Space-Time Super-Resolution

Eli Shechtman, Yaron Caspi, and Michal Irani, *Member*, *IEEE*

Abstract—We propose a method for constructing a video sequence of high space-time resolution by combining information from multiple low-resolution video sequences of the same dynamic scene. Super-resolution is performed simultaneously in time and in space. By "temporal super-resolution," we mean recovering rapid dynamic events that occur faster than regular frame-rate. Such dynamic events are not visible (or else are observed incorrectly) in any of the input sequences, even if these are played in "slow-motion." The spatial and temporal dimensions are very different in nature, yet are interrelated. This leads to interesting visual trade-offs in time and space and to new video applications. These include: 1) treatment of *spatial* artifacts (e.g., motion-blur) by increasing the *temporal* resolution and 2) combination of input sequences of different space-time resolutions (e.g., NTSC, PAL, and even high quality still images) to generate a high quality video sequence. We further analyze and compare characteristics of temporal super-resolution to those of spatial super-resolution. These include: How many video cameras are needed to obtain increased resolution? What is the upper bound on resolution improvement via super-resolution? What is the temporal analogue to the spatial "ringing" effect?

Index Terms—Super-resolution, space-time analysis, temporal resolution, motion blur, motion aliasing, high-quality video, fast cameras.

1 Introduction

A video camera has limited spatial and temporal resolution. The spatial resolution is determined by the spatial density of the detectors in the camera and by their induced blur. These factors limit the minimal size of spatial features or objects that can be visually detected in an image. The temporal resolution is determined by the frame-rate and by the exposure-time of the camera. These limit the maximal speed of dynamic events that can be observed in a video sequence.

Methods have been proposed for increasing the spatial resolution of images by combining information from multiple low-resolution images obtained at subpixel displacements (e.g., [1], [2], [3], [6], [7], [11], [13], [14], [15], [16]. See [4] for a comprehensive review). An extension of [15] for increasing the spatial resolution in three-dimensional (x, y, z) medical imagery has been proposed in [12], where MRI data was reconstructed both within image slices (x and y axis) and between the slices (z axis).

The above-mentioned methods, however, usually assume static scenes with limited *spatial* resolution and do not address the limited *temporal* resolution observed in dynamic scenes. In this paper, we extend the notion of super-resolution to the *space-time* domain. We propose a unified framework for increasing the resolution both in time and in space by combining information from multiple *video sequences* of dynamic scenes obtained at (subpixel) spatial and (subframe) temporal misalignments. As will be shown, this enables new visual capabilities of dynamic events, gives rise to visual trade-offs between time and space, and leads to new video applications. These are substantial in the presence of very fast dynamic events. From here on, we will use SR as an abbreviation for the frequently used term "super-resolution."

Rapid dynamic events that occur faster than the frame-rate of video cameras are not visible (or else captured incorrectly) in the recorded video sequences. This problem is often evident in sports videos (e.g., tennis, baseball, hockey), where it is impossible to see the full motion or the behavior of the fast moving ball/puck. There are two typical visual effects in video sequences which are caused by very fast motion. One effect (motion blur) is caused by the exposure-time of the camera and the other effect (motion aliasing) is due to the temporal subsampling introduced by the frame-rate of the camera:

1. *Motion Blur*: The camera integrates the light coming from the scene during the exposure time in order to generate each frame. As a result, fast moving objects produce a noted blur along their trajectory, often resulting in distorted or unrecognizable object shapes. The faster the object moves, the stronger this effect is, especially if the trajectory of the moving object is not linear. This effect is notable in the distorted shapes of the tennis ball shown in Fig. 1. Note also that the tennis racket also "disappears" in Fig. 1b. Methods for treating motion blur in the context of image-based SR were proposed in [2], [1]. These methods, however, require prior segmentation of moving objects and the estimation of their motions. Such motion analysis may be impossible in the presence of severe shape distortions of the type shown in Fig. 1. We will show that, by increasing the *temporal resolution* using information from multiple video sequences, *spatial artifacts* such as motion blur can be handled without needing to separate static

- *E. Shechtman is with the Department of Computer Science and Applied Math, Ziskind Building, Room 36, The Weizmann Institute of Science, Rehovot, 76100 Israel. E-mail: eli.shechtman@weizmann.ac.il.*
- *Y. Caspi is with the School of Computer Science, Tel Aviv University, Kaplun Building, Room 330, Tel Aviv 69978 Israel. E-mail: caspiy@tau.ac.il.*
- *M. Irani is with the Department of Computer Science and Applied Math, Ziskind Building, Room 224, The Weizmann Institute of Science, Rehovot, 76100 Israel. E-mail: michal.irani@weizmann.ac.il.*

Manuscript received 8 Jan. 2004; revised 25 June 2004; accepted 10 Sept. 2004; published online 10 Feb. 2005.
Recommended for acceptance by S. Soatto.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0022-0104.

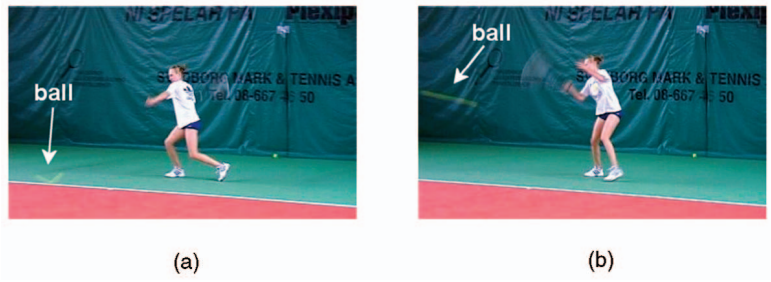

Fig. 1. Motion blur. Distorted shape due to motion blur of very fast moving objects (the tennis ball and the racket) in a real tennis video. The perceived distortion of the ball is marked by a white arrow. Note the "V"-like shape of the ball in (a) and the elongated shape of the ball in (b). The racket has almost "disappeared."

and dynamic scene components or estimate their motions.

2. *Motion-Based (Temporal) Aliasing*: A more severe problem in video sequences of fast dynamic events is false visual illusions caused by aliasing in time. Motion aliasing occurs when the trajectory generated by a fast moving object is characterized by frequencies which are higher than the frame-rate of the camera (i.e., the temporal sampling rate). When that happens, the high temporal frequencies are "folded" into the low temporal frequencies. The observable result is a distorted or even false trajectory of the moving object. This effect is illustrated in Fig. 2, where a ball moves fast in sinusoidal trajectory of high frequency (Fig. 2a). Because the frame-rate is much lower (below the Nyquist frequency of the trajectory), the *observed* trajectory of the ball over time is a straight line (Fig. 2b). Playing that video sequence in "slow-motion" will not correct this false visual effect (Fig. 2c). Another example of motion-based aliasing is the well-known visual illusion called the "wagon wheel effect": When a wheel is spinning very fast, beyond a certain speed it will appear to be rotating in the "wrong" direction.

Neither the motion-based aliasing nor the motion blur can be treated by playing such video sequences in "slow-motion," even when sophisticated temporal interpolations are used to increase the frame-rate (as in video format conversion or "retiming" methods [10], [20]). This is because the information contained in a single video sequence is insufficient to recover the missing information of very fast dynamic events. The high temporal resolution has been lost due to excessive blur and excessive subsampling in time. Multiple video sequences, on the other hand, provide additional samples of the dynamic space-time scene. While none of the individual sequences provides enough visual information, combining the information from all the sequences allows us to generate a video sequence of high space-time resolution which displays the correct dynamic events. Thus, for example, a reconstructed high-resolution sequence will display the correct motion of the wagon wheel despite it appearing incorrectly in *all* of the input sequences.

The spatial and temporal dimensions are very different in nature, yet are interrelated. This introduces visual trade-offs between space and time, which are unique to spatio-temporal SR, and are not applicable in traditional spatial (i.e., image-based) SR. For example, output sequences of different space-time resolutions can be generated from the same input sequences. A large increase in the temporal resolution usually comes at the expense of a large increase in the spatial resolution and vice versa.

Furthermore, input sequences of different space-time resolutions can be meaningfully combined in our framework. In traditional image-based SR, there is no benefit in combining input images of different spatial resolutions since a high-resolution image will subsume the information contained in a low-resolution image. This, however, is not the case here. Different types of cameras of different space-time resolutions may provide *complementary* information. Thus, for example, we can combine information obtained by high-quality still cameras (which have very high spatial-resolution, but extremely low "temporal resolution") with information obtained by standard video cameras (which have low spatial-resolution but higher temporal resolution) to obtain an improved video sequence of high spatial and high temporal resolution.

Differences in the physical properties of temporal versus spatial imaging lead to marked differences in performance and behavior of temporal SR versus spatial SR. These include issues such as: the upper bound on improvement in resolution, synchronization configurations, and more. These issues are also analyzed and discussed in this paper.

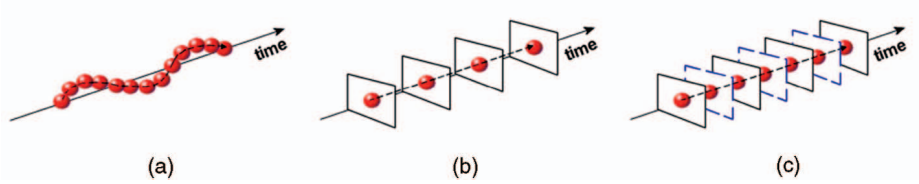

Fig. 2. Motion aliasing. (a) shows a ball moving in a sinusoidal trajectory over time. (b) displays an image sequence of the ball captured at low frame-rate. The preceived motion is along a straight line. This false perception is referred to in the thesis as "motion aliasing." (c) Illustrates that, even using an ideal temporal intyerpolation for "slow motion" will not produce the correct motion. The filled-in frames are indicated by dashed blue lines. In other words, the false perception cannot be corrected by playing the video sequence in slow-motion as the information is already lost in the video recording (b).

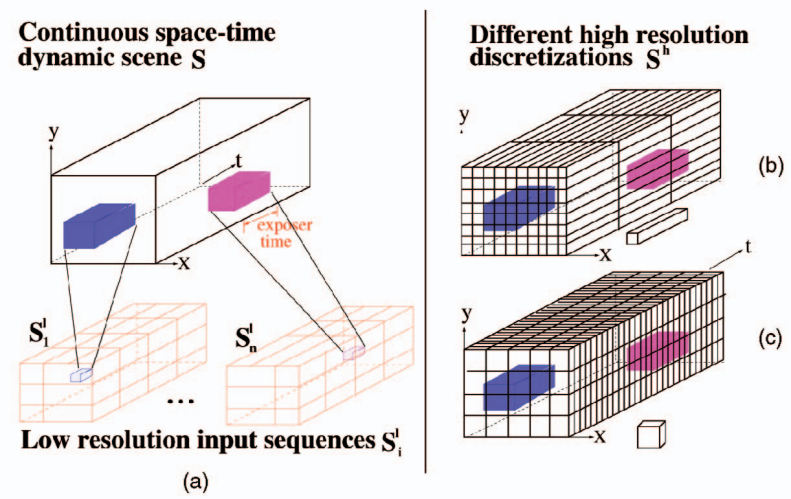

Fig. 3. The space-time imaging process. (a) illustrates the continuous space-time scene and two of the low resolution sequences. The large pink and blue boxes are the support regions of the space-time blur correspodning to the low resolution space-time measurements marked by the respective small boxes. (b), (c) show two different possible discretizations of the continuous space-time volume S resulting in two different possible types of resolution output sequences S^h. (b) has a low frame-rate and high spatial resolution, whereas (c) has a high frame-rate but low spatial resolution.

The rest of this paper is organized as follows: Section 2 describes our space-time SR algorithm. Section 3 shows some examples of handling motion aliasing and motion blur in dynamic scenes. Section 4 analyzes how temporal SR can resolve motion blur and derives a lower bound on the minimal number of input cameras required for obtaining an effective motion deblurring. Section 5 explores the potential of combining input sequences of different space-time resolutions (e.g., video and still). Finally, in Section 6, we analyze the commonalities and the differences between spatial SR and temporal SR.

A shorter version of this paper appeared in [22].

2 Space-Time Super-Resolution

Let S be a dynamic space-time scene. Let $\{S_i^l\}_{i=1}^n$ be n video sequences of that dynamic scene recorded by n different video cameras. The recorded sequences have limited spatial and temporal resolution (the subscript "l" stands for "low" space-time resolution). Their limited resolutions are due to the space-time imaging process, which can be thought of as a process of blurring followed by sampling both in time and in space.

We denote each pixel in each frame of the low resolution sequences by a "space-time point" (marked by the small boxes in Fig. 3a). The blurring effect results from the fact that the value at each space-time point is an integral (a weighted average) of the values in a space-time *region* in the dynamic scene S (marked by the large pink and blue boxes in Fig. 3a). The temporal extent of this region is determined by the exposure-time of the video camera (i.e., how long the shutter is open) and the spatial extent of this region is determined by the spatial point-spread-function (PSF) of the camera (determined by the properties of the lens and the detectors [5]).

The sampling process also has a spatial and a temporal component. The spatial sampling results from the fact that the camera has a discrete and finite number of detectors (the output of each detector is a single pixel value) and the temporal sampling results from the fact that the camera has a finite frame-rate resulting in discrete frames (typically 25 $frames/sec$ in PAL cameras and 30 $frames/sec$ in NTSC cameras).

The above space-time imaging process inhibits high spatial and high-temporal frequencies of the dynamic scene, resulting in video sequences of low space-time resolutions. Our objective is to use the information from all these sequences to construct a new sequence S^h of high space-time resolution. Such a sequence will ideally have smaller blurring effects and finer sampling in space and in time and will thus capture higher space-time frequencies of the dynamic scene S. In particular, it will capture fine spatial features in the scene and rapid dynamic events which cannot be captured (and are therefore not visible) in the low-resolution sequences.

The recoverable high-resolution information in S^h is limited by its spatial and temporal sampling rate (or discretization) of the space-time volume. These rates can be different in space and in time. Thus, for example, we can recover a sequence S^h of very high spatial resolution but low temporal resolution (e.g., see Fig. 3b), a sequence of very high-temporal resolution but low spatial resolution (e.g., see Fig. 3c), or a bit of both. These trade-offs in space-time resolutions and their visual effects will be discussed in more detail later in Section 6.1.

We next model the geometrical relations (Section 2.1) and photometric relations (Section 2.2) between the unknown high-resolution sequence S^h and the input low-resolution sequences $\{S_i^l\}_{i=1}^n$.

2.1 The Space-Time Coordinate Transformations

In general, a space-time dynamic scene is captured by a 4D representation (x, y, z, t). For simplicity, in this paper, we deal with dynamic scenes which can be modeled by a 3D space-time volume (x, y, t) (see in Fig. 3a). This assumption is valid if one of the following conditions holds: 1) The scene is planar and the dynamic events occur within this plane or 2) the scene is a general dynamic 3D scene, but the distances between the recording video cameras are small relative to their distance from the scene. (When the camera

centers are very close to each other, there is no relative 3D parallax.) Under those conditions, the dynamic scene can be modeled by a 3D space-time representation. Note that the cameras need not have the same viewing angles or zooms.

Without loss of generality, let S_1^l (one of the input low-resolution sequences) be a "reference" sequence. We define the coordinate system of the continuous space-time volume S (the unknown dynamic scene we wish to reconstruct) so that its x, y, t axes are parallel to those of the reference sequence S_1^l. S^h is a discretization of S with a higher sampling rate than that of S_1^l (see Fig. 3b). Thus, we can model the transformation T_1 from the space-time coordinate system of S_1^l to the space-time coordinate system of S^h by a scaling transformation (the scaling can be different in time and in space). Let $T_{i\to 1}$ denote the space-time coordinate transformation from the ith low resolution sequence S_i^l to the reference sequence S_1^l (see below). Then, the space-time coordinate transformation of each low-resolution sequence S_i^l is related to that of the high-resolution sequence S^h by $T_i = T_1 \cdot T_{i\to 1}$.

The space-time coordinate transformations $T_{i\to 1}$ between input sequences (and, thus, also the space time transformations from the low resolution sequences to the high resolution sequence) result from the different settings of the different cameras. A *temporal misalignment* between two video sequences occurs when there is a time-shift (offset) between them (e.g., if the two video cameras were not activated simultaneously) or when they differ in their frame rates (e.g., one PAL and the other NTSC). Such temporal misalignments can be modeled by a 1D affine transformation in time and are typically at subframe time units. The *spatial misalignment* between the sequences results from the fact that the cameras have different external (e.g., rotation) and internal (e.g., zoom) calibration parameters. In our current implementation, as mentioned above, because the camera centers are assumed to be very close to each other or else the scene is planar, the spatial transformation between the two sequences can thus be modeled by an intercamera homography (even if the scene is a cluttered 3D scene). We computed these space-time coordinate transformations using the method of [9], which provides high subpixel and high subframe accuracy.

Note that, while the space-time coordinate transformations ($\{T_i\}_{i=1}^n$) *between the sequences* are very simple (a spatial homography and a temporal affine transformation), the motions occurring over time *within* each sequence (i.e., within the dynamic scene) can be very complex. Our space-time SR algorithm does *not* require knowledge of these complex intrasequence motions, only knowledge of the simple intersequence transformations $\{T_i\}_{i=1}^n$. It can thus handle very complex dynamic scenes. For more details, see [9].

2.2 The Space-Time Imaging Model

As mentioned earlier, the space-time imaging process induces spatial and temporal blurring in the low-resolution sequences. The temporal blur in the low-resolution sequence S_i^l is caused by the exposure-time (shutter-time) of the ith video camera (denoted henceforth by τ_i). The spatial blur in S_i^l is due to the spatial point-spread-function (PSF) of the ith camera, which can be approximated by a 2D spatial Gaussian with std σ_i. (A method for estimation of the PSF of a camera may be found in [14].)

Let $B_i = B_{(\sigma_i,\tau_i,p_i^l)}$ denote the combined space-time blur operator of the ith video camera corresponding to the low resolution space-time point $p_i^l = (x_i^l, y_i^l, t_i^l)$. Let $p^h = (x^h, y^h, t^h)$ be the corresponding high resolution space-time point $p^h = T_i(p_i^l)$ (p^h is not necessarily an integer grid point of S^h, but is contained in the continuous space-time volume S). Then, the relation between the *unknown* space-time values $S(p^h)$, and the *known* low resolution space-time measurements $S_i^l(p_i^l)$ can be expressed by:

$$S_i^l(p_i^l) = \big(S * B_i^h\big)(p^h) = \int_x \int_y \int_t \underset{p=(x,y,t)\in Support(B_i^h)}{} S(p)\ B_i^h(p - p^h)dp, \tag{1}$$

where $B_i^h = T_i(B_{(\sigma_i,\tau_i,p_i^l)})$ is a point-dependent space-time blur kernel represented in the high resolution coordinate system. Its support is illustrated by the large pink and blue boxes in Fig. 3a. This equation holds wherever the discrete values in the left-hand side are defined. To obtain a linear equation in terms of the *discrete unknown* values of S^h, we used a discrete approximation of (1). See [7], [8] for a discussion of the different spatial discretization techniques in the context of image-based SR. In our implementation, we used a nonisotropic approximation in the temporal dimension and an isotropic approximation in the spatial dimension (for further details, refer to [21]). Equation (1) thus provides a linear equation that relates the unknown values in the high resolution sequence S^h to the *known* low resolution measurements $S_i^l(p_i^l)$.

When video cameras of different photometric responses are used to produce the input sequences, then a preprocessing step is necessary. We used a simple histogram specification to equalize the photometric responses of all sequences. This step is required to guarantee consistency of the relation in (1) with respect to all low resolution sequences.

2.3 The Reconstruction Step

Equation (1) provides a single equation in the high resolution unknowns for each low resolution space-time measurement. This leads to the following huge system of linear equations in the unknown high resolution elements of S^h:

$$A\vec{h} = \vec{l}, \tag{2}$$

where $\vec{h}$ is a vector containing all the unknown high resolution values (grayscale or color values in YIQ) of S^h, $\vec{l}$ is a vector containing all the space-time measurements from all the low resolution sequences, and the matrix A contains the relative contributions of each high resolution space-time point to each low resolution space-time point, as defined by (1).

When the number of low resolution space-time measurements in $\vec{l}$ is greater than or equal to the number of space-time points in the high-resolution sequence S^h (i.e., in $\vec{h}$), then there are more equations than unknowns and (2) is typically solved using LSQ methods. This is obviously a necessary requirement, however, not sufficient. Other issues, such as dependencies between equations or noise

magnification, may also affect the results (see [3], [18] and also Section 6.2). The above-mentioned requirement on the number of unknowns implies that a large increase in the spatial resolution (very fine spatial sampling in S^h) will come at the expense of a significant increase in the temporal resolution (very fine temporal sampling in S^h) and vice versa. This is because, for a given set of input low-resolution sequences, the size of $\vec{l}$ is fixed, thus dictating an upper bound on the number of unknowns in S^h. However, the number of high resolution space-time points (unknowns) can be distributed differently between space and time, resulting in different space-time resolutions (this issue is discussed in more detail in Section 6.1).

2.3.1 Directional Space-Time Regularization

When there is an insufficient number of cameras relative to the required improvement in resolution (either in the entire space-time volume or only in portions of it), then the above set of equations (2) becomes ill-posed. To constrain the solution and provide additional numerical stability (as in image-based Super-Resolution [6], [11], [19]), a space-time regularization term can be added to impose smoothness on the solution S^h in space-time regions which have insufficient information. We introduce a *directional* (or steerable [16]) space-time regularization term which applies smoothness only in directions within the space-time volume where the derivatives are low and does *not* smooth across space-time "edges." In other words, we seek $\vec{h}$ which minimizes the following error term:

$$min\left(||A\vec{h}-\vec{l}||^2+||W_xL_x\vec{h}||^2+||W_yL_y\vec{h}||^2+||W_tL_t\vec{h}||^2\right), \tag{3}$$

where L_j $(j=x,y,t)$ is a matrix capturing the second-order derivative operator in direction j and W_j is a diagonal weight matrix which captures the degree of desired regularization at each space-time point in the direction j. The weights in W_j prevent smoothing across space-time "edges." These weights are determined by the location, orientation, and magnitude of space-time edges and are approximated using space-time derivatives in the low resolution sequences. Thus, in regions that have high *spatial resolution* but small motion (or no motion), the regularization will be stronger in the temporal direction (thus preserving sharp spatial features). Similarly, in regions that have *fast dynamic changes* but low spatial resolution, the regularization will be stronger in the spatial direction. This is illustrated in Fig. 4. In a smooth and static region, the regularization will be strong both in time and in space.

2.3.2 Solving the Equation

The optimization problem of (3) has a very large dimensionality. For example, even for a simple case of four low resolution input sequences, each of one-second length (25 frames) and of size 128×128 pixels, we get: $128^2 \times 25 \times 4 \approx 1.6 \times 10^6$ equations from the low resolution measurements alone (without regularization). Assuming a similar number of high resolution unknowns poses a severe computational problem. However, because matrix A is sparse and local (i.e., all the nonzero entries are located in a few diagonals), the system of equations can be solved using "box relaxation" [23], [19]. For more details, see [21].

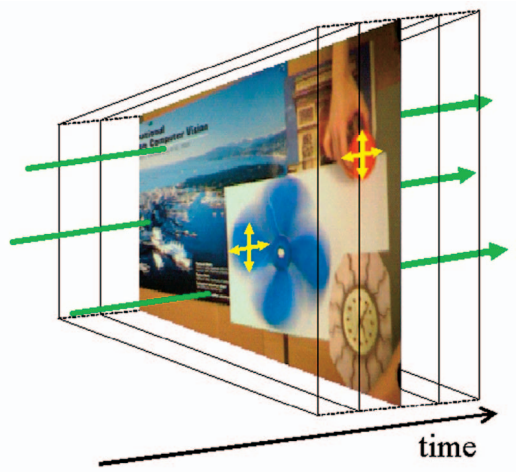

Fig. 4. Space-time regularization. The figure shows the space-time volume with one high resolution frame from the example of Fig. 6. It illustrates a couple of interesting cases where the space-time regularization is applied in a physically meaningful way. In regions that have high spatial resolution but small (or no) motion (such as in the static background), the temporal regularization is strong (green arrow). Similarly, in regions with fast dynamic changes and low spatial resolution (such as in the rotating fan), the spatial regularization is strong (yellow arrows).

Fig. 5 shows a result of applying our space-time SR algorithm to 36 low resolution sequences. The increase in dimension of the output sequence relative to the input sequences is $\times 2 \times 2 \times 8$ (an increase of $\times 2$ in x, $\times 2$ in y, and $\times 8$ in t). In other words, the output frames are twice as big in each spatial dimension and the frame-rate was increased by a factor of 8. The increase both in spatial and temporal resolution relative to the input sequences is evident. Note the increase in spatial resolution of the text in the static background, as well as the increase in resolution and reduction in the motion-blur of the moving object (the Coke can). It is important to emphasize that this is achieved *without* any motion segmentation or estimation of the object motion. The notion of spatial SR is quite familiar and intuitive. However, not so is the notion of temporal SR. We devote the next two sections to explaining the significance and properties of temporal SR in video.

3 Examples of Temporal SR

Before proceeding with more in-depth analysis and details, we first show a few examples of applying the above algorithm for recovering higher *temporal* resolution of fast dynamic events. In particular, we demonstrate how this approach provides a solution to the two previously mentioned problems encountered when fast dynamic events are recorded by slow video cameras: 1) motion aliasing and 2) motion blur.

3.1 Example 1: Handling Motion Aliasing

We used four independent PAL video cameras to record a scene of a fan rotating clockwise very fast. The fan rotated faster and faster until, at some stage, it exceeded the maximal velocity that can be captured correctly by the video frame-rate. As expected, at that moment, all four input sequences display

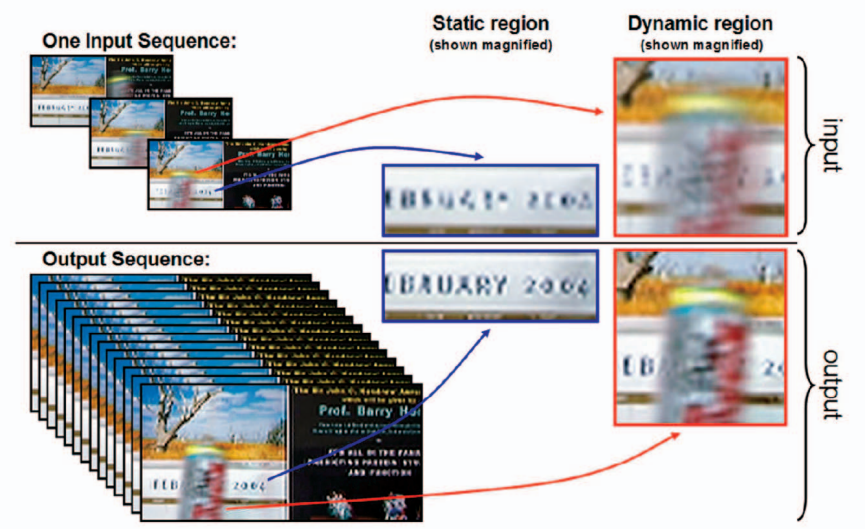

Fig. 5. Space-time super-resolution. This figure shows a result of simultaneous resolution enhancement in time and in space using the algorithm described in Section 2. The top part shows one out of the 36 simulated low resolution sequences used as input to the algorithm. The lower part shows the output sequence whose frames are 2×2 larger and whose frame-rate is $\times 8$ larger. The right-hand side shows the resolution enhancement obtained in space (text) and in time (moving coke can). These results are contrasted with the corresponding places in the low resolution input sequence (regions are magnified). It should be appreciated that no object motion estimation or segmentation was involved in generating these results.

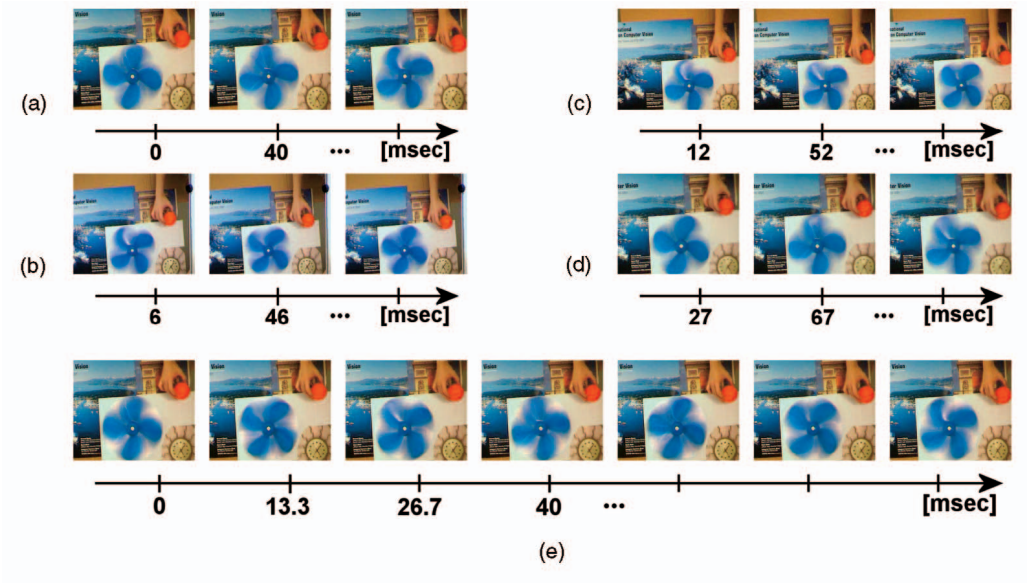

Fig. 6. Example 1: Handling motion aliasing—The "wagon wheel effect." (a)-(d) display three successive frames from four PAL video recordings of a fan rotating clockwise. Because the fan is rotating very fast (almost 90° between successive frames), the motion aliasing generates a false perception of the fan rotating slowly in the opposite direction (counterclockwise) in all four input sequences. The temporal misalignments between the input sequences were computed at subframe temporal accuracy and are indicated by their time bars. The spatial misalignments between the sequences (e.g., due to differences in zoom and orientation) were modeled by a homography and computed at subpixel accuracy. (e) shows the reconstructed video sequence in which the temporal resolution was increased by a factor of 3. The new frame rate $(75\frac{frames}{sec})$ is also indicated by time bars. The correct clockwise motion of the fan is recovered. For video sequences, see: www.wisdom.weizmann.ac.il/~vision/SuperRes.html.

the classical "wagon wheel effect" where the fan appears to be falsely rotating backwards (counterclockwise). We computed the spatial and temporal misalignments between the sequences at subpixel and subframe accuracy using [9] (the recovered temporal misalignments are displayed in Figs. 6a, 6b, 6c, and 6d using a time-bar). We used the SR method of Section 2 to increase the temporal resolution by a factor of 3 while maintaining the same spatial resolution. The resulting high-resolution sequence displays the true forward (clockwise) motion of the fan as if recorded by a high-speed camera (in this case, 75 frames/sec). Examples of a few successive frames from each low resolution input sequence are shown in Figs. 6a, 6b, 6c, and 6d for the portion where the fan falsely appears to be rotating counterclockwise. A few successive frames from the reconstructed high temporal-resolution sequence corresponding to the same time are shown in Fig. 6e, showing the correctly recovered (clockwise) motion. It is difficult to perceive these strong dynamic effects via a static figure. We therefore urge the reader to view the video clips

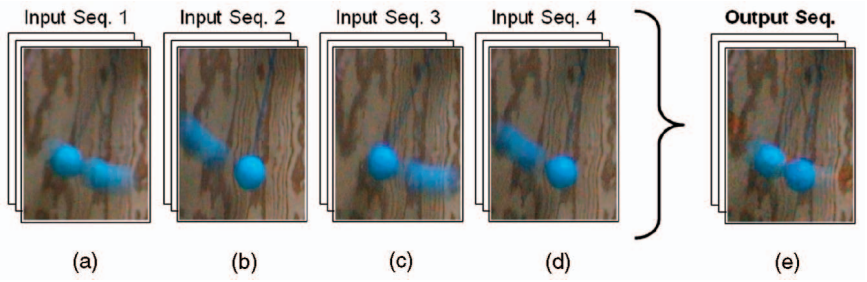

Fig. 7. Example 2: Handling motion blur via temporal SR. A "tic-tac" toy (two balls hanging on strings and bouncing against each other) was shot by four video cameras. (a)-(d) display the four frames, one from each of the input sequences, which were closest to the time of collision. In each one of these frames, at least one of the balls is blurred. The four input sequences were plugged into the temporal SR algorithm and the frame-rate was increased by a factor of 4. (e) shows the frame from the output, closest to the time of collision. Motion-blur is evidently reduced.

www.wisdom.weizmann.ac.il/~vision/SuperRes.html, where these effects are very vivid.

Note that playing the input sequences in "slow-motion" (using any type of temporal interpolation) will *not* reduce the perceived false motion effects as the information is already lost in any individual video sequence (as illustrated in Fig. 2). It is only when the information is combined from all the input sequences that the true motion can be recovered.

3.2 Example 2: Handling Motion Blur

In the following example, we captured a scene of fast moving balls using four PAL video cameras of 25 frames/sec and exposure-time of 40 msec. Figs. 7a, 7b, 7c, and 7d show four frames, one from each low-resolution input sequence, that were the *closest* to the time of collision of the two balls. In each of these frames, at least one of the balls is blurred. We applied the SR algorithm and increased the frame-rate by a factor of 4. Fig. 7e shows an output frame at the time of collision. Motion-blur is reduced significantly. Such a frame did not exist in any of the input video sequences. Note that this effect was obtained by increasing the *temporal* resolution (not the spatial) and, hence, did *not* require estimation of the motions of the balls. This phenomena is explained in more detail in Section 4.

To examine the performance of the algorithm under *severe* effects of motion-blur of the kind shown in Fig. 1, one needs many (usually more than 10) video cameras. A quantitative analysis of the amount of input data needed appears in Section 4. Since we do not have so many video cameras, we resorted to simulations, as described in the next example.

3.3 Example 3: Handling Severe Motion Aliasing & Motion Blur

In the following example, we simulated a sports-like scene with an extremely fast moving object (of the type shown in Fig. 1) recorded by many video cameras (in our example, 18 cameras). We examined the performance of temporal SR in the presence of both strong motion aliasing and strong motion blur.

To simulate such a scenario, we recorded a single video sequence of a slow moving object (a basketball bouncing on the ground). We temporally blurred the sequence using a large (9-frame) blur kernel (to simulate a large exposure time), followed by a large subsampling in time by a factor of $1:30$ (to simulate a low frame-rate camera). Such a process results in 18 low temporal-resolution sequences of a very fast dynamic event having an "exposure-time" of about $\frac{1}{3}$ of its frame-time and temporally subsampled with *arbitrary* starting frames. Each generated "low-resolution" sequence contains seven frames. Three of the 18 sequences are presented in Figs. 8a, 8b, and 8c. To visually display the dynamic event, we superimposed all seven frames in each sequence. Each ball in the superimposed image represents the location of the ball at a different frame. None of the 18 low-resolution sequences captures the correct trajectory of the ball. Due to the severe motion aliasing, the perceived ball trajectory is roughly a smooth curve, while the true trajectory was more like a cycloid (the ball jumped five times on the floor). Furthermore, the shape of the ball is completely distorted in all of the input image frames due to the strong motion blur.

We applied the SR algorithm of Section 2 on these 18 low-resolution input sequences and constructed a high-resolution sequence whose frame-rate is 15 times higher than that of the input sequences. (In this case, we requested an increase only in the temporal sampling rate.) The reconstructed high-resolution sequence is shown in Fig. 8d. This is a superimposed display of some of the reconstructed frames (every eighth frame). The true trajectory of the bouncing ball has been recovered. Furthermore, Figs. 8e and 8f show that this process has significantly reduced the effects of motion blur and the true shape of the moving ball has been automatically recovered, although no single low resolution frame contains the true shape of the ball. Note that no estimation of the ball motion was needed to obtain these results.

The above results obtained by temporal SR cannot be obtained by playing any low-resolution sequence in "slow-motion" due to the strong motion aliasing. While interleaving and interpolating between frames from the 18 input sequences may resolve some of the motion aliasing, it will not handle the severe motion-blur observed in the individual image frames. Note, however, that, even though the frame rate was increased by a factor of 15, the effective reduction in motion blur in Fig. 8 is only by a factor of ~ 5. These issues are explained in the next section.

A method for treating motion blur in the context of *image-based* SR was proposed by [2], [1]. However, these methods require a prior segmentation of the moving objects and the estimation of their motions. These methods will have difficulties handling complex motions or motion aliasing. The distorted shape of the object due to strong blur will pose severe problems in motion estimation. Furthermore, in the presence of motion aliasing, the

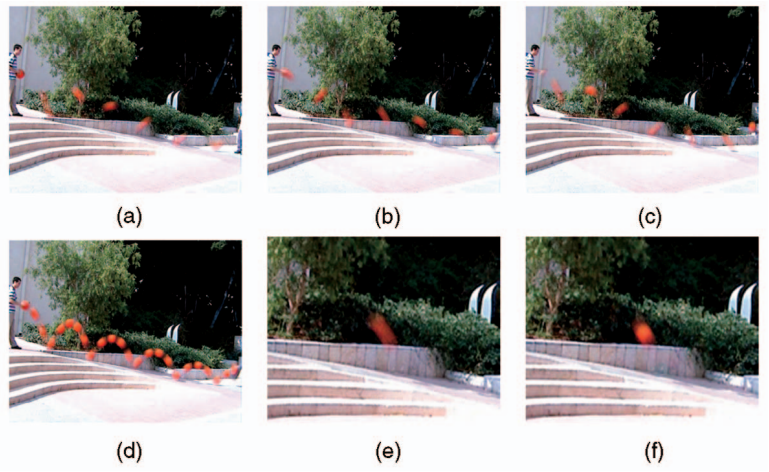

Fig. 8. Example 3: Handling motion blur & motion aliasing. We simulated 18 low-resolution video recordings of a rapidly bouncing ball inducing strong motion blur and motion aliasing (see text). (a)-(c) display the dynamic event captured by three representative low-resolution sequences. These displays were produced by superposition of all seven frames in each low-resolution sequence. All 18 input sequences contain severe motion aliasing (evident from the falsely perceived curved trajectory of the ball) and strong motion blur (evident from the distorted shapes of the ball). (d) The reconstructed dynamic event as captured by the recovered high-resolution sequence. The true trajectory of the ball is recovered, as well as its correct shape. (e) A close-up image of the distorted ball in one of the low resolution sequences. (f) A close-up image of the ball at the exact corresponding frame in time in the high-resolution output sequence. For video sequences see: www.wisdom.weizmann.ac.il/~vision/SuperRes.html.

direction of the estimated motion will not align with the direction of the induced blur. For example, the motion blur in Figs. 8a, 8b, and 8c is along the true trajectory and not along the perceived one. In contrast, our approach does not require separation of static and dynamic scene components, nor their motion estimation, and therefore can handle very complex scene dynamics. However, we require multiple cameras. These issues are explained and analyzed next.

4 Resolving Motion Blur

A crucial observation for understanding why temporal SR reduces motion blur is that motion blur is caused by *temporal* blurring and *not* by spatial blurring. The blurred colors induced by a moving object (e.g., Fig. 9) result from blending color values along time and not from blending with spatially neighboring pixels. This observation and its implications are the focus of Section 4.1. Section 4.2 derives a bound on the best expected quality (temporal resolution) of a high resolution output sequence which yields a practical formula for the recommended number of input cameras.

4.1 Why Is Temporal Treatment Enough?

The observation that motion blur is a purely *temporal* artifact is nonintuitive. After all, the blur is visible in a single image frame. This is misleading, however, as even a single image frame has a *non*infinitesimal exposure time. The first attempts to reduce motion blur, which date back to the beginning of photography, tried to reduce the exposure time by increasing the amount of light.

Figs. 9a and 9b display the same event (a ball falling) captured by two cameras with different exposure times. Since the exposure time of the camera in Fig. 9a was longer than that of Fig. 9b, its shape is more elongated. The amount of induced motion blur is linearly proportional to the temporal length of the exposure time. The longer the exposure-time is, the larger the induced spatial effect of motion blur is.

Another source of confusion is the indirect link to motion. The blur results from integration over time (due to the exposure time). In general, it captures any temporal changes of intensities. In practice, most of the temporal changes result from moving objects (which is why this temporal blur is denoted as "motion blur"). However, there exist temporal changes that do not result from motion, e.g., in a video recording of a flashing light spot. With sufficiently long exposure time, the spot of light will be observed as a constant dim light in all frames. This light dimming effect and motion blur are both caused directly by temporal blur (integration over exposure time), but with different indirect causes of temporal change. Our algorithm addresses the direct cause (i.e., temporal blur) and thus does not need to analyze which of the indirect causes where involved (i.e., motion, light-changes, etc.).

To summarize, we argue that all pixels (static and dynamic) experience the same temporal blur—a convolution with a temporal rectangular function. However, the temporal

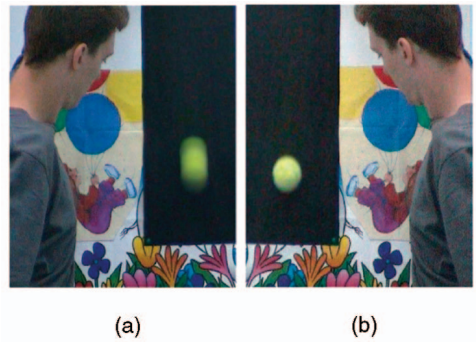

Fig. 9: What is motion blur? A spatial smear induced by temporal integration. A free-falling tennis ball was shot by two cameras through a beam-splitter (hence, the flip). The camera on the left (a) had a long exposure time and the camera on the right (b) had a short one. The longer the exposure time is, the larger the spatial smear of moving objects is.

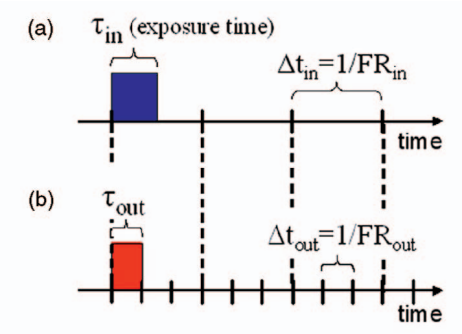

Fig. 10. Frame-time and exposure time. This figure graphically illustrates the quantities used in Section 4.2. The time-bars in (a) and (b) denote the frame-rates of the input and output sequences, respectively (the same time scale is used). The quantity Δt denotes the elapsed time between successive frames (frame-time), where FR is the video frame-rate. The quantity τ denotes the exposure times.

blur is visible only in image locations where there are temporal changes (e.g., moving objects). Our algorithm addresses this temporal blur directly by reducing the exposure time and, thus, does not require motion analysis, segmentation, or any scene interpretation.

4.2 What Is the Minimal Number of Required Video Cameras?

Denote by Δt_{in} and Δt_{out} the elapsed time between successive frames in the input and output sequences, respectively, i.e., $\Delta t = \frac{1}{FR}$, where FR is the frame-rate. Δt_{in} is a physical property of the input cameras. Δt_{out} is dictated by the output frame rate (specified by the user). Similarly, denote by τ_{in} and τ_{out} the exposure time of the input and output sequences. All these quantities are illustrated in Fig. 10. τ_{in} is a physical quantity—the exposure time of the input sequences. On the other hand, τ_{out} is *not* a physical quantity. It is a measure of the quality of the output sequence. Its units are the exposure time of a real camera that will generate an equivalent output (an output with the same motion-blur). Thus, τ_{out} is denoted here as the "induced exposure time," and quantifying it is the objective of this section.

We have shown analytically in [21] that, under ideal conditions (i.e., uniform sampling in time and no noise), τ_{out} is bounded by:

$$\tau_{out} \geq \Delta t_{out}. \tag{4}$$

Equation (4) should be read as follows: The residual motion blur in the output sequence is at least the same as the motion blur caused by a true camera with exposure time Δt_{out}.

Furthermore, the experiment in Fig. 11 shows that, in ideal conditions (i.e., optimal sample distribution and ignoring noise amplification), this bound may be reached. Namely, $\tau_{out} \approx \Delta t_{out}$. Rows (b) and (c) compare the SR output to the "ground truth" temporal blur for various specified Δt_{out}. These "ground truth" frames were synthesized by temporal blurring the original sequence (in the same way in which the low resolution sequence was generated) such that their exposure time is Δt_{out}. One can see that the induced motion blur in the reconstructed sequences is similar to the motion blur caused by the imaging process with the same exposure time (i.e., $\tau_{out} \approx \Delta t_{out}$).

Given the above observation, we obtain a practical constraint on the required number of input video sequences (cameras) N_{cam}.

Naturally, the smaller the exposure time, the smaller the motion blur. In our case, the output induced exposure time (τ_{out}) is dictated by the user-selected output frame-rate ($\tau_{out} \approx \Delta t_{out} = \frac{1}{FR_{out}}$). However, there is a limit on how much the output frame-rate (FR_{out}) can be increased (or, equivalently, how much Δt_{out} can be decreased). This bound is determined by the number of low resolution input sequences and their frame rate:

$$FR_{out} \leq N_{cam} \cdot FR_{in}, \tag{5}$$

or, equivalently, $\Delta t_{out} \geq \Delta t_{in}/N_{cam}$. If the frame-rate is further increased, we will get more equations than unknowns. Fig. 12 displays several results of applying our algorithm, but with different specified FR_{out}. The minimal number of required cameras for each such reconstruction quality is indicated below each example. These numbers are dictated by the required increase in frame-rate (5).

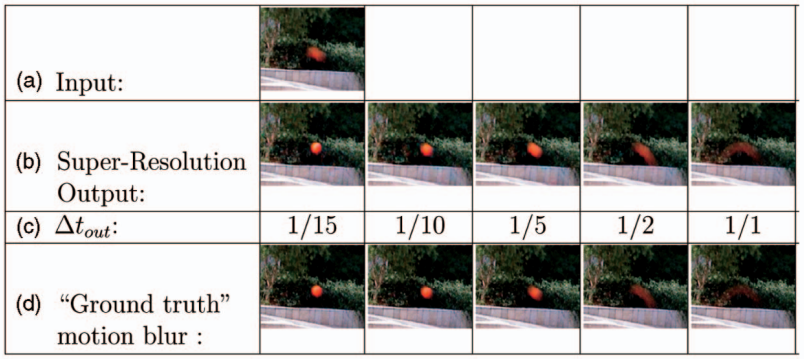

Fig. 11. Residual motion blur as a function of Δt_{out}. Using the same number of input cameras (18), we have reconstructed several output sequences with increasing Δt_{out}. (a) shows a frame from one of the input sequences in Fig. 8. (b) displays frames from the reconstructed outputs. A user-defined Δt_{out} in each case is indicated below in row (c). As can be seen, the amount of motion-blur increases as Δt_{out} increases. Row (d) displays frames from "ground truth" blur. These are frames from synthesized blurred sequences, each with exposure time that is equal to the corresponding Δt_{out} (i.e., (c) above). The similarity between the observed motion blur in rows (b) and (d) argues that, in this example, $\tau_{out} \approx \Delta t_{out}$.

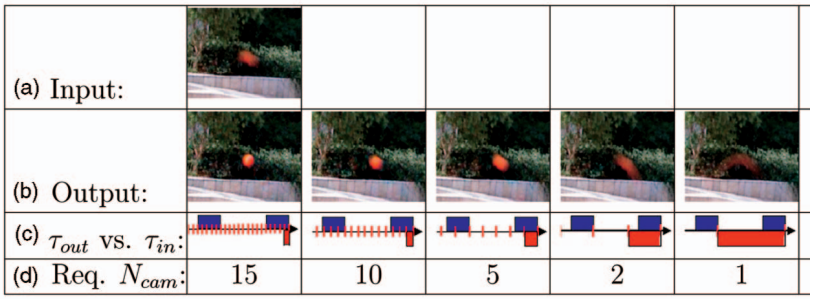

Fig. 12. The required number of input cameras. This figure shows the relation between the number of cameras in use and the reduction in motion blur. We have reconstructed several output sequences with decreasing FR_{out}. (a) shows a frame from one of the input sequences in Fig. 8. (b) displays frames from reconstructed outputs. (c) The blue and red rectangles illustrate the input and output exposure-time, respectively (the input is identical in all cases and the output is induced by the output frame-rate FR_{out}—the red bars). Row (d) indicates the minimal number of cameras required for obtaining the corresponding FR_{out} (5). In order to reduce motion blur with respect to the input frame (a), τ_{out} (the red rectangles) should be smaller than τ_{in} (the blue rectangles). Therefore, in the left three sequences, the motion blur is decreased, while in the right two sequences, although we increase the frame-rate, the motion blur is increased.

It is interesting to note that, in some cases, the "SR" *increases* the motion blur. Such an undesired outcome occurs when the input exposure time τ_{in} is smaller than the induced output exposure time τ_{out}. Thus, requiring that the output motion blur be better than the input motion blur ($\tau_{out} < \tau_{in}$) yields the following constraint on the minimal number of input cameras:

$$\tau_{in} \geq \tau_{out} \approx \Delta t_{out} = \frac{\Delta t_{in}}{N_{cam}}. \tag{6}$$

Substituting input frame-rate and reordering terms provides:

$$N_{cam} \geq \frac{1}{\tau_{in} FR_{in}}. \tag{7}$$

A numerical example of this constraint is illustrated in Fig. 12. Rows (a) and (b) display input and output frames as in Fig. 11. Row (c) illustrates graphically the ratio between τ_{out} (the red rectangle) and τ_{in} (the blue rectangle). It is evident from the figure that, if $\tau_{in} > \tau_{out}$ (the three left images), the output quality outperforms the input quality (a). Similarly, when $\tau_{in} < \tau_{out}$ (the two most right images), then the output motion blur is worse than the input motion blur. In this example, the input frame rate is $FR_{in} = 1\frac{frames}{sec}$ and $\tau_{in} = \frac{1}{3} sec$ (see Section 3 for details), thus the minimal number of required cameras to outperform the input motion blur is at least three, exactly as observed in the example.

Finally, the analysis in this section assumed ideal conditions. It did not take into account the following factors: 1) There may be errors due to inaccurate sequence alignment in space or in time and 2) nonuniform sampling of sequences in time may increase the numerical instability. This analysis therefore provides only a lower bound on the number of required cameras. In practice, the actual number of required cameras is likely to be slightly larger. For example, in our experiments, we sometimes used more cameras than the computed minimum, but never more than twice this lower bound.

5 Combining Different Space-Time Inputs

So far, we assumed that all input sequences were of similar spatial and temporal resolutions. However, the space-time SR algorithm of Section 2 is not restricted to this case and can also handle input sequences of different space-time resolutions. Such a case is meaningless in *image-based* SR (i.e., combining information from *images* of varying spatial resolution) because a high-resolution input image would always contain the information of a low-resolution image. In space-time SR, however, this is not the case. One camera may have high spatial resolution but low temporal resolution, and the other vice versa. Thus, for example, it is meaningful to combine information from NTSC and PAL video cameras. NTSC has higher temporal resolution than PAL (30 frames/sec versus 25 frames/sec), but lower spatial resolution (640×480 pixels versus 768×576 pixels). An extreme case of this idea is to combine information from *still* and *video* cameras. Such an example is shown in Fig. 13. Two high quality still images (Fig. 13a) of high spatial resolutions ($1{,}120 \times 840$ pixels) but extremely low "temporal resolution" (the time gap between the two still images was 1.4 sec.) were combined with an interlaced (PAL) video sequence using the algorithm of Section 2. The video sequence (Fig. 13b) has three times lower spatial resolution (we used fields of size 384×288 pixels), but a high temporal resolution (50 fields/sec). The goal is to construct a new sequence of high spatial and high temporal resolutions (i.e., $1{,}120 \times 840$ pixels at 50 fields/sec). The output sequence shown in Fig. 13c contains the high spatial resolution from the still images (the sharp text) and the high temporal resolution from the video sequence (the rotation of the toy dog and the brightening and dimming of illumination).

In the example of Fig. 13, the number of unknowns was significantly larger than the number of low resolution measurements (the input video and the two still images). Although, theoretically, this is an ill-posed set of equations, the reconstructed output is of high quality. This is achieved by applying physically meaningful space-time directional regularization (Section 2.3), that exploits the high redundancy in the video sequence.

In the example of Fig. 13, we used only one input video sequence and two still images, thus we did not attempt to exceed the temporal resolution of the video or the spatial resolution of the stills. However, when multiple video sequences and multiple still images are used (so that the number of input measurements exceeds the number of output high resolution unknowns), then an output sequence can be recovered that exceeds the spatial resolution of the still images and temporal resolution of the video sequences.

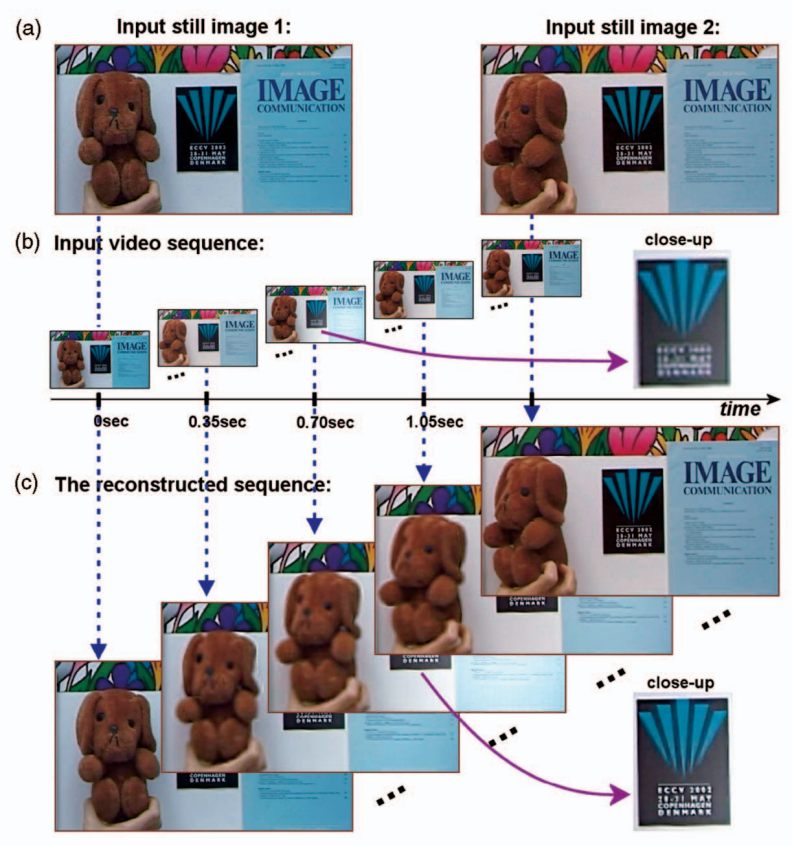

Fig. 13. Combining Still and Video. A dynamic scene of a rotating toy-dog and varying illumination was captured by: (a) a still camera with spatial resolution of $1,120 \times 840$ pixels and (b) a video camera with 384×288 pixels at 50 f/sec. The video sequence was 1:4sec. long (70 frames) and the still images were taken 1:4sec. apart (together with the first and last frames). The algorithm of Section 2 is used to generate the high resolution sequence (c). The output sequence has the spatial dimensions of the still images and the frame-rate of the video ($1120 \times 840 \times 50$). It captures the temporal changes correctly (the rotating toy and the varying illumination), as well as the high spatial resolution of the still images (the sharp text). Due to lack of space, we show only a portion of the images, but the proportions between video and still are maintained. For video sequences, see: www.wisdom.weizmann.ac.il/~vision/SuperRes.html.

6 Temporal versus Spatial Super-Resolution: Differences and Similarities

Unlike in image-based SR, where both x and y dimensions are of the same type (spatial), different types of dimensions are involved in space-time SR. The spatial and temporal dimensions are very different in nature, yet are interrelated. This leads to different phenomena in space and in time, but also to interesting tradeoffs between the two dimensions. In Section 5, we saw one of the differences between spatial SR and space-time SR. In this section, we will discuss more differences, as well as similarities between space and time that lead to new kinds of phenomena, problems and challenges.

6.1 Producing Different Space-Time Outputs

The mix of dimensions introduces visual tradeoffs between space and time, which are unique to spatio-temporal SR, and are not applicable to the traditional spatial (image-based) SR. In spatial SR, the increase in the sampling rate is equal in all spatial dimensions. This is necessary to maintain the aspect ratio of image pixels. However, this is not the case in space-time SR. The increase in sampling rate in the spatial and temporal dimensions need not be the same. Moreover, increasing the sampling rate in the spatial dimension comes at the expense of increase in the temporal frame rate and the *temporal resolution*, and vice-versa. This is because the number of equations provided by the low resolution measurements is fixed, dictating the maximal number of possible unknowns (the practical upper limit on the number of unknowns is discussed later, in Section 6.2). However, the arrangement of the unknown high-resolution space-time points in the high-resolution space-time volume depends on the manner in which this volume is discretized.

For example, assume that eight video cameras are used to record a dynamic scene. One can increase the temporal frame-rate alone by a factor of 8 or increase the spatial sampling rate alone by a factor of $\sqrt{8}$ in x and in y (i.e., increase the number of pixels by a factor of 8) or do a bit of both: Increase the sampling rate by a factor of 2 in all three dimensions x, y, t. These options are graphically illustrated in Fig. 14. For more details, see [22].

6.2 Upper Bounds on Temporal versus Spatial Super-Resolution

The limitations of spatial SR have been discussed in [3], [18]. Both showed that the noise that is amplified by the SR algorithm grows quadratically with the magnification factor. Thus, large magnification factors in image-based SR are not practical. Practical assumptions about the initial noise in real images [18] lead to a realistic magnification factor of 1.6 (and a theoretical factor of 5.7 is claimed for synthetic images with quantization noise). Indeed, many

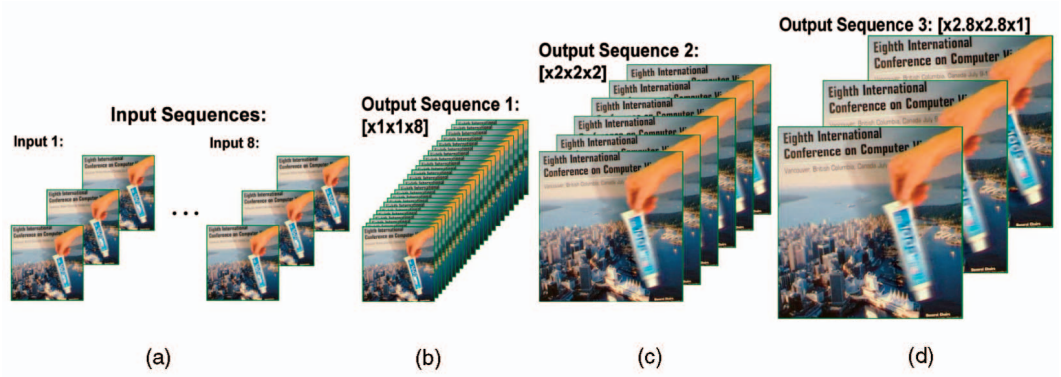

Fig. 14. Trade-offs between spatial and temporal resolution. (a) Two out of eight input sequences. (b)-(d) graphically display output discretization options. (b) One option—apply SR to increase the density by a factor of 8 in time only. The spatial resolution remains the same. (c) Another option—apply SR to increase the density by a factor of 2 in all three dimensions x, y, t. (d) A third option—increase the spatial resolution alone by a factor of $\sqrt{8}(\approx 2:8)$ in each of the spatial dimensions (x and y) maintaining the same frame-rate. Note that (b)-(d) are not the actual output of the SR algorithm. The results appear in [22].

image-based SR algorithms (e.g., [1], [2], [3], [6], [7], [11], [13], [14], [15], [16]) illustrate results with limited magnification factors (usually up to 2). In this section, we will show and explain why we get significantly larger magnification factors (and resolution enhancement) for *temporal* SR.

The analysis described in [3], [18] applies to temporal SR as well. The differences result from the different types of blurring functions used in temporal and in spatial domains: 1) The temporal blur function induced by the exposure time has approximately a rectangular shape, while the spatial blur function has a Gaussian-like shape. 2) The supports of spatial blur functions typically have a diameter larger than one pixel, whereas the exposure time is usually smaller than a single frame-time (i.e., $\tau < \Delta t$). These two differences are depicted in Fig. 15. 3) Finally, the spatial blurring acts along two dimensions (x and y), while temporal blurring is limited to a single dimension (t). These differences in shape, support, and dimensionality of the blur kernels are the cause of having a significantly larger upper bound in temporal SR, as explained below.

When the blur function is an "ideal" low-pass filter, no SR can be obtained since *all* high frequencies are eliminated in the blurring process. On the other hand, when high frequencies are not completely eliminated and are found in aliased form in the low resolution data, SR can be applied (those are the frequencies that are recovered in the SR process). The spatial blur function (the point spread function) has a Gaussian shape and its support extends over *several* pixels (samples). As such, it is a much "stronger" low-pass filter. In contrast, the temporal blur function (the exposure time) has a rectangular shape and its extent is *sub*frame (i.e., less than one sample), thus preserving more high temporal frequencies. Figs. 15c and 15d illustrate this difference. In addition to the above, it was noted in [3] that the noise in image-based SR (2D signals) tends to grow quadratically with the increase of the spatial magnification. Using similar arguments and following the same derivations, we deduce that the noise in one-dimensional *temporal* SR grows only *linearly* with the increase in the temporal magnification. Hence, larger effective magnification factors are expected. Note that, in the case of SR in space and in time simultaneously, the noise amplification grows *cubically* with the magnification factor.

The next experiment illustrates that: 1) Large magnification factors in time are feasible, (i.e., recovery of high temporal frequencies, and not just a magnification of the frame-rate). 2) The noise growths linearly with temporal magnification factor.

To show this, we took four sets of 30 input sequences with different exposure times. Each set was synthetically generated by temporal blurring followed by temporal subsampling, similar to the way described in Section 3 (Example 3). Small Gaussian noise was added to the input sequences in a way that, in all of them, the temporal noise would be the same ($\sigma_{in} \approx 2.3$ gray-levels, RMS). Figs. 16a, 16b, 16c, and 16d show matching frames from each set of the simulated sequences with increasing exposure times.

We increased the frame-rate by factor 15 in each of the sets using the temporal SR algorithm. *No regularization* was applied to show the "pure" output noise of the temporal SR algorithm without any smoothing. Figs. 16e, 16f, 16g, and 16h are the corresponding frames in the reconstructed sequences. It is vivid that: 1) The size of the ball is similar in all cases, thus, the residual motion-blur in the output sequences is similar regardless of the SR magnification (the reconstructed shape of the ball is correct). Note that the SR magnification factors $M = \frac{\tau_{in}}{\tau_{out}}$ are defined as the increase of temporal resolution (the reduction in the exposure-time)

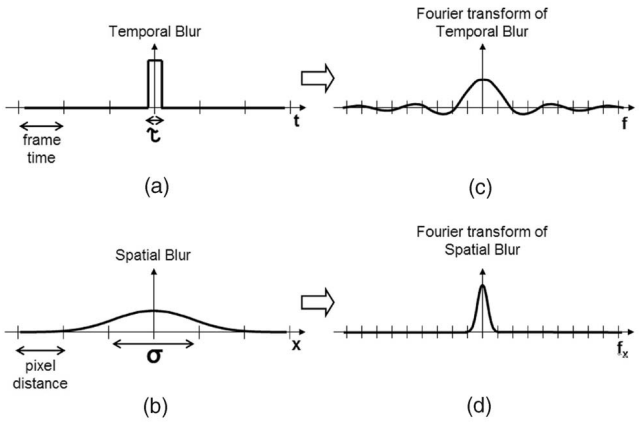

Fig. 15. Temporal versus spatial blur kernels.

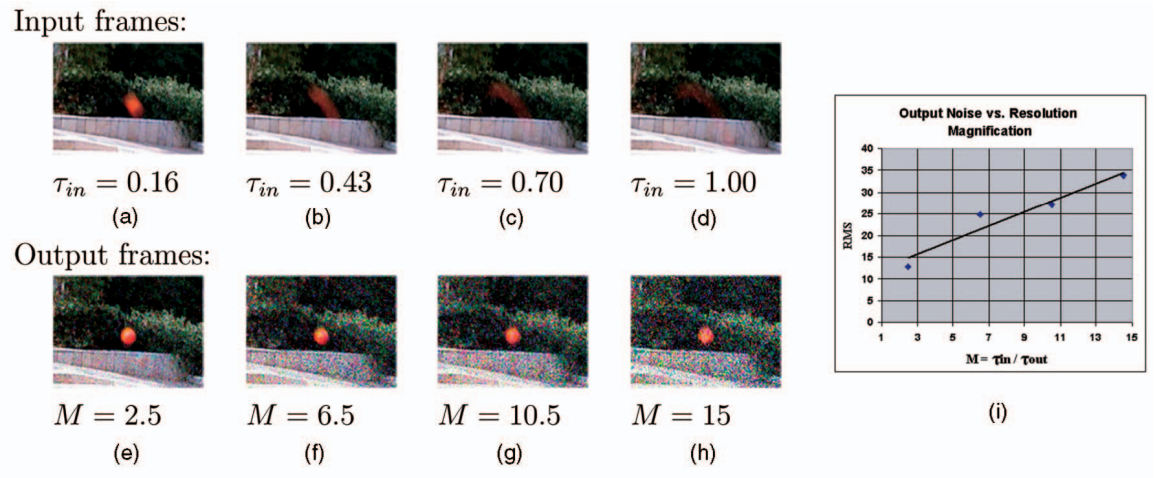

Fig. 16. Temporal SR with large magnification factors. In the following example, we simulated four sets of 30 sequences with different exposure times for each set. (a)-(d) display the corresponding frame from each set of the simulated low resolution sequences. The corresponding exposure-times are indicated below each frame (where "1" = Δt_{in}). (e)-(h) display the corresponding frames in the reconstructed high resolution sequence with frame-rate increased by a factor of 15. The resulting temporal SR magnification factors, $M = \frac{\tau_{in}}{\tau_{out}}$, are indicated below. Note that we denote the magnification as the increase of temporal resolution (captured by the change in exposure time τ) and not as the increase of the frame-rate (which is captured by Δt and is $\frac{\Delta t_{in}}{\Delta t_{out}} = 15$ in all cases). The graph in (i) shows the RMS of the output temporal noise τ_{out} as a function of M, where the noise level of all input sequences was $\sigma_{in} = 2.3$.

and not as the increase of the frame-rate[1] $\Delta t_{in}/\Delta t_{out}$. 2) The measured noise was amplified linearly with the SR magnification factor (Fig. 16i).

To conclude, it is evident that typical magnification factors of temporal SR are likely to be much larger than in spatial super-resolution. Note, however, that spatial and temporal SR are inherently the same process. The differences reported above result from the different spatial and temporal properties of the sensor. In special devices (e.g., a nondiffraction limited imaging system [17]), where the spatial blur of the camera lens is smaller than the size of a single detector pixel, spatial SR can reach similar bounds as temporal SR of regular video cameras.

6.3 Temporal versus Spatial "Ringing"

So far, we have shown that the rectangular shape of the blur kernel has an advantage over a Gaussian shape in the ability to increase resolution. On the other hand, the rectangular shape of the temporal blur is more likely to introduce a temporal artifact which is similar to the spatial "ringing" ([11], [7], [3]). This effect appears in temporal superresolved video sequences as a trail that is moving before and after the fast moving object. We refer to this temporal effect as "ghosting." Figs. 17a, 17b, and 17c show an example of the "ghosting" effect resulting in the basketball example when temporal SR is applied *without* any space-time regularization. (The effect in Fig. 17d is *magnified by a factor of 5*, to make it more visible.)

The explanation of the "ghosting" effect is simple if we look at the frequencies of the temporal signals. The SR algorithm (spatial or temporal) can reconstruct the true temporal signal at all frequencies except for specific frequencies that have been set to zero by the temporal rectangular blur. The system of equations (2) does not provide any constraints on those frequencies. If such frequencies are somehow "born" in the iterative process due to noise, they will stay in the solution and will not be suppressed. These "unsuppressed" frequencies are connected directly to the shape of the rectangular blur kernel through its exposure-time. If the exposure-time width is an integer multiple of the wavelength of a periodic signal (thus, the integral over the periodic signal is 0), then such a signal cannot be handled by the SR algorithm. This is illustrated in Fig. 17e where the "unsuppressed" frequencies are shown as temporal sinusoidal signals in one of the pixels of the "ghosting" trail.[2] These frequencies can be predicted (see [21]) from the number of input cameras, their frame-rate, and the frame-rate of the output sequence.

The "ghosting" effect is significantly reduced by the space-time regularization. It smoothes those trails in regions where no spatial or temporal edges are expected. This is why the ghosting effect is barely visible in our example output videos.

7 Conclusions

We have shown that space-time SR can improve the resolution of both static and dynamic scene components without needing to segment them or to perform any motion estimation within the sequence. We further showed that, because motion blur is inherently a temporal phenomenon, temporal SR (and not spatial SR) is the correct way to address it. Space-time SR can be used for generating a high-speed video camera from multiple slow video cameras, as well as for combining information from still and video cameras for producing high-quality video. We have further provided analysis of the limitations and bounds of this method.

1. In spatial SR, there is no difference between the two since the diameter of a typical PSF is close to a pixel size.

2. Those frequencies are also upperbounded by the frame-rate of the output sequence.

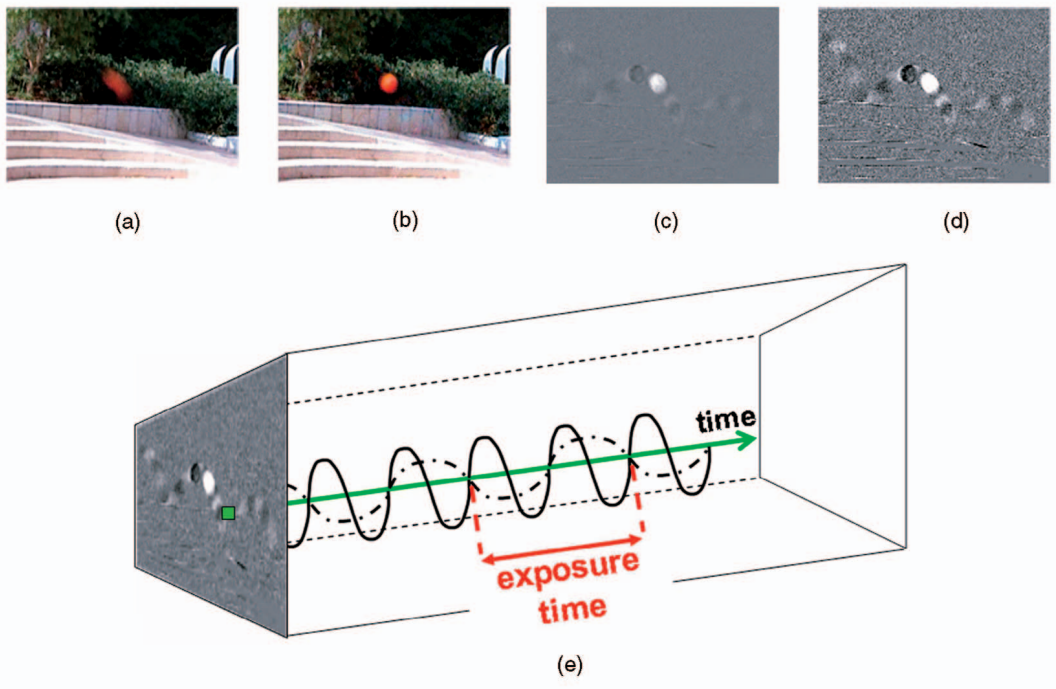

Fig. 17: "Ghosting" effect in video sequence. In order to show the "ghosting" effect caused by temporal SR, we applied the algorithm *without* any regularization to the basketball example (see Section 3). One input frame of the blurred ball is shown in (a). The temporal SR matching frame is shown in (b). (c) is the difference between the frame in (b) and a matching frame of the background. The "ghosting" effect is hard to see in a single frame (c) but is observable when watching a video sequence (due to the high sensitivity of the eye to motion). In order to show the effect in a single frame, we magnified the differences by a factor of 5. The resulting "ghosting" trail of the ball is shown in (d). Note that some of the trail values are positive (bright) and some are negative (dark). (e) illustrates that, although this effect has spatial artifacts, its origin is purely temporal. As explained in the text, due to the rectangular shape of the temporal blur, for each pixel (as the one marked in green), there are some specific temporal frequencies (e.g., the sinusoids marked in black) that will remain in the reconstructed sequence.

ACKNOWLEDGMENTS

The authors wish to thank Merav Galun and Achi Brandt for their helpful suggestions. This research was supported in part by the Israel Science Foundation grant no. 267/02. This research was done while Yaron Caspi was still at the Weizmann Institute of Science.

REFERENCES

[1] M.I. Sezan, A.J. Patti, and A.M. Tekalp, "Superresolution Video Reconstruction with Arbitrary Sampling Lattices and Nonzero Aperture Time," *IEEE Trans. Image Processing,* vol. 6, pp. 1064-1076, Aug. 1997.

[2] A. Blake, B. Bascle, and A. Zisserman, "Motion Deblurring and Super-Resolution from an Image Sequence," *Proc. European Conf. Computer Vision,* pp. 312-320, 1996.

[3] S. Baker and T. Kanade, "Limits on Super-Resolution and How to Break Them," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 9, Sept. 2002.

[4] S. Borman and R. Stevenson, "Spatial Resolution Enhancement of Low-Resolution Image Sequences—A Comprehensive Review with Directions for Future Research," technical report, Laboratory for Image and Signal Analysis (LISA), Univ. of Notre Dame, July 1998.

[5] M. Born and E. Wolf, *Principles of Optics.* Permagon Press, 1965.

[6] D. Capel and A. Zisserman, "Automated Mosaicing with Super-Resolution Zoom," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 885-891, June 1998.

[7] D. Capel and A. Zisserman, "Super-Resolution Enhancement of Text Image Sequences," *Proc. Int'l Conf. Pattern Recognition,* pp. 600-605, 2000.

[8] D.P. Capel, "Image Mosaicing and Super-Resolution," PhD thesis, Dept. of Eng. Science, Univ. of Oxford, 2001.

[9] Y. Caspi and M. Irani, "Spatio-Temporal Aignment of Sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 11, pp. 1409-1425, Nov. 2002.

[10] G. de Haan, "Progress in Motion Estimation for Video Format Conversion," *IEEE Trans. Consumer Electronics,* vol. 46, no. 3, pp. 449-459, Aug. 2000.

[11] M. Elad, "Super-Resolution Reconstruction of Images," PhD thesis, Technion Israel Inst. of Technology, Dec. 1996.

[12] H. Greenspan, G. Oz, N. Kiryati, and S. Peled, "MRI Inter-Slice Reconstruction Using Super Resolution," *Magnetic Resonance Imaging,* vol. 20, pp. 437-446, 2002.

[13] T.S. Huang and R.Y. Tsai, "Multi-Frame Image Restoration and Registration," *Advances in Computer Vision and Image Processing,* T.S. Huang, ed., vol. 1, pp. 317-339, JAI Press Inc., 1984.

[14] M. Irani and S. Peleg, "Improving Resolution by Image Registration," *CVGIP: Graphical Models and Image Processing,* vol. 53, pp. 231-239, May 1991.

[15] M. Irani and S. Peleg, "Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *J. Visual Comm. and Image Representation,* vol. 4, pp. 324-335, Dec. 1993.

[16] J.R. Price, J. Shin, J. Paik, and M.A. Abidi, "Adaptive Regularized Image Interpolation Using Data Fusion and Steerable Constraints," *SPIE Visual Comm. and Image Processing,* vol. 4310, Jan. 2001.

[17] V. Laude and C. Dirson, "Liquid-Crystal Active Lens: Application to Image Resolution Enhancement," *Optics Comm.,* vol. 163, pp. 72-78, May 1999.

[18] Z. Lin and H.Y. Shum, "On the Fundamental Limits of Reconstruction-Based Super-Resolution Algorithms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2001.

[19] N. Nguyen, P. Milanfar, and G. Golub, "A Computationally Efficient Superresolution Image Reconstruction Algorithm," *IEEE Trans. Image Processing,* vol. 10, no. 4, pp. 573-583, Apr. 2001.

[20] REALVIZ™, Retimer, www.realviz.com/products/rt, 2000.

[21] E. Shechtman, "Space-Time Super-Resolution in Video," MSc thesis, Dept. of Computer Science and Applied Math., Weizmann Inst. Science, Feb. 2003.

[22] E. Shechtman, Y. Capsi, and M. Irani, "Increasing Space-Time Resolution in Video," *Proc. European Conf. Computer Vision,* vol. 1, pp. 753-768, 2002.

[23] U. Trottenber, C. Oosterlee, and A. Schüller, *Multigrid.* Academic Press, Nov. 2000.

Eli Shechtman received the BSc degree cum laude in electrical engineering from Tel Aviv University in 1996 and the MSc degree summa cum laude in mathematics and computer science from the Weizmann Institute of Science in 2003. He received the Weizmann Institute excellence award for his MSc thesis, the best paper award at ECCV '02 (European Conference on Computer Vision), and a best poster award at CVPR '04 (IEEE Computer Vision and Pattern Recognition). He is currently a PhD degree student at the Weizmann Institute of Science. His current research interests are video analysis and synthesis, and motion-based recognition and classification.

Yaron Caspi received the BSc degree in mathematics and computer science from the Hebrew University of Jerusalem in 1991 and the MSc degree in mathematics and computer science from the Weizmann Institute of Science in 1993, working on human-like line-drawing interpretation. From 1994-1999, he worked at several computer-vision companies. He received the PhD degre in 2002 from the Weizmann Institute of Science, working on sequence-to-sequence alignment. For his PhD research, he received the J.F. Kennedy award and the Israeli Knesset (Israeli parliament) outstanding student award. He received the best paper award at ECCV '02 (European Conference on Computer Vision), and the honorable mention for the Marr Prize at ICCV '01 (IEEE International Conference on Computer Vision). During 2003-2004, he shared his time between Microsoft Research (Redmond), Washington, and the Hebrew University of Jerusalem. He is currently at Tel-Aviv University.

Michal Irani received the BSc degree in mathematics and computer science in 1985 and the MSc and PhD degrees in computer science in 1989 and 1994, respectively, all from the Hebrew University of Jerusalem. During 1993-1996, she was a member of the technical staff in the Vision Technologies Laboratory at the David Sarnoff Research Center, Princeton, New Jersey. She is currently an associate professor in the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science, Israel. Her research interests are in the areas of computer vision and video information analysis. Michal Irani's prizes and honors include the David Sarnoff Research Center Technical Achievement Award (1994), the Yigal Allon three-year fellowship for outstanding young scientists (1998), and the Morris L. Levinson Prize in Mathematics (2003). She also received the best paper awards at ECCV '00 and ECCV '02 (European Conference on Computer Vision), the honorable mention for the Marr Prize at ICCV '01 (IEEE International Conference on Computer Vision), and a best poster award at CVPR '04 (Computer Vision and Pattern Recognition). She is a member of the IEEE.