Flexible Voxels for Motion-Aware Videography

Mohit Gupta[1], Amit Agrawal[2],
Ashok Veeraraghavan[2], and Srinivasa G. Narasimhan[1]

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
[2] Mitsubishi Electrical Research Labs, Cambridge, USA

Abstract. The goal of this work is to build video cameras whose spatial and temporal resolutions can be changed post-capture depending on the scene. Building such cameras is difficult due to two reasons. First, current video cameras allow the same spatial resolution and frame rate for the entire captured spatio-temporal volume. Second, both these parameters are fixed *before* the scene is captured. We propose different components of video camera design: a sampling scheme, processing of captured data and hardware that offer post-capture variable spatial and temporal resolutions, independently at each image location. Using the motion information in the captured data, the correct resolution for each location is decided automatically. Our techniques make it possible to capture fast moving objects without motion blur, while simultaneously preserving high-spatial resolution for static scene parts within the same video sequence. Our sampling scheme requires a fast per-pixel shutter on the sensor-array, which we have implemented using a co-located camera-projector system.

1 Introduction

Traditional video cameras offer a fixed spatial resolution (SR) and temporal resolution (TR) independent of the scene. Given a fixed number of measurements (voxels) to sample a space-time volume, the shape of the voxels can vary from 'thin and long' (high SR, low TR) to 'fat and short' (high TR, low SR) as shown in Figure 1. For conventional cameras, the shape of the voxels is fixed *before* capture (scene independent), and is the same for the entire spatio-temporal volume. Can we design video cameras that can choose different spatio-temporal resolutions post-capture, depending on the scene content? We show that it is achievable by a careful choice of per-pixel temporal modulation along with well-designed reconstruction algorithms.

While a high spatial resolution camera captures the fine detail in the static scene parts, it blurs fast moving objects. On the other hand, a high-speed camera captures fast temporal variations but unnecessarily trades off light throughput and spatial resolution for the static and slowly moving scene parts. This fundamental capture limitation can be overcome by designing video cameras with the following two properties: (a) The flexibility to decide the spatio-temporal resolution *post-capture* in a content-aware (scene dependent) manner, and (b)

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 100–114, 2010.

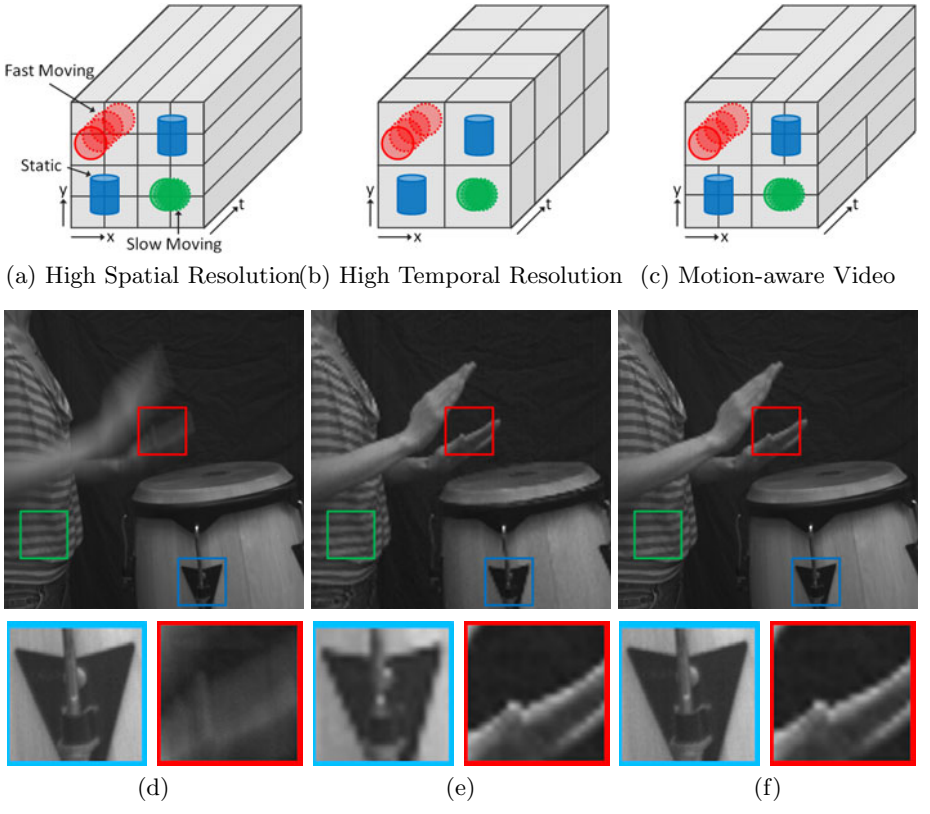


Fig. 1. Different samplings of the space-time volume: For conventional video cameras, the sampling of the space-time volume is decided *before* the scene is captured. Given a fixed voxel budget, a high spatial resolution (SR) camera (a) results in large motion blur and (d) aliasing. A high-speed camera (b) results in low SR even for the static/slow-moving parts of the scene (drums in (e)). With our sampling and reconstruction scheme, the spatio-temporal resolution can be decided *post-capture, independently at each location* in a content-aware manner (c): notice the reduced motion blur for the hands (f) and high SR for the slow-moving parts of the scene.

the ability to make this choice *independently* at each video location. In this paper, we take an initial step towards achieving these goals by demonstrating a hardware setup that enables fast per-pixel temporal modulation, by designing a necessary space-time sampling scheme and by developing simple yet effective motion-aware post-processing interpolation schemes.

We determine necessary conditions for a sampling scheme to allow capturing multiple space-time resolutions simultaneously. Data captured with a sampling scheme which satisfies these conditions can be reconstructed at different spatio-temporal resolutions, independently at each image location. The reconstruction problem is posed as interpolation of scattered samples using well-known anisotropic diffusion techniques. Since the shape of diffusion tensor determines

the local smoothing orientations, by designing different diffusion tensors, we can essentially achieve a *continuum* of effective spatio-temporal resolutions. The *correct* resolution is automatically determined by designing spatially and temporally varying local diffusion tensors based on motion information in the captured data.

Hardware implementation of our sampling scheme requires fast *independent* shutter control of each pixel, which is not possible with available commercial cameras. We have built a prototype using a projector-camera setup which achieves rapid per-pixel temporal modulation during camera integration time. This setup emulates a flexible spatio-temporal resolution camera with a maximum frame rate of 240 Hz, even though the frame rate of the original camera is only 15 Hz. We show several real results that demonstrate variable resolution trade-off in space and time post capture.

1.1 Related Work

Content-based re-sampling and compressive sampling: Content-based re-sampling and representation of data is central to most image/video compression algorithms. Adaptive sampling of data has been used for building content-aware multi-resolution image and video pyramids for fast data transmission [1]. Recently, the field of compressive sensing has exploited sparsity in data at acquisition time, thus reducing the sensing over-head significantly [2,3]. In contrast, our sampling scheme allows re-allocating the saved resources to another dimension in a content-aware manner. If the captured video-stream is sparse in spatial domain, high-frequency detail can be preserved in the temporal dimension and vice-versa.

Multi-dimensional imaging: Several methods trade off spatial resolution to sample other dimensions such as dynamic range [4], wavelength [5], angular dimensions in lightfield [6] and color/polarization [7]. Ben-Ezra et al. [8] used precise sub-pixel detector shifts for increasing the spatial resolution of a video camera. In contrast, our goal is to increase TR much beyond the native frame rate of the camera by trading off SR. Recently, a variety of approaches [9,10,11] which increase TR by trading off SR have been introduced. However, these methods provide *the same* spatio-temporal resolution tradeoff over the entire image. Further, the technique in [11] requires long integration time for a single image capture, making it ill-suited for videos. The method presented in [9] simply rearranges/rebins the captured samples to produce different spatio-temporal resolutions, leading to visual artifacts due to aliasing. Our implementation allows choosing different resolutions for each image location independently, performs fast acquisition (results on dynamic scenes with up to 240 Hz), requires no masks, mitigates aliasing, and is simpler to implement with a regular camera, projector and a beam-splitter.

Spatio-temporal super-resolution using multiple cameras: Hybrid resolution imaging has been used for enhancing the resolution of videos with still

images [12], and for motion deblurring [13]. Wilburn et al. [14] used an array of cameras with temporally staggered short exposures to simulate a high-speed camera. Shechtman et al. [15] combined a set of videos captured at different spatial and temporal resolutions to achieve space-time super-resolution. Agrawal et al. [16] used multiple cameras with multiplexed coding for temporal super-resolution. All these techniques use multiple cameras for capturing videos at different resolutions that need to be decided pre-capture. The number of required cameras scales (at least linearly) with the required temporal speed-up. In contrast, our implementation requires only a single camera and projector, even for large temporal speed-ups.

2 Multi-resolution Sampling of the Space-Time Volume

In this section, we present our multi-resolution space-time sampling scheme. We show that this sampling can provide us with multiple spatio-temporal resolutions at each video location independently, using the same number of measurements as a conventional camera. Consider the group of 4 pixels in Figure 2a. We divide the integration time of each pixel into 4 equal intervals. Each of the 4 pixels is on for only one of the intervals (white indicates on, black indicates off). By switching on each pixel during a different time-interval, we ensure that each pixel samples the space-time volume at different locations.

Different spatio-temporal resolutions can be achieved by simply re-binning these measurements, as illustrated in Figure 2b. For example, the four measurements can be arranged as temporal blocks (marked in red), spatial blocks (marked in blue) or as 2×2 spatio-temporal blocks (marked in green). We define the [TR, SR] factors for a reconstruction as the gain in temporal and spatial resolution respectively over the acquired video. Thus, the [TR, SR] factors for these arrangements are $[4, \frac{1}{4}]$, $[1, \frac{1}{1}]$ and $[2, \frac{1}{2}]$ respectively.

In general, consider the space-time volume V_{mn} defined by a neighborhood of $m \times n$ pixels and one camera integration time, as illustrated in Figure 2, bottom-left. The integration time is divided into $K = mn$ distinct sub-intervals, resulting in K^2 distinct space-time locations. Different divisions of this volume into K equal rectilinear blocks correspond to different spatio-temporal resolutions. An illustration is shown in Figure 2. For the rest of the paper, we will use K for the pixel neighborhood size.

Each division of the volume corresponds to a spatio-temporal resolution[1]. A sampling scheme which facilitates all the resolutions corresponding to the different divisions should satisfy the following property: each block in every division must contain at least one measured sample. Since the total number of measured samples is only K (one for each pixel), each block will contain exactly one sample. Let x_p be the indicator variable for location $p \in \{1, 2, \ldots, K^2\}$, such that x_p is 1 if the p^{th} location is sampled; it is 0 otherwise. Let B_{ij} be the i^{th} block in

[1] The total number of such divisions is the number of distinct factors of K. For example, for $K = 16$, we can have 5 distinct resolutions.

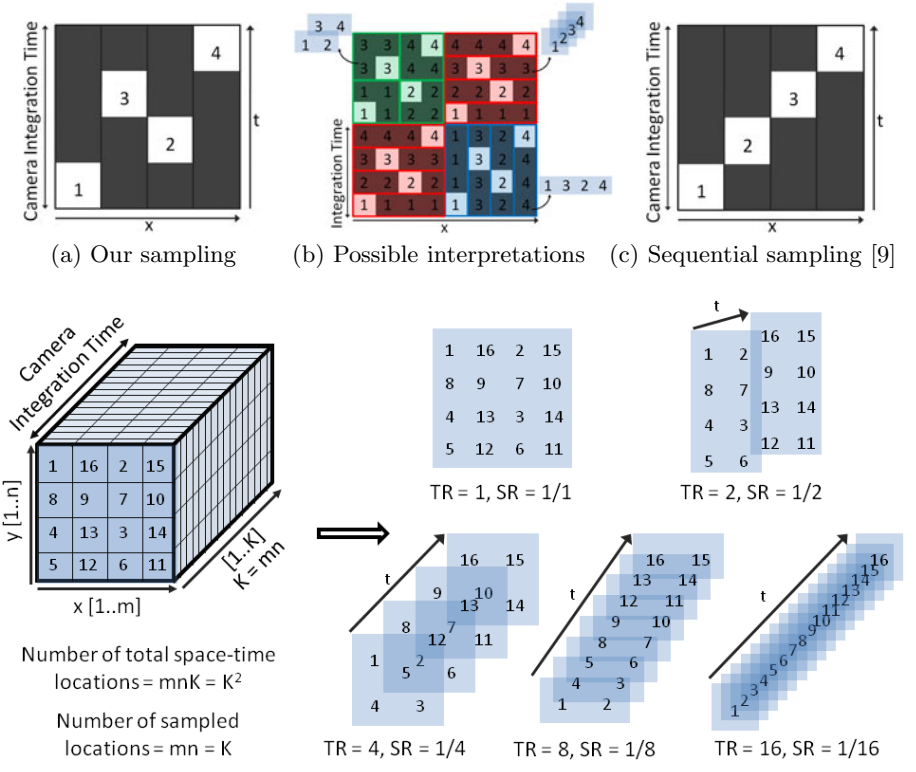


Fig. 2. Simultaneously capturing multiple spatio-temporal resolutions: (a) For a group of K neighboring pixels, each pixel is on for a temporal sub-segment of length $\frac{1}{K}$ (white indicates on, black indicates off). For top row, $K = 4$. (b) These measurements can be interpreted post-capture as 4 temporal measurements (red), 4 spatial measurements (blue) or 4 spatio-temporal measurements (green). (c) Sequential sampling captures only a small sub-set of possible spatio-temporal resolutions. **Bottom row**: The temporal firing order for a group of 4×4 pixels ($K = 16$) and the possible resulting interpretations. With this sampling, we can achieve a temporal resolution gain of up to 16X.

the j^{th} division of the volume. Then, for any pixel-neighborhood of a given size, a multi-resolution sampling can be computed by solving the following binary integer program:

$$\sum_{p \in B_{ij}} x_p = 1 \quad \forall B_{ij}, \qquad \sum_{p=1}^{K^2} x_p = K, \qquad x_p \in \{0,1\} \quad \forall p \tag{1}$$

The first constraint ensures that every block in every division contains exactly one sample. The second constraint enforces the total number of samples to be equal to the number of pixels. For any given V_{mn}, the constraints can be generated automatically by computing different recti-linear divisions of the volume. The bottom row of Figure 2 shows the sampling order for a group of 4×4 pixels

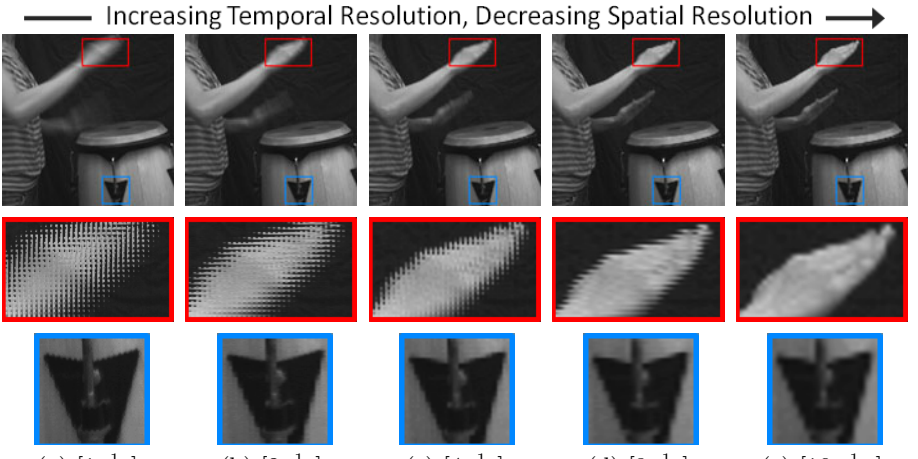
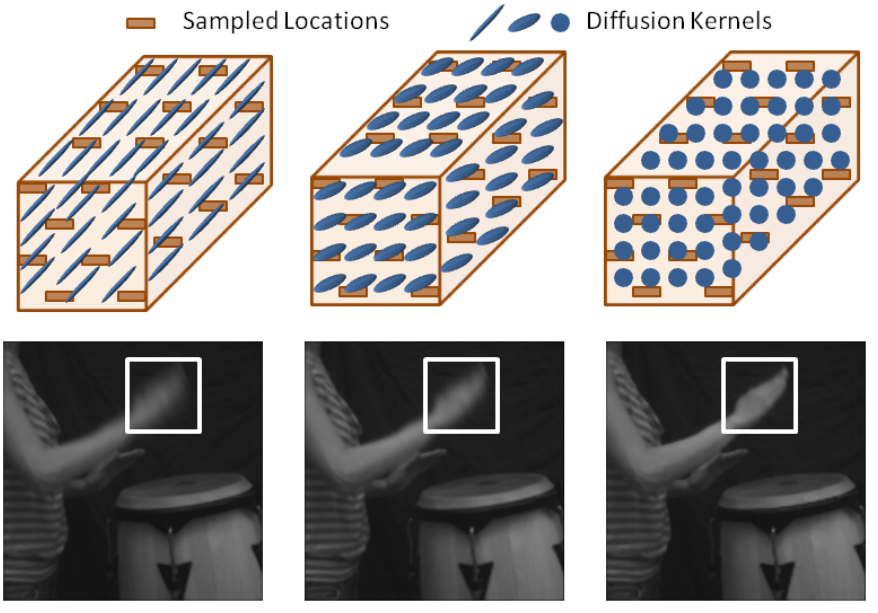


Fig. 3. Generating multiple spatio-temporal resolutions by re-binning captured data: (a) An image acquired with the temporal firing order given in Figure 2 bottom-left. The pixel neighborhood size is 4×4. (a-e) Different re-arrangements of the measurements, as given in Figure 2, and the corresponding [TR, SR] factors. From left to right, motion blur decreases but spatial resolution decreases as well. Simple re-binning of samples results in coded blur artifacts in the reconstructions.

computed by solving the integer program (1). The numbers denote the temporal firing order within an integration time. With this firing order, the samples can be arranged into 5 different spatio-temporal arrangements, shown on the bottom right. These arrangements correspond to resolutions with [TR, SR] factors of $[1, \frac{1}{1}]$, $[2, \frac{1}{2}]$, $[4, \frac{1}{4}]$, $[8, \frac{1}{8}]$ and $[16, \frac{1}{16}]$ as compared to the acquired image. In contrast, sequential sampling [9] does not satisfy the constraints of the above binary integer program. As a result, it is amenable to a small sub-set of possible spatio-temporal resolutions. For the sequential sampling given in Figure 2c, the 2×2 arrangement is not possible since not all the blocks are sampled.

Simulating multi-resolution sampling: To verify the feasibility of our multi-resolution sampling scheme, we used a Photron 1024 PCI camera to capture high-speed images at 480 Hz. The spatial resolution of the images is 640×480. The image is divided into neighborhoods of 4×4 pixels. For each set of 16 consecutive frames, we weighted them according to the per-pixel code given on the bottom-left of Figure 2 and added them together. The resulting video is as if captured by a 30 Hz camera with a per-pixel shutter operating at 480 Hz. The scene consists of a person playing drums. While the hands move rapidly, the rest of the body moves slowly, and the drums move only on impact. An example image from the sequence is given in Figure 3 (top-left). Notice the per-pixel coded blur on the captured image (Figure 3a) as compared to usual smooth motion blur in regular cameras. This is because pixels encode temporal information as well. By rearranging the pixels according to Figure 2, we get sequences with different



(a) Low TR, High SR (b) Medium TR, Medium SR (c) High TR, Low SR

Fig. 4. Anisotropic diffusion for generating multiple spatio-temporal resolutions: By interpolating the captured data with diffusion tensors of varying spectral shapes, we can achieve multiple spatio-temporal resolutions. The diffusion process also mitigates the effects of aliasing. Notice that coded blur artifacts are significantly reduced in comparison to the simple rebinning scheme of Figure 3.

combinations of spatio-temporal resolutions, as shown in Figure 3 (b-e) . From left to right, temporal resolution increases but the spatial resolution decreases.

3 Interpreting the Captured Data

In this section, we present post-capture algorithms for interpreting the data captured using our sampling scheme. One approach is simply re-arranging the measured samples to generate different spatio-temporal resolutions, as mentioned in the previous section. This scheme has two disadvantages: first, it restricts the possible spatio-temporal resolutions of the reconstructions to a few discrete choices. Second, it does not account for aliasing due to sub-sampling. Consequently, we witness disturbing visual artifacts such as coded blur (Figure 3) and temporal incoherence (pixel swimming). Such artifacts are specially noticeable in the presence of highly textured scene objects. In the following, we present a reconstruction algorithm which effectively addresses these limitations.

3.1 Interpolation of Sub-sampled Data Using Anisotropic Diffusion

Let $I_{(0)}$ be the initial space-time volume defined over a regular $3D$ grid. Our sampling scheme measures samples at a few locations in this volume. The re-

maining locations are considered missing data, as illustrated in Figure 4. We pose the reconstruction problem as *inpainting* the missing data by interpolating the measured samples using anisotropic diffusion [17,18]. The key idea is that by diffusing the intensities with tensors T of different spectral shapes (orientation), we can achieve different *effective* spatio-temporal resolutions. Consider the evolution of the image data with the number of iterations n:

$$\frac{\partial I}{\partial n} = \mathbf{trace}(TH) \ , \quad \text{where} \ \ H = \begin{bmatrix} I_{xx} & I_{xy} & I_{xt} \\ I_{yx} & I_{yy} & I_{yt} \\ I_{tx} & I_{ty} & I_{tt} \end{bmatrix} \tag{2}$$

is the 3×3 Hessian matrix of the $3D$ image data I. The 3×3 diffusion tensor defined by $T = c_1 \lambda \lambda^T + c_2 \psi \psi^T + c_3 \gamma \gamma^T$ [18] is characterized by its eigen values c_1, c_2, c_3 and eigen vectors λ, ψ, γ. The solution of the PDE of Eqn. 2 is [18]:

$$I_{(n)} = I_{(0)} * G^{(T,n)} \ , \quad \text{where} \ \ G^{(T,n)}(\mathbf{x}) = \frac{1}{4\pi n} \exp(-\frac{\mathbf{x}^T T^{-1} \mathbf{x}}{4n}) \ , \tag{3}$$

where $\mathbf{x} = (x \ y \ t)^T$. Starting with the initial volume $I_{(0)}$, this PDE has the effect of progressively smoothing the data with oriented $3D$ Gaussians[2] defined by the tensor T. The PDE is repeatedly applied only on the missing data locations until the intensities from the measured samples diffuse to fill in the holes.

A continuum of spatio-temporal resolutions: By designing diffusion tensors of different spectral shapes, we can achieve different spatio-temporal resolutions of the reconstructed volume. Consider the set of axis-aligned ellipsoidal kernels $T = diag\,(c_1, c_2, c_3)$. If $c_3 >> c_1$ and $c_3 >> c_2$, low-pas filtering occurs primarily in the temporal direction. Consequently, high-frequency content in the spatial direction is preserved. The resulting reconstruction, thus, has high spatial resolution and low temporal resolution, as illustrated in Figure 4a. On the other hand, if $c_3 << c_1$ and $c_3 << c_2$, then most of the smoothing happens in the spatial direction, thus preserving high-frequency content in the temporal direction (Figure 4c). With $c_1 = c_2 = c_3$, the data is diffused isotropically in all three directions (Figure 4b). The reconstructions achieved with the simple scheme of re-arranging samples correspond to special cases of the diffusion tensor. For example, the $[1, \frac{1}{1}]$ reconstruction can be achieved by using a tensor with $c_1 = c_2 = 0, c_3 = 1$. Similarly, with $c_1 = c_2 = 1, c_3 = 0$, we can achieve the $[16, \frac{1}{16}]$ reconstruction.

Aliasing artifacts: The diffusion process interpolates and regularizes the data on the $3D$ grid, thus mitigating the effects of aliasing due to sub-sampling. Consequently, coded blur and temporal coherence artifacts are significantly reduced in the reconstructions. **See the project web-page [19] for comparisons**.

[2] An equivalent representation of the tensor T is in terms of oriented ellipsoids.

(a) (b) (c)

Fig. 5. Motion-aware video reconstruction: (a) Quiver plot of the optical flow between two successive frames of a high TR reconstruction. (b) Color coded magnitude of the optical flow. Red indicates fast moving objects, green indicates slow moving and blue indicates stationary objects. Raw data is interpolated with diffusion tensors oriented along the optical flow vectors (c) to achieve a motion aware reconstruction. The resulting frame is shown in Figure 1f.

4 Motion-Aware Video

The reconstruction algorithms discussed so far are independent of the captured data, which, although sparse, can provide useful information about the scene. In this section, we present an algorithm to use the motion information in the captured data to drive the reconstruction process. We call the resulting reconstruction *motion-aware*: the spatio-temporal resolution trade-off at each location is resolved according to the motion information at that location. Such a reconstruction would minimize the motion blur for fast moving objects while simultaneously maximizing the spatial frequency content for slow moving or static objects. Following is the algorithm we use for computing such a reconstruction:

Step 1: High TR reconstruction: It can be extremely difficult to recover faithful motion information in the presence of large motion blur. Thus, a high temporal resolution reconstruction is imperative for computing accurate motion information. Our first step is to do a high TR reconstruction using an axis-aligned tensor $T = diag\,(c_1, c_2, c_3)$ with $(c_1, c_2, c_3) = (1.0, 1.0, 0.05)$. Such a reconstruction would smooth primarily in the spatial dimensions, thus preserving high-frequency temporal content. A small value is assigned to c_3 to mitigate temporal flickering artifacts.

Step 2: Computing optical flow: We compute motion information in the form of optical flow between successive frames of the high TR reconstruction. For this, we used an implementation of the optical flow method given by Brox et al [20]. Since computed on a high TR reconstruction, the optical flow estimates are fairly robust, even for fast moving objects. Figures 5a and 5b illustrate the optical flow between two successive frames of the drums sequence using a quiver plot and color coded magnitudes respectively. Red indicates fast moving objects, green indicates slow moving and blue indicates stationary objects. Although the

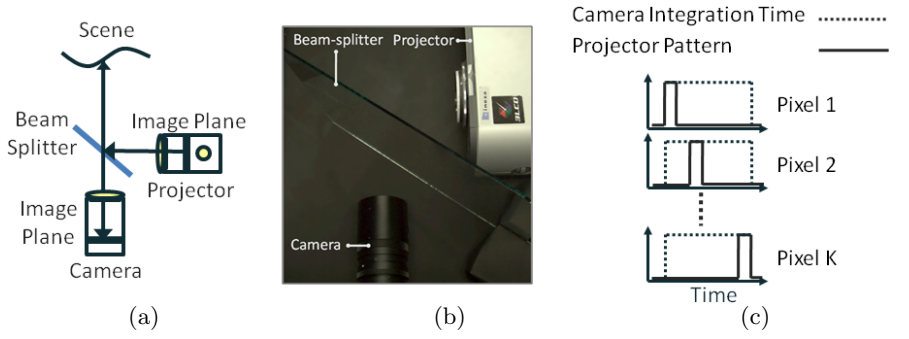


Fig. 6. Hardware setup for simulating per-pixel shutter: (a-b) Our setup consists of co-locating and temporally synchronizing a camera (15 Hz) and a projector (240 Hz). Under no global illumination, a camera pixel receives light only when the corresponding projector pixel is on. (c) The observed irradiance at a camera pixel is modulated according to the binary pattern on the corresponding projector pixel.

optical flow vectors have high temporal resolution, their spatial resolution is much lesser than that of the scene itself. Thus, computing optical flow at a low spatial resolution does not result in significant spatial aliasing. In contrast, optical flow estimates on the original captured data are unreliable due to the presence of large, albeit coded motion blur.

Step 3: Motion driven diffusion: The key idea is to design diffusion tensors at each location so that they smooth along the motion direction. Let $(u, v, 1)$ be the optical flow vector at a given location. We define the diffusion tensor as $T = c_1\lambda\lambda^T + c_2\psi\psi^T + c_3\gamma\gamma^T$, where

$$\lambda = \frac{(u, v, 1)}{\sqrt{u^2 + v^2 + 1}}, \quad \psi = \lambda \times (0, 0, 1), \quad \gamma = \lambda \times \psi \tag{4}$$

form an ortho-normal set of unit vectors. By choosing $c_1 = 0.95, c_2 = 0.05, c_3 = 0.05$, we orient the diffusion tensor sharply along λ, the motion direction. Note that this results in a **variable** diffusion tensor field over the space-time volume (Figure 5c) as different locations have different optical flow vectors. An example frame from the motion-aware reconstruction of the drums sequence is given in Figure 1f. Note that the motion blur is minimized on the fast moving hands while the drums and the body retain high spatial resolution. Results with real experimental data are given in Figures 7 and 8.

5 Hardware Implementation of Per-Pixel Shutter

The sampling scheme discussed in the previous sections requires a fast (K times the frame-rate of the camera) per-pixel shutter on the sensor array. Currently

available cameras have fast global shutters[3] implemented as external trigger modes [22]. However, these modes do not provide per-pixel control. Recently, DMD arrays have been used to provide precise, per-pixel temporal modulation [9,23]. These devices are commonly used as light modulators in off-the shelf DLP projectors. We have implemented per-pixel shutter using a DLP projector in conjunction with a camera. The projector is used to provide fast, per-pixel light modulation externally.

The projector and the camera are *co-located* using a beam-splitter, as shown in Figure 6. The setup is placed in a dark room. We assume that there is no ambient or global illumination. Co-location is achieved by aligning the camera and the projector so that the camera does not observe any shadows cast by the projector. This procedure takes about 15 mins. Co-location ensures that the camera and the projector image planes are related by a single homography irrespective of the scene.

The camera and the projector are temporally synchronized so that for each camera integration time, the projector cycles through K binary patterns. The binary patterns consist of tiles of K pixels repeated spatially. Each tile encodes the sampling scheme being used. Since there is no ambient illumination, a camera pixel receives light only when the corresponding projector pixel is on. Consequently, the irradiance at a camera pixel is modulated according to the binary pattern on the corresponding projector pixel. An illustration is shown in Figure 6c. This modulation acts as per-pixel shutter. The temporal frequency of modulation (hence the shutter), is given by the frame rate of the projector.

We used a Point-Grey Flea2 camera and a Multi-Use-Light-Engine (MULE) projector [24]. With a 60 Hz video input, the MULE projector can project binary bit-planes at up to $60 \times 24 = 1440$ Hz. To implement the coding scheme given in Figure 3a, we operated the projector at 240 Hz, thus achieving a frame-rate of 240 Hz even though the frame rate of the camera is 15 Hz.

5.1 Real Experiments and Results

Fan rotating scene (Figure 7): The first sequence consists of a rotating fan acquired with a camera running at 7.5 Hz. The frames have significant motion blur and temporal aliasing. In this case, the pixel neighborhood size was 2×4; thus, $K = 8$. The second and the third columns show 1 frame each from two reconstructions done with the diffusion tensors $T = diag\,(0.05, 0.05, 1)$ and $T = diag\,(1, 1, 0.05)$ respectively. We call these motion-independent reconstructions, as these reconstructions do not use any motion information. The high TR reconstruction has a temporal resolution of $7.5 \times 8 = 60$ Hz. The fourth column shows optical flow magnitudes between two successive frames of the high TR reconstruction. The optical flow information is used for computing a motion-aware reconstruction (last column), as discussed in Section 4.

[3] Fast global shutters have been used in the past for motion deblurring [21], but modulate all pixels simultaneously.

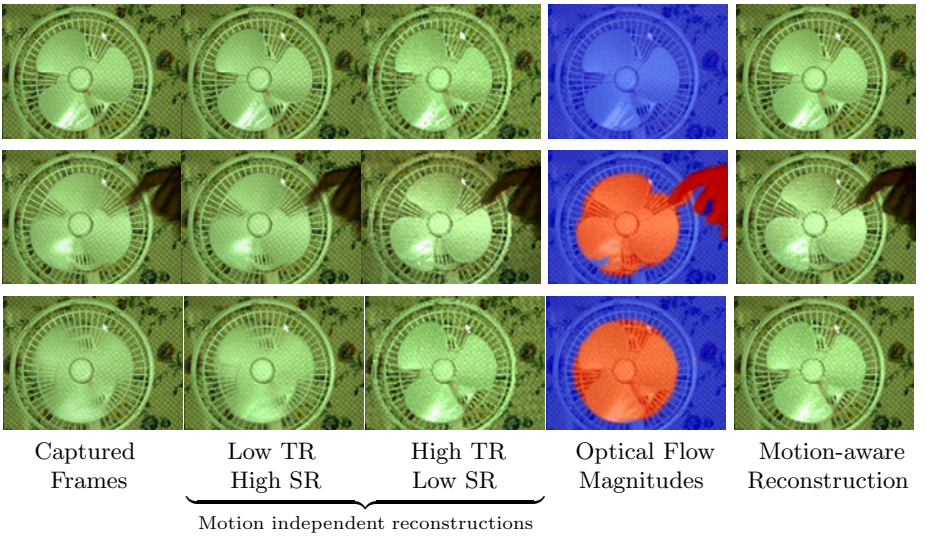


Fig. 7. Motion-aware video of rotating fan: (First column) Raw frames from the captured sequence. (Second and the third columns) One frame each from two reconstructions done with different diffusion tensors. (Fourth column) Optical flow magnitudes between two successive frames of the high TR reconstruction. (Last column) Motion aware reconstruction. Notice the much reduced motion blur on the fan and high-spatial resolution on the static background. Zoom in for details.

Multiple Balls Bouncing (Figure 8): This sequence consists of multiple balls colliding with each other at high velocities. The camera is running at 15 Hz. We used a pixel neighborhood of 4×4; thus, $K = 16$. The second and third columns show one frame each from reconstructions with tensors $T = diag\,(0.05, 0.05, 1)$ and $T = diag\,(1, 1, 0.05)$ respectively. The last column shows motion-aware reconstruction. Notice that one of the balls is almost invisible in the captured frame of third row due to large motion blur. In the motion aware reconstruction, it can be easily localized.

6 Discussion and Limitations

The goal of this work was to build video cameras whose spatial and temporal resolutions can be changed post-capture depending on the scene. We have presented the first example of an imaging system which allows multiple space-time resolutions at each image location independently - using programmable, fast per-pixel shutters and a content-aware post-processing scheme.

A limitation of our sampling scheme is that the pixels collect light over only a fraction of the integration time leading to low signal-to-noise ratio (SNR). The trade-off between temporal resolution and SNR is well known for video cameras. High-speed cameras suffer from significant image noise in low-light

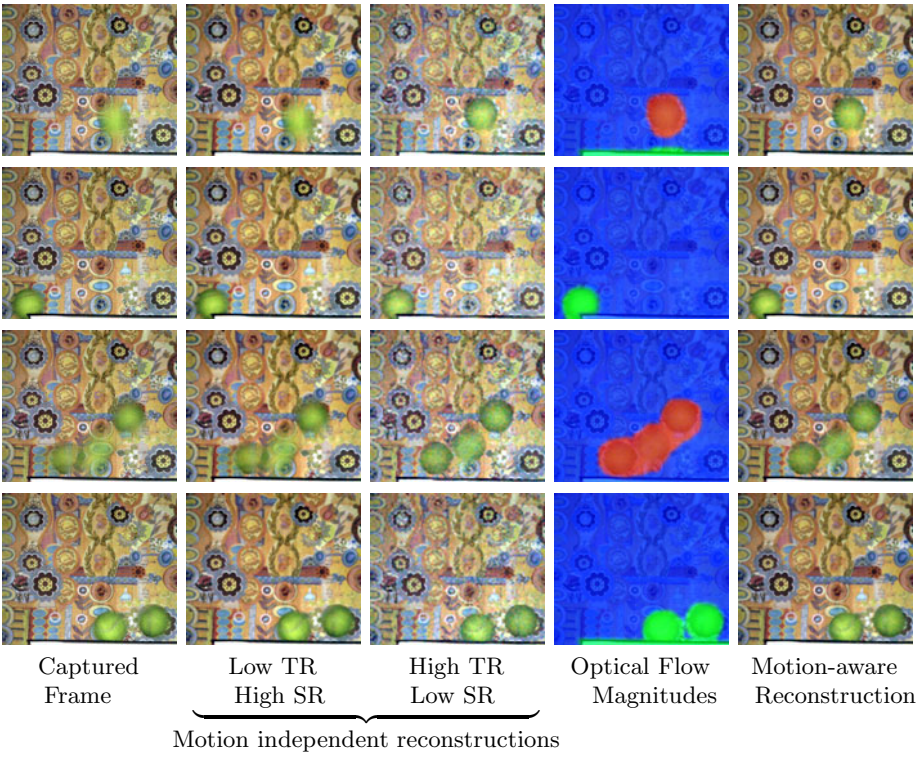


Fig. 8. Motion-aware video of multiple bouncing balls: (First column) Raw frames from the captured sequence. (Second-third columns) One frame each from two reconstructions done with different diffusion tensors. (Fourth column) Optical flow magnitudes between two successive frames of the highest TR reconstruction. (Last column) Motion aware reconstruction.

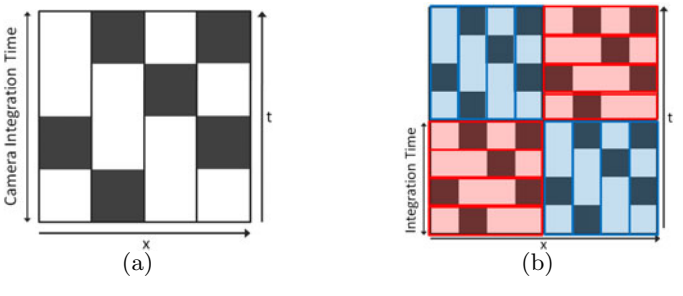


Fig. 9. Multiplexed sampling: By using multiplexed codes (a), each pixel gathers more light resulting in higher SNR (white indicates on, black indicates off). Post-capture reshaping of voxels (b) can be achieved by de-multiplexing the captured data.

conditions. This trade-off can be countered by incorporating multiplexing into our sampling scheme. With multiplexed codes, as shown in Figure 9a, each pixel gathers more light as compared to identity codes (Figure 2a). This is similar in spirit to capturing images using multiplexed illumination for achieving higher SNR [25]. Post-capture reshaping of voxels can be achieved by de-multiplexing.

Our implementation of per-pixel shutter using a projector-camera system is limited to scenes with low global and ambient illumination. Passive implementations using either a DMD array [9,23] or variable integration on sensor chip can effectively address these limitations.

Acknowledgments. This research was supported in parts by ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762, Okawa Foundation Grant and NSF CAREER award IIS-0643628 and Mitsubishi Electrical Research Labs. Authors thank Jinwei Gu and Shree K. Nayar for use of the MULE projector.

References

1. Zabrodsky, H., Peleg, S.: Attentive transmission. J. of Visual Comm. and Image Representation 1 (1990)
2. Baraniuk, R.: Compressive sensing. IEEE Signal Processing Magazine 24 (2007)
3. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. ACM Trans. Graph 28 (2009)
4. Nayar, S.K., Mitsunaga, T.: High dynamic range imaging: spatially varying pixel exposures. In: IEEE CVPR (2000)
5. Narasimhan, S.G., Nayar, S.K.: Enhancing resolution along multiple imaging dimensions using assorted pixels. PAMI 27 (2005)
6. Ng, R.: Fourier slice photography. ACM Trans. Graphics 24, 735–744 (2005)
7. Horstmeyer, R., Euliss, G., Athale, R., Levoy, M.: Flexible multimodal camera using a light field architecture. In: ICCP (2009)
8. Ben-Ezra, M., Zomet, A., Nayar, S.K.: Video super-resolution using controlled subpixel detector shifts. PAMI 27 (2005)
9. Bub, G., Tecza, M., Helmes, M., Lee, P., Kohl, P.: Temporal pixel multiplexing for simultaneous high-speed, high-resolution imaging. Nature Methods (2010)
10. Gu, J., Hitomi, Y., Mitsunaga, T., Nayar, S.K.: Coded rolling shutter photography: Flexible space-time sampling. In: ICCP (2010)
11. Agrawal, A., Veeraraghavan, A., Raskar, R.: Reinterpretable imager: Towards variable post capture space, angle & time resolution in photography. In: Eurographics (2010)
12. Gupta, A., Bhat, P., Dontcheva, M., Curless, B., Deussen, O., Cohen, M.: Enhancing and experiencing spacetime resolution with videos and stills. In: ICCP (2009)
13. Ben-Ezra, M., Nayar, S.K.: Motion-based motion deblurring. PAMI 26 (2004)
14. Wilburn, B., Joshi, N., Vaish, V., Levoy, M., Horowitz, M.: High speed video using a dense camera array. In: IEEE CVPR (2004)
15. Shechtman, E., Caspi, Y., Irani, M.: Space-time super-resolution. PAMI 27 (2005)
16. Agrawal, A., Gupta, M., Veeraraghavan, A., Narasimhan, S.G.: Optimal coded sampling for temporal super-resolution. In: IEEE CVPR (2010)

17. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE PAMI 12 (1990)
18. Tschumperle, D., Deriche, R.: Vector-valued image regularization with pdes: A common framework for different applications. PAMI 27 (2005)
19. Project web-page, http://graphics.cs.cmu.edu/projects/FlexibleVoxels/
20. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
21. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans. Graphics 25, 795–804 (2006)
22. External trigger modes supported by point grey cameras, http://www.ptgrey.com/support/kb/
23. Nayar, S.K., Branzoi, V., Boult, T.E.: Programmable imaging: Towards a flexible camera. In: IJCV (2006)
24. McDowall, I., Bolas, M.: Fast light for display, sensing and control applications. In: IEEE VR 2005 Workshop on Emerging Display Technologies (March 2005)
25. Schechner, Y., Nayar, S.K., Belhumeur, P.: A theory of multiplexed illumination. In: ICCV (2003)