, 4, 8, 9, 10, and 12.

Blur caused by camera shake or the wrong focal settings is one of the most common problems with photographs. While the short exposure time of video clips alleviates the effect of camera motion to an extent,



(a) Example video frames

(b) Single-image super-resolution (X 2) [13]

(c) Multi-image super-resolution (X 2) [5]

(c) Video snapshot (X 2) with motion confidence

(d) Video snapshot (X 2) with saliency, sampling, and overlaying

Fig. 4. (a) In this clip of a basketball player dunking (25 frames, 640 x 480 resolution), the original video suffers from low resolution and high noise. (b) Upsampling the reference frame using the single-image super-resolution technique of Yang et al. [13], improves the resolution a little but does not reduce the noise. (c) By combining multiple frames, multi-frame super-resolution [5] produces a result that has more detail on the wall and low noise, but motion of the player causes him to disappear from the result. (d) Using the motion confidence as weights preserves the high-resolution, low-noise background and captures the player. (e) Using saliency weights and temporal-overlaying summarizes the player's movement while retaining the high-quality background. Credit: Vimeo user *A.S. Saint Pantaléon Basket*.

it is not unusual to capture a video sequence and to later find out that intermittent frames are blurred. Estimating the blur kernel (which is spatially-varying in most cases) and deconvolving the image is a very difficult vision problem. Instead, we use local sharpness weights to automatically identify and reconstruct the output snapshot from only the sharpest pixels in the video clip. This approach also has the advantage that it handles variation in scene texture gracefully; smooth, low-texture regions will have uniformly low sharpness values and can be estimated from many frames, while textured regions and strong edges are reconstructed from only the sharpest pixels. Local sharpness weights can be used to create the sharpest possible snapshot in the case of motion blur (Fig. 5), as well as an all-in-focus image from a clip with varying defocus blur (Fig. 6).

Motion is often a critical component of video sequences, and the depiction of motion in static images has a long history in artistic and scientific visualization. However, most work on image enhancement avoids the issue of moving objects in a video. By using saliency weights in our framework, we are able to combine multiple frames and create a high-resolution, low-noise, sharp background while retaining the salient moving objects from individual frames. This results in high-quality still images that summarize the entire video clip in a single static snapshot (Figs. 1, 3, 4, 8, 9, 10, 11, 12, and 13). We can also use saliency in conjunction with time-based weighting to create different depictions of motion (Fig. 11). Finally, we can also use anti-saliency weights to completely remove transient elements of the video clip and produce high-quality snapshots of just the background (Fig. 12, and 13).

We have compared the quality of our results against single-image super-resolution and multi-image super-resolution. For single-image super-resolution, we compare against the work of Yang et al. [13], that uses a learned sparse dictionary of image patches to super-resolve images. As is expected, leveraging multiple frames almost always produces higher quality results than using a single image. For multi-image enhancement, we compare against the standard super-resolution technique of Irani and Peleg [5] that models the image formation process in a way that is similar to ours, and can be thought of as the standard approach to multi-image super-resolution without the use of our importance-based weights. By weighting the important pixels in the video appropriately, our framework produces snapshots with the same or better quality as standard multi-image super-resolution. We also compare our technique to a recent state-of-the-art video super-resolution method proposed by Liu and Sun [11]. This technique iteratively solves for the underlying motion, blur kernel and noise level while using a sparse image prior as well as priors on the motion and kernel. Fig. 7 shows the results of this comparison for two datasets

(a) Example video frames (b) Multi-image super-resolution (X 2) [5] (c) Video snapshot (X 2) with sharpness weights

Fig. 5. When photographing a scene with a moving camera, it is often the case that some of the frames, possibly even the desired frames captured, are motion blurred. (a) This is illustrated on this video clip of a mural captured with a hand-held video camera (21 frames, 640 x 360 resolution), where the reference frame has the best composition of the scene, but is motion blurred. (b) Most of the frames in this video clip are blurred and combining all of them to super-resolve the reference frame [5] results in a blurry image. (c) Using the local sharpness weights in our framework ensures that pixels from only the sharp frames are propagated to the reference frame, resulting in a sharp snapshot.

from their work. As can be seen from the results, when our assumption of approximately affine camera motion is met, our technique produces results that are qualitatively similar to those of Liu and Sun. In addition, our technique gives the user the freedom to go beyond basic enhancement, and depict interesting events and actions in the final snapshot.

6 Discussion and Future Work

In this paper, we have shown how to generate sharp, high-quality snapshots from lower-resolution, lower-quality videos. Our framework aligns images, computes per-pixel weights based on temporal saliency, alignment, and local image statistics, and fuses them. Our approach is flexible and can perform super-resolution, noise reduction, sharpening, and spatio-temporal summarization by changing only a few parameters. We believe this is a big step forward in increasing the ease with which users can create high-quality still photographs from short video clips. The importance of this work increases as the cost and effort of capturing video continues to decrease thanks to inexpensive, and portable consumer devices.

Our results suggest several areas for future work. While our approximation of camera motion using an affine transformation worked well for our video clips, motion estimation in complex videos is still a challenging task. As the alignment quality degrades, fewer samples can be aligned and averaged, reducing our method's ability to enhance image quality. We are investigating hierarchical motion estimation algorithms, e.g., Kang et al. [38], to address this issue. We are also investigating extensions of our importance-based weighting schemes to image enhnacement methods that don't require explicit motion estimation [25], [16]. Extremely poor quality videos pose a challenge to our system because very high noise levels and compression artifacts corrupt both the alignment as well as the importance measures.

In addition to the weights discussed in this paper, there are other weights that would be interesting to use in our framework, such as resampling / distortion weights [28], [2]. Using feature detection methods, one could also automatically find weights that indicate the presence of faces, smiles, and open / closed eyes. Our framework is general and allows any type of importance weights and user-defined combinations thereof to be used to create compelling video snapshots. Our importance-based enhancement can also be generalized to any application that involves a linear processing of video pixels. In the future we would like to investigate applications such as image stitching and compositing. It would also be interesting to perform some of our processing in the gradient domain; certain enhancements, e.g., removing blocking artifacts in compressed video, could benefit from the seamless edits that are possible with gradient domain methods.

Lastly, our final snapshots are based on a user-specified reference-frame. This could be replaced by an algorithm that automatically selects "good" reference frames (for e.g., Fiss et al. [39]) based on factors such as image quality and scene semantics.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 294–302, 2004.

[2] N. Joshi and M. F. Cohen, "Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal," in *IEEE International Conference on Computational Photography (ICCP)*, 2010.

(a) Example video frames (b) Multi-image super-resolution (X 2) [5] (c) Video snapshot (X 2) with sharpness weights

Fig. 6. (a) In this video clip (21 frames, 640 x 360 resolution), shot with a handheld video camera, the focal plane is being moved from the back to the front to create an unstabilized focal stack. (b) Naive multi-image super-resolution [5] combines both sharp and blurry frames, and produces a result that is only marginally sharper because it does not model the defocus blur in the video properly. (c) Our result uses local sharpness weights to identify and combine the sharpest pixels in the input video clip to produce an all-in-focus super-resolved snapshot.

[3] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," in *Advances in CVIP*, 1984.

[4] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[5] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, pp. 231–239, May 1991.

[6] M. E. Tipping and C. M. Bishop, "Bayesian image super-resolution," in *Neural Information Processing Systems*, 2002, pp. 1279–1286.

[7] L. Pickup, D. Capel, S. Roberts, and A. Zisserman, "Bayesian methods for image super-resolution," *The Computer Journal*, 2007.

[8] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 1167 – 1183, September 2002.

[9] Z. Lin and H.-Y. Shum, "Fundamental limits of reconstruction-based superresolution algorithms under local translation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 83 –97, jan. 2004.

[10] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *Transactions on Image Processing*, vol. 18, pp. 1958–1975, September 2009.

[11] C. Liu and D. Sun, "A Bayesian approach to adaptive video super resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[12] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, pp. 56–65, March 2002.

[13] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, pp. 2861–2873, Nov 2010.

[14] M. F. Tappen, B. C. Russell, and W. T. Freeman, "Exploiting the sparse derivative prior for super-resolution and image demosaicing," in *IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.

[15] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *International Conference on Computer Vision (ICCV)*, 2009.

[16] O. Shahar, A. Faktor, and M. Irani, "Super-resolution from a single video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[17] P. Chatterjee and P. Milanfar, "Is denoising dead?" *IEEE Transactions on Image Processing*, vol. 19, pp. 895–911, Apr 2010.

[18] E. Simoncelli and E. Adelson, "Noise removal via Bayesian wavelet coring," in *International Conference on Image Processing (ICIP)*, vol. 1, sep 1996, pp. 379–382.

[19] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[20] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[21] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the International Conference on Computer Vision*, 1998, pp. 839–846.

[22] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 860–867.

[23] M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Transactions On Signal Processing*, vol. 54, pp. 4311–4322, Nov 2006.

[24] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions On Image Processing*, vol. 15, pp. 3736–3745, Dec 2006.

[25] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," *International Journal of Computer Vision*, vol. 76, pp. 123–139, February 2008.

[26] E. P. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 845–852, 2005.

[27] C. Liu and W. T. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *European Conference on Computer Vision (ECCV)*. Springer-Verlag, 2010, pp. 706–719.

[28] L. Teodosio and W. Bender, "Salient video stills: content and context preserved," in *Proceedings of ACM International Conference on Multimedia*, 1993, pp. 39–46.

[29] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 817–834, 1999.

[30] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 70, 2007.

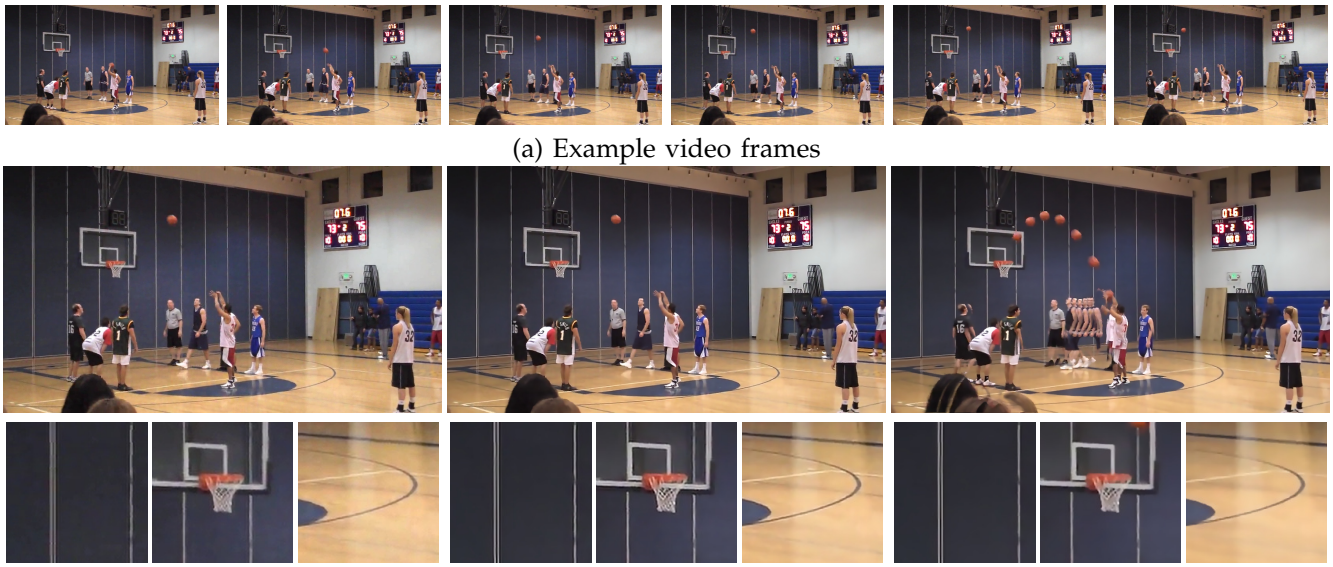
[31] P. Meer, "Robust techniques for computer vision," in *Emerging Topics in Computer Vision*, G. Medioni and S. B. Kang, Eds. Prentice Hall, July 2004, ch. 4.

[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis

(a) Example video frames (b) Bicubic upsampling (X 4) (c) Liu and Sun [11] (X 4) (c) Video snapshot (X 4)

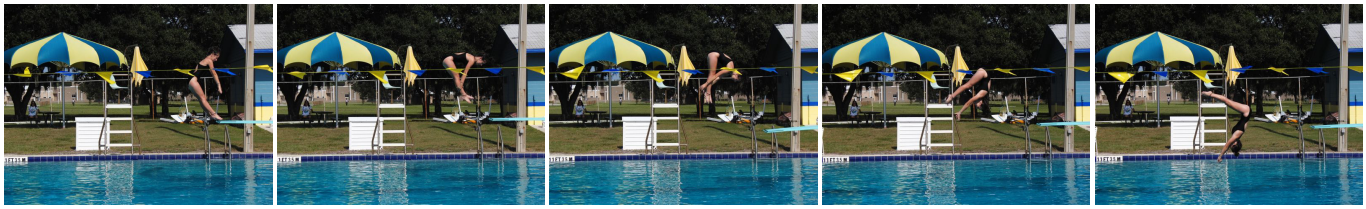
Fig. 7. Comparing our results with a state-of-the-art video super-resolution technique [11], shows that, when the camera motion in the input video is approximately affine, our technique is able to produce results that are qualitatively similar. The top row shows the *calendar* sequence, where the camera zooms outs and translates. Here both the techniques produce results where the details are resolved clearly (for e.g., the text “Maree Fine”). The bottom row shows the *foliage* sequence, where the camera pans, and there is scene motion in the video. We are able to suppress this scene motion by using our motion confidence weights, and produce a snapshot that has more detail than the bicubic-upsampled result.



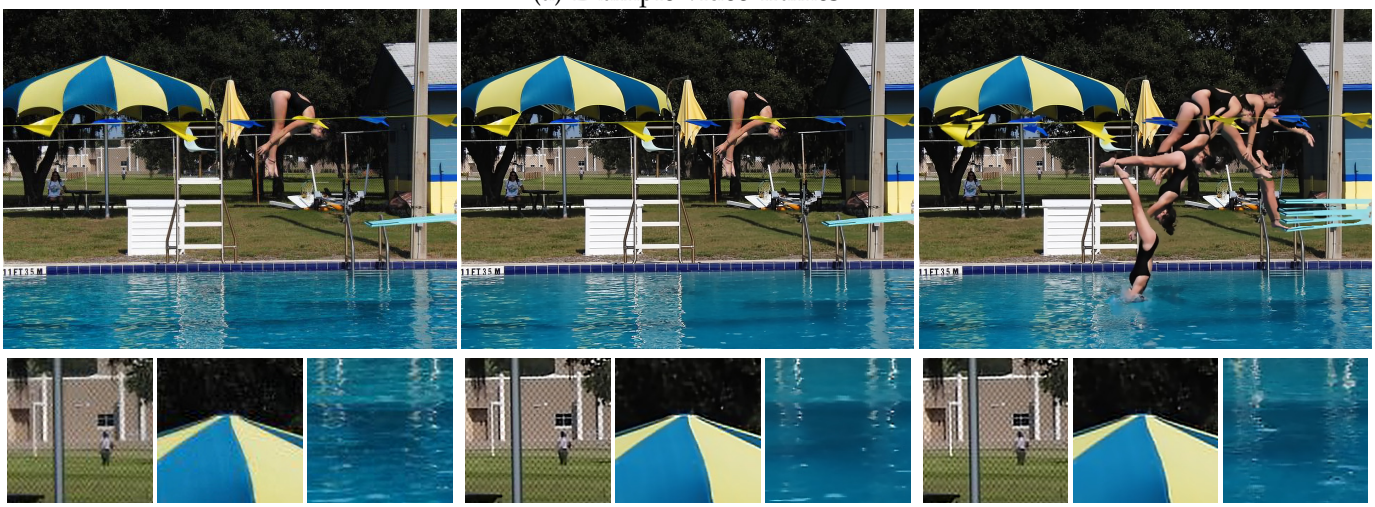
(a) Example video frames

(b) Single-image super-resolution (X 2) [13] (c) Video snapshot (X 2) with motion confidence (d) Video snapshot (X 2) with saliency and sampling

Fig. 8. (a) This video clip of a basketball game (31 frames, 640 x 360 resolution) has a lot of camera noise as can be seen on the walls. (b) Single-image super-resolution (using the method of Yang et al. [13]) can sharpen the reference frame but is unable to remove the noise and blocking artifacts. (c) Using our framework in combination with motion confidence weights produces a low-noise 1280 x 720 snapshot where the moving players and the basketball are preserved in their positions in the reference frame. (d) By using saliency weights in combination with time-sampling we can retain the high-quality background from (b) while clearly showing which players moved, and how the basketball arcs. Credit: Vimeo user *Charles Skoda*.



(a) Example video frames



(b) Single-image super-resolution (X 2) [13]

(c) Video snapshot (X 2) with motion confidence

(d) Video snapshot (X 2) with saliency, time-sampling and overlaying

Fig. 9. (a) This video clip of a diving girl (28 frames, 640 x 480 resolution) has noise and compression artifacts. (b) Upsampling the reference frame using the single-image technique of Yang et al. [13] marginally improves the resolution but can not handle the noise and blocking artifacts. Using our framework, we can combine the original video frames to upsample and denoise the reference frame. We do this while either (c) suppressing the motion, or while (d) summarizing the entire dive in the snapshot. Credit: Vimeo user *DHS Swim & Dive*.

(a) Example video frames

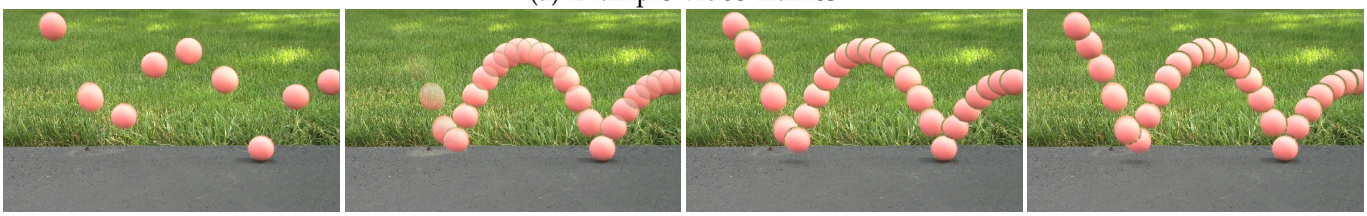
(b) Single-image super-resolution (X 2) [13]

(c) Video snapshot (X 2) with motion confidence

(d) Video snapshot (X 2) with saliency and time-sampling

Fig. 10. (a) This juggling video (24 frames, 640 x 480 resolution) has low resolution and a high noise level. (b) Single-image super-resolution [13] only improves the resolution marginally and can not handle the noise. Our method is able to improve the resolution markedly (note how the letters on the blackboard are clearer), while also lowering the noise in the image (the noise near the duster in the bottom right is reduced). We do this while either (c) capturing the moment in the reference frame, or while (d) depicting the motion of the ball and the hands. Credit: Vimeo user *BCCP Video*.

(a) Example video frames



(b) Video snapshot with saliency and time-sampling

(c) Video snapshot with saliency and linear time weights

(d) Video snapshot with saliency and time overlaying

(e) Video snapshot with saliency and reverse-time overlaying

Fig. 11. Our framework enables video snapshots with time-based effects. (a) In this clip of a bouncing ball (11 frames, 960×540 resolution), the input frames (shown at half size) can be combined with (b) time-sampling weights to discretely sample some of the frames, with (c) temporal weights that increase linearly to emphasize the direction of motion, or with (d,e) weights that overlay each instant of the ball on top of the previous or next instances. While these effects are applied to the ball, the other (background) pixels in the snapshots are generated by combining all the frames.

(a) Example video frames

(b) Video snapshot (X 2) with motion confidence

(c) Video snapshot (X 2) with saliency and sampling

(d) Video snapshot (X 2) with anti-saliency

Fig. 12. (a) In this video clip of a man walking from the left to the right (13 frames, 640 x 360 resolution), (b) using motion-confidence weights produces a result where the man is preserved as in the original frame, while the background is super-resolved. (c) Using saliency-based weights captures the man in each position, and summarizes his motion. (d) Using anti-saliency weights removes the man completely to produce a high-quality image of only the background.

and automated cartography," *Communications of ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[35] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 631–637.

[36] J. E. Cutting, "Representing motion in a static image: constraints and parallels in art, science, and popular culture," *Perception*, vol. 31, no. 10, pp. 1165–1193, 2002.

[37] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 299–314, 2008.

[38] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High dynamic range video," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 319–325, 2003.

[39] J. Fiss, A. Agarwala, and B. Curless, "Candid portrait selection from video," *ACM Transactions on Graphics*, vol. 30, pp. 128:1–128:8, Dec. 2011.

(a) Example video frames

(b) Video snapshot (X 2) with saliency

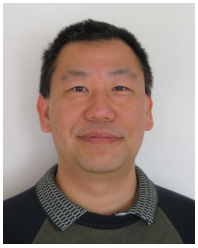
(c) Video snapshot (X 2) with saliency, sampling, and overlaying

(d) Video snapshot (X 2) with anti-saliency

Fig. 13. (a) This is video clip of traffic at a busy roundabout (20 frames, 640 x 360 resolution). (b) Using saliency weights produces a snapshot that captures all the vehicles in the video. However, because of the number of moving objects in the scene, this result looks crowded. (c) By using saliency with time-based sampling and overlaying, we can reduce this clutter while capturing the traffic in the clip. (d) We can also create a "clean-plate" snapshot of just the background by using anti-saliency weights to remove most of the moving objects. Credit: Vimeo user *Vietnam720*.

Kalyan Sunkavalli is a Ph.D. candidate at the School of Engineering and Applied Sciences at Harvard University. He received his B.Tech. in Electrical Engineering from the Indian Institute of Technology, New Delhi in 2003, and his M.S. in Computer Science from Columbia University in 2006. Kalyan's research interests lie at the intersection of computer graphics, computer vision, and image and video processing; his specific focus is on understanding the different aspects of visual appearance, and building tools to easily analyze and edit them in images and videos.

Neel Joshi is currently a Researcher in the Graphics Group at Microsoft Research. His work spans computer vision and computer graphics, focusing on imaging enhancement and computational photography. Neel earned an Sc.B. from Brown University, an M.S. from Stanford University, and his Ph.D. in computer science from U.C. San Diego in 2008. He has held internships at Mitsubishi Electric Research Labs, Adobe Systems, and Microsoft Research, and he was recently a visiting professor at the University of Washington. He is a member of the IEEE.



Sing Bing Kang received his Ph.D. in robotics from Carnegie Mellon University in 1994. He is currently Principal Researcher at Microsoft Corporation, and his research interests are image and video enhancement as well as image-based modeling. In addition to co-editing two books ("Panoramic Vision" and "Emerging Topics in Computer Vision"), he has co-authored two books ("Image-Based Rendering" and "Image-Based Modeling of Plants and Trees"). He has served as area chair and member of technical committee for the major computer vision conferences. He has also served as papers committee member for SIGGRAPH and SIGGRAPH Asia. Sing Bing was program co-chair for ACCV 2007 and CVPR 2009, and is currently Associate Editor-in-Chief for IEEE TPAMI and IPSJ Transactions on Computer Vision and Applications. He is an IEEE Fellow (class of 2012).

Michael F. Cohen is a Principal Researcher at Microsoft Research and an Affilate Professor in the CSE Dept. at the Unniversity of Washington. He has previously served on the Computer Science faculty at Princeton University and the Architecture faculty at Cornell University. Michael received his Ph.D. in 1992 from the University of Utah. He also holds undergraduate degrees in Art and Civil Engineering from Beloit College and Rutgers University respectively, and an M.S. in Computer Graphics from Cornell. He co-authored the book “Radiosity and Image Synthesis”. At Microsoft, Dr. Cohen has worked on a number of projects ranging from image based rendering (using the Lumigraph), to animation, to camera control, to more artistic non-photorealistic rendering. His recent work has focused on computational photography applications including new methods for low bandwidth teleconferencing, segmentation and matting of images and video, technologies for combining a set of “image stacks” as a Photomontage, and the creation of very high resolution panoramas. Michael received the 1998 SIGGRAPH Computer Graphics Achievement Award for his contributions to the Radiosity method for image synthesis. Dr. Cohen also served as paper’s chair for SIGGRAPH ’98. He is an ACM Fellow.

Hanspeter Pfister is Gordon McKay Professor of Computer Science in the School of Engineering and Applied Sciences at Harvard University. His research in visual computing lies at the intersection of visualization, computer graphics, and computer vision. It spans a wide range of topics, including bio-medical visualization, 3D reconstruction, GPU computing, and data-driven methods in computer graphics. Before joining Harvard he worked for over a decade at Mitsubishi Electric Research Laboratories where he was most recently Associate Director and Senior Research Scientist. Dr. Pfister has a Ph.D. in Computer Science from the State University of New York at Stony Brook and an M.S. in Electrical Engineering from ETH Zurich, Switzerland. He is the recipient of the 2010 IEEE Visualization Technical Achievement award. He has authored over 40 US patents and over 70 peer-reviewed publications and book chapters, including 18 ACM SIGGRAPH papers, the premier forum in Computer Graphics. He is co-editor of the first textbook on Point-Based Computer Graphics, published by Elsevier in 2007.