# Compact Representations of Dynamic Video Background Using Motion Sprites

Solomon Garber, Aaditya Prakash, Ryan Marcus, Antonella DiLillo and James Storer

Brandeis University

{solomongarber,aprakash,rcmarcus,dilant,storer}@brandeis.edu

**Abstract:** We present a method to extend the idea of sprite coding to videos containing a wide variety of naturally occurring background motion, which could potentially be incorporated into existing and future video standards. The existing MPEG-4 part 2 standard, now almost 20 years old, provides the ability to store objects in separate layers, and includes a sprite mode where the background layer is generated by cropping a still image based on frame-wide global motion parameters, but videos containing more general background motion cannot be effectively encoded with sprite mode. We propose a perceptually motivated lossy compression algorithm, where oscillatory background motion can be compactly encoded. Our model achieves a low bit rate by referencing a time-invariant representation of the optical flow with only a few added parameters per frame. At very low bit rates, our technique can provide dynamic backgrounds at a visual quality that may not be achievable by traditional methods which are known to produce unacceptable blocking and ringing artifacts.

## 1 Introduction

Among the most visually salient features of a video is the motion of the objects in the frame. Unfortunately, from a video compression standpoint, the information which encodes this motion makes up a large portion of the data contained in a video file, even when the motion is background motion of secondary importance. For instance, in a video of an outdoor soccer game, the swaying of trees in the background is of far less importance to the viewer than the players and the ball in the foreground. In some instances, background motion can be heavily compressed. The existing MPEG-4 part 2 video standard, now almost 20 years old, offers the option of *sprite coding*, where a static frame can be used to encode the background of an entire video clip. In this mode, the background of each video frame is cropped from this sprite frame based on global motion parameters, which specify the motion of the entire background with only a few parameters.

Traditional sprite coding is only intended for videos with static backgrounds, where background motion is the result of either a camera pan, tilt or zoom. However, many videos contain background motion which is approximately oscillatory, where stationary background objects such as trees, foliage, telephone wires, and flags move around a stationary fixed point, often due to excitation from the wind. We present a method for extracting and compactly encoding global motion parameters for perceptually realistic oscillatory background motion. By filtering the motion in the background pane of a video, we can find the modal shapes, amplitudes, frequencies

and directions of the background oscillations. We can then subsample the oscillations in the frequency domain, a process we refer to as *motion palette optimization*. The generated modes of background motion are spatially smooth, which allows our system to downsample and compress them to a size that is generally much smaller than even the background sprite itself. Beyond this background motion mode only two complex modal parameters are required per video frame to decode each mode into a dense frame of motion vectors. This amounts to only 4 floats of data per frame of global background motion parameters, which can be easily encoded as metadata, to reproduce smooth background motion perceptually similar to the uncompressed video. In addition, with a modest increase in decoding time (as little as four multiplies per pixel), a motion-sprite enabled video decoder can produce an unlimited amount of background motion with almost no added data per frame. This technique enables compact sprite coding of windy outdoor scenes, not normally possible with a sprite, and thus frees up bandwidth for the foreground layers of the video. We present experiments that show compact, visually pleasing representations that are not possible with traditional video encoding techniques.

## 2 Related Work

Our work builds on Eulerian motion processing [1, 2, 3, 4, 5], which refers to processing motion information separately at each spatial location over time, as contrasted with Lagrangian methods such as [6], where individual particles are tracked through time along their motion trajectories. As demonstrated by [7], the dense velocity and direction of the motion contained in image sequences can be robustly and efficiently estimated with sub-pixel accuracy by tracking level curves of the phase response of complex-valued Gabor wavelets over time. As noted by [7], sensor noise, changes in lighting and contrast, and nonuniform motion all cause pixel values and filter responses to translate imperfectly. The phase response, which is determined solely based on the relative amplitudes of the symmetric and antisymmetric parts of the complex-valued filter responses and has a linear relationship with the shift parameters, is much more robust to these sorts of variations in the input signal [7].

By using the changes in local phase over time as a proxy for the motion signal at each point, much can be revealed about the properties of oscillatory motions in a video. In [2] it was shown how these phase variations could be temporally filtered and band-amplified to accentuate different temporal frequency bands of periodic motion within the video. The authors of [8] showed that the high speed vibrations of the objects in a video, pooled over all of space to boost the signal, could be used to approximately reconstruct the sound waves in the room which caused the vibrations.

The authors of [1], [2], [8], [3], and [9] were all exploring Eulerian methods for video motion processing in the context of studying small motions, which may be too subtle for a human observer to notice. Eulerian methods lend themselves nicely to the motion amplification task because motion can be detected and amplified in a pointwise manner, but this breaks down in the presence of large motion where the artifacts of Eulerian motion synthesis are most noticeable. In [10] it was noted that Eulerian techniques like the ones described in [2] could be used to process macroscopic, visible motion, provided that motion synthesis was performed in a different way.

In [9], a physical interpretation was given of the motion dynamics in a video, allowing for eigenmode analysis of video motion. In [10], it was demonstrated that by interpreting the eigenmodes [9] of video object motion as dense motion fields oscillating over time around their central frequency, and by specifying environmental forcing interactions and modal damping factors for these modes, static images can be warped to generate perceptually plausible physical simulations of object motion in video. We present a method to generate modes automatically, without requiring user interactions, and correlate them against the input signal to estimate their modal coordinates at each frame of an input video. In this way, we can use the insights of [10] to approximately reproduce the original video motions.

## 3 Background

In modal analysis, object deformations are approximated by point masses connected by springs and dampers[9]. The evolution of the state of such a system is given by the differential equation $M\ddot{x} + C\dot{x} + Kx = 0$, where $M$ is the matrix representing the inertial masses between points, $C$ gives the damping values between points, $K$ the spring stiffness between points, and $\ddot{x}$, $\dot{x}$, and $x$ are the accelerations, velocities, and displacements of the point masses. This formulation allows dynamically oscillating motion vector fields to be decoupled into a sum of a small set of complex-valued eigenmodes (shown in Figure 1) whose states evolve over time according their modal masses, damping factors, and any external forcing coming into the system. We will use these eigenmodes as a lossy model of oscillatory motion of video backgrounds.

While the transfer function of the spatial displacement of an undamped oscillator over time is a Dirac delta at the resonant frequency, the impulse response of a damped oscillator is given by an exponentially decaying sinusoid, which has a transfer function of a Lorentzian distribution $\frac{s^2}{(x-\omega)^2+s^2}$, where $\omega$ is the temporal frequency of the mode and $s$ can be computed from the damping (internal friction) and modal mass of the mode[9] . Peaks in the amplitude response correspond to periodic motion in the video around some frequency. Fourier analysis can be used to infer the mode shapes and frequencies from the local motion signals.

## 4 Motion Palette Optimization

The results of [11] support the notion that human vision has a preference for lower velocities and for temporally consistent motion (low frequency or smooth), especially in regions of high contrast. Our method filters motions in space and time to isolate the low frequency components. We use contrast cues as our motion signal, so these are the areas where the modes are strongest and contain the least noise. While our technique may struggle to encode and reconstruct fluttering motions, it can pick up on the emergent large scale motion signal, and provide a compact representation of the most visually salient aspects of oscillatory background motion in a video.

Our method compresses background motions into a set of static motion modes and a complex valued time domain signal representing the state of each mode at each video frame, as shown in Figure 3. In order to estimate the state of the modes at each input frame, first we need to find the modes that we will use, and estimate the

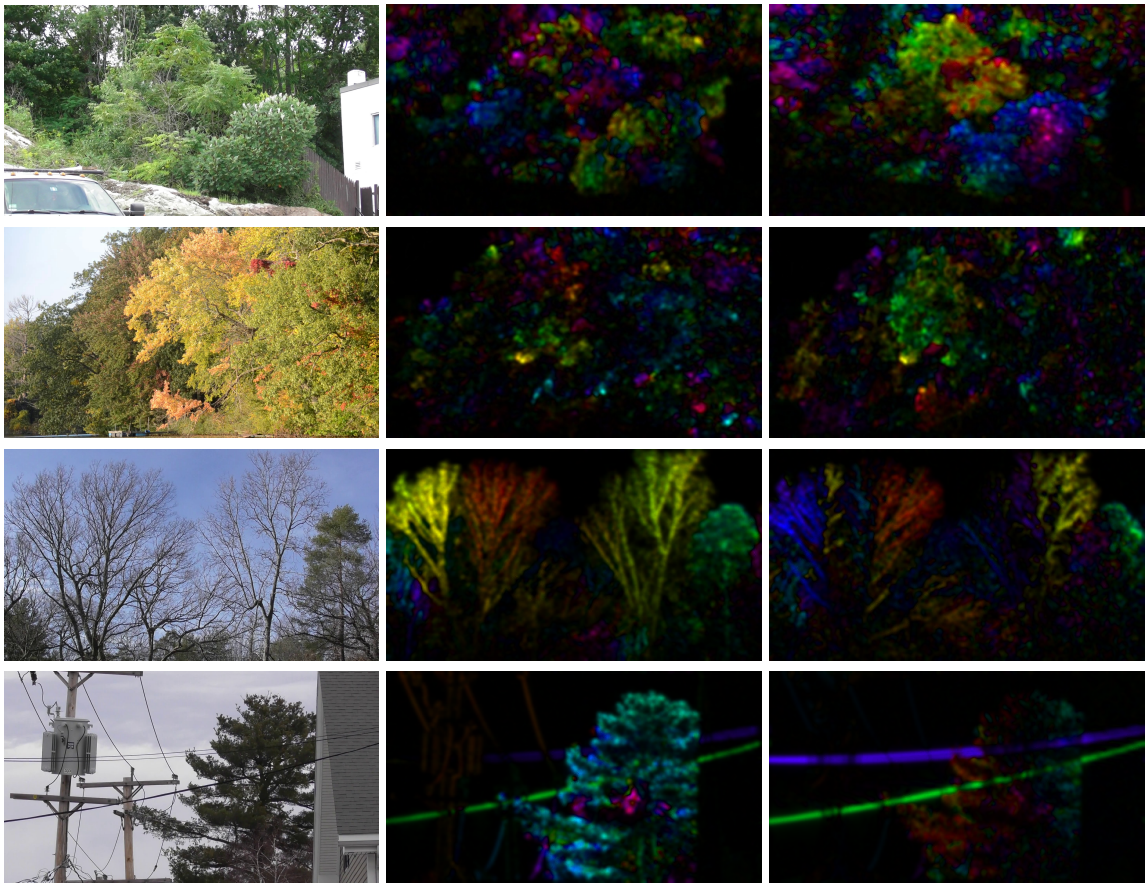

Figure 1: (left) Sample still frames taken from decompressed video as encoded in camera (click to see full size). (middle) complex valued mode of horizontal motions and (right) vertical motions, taken from a still frame of filtered motions in the time domain as described in Section 4. To represent complex numbers using color, complex valued images are decoded as though they were in HSV format, with phase mapped to hue and amplitude mapped to value.

frequency and damping parameters of those modes. The final set of modes represent an *optimized motion palette*, where the motion at each frame is approximated by its coordinates in the modal motion space.

To find the modes, we begin by filtering the images in space using the complex steerable pyramid [12]. We process the local phase signals temporally by taking a difference of Butterworth filters to estimate the local displacements. We employ a local contrast weighted blur [2] to boost the quality of the phase signal.

Once we have approximate local displacements, we can proceed in generating our complex modal images in two ways. If the modal parameters are known in advance, for instance if the same scene is recorded on multiple occasions, the modal images can easily be generated on the fly, using the complex valued temporal IIR filtering described later on in this section. In order to estimate the modal parameters (frequency and damping) used to determine the vibration state as described in Section 5, the

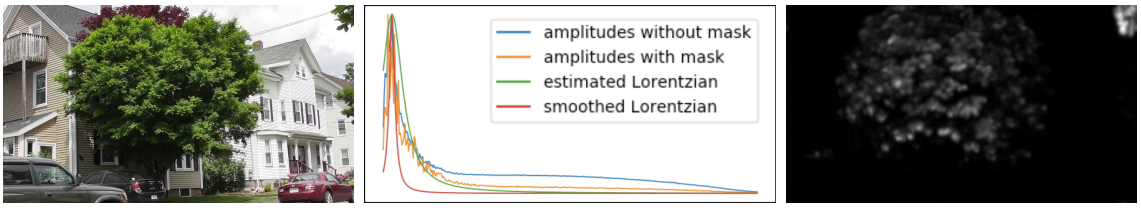


Figure 2: (left) Sample still frame taken from video. (middle) The amplitude responses of the video. (right) Spatial mask for denoising as described in Section 4.

FFT of the local displacements at each spatial location is taken over all of time, and the amplitudes are pooled over all of space, as proposed by [8]. In practice, taking an FFT at each pixel over all of time may be prohibitive, and in theory it is unnecessary because the shape and transfer function of a mode depend only on the intrinsic physical properties of the object, properties which generally don't change over the course of a video. For these reasons, we propose that this stage be performed on a short video clip.

As mentioned earlier, spikes in the amplitude response in the shape of a Lorentzian distribution correspond to eigenmodes of oscillation in the video around some frequency. The internal friction of the oscillating objects cause modal damping which determines the width of these spikes (broad spikes are heavily damped modes which will not resonate with an incoming force for very long, narrow spikes correspond to modes which retain their energy for longer).

Solving for the parameters of an unknown number of Lorentzian distributions which sum up to a given amplitude response is not easy, even outside the presence of noise. No less challenging is solving for the parameters of several damped sinusoids which sum together to make a time domain signal [13]. The authors of [9] and [10] propose a graphic user interface to select modes manually, we solve for the parameters of the vibration directly by fitting a single Lorentzian to the amplitude response using gradient descent. In order to isolate the most salient mode, the amplitude response is scaled to have a maximum amplitude of one, and we exclude the amplitude of the Lorentzian from the parameters to be optimized. The Lorentzian is initialized to have its mode at the frequency with highest amplitude. The noise in the predicted motion vectors causes the amplitude response to differ from the idealized Lorentzian distribution, particularly in the high frequencies where there are spurious motion vectors scattered all over the frame. To reduce this noise, we create a mask by summing together the amplitudes of the top frequencies. An example mask is shown in Figure 2. We use this mask as a spatial window to suppress frequency domain noise outside the region of interest, and recompute the amplitude response, which is now closer to the idealized distribution. Even with the reduced noise, we found our technique still tends to overestimate the damping, often due to combining multiple modes not well separated in frequency. This can lead to unrealistically abrupt motion in the reconstructions. Experimentally, we found that reducing the estimated damping by a constant factor of 4 often improved the perceptual quality of the results, but determining this factor automatically remains future work.

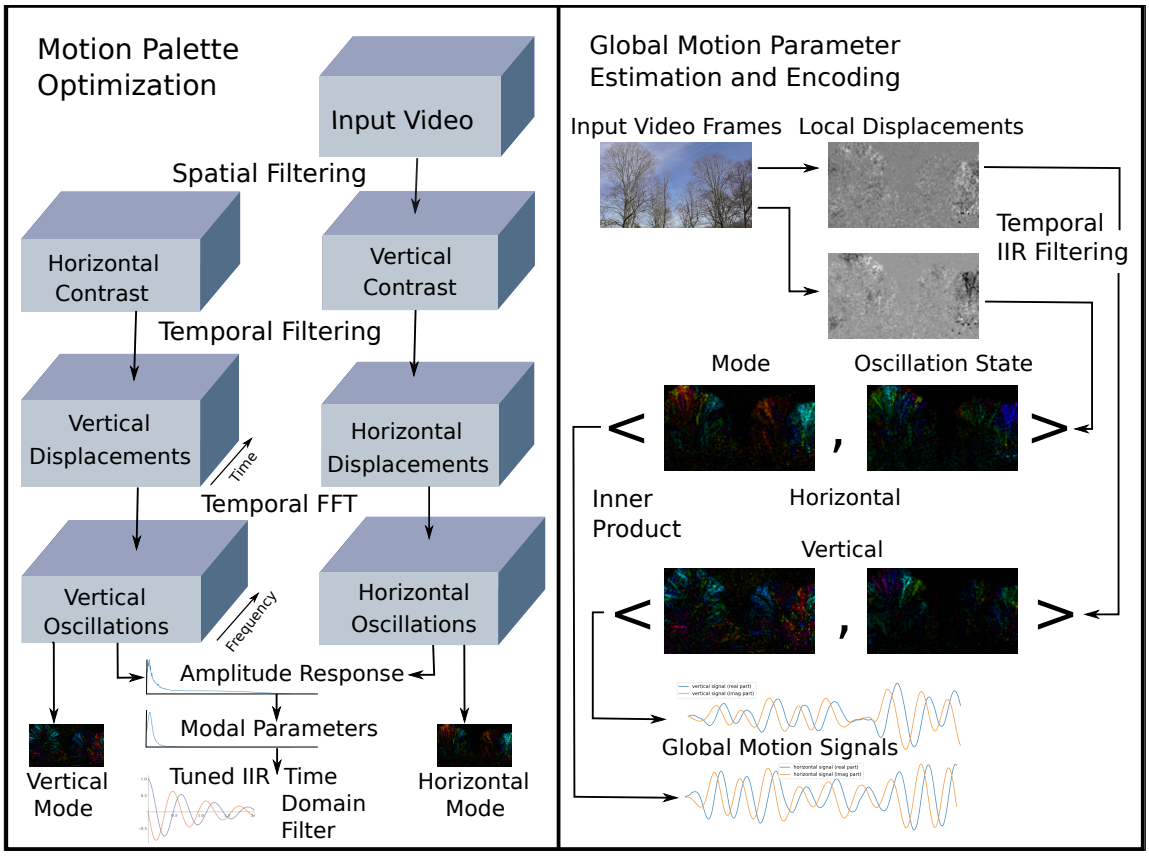


Figure 3: Our video encoding pipeline. Left: Modal parameters are estimated from a short video clip, as described in Section 4. Right: Motion vectors are filtered and projected onto the modal basis to generate global motion parameters as described in Section 5.

After selecting the central frequencies of the modes, the corresponding slices of the FFT can be used as a complex modal motion basis for synthesizing the vibrations of the video. Each mode contains 2 complex valued panes which cover the image plane - one mode for horizontal oscillations, and another for vertical. If the oscillations were truly due to a single vibrating object, the phases in the mode would only take on two values, corresponding to forward and reverse oscillations, as you would see in a standing wave. If this held in practice, we could use real valued modal coordinates and bases. However, many scenes contain multiple oscillating objects which may have arbitrary and even non-static phase relationships to one another, and often even a single tree can exhibit non-binary phases as a result of longitudinal waves rippling through the tree or distant branches which are relatively decoupled. For these reasons we use complex valued modes, which are able to generate more varied, realistic motion.

In our videos, there are often multiple objects in the frame oscillating around similar frequencies. This means that while we can get some benefit from selecting multiple modes around the same frequency (which can help decouple motions of disconnected objects), there is little to be gained by choosing more than one target frequency. In order to generate multiple modes around the same target frequency,

the input displacement signals can be filtered (either in the frequency domain or the time domain) with a complex valued temporal filter with the same impulse response or transfer function as the mode itself. This means that once the parameters of the Lorentzian have been estimated, the distribution can be used as a frequency domain filter. By filtering in the frequency domain and inverting the FFT, the state of the oscillations at each input frame can be estimated. Any still frame from this stream can also be used as a mode for that video with the same modal parameters, just as any background frame from a video can be used as the sprite image. Alternatively, for longer videos where the exemplar was just a fraction of the length of the entire video, filtering and analysis can be kept to the time domain with a complex valued IIR filter of the form:

$$y_t = se^{-i\omega}y_{t-1} + (1-s)x_t$$

where $\omega$ and $s$ are the target frequency and damping factor. For our purposes, we find that plausible motion can be reconstructed from a video using only one mode, however, increasing the number of modes does increase the fidelity, variety, and apparent smoothness of the final decoded motion. These improvements were minimal, and there was no observable benefit with our technique beyond 4 modes.

## 5 Encoder

Once we have selected the modal parameters (frequency and damping) of the modes of vibration we wish to encode, and obtained a suitable modal basis for these parameters, we estimate the state of the vibrations (amplitude and phase) for each frame in the input video, so that the approximate displacements can be reconstructed at each frame. To find the local displacements, we again adopt the technique of [2], shown in the left column of Figure 3, although depending on resource constraints many other techniques could be used, including upsampling the macro block motion vectors in the Pframes and Bframes of standard video codecs like AVC (H.264, MPEG-4 part 10) or HEVC (H.265). In line with the observations of [10], we find that the moving image regions contain a broad spectrum of spatial frequencies, and thus the flow can be measured at a wide variety of scales with similar results. For this reason, we process video frames at two different orientations of the same spatial octave of image contrast. For the octave we used, the signal could be decimated by 4 times in both directions of the image plane, meaning each filter response was 16 times smaller than the input image, allowing for faster processing and a smaller final encoding size for the motion mode.

We further filter the local displacements using the complex IIR filter defined in Section 4. If multiple modes at a several different frequencies and damping factors are desired, this requires the use of a different filter for each frequency, increasing the time and resources required during encoding, however, decoding remains the same and only depends on the number of modal bases chosen, and has no dependence on the number of frequencies those bases represent.

The final mode of motion is encoded as a complex number with a phase and amplitude, where the real and imaginary parts are represented as floats. Since each mode represents motion around a single temporal frequency, the relative amplitudes at each location in the mode represent differences in relative velocity. The floating point

numbers capture a broad range of velocities where some objects may be moving at the same frequency, but with several orders of magnitude less displacement. Because of this, directly quantizing these values visibly diminishes the quality of the simulated motion. However, we found that quantizing the log of the amplitude did a faithful job of producing motion indistinguishable from that of the unquantized modes. We store each mode in four greyscale JPG files, one each for the phases and log amplitudes of the horizontal and vertical modes.

In order to compute the modal coordinate at each time frame, we project onto the modal basis by taking a dot product of the mode with the frame of filtered oscillations at that time. We can then throw away the filtered oscillations and keep only the modes and their coordinates over time.

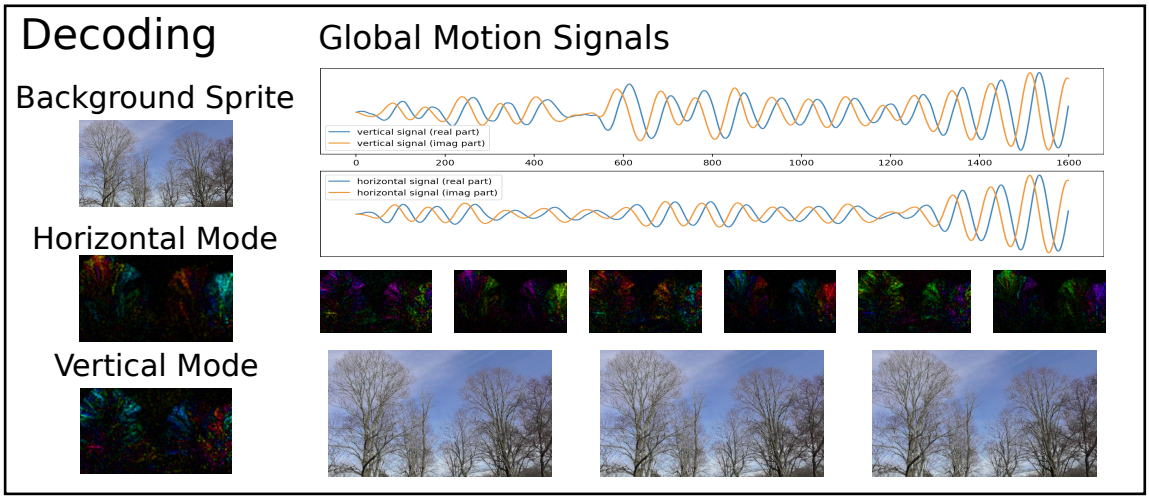


Figure 4: Modal bases are scaled and phase shifted according to the global motion signal, and the background sprite is warped to generate motion.

## 6 Decoder

Given modes $m_i$ and their modal coordinates $\phi_{i,t}$ we can generate a spatial map of displacement vectors to guide our image warping, as shown in Figure 4. This displacement map is computed as $D_t = \sum_i Real(m_i \phi_{i,t})$. Once the modal bases have been multiplied by their respective modal coordinates, summed together and had the imaginary part discarded, they represent motion vectors, and the image can be warped according to these motion vectors. While previous approaches [10, 6] follow a Lagrangian approach to warping the image, by treating the image as a mesh grid that is warped by pushing each pixel along its respective motion vector, we follow a more Eulerian approach, where motion vectors specify the pixel to read from rather than the place to write to. This means our motion decoding works essentially the same as macro block motion compensation, with single pixel macro blocks. This approach is faster and more in line with computations performed by preexisting video codecs. Warping artifacts can still be seen when the motion is large, particularly at object boundaries. These can be diminished if object masks are available. Figure 5 shows the output of our decoder on five test sequences. Note that in columns 2 through 4,


Figure 5: Left column: Sample frames from the original MP4 videos as encoded by the camera. Middle Left column: Expanded view of the box portion shown in the left column taken from AVC encodings. Middle Right Column: Expanded view of the box portion shown in the left column taken from HEVC encodings. Right column: Expanded view of the box portion shown in the left column taken from our decoded videos. Numbers below each frame represent file size (left) and PSNR compared with the original video as compressed in camera (right). PSNR numbers are an average over the video frames of the luminance channel as computed by ffmpeg. Click on any one of these sample frames to see the corresponding video (or you may also go to **rm.cab/sprites**).

the PSNR numbers are poor. However our technique maintains sharp image quality and smooth motion while the AVC and HEVC artifacts result in blurry image quality and jerky motion.

## 7 Conclusion

Our method can represent a broad class of dynamic video backgrounds using a static representation and global motion parameters to provide acceptable visual quality for many applications. By using dense flow fields, our technique avoids blocking artifacts and can provide dynamic backgrounds at a bit rate for which acceptable quality may not be achievable by traditional video compression methods. By projecting the displacements onto our modal basis we are able to reconstruct a visually pleasing approximation to the motion that actually took place in the video. An obvious application of our technique is to the production of videos where foreground and background material are created on different layers. With advances in foreground-background separation techniques, including those employing deep learning e.g.[14, 15, 16], a promising area of future research is to provide low bandwidth versions of existing videos, especially those with persistent but unimportant dynamic backgrounds (e.g., an outdoor soccer game) by essentially eliminating the bandwidth needed for the background. MPEG-4 part 2 remains a supported standard, and our motion sprite video format can be easily integrated into its decoding pipeline. Although sprite mode is not explicitly included in the AVC or HEVC standards, for these and newly emerging standards, it can be approximated by using separate reference frames for foreground and background in the decoding of inter-frames.

## References


[1] H.-Y. Wu et al. "Eulerian video magnification for revealing subtle changes in the world". *TOG* 31 (2012), 65:1–65:8.

[2] N. Wadhwa et al. "Phase-based video motion processing". *TOG* 32.4 (2013), p. 80.

[3] N. Wadhwa et al. "Riesz pyramids for fast phase-based video magnification". *ICCP* (2014).

[4] J. G. Chen et al. "Video camera–based vibration measurement for civil infrastructure applications". *Journal of Infrastructure Systems* 23.3 (2016).

[5] E. Prashnani et al. "A Phase-Based Approach for Animating Images Using Video Examples". *Computer Graphics Forum* 36.6 (2017).

[6] C. Liu et al. "Motion magnification". *TOG* 24.3 (2005), pp. 519–526.

[7] D. J. Fleet and A. D. Jepson. "Computation of component image velocity from local phase information". *IJCV* 5.1 (1990), pp. 77–104.

[8] A. Davis et al. "The visual microphone: passive recovery of sound from video". *TOG* 33.4 (2014), p. 79.

[9] A. Davis et al. "Visual vibrometry: Estimating material properties from small motion in video". *CVPR* (2015), pp. 5335–5343.

[10] A. Davis, J. G. Chen, and F. Durand. "Image-space modal bases for plausible manipulation of objects in video". *TOG* 34.6 (2015), p. 239.

[11] A. A. Stocker and E. P. Simoncelli. "Noise characteristics and prior expectations in human visual speed perception". *Nature neuroscience* 9.4 (2006).

[12] J. Portilla and E. P. Simoncelli. "A parametric texture model based on joint statistics of complex wavelet coefficients". *IJCV* 40.1 (2000).

[13] Y. Hua and T. K. Sarkar. "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise". *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38.5 (1990), pp. 814–824.

[14] K. Rematas et al. "Soccer on Your Tabletop". *CVPR* (2018), pp. 4738–4747.

[15] K. He et al. "Mask R-CNN". *ICCV* (2017), pp. 2980–2988.

[16] A. Prakash et al. "Semantic Perceptual Image Compression Using Deep Convolution Networks". *DCC* (2017), pp. 250–259.