J Real-Time Image Proc (2007) 2:11–22
DOI 10.1007/s11554-007-0037-x

ORIGINAL RESEARCH PAPER

Real-time stabilization of long range observation system turbulent video

Barak Fishbain · Leonid P. Yaroslavsky · Ianir A. Ideses

Received: 18 March 2007 / Accepted: 1 August 2007 / Published online: 7 September 2007

Abstract The paper presents a real-time algorithm that compensates image distortions due to atmospheric turbulence in video sequences, while keeping the real moving objects in the video unharmed. The algorithm involves (1) generation of a "reference" frame, (2) estimation, for each incoming video frame, of a local image displacement map with respect to the reference frame, (3) segmentation of the displacement map into two classes: stationary and moving objects; (4) turbulence compensation of stationary objects. Experiments with both simulated and real-life sequences have shown that the restored videos, generated in real-time using standard computer hardware, exhibit excellent stability for stationary objects while retaining real motion.

Keywords LOROS · Turbulence compensation · Real-time processing · Rank filtering · Optical-flow

1 Introduction

Long range observation systems (LOROS) is a domain, which carries a lot of interest in many fields such as astronomy (i.e. planet exploration), geology (i.e. topographical measurements), ecology, traffic control, remote sensing, and homeland security (surveillance and military intelligence). Ideally, image quality in LOROS would be limited only by the optical setup used, but the major cause for image distortion in such systems is atmospheric turbulence. Atmospheric turbulence causes spatially and temporally chaotic fluctuations in the index of refraction of the atmosphere [1]. In remote sensing applications, light propagates long paths through the lower atmospheric regions (Troposphere) and the effect can be very destructive to the acquired image.

The troposphere layer is in constant motion due to winds and local temperature variations [2]. These variations cause formation air pockets, which have a uniform index of refraction and can be modeled as spherically shaped turbulent cells in a range of sizes and densities (referred to as "turbulent eddies"). This causes small neighborhoods in the image sequences to chaotically move in different directions in different frames. As a result, videos captured by optical sensors in the presence of atmospheric turbulence are very unstable and degraded in their resolution and geometry. One, who observes a turbulent affected scene, might feel uncomfortable watching the wavering output. Difficulties to observe real-moving objects when the entire image contains motion, due to turbulence, are also present. Therefore it is desirable to mitigate the turbulence distortions making the observation task more comfortable. Precise image instability compensation algorithm working in real-time will eventually allow integration of this algorithm into an embedded computer in the field.

In astronomical systems, the observed scene has a narrow field of view and the entire frame can be modeled by the convolution of the object with a single, though random, point spread function. Adaptive optics methods were suggested for dealing with turbulence in astronomical imaging systems [3–5], where aberrations induced by the atmosphere were compensated using mechanical means.

Remote sensing in terrestrial observations is affected by turbulence in a different way than astronomical systems. Long distant troposphere observations have wider field of view and are modeled by convolution with space variant

B. Fishbain (✉) · L. P. Yaroslavsky · I. A. Ideses
Department of Interdisciplinary Studies, the Iby and Aladar
Fleischman Faculty of Engineering, Tel-Aviv University,
Tel Aviv 69978, Israel
e-mail: barak@eng.tau.ac.il

and random point spread functions [6]. This is referred to as anisoplanatic imaging. For a short exposure of the sensor, the turbulence effect manifests itself mostly in a geometric distortions and less so in terms of blur. In long exposures, the images are much more blurred. In anisoplanatic imaging, light from each of the points in the scene acquired a slightly different tilt and low order aberration, causing the images of these points to be randomly dislocated from their correct geometrical positions. Hardware solutions involving adaptive optics fundamentally cannot meet the goal of improving the entire image, since even if we were to assume that a suitable wave front sensor beacon were present, adaptive optical systems can correct only a single isoplanatic patch of the image.

Some methods for turbulence distortions compensation of images acquired through the troposphere layer suggest finding the modulation transfer function (MTF) of the observed scene and inverse filtering the output image to obtain turbulent free images [7–9]. Extracting the MTF from the video sequences, however, requires some particular knowledge of the image content beforehand.

The effects of turbulence on imaging systems were widely recognized in the 1950s and 1960s. A number of methods were proposed to mitigate these effects [1, 3–5, 7–14]. The research that has been carried out until now dealt with turbulence compensation by optical and physical terms. Adaptive optics methods, as described earlier, are not adequate for this task. The mentioned MTF methods, on the other hand, require some prior knowledge about the sequence and can not cope with space variant distortions.

An image processing based turbulence compensation algorithm applying local neighborhood methods without prior knowledge about the scene observed, was suggested in [15–18]. The principal idea is using, for reconstructing distortion-compensated image frames, an adaptive control grid interpolation controlled by estimations of the spatially local displacement vectors. The algorithm also manages to preserve the genuine motion of the object by evaluating its motion vectors characteristics and making a decision whether to make the correction (turbulent motion) or not (real motion). However, the method for distinguishing real from turbulent motion presented in the published algorithm has quite high computational complexity and did not allow real-time implementation.

The present paper suggests, for this class of problems, a real-time algorithm that consists of three building blocks: (1) estimation of the stationary scene, (2) real motion extraction, and (3) generation of stabilized frames. To preserve real motion in the scene, the moving objects are located and the compensation for the distortion of the turbulence is applied only to the stationary areas of images. Each block is purposely designed for matching real-time processing requirements.

For real motion extraction, a reference image, which is an estimate of the stable image of the scene, is computed. An element-wise rank filtering in a temporal sliding window is suggested for obtaining the reference stable image [19]. For matching real-time requirements, the paper proposes using a fast recursive algorithm for calculating a specific rank order statistics. The reference image computation is addressed in Sect. 2.

In order to achieve real motion detection, it is necessary, for each pixel in the processed frame, to decide if it belongs to a moving or to a stationary object. For the decision-making, the reference image is used, as it keeps only the non-moving stationary objects. To speed up the processing, a two-step hierarchical decision mechanism is suggested:

- The first step is aimed at extracting areas, such as background, that are most definitely stationary. In most cases, a great portion of the pixels located in stable parts of the scene will be extracted at this stage. This first phase is detailed under Sect. 3.1.
- At the second step, the rest of the pixels are dealt. It improves the accuracy of detecting real movement at the expense of higher computational complexity. As it handles a small portion of the pixels, its higher computational complexity does not substantially reduce the total computation speed. This stage uses computing and processing of optical flow data. Discriminating real motion from turbulent one is achieved through statistical analysis of the magnitude and angle of the motion field. This is presented in Sect. 3.2.

Finally, all pixels in the incoming frame that are tagged as stationary are found by means of a pixel-wise combination, through fuzzy logic, of the input frame and the reference one. In this way, turbulence distortions are compensated while keeping the real moving objects in the video unharmed. In the resulting stabilized scene, intended for visual analysis, moving objects are left unharmed and are easily detectable and tracked in a stable background. Section 4 describes the generation of the stabilized sequence.

2 Computation of the reference stable image

In the turbulence compensation process, the reference image is an estimate of the stable scene. Such an image has to be obtained from the input sequence itself [9, 20, 21]. In order to achieve optimal results, the reference image should have the following properties:

- The reference image should contain only the static background with no moving objects in it.
- It should contain no turbulent induced geometric distortion.

In order to generate a reference image, one has to, for each time window, (1) take the current frame as the reference; (2) compute motion vectors for each frame in the time window with respect to this reference frame; (3) find pixel-wise means for motion vectors; (4) resample pixels of each frame to the positions defined be the means; this will generate a stabilized sequence. This process presents high computational load, and therefore an approximate solution is suggested.

The suggested solution generates the reference image using a pixel-wise rank filtering in a temporal sliding window. Specifically, temporal adaptive median filter is used for estimation of the stable scene, although other rank filters such as alpha-trimmed mean [15, 22] can also be considered. The size of the window, or the number of frames $\boldsymbol{N}$, over which the temporal rank filtering is carried out, is determined by the atmospheric turbulence correlation over the time domain, meaning that the longer the time correlation of the turbulence effect, the larger the temporal sliding window. In addition, it is important that the number of the images be sufficient enough to efficiently eliminate the moving objects. Figure 1 illustrates such an estimation of a stable scene from a real life turbulent video. Figure 1a presents a single frame taken from a turbulent distorted sequence (the entire sequence can be downloaded from [23]), while Fig. 1b presents the estimation of the stable scene calculated by temporal median over 117 frames. One can clearly see that chaotic geometrical distortions in (a) are removed in the stable estimation (b).

Generally, the rank filtering and, specifically, median filtering has high computational complexity since a sorting operation is carried out for each median computation. Exploiting the fact that the median computation is held over a sliding time window allows utilizing a fast recursive method for median filtering. The fast median recursive computation method used is a variation of the method described in [24]. First it holds an initialization process, in which odd number of $\boldsymbol{N}$ frames comprising the temporal window are read and the temporal histograms are computed for each pixel. Having completed the initialization process, for each pixel the following data is maintained: (a) the temporal window gray-level histogram; (b) the actual gray-level median value (***b***-parameter); (c) the number of pixels in the window having the same intensity level as the median (***c***-parameter); (d) the number of pixels in the temporal window, which have lower gray-level values than the median (***d***-parameter).

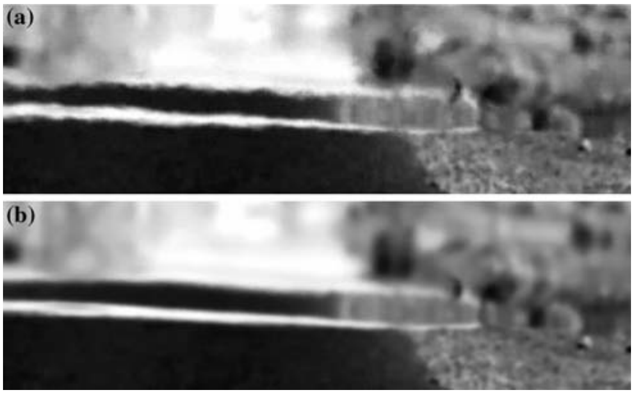


Fig. 1 Temporal median rank filtering for estimation of the stable scene: **a** a sample frame taken from a turbulent distorted video; **b** the corresponding stable scene estimation

Upon new frame acquisition, for each pixel, these data are updated according to the new and departing values of the moving window. If the departing or new values equal to the median then the ***c***-parameter for that pixel is decreased or, respectively, increased by one; if the departing or new values have lower value than the current median then ***d*** -parameter is decreased or increased in the same manner. Upon completion of the update process, one can be determined if the conditions described by Eqs. (1) and (2) are fulfilled.

$$d < \frac{N+1}{2} \tag{1}$$

$$[c+d] \geq \frac{N+1}{2} \tag{2}$$

If this is the case, then the pixel has preserved its median value (b). If this is not the case, for a given pixel, the median is calculated using the temporal histogram (a) and the pixel's corresponding ***b***-, ***c***- and ***d***- parameters are updated. Typically, the median values do not change significantly over time. Therefore the need for calculating the histogram for every pixel in every incoming frame will be notably smaller, thus decreasing the load derived due to the median computation.

Figure 2 contains three different frames (a), (c) and (e), taken in different times, from a real-life turbulent sequence (see [23] for the entire sequence). The same location is marked with a cross on each of those images. Images (a), (c) and (e) are taken before, throughout and after the bird passes through the marked pixel. The temporal histogram (over 117 frames) of this location is given in (b), (d) and (f). The gray-level histogram is computed in a temporal window centered at the corresponding frame. The gray-level of the median value is marked with a dashed line. The median gray-level's standard deviation throughout the entire sequences was 3.05 gray-levels, which, for the human observer, is an insignificant difference. The median value itself rarely changes throughout the sequence. In this specific example, the median was recalculated for less than 1% of the pixels.

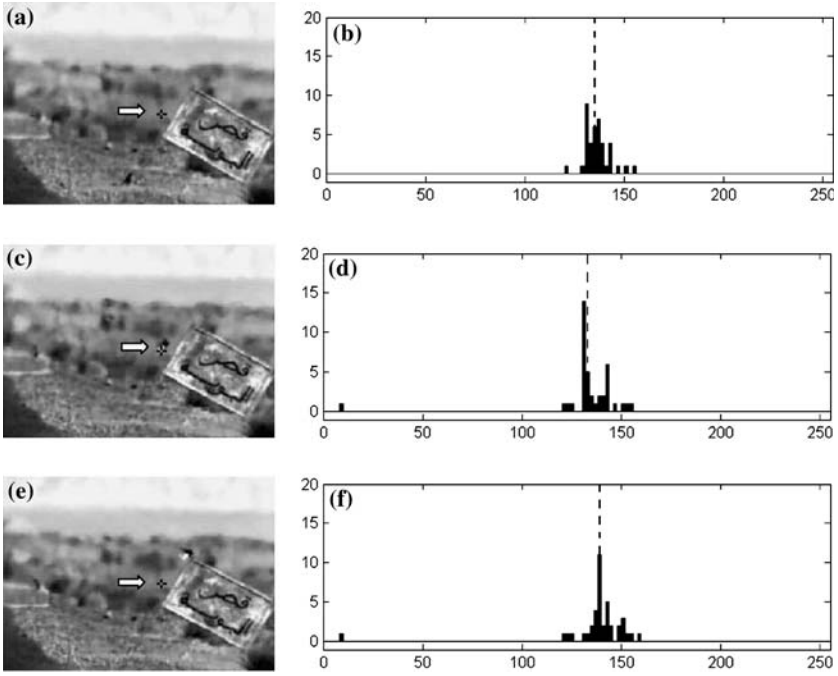


Fig. 2 Temporal median. **a**, **c** and **e** are images taken from a turbulent degraded real-life sequence. **b**, **d** and **f** are the corresponding temporal histograms for the pixel marked by a cross and pointed by an *arrow*. The *dashed line* on the histogram represents the temporal median gray-level value

3 Real motion extraction

In order to avoid, in course of the turbulence compensation process, compensation of real motion, pixels that represent real moving objects must be extracted from the observed frames. A following two-stage decision-making algorithm for moving object extraction meets requirements of efficient real-time computation. The first step is aimed at extracting areas, such as background, that are most definitely stationary and can be extracted in a very simple and fast way. In most cases, a great portion of the stable parts in the scene will be extracted at this stage. The rest of the pixels are dealt with at the second step. The second step improves extraction accuracy at the expense of higher computational complexity, but it handles a substantially smaller portion of the pixels. This stage uses computing and statistical analysis of optical flow for motion segmentation [15–18].

3.1 Real motion extraction—stage I

At this phase, the gray-level difference between running value of each pixel of the incoming frame and its temporal median (Distance From Median, $\mathrm{MDM}_{x,y,t}$), is calculated as "real-motion measure" (Eq. 3).

$$\mathrm{DFM}_{x,y,t} = I_{x,y,t} - \bar{I}_{x,y,t} \tag{3}$$

where t is index of the current processed frame and $\bar{I}_{x,y,t}$ is its median over the temporal window (Ω) centered at t

$$\bar{I}_{x,y,\Omega} = \mathrm{MED}_{x,y}\left\{I_{x,y,\Omega}\right\} \tag{4}$$

If the distance $DFM_{x,y,t}$ is below a given pre-defined threshold, the pixel is considered to be of a stationary object. The threshold is determined by exploiting the observer's limitation of distinguishing between close gray-levels. Background areas, which do not belong to a moving object nor are located near edges, will be resolved in this way, since these areas suffer less from turbulent distortion. All other pixels that are not resolved at this stage are processed at the next phase.

Figure 5 illustrates real-motion extraction using the above distance measure from the reference frame. Figure 5a is a single frame taken from a turbulent degraded sequence. The frame presented in Fig. 5b is the stable scene median estimation. Applying a difference threshold of 10 gray-levels, on the pixel-wise absolute difference between figure (a) and (b), results in (c) (higher difference values are printed in white); 10 gray-levels is insignificant difference to the observer, while it filters most stationary areas effectively. One can see that the car is detected as real moving object, and 75% of the pixels in the frame are tagged as stationary and will not, therefore, be further processed. Along with that, some areas on the background that contain no motion are also tagged as real moving objects, though; this movement is due to the turbulent motion. Those areas will be dealt with at the following stage.

3.2 Real motion extraction—stage II

The second motion extraction stage uses more sophisticated optical flow analysis methods in order to achieve better discrimination accuracy. The mapping of one turbulent image to a stable one can be obtained by registering a spatial neighborhood surrounding each pixel in the image in the reference image. Such a registration can be implemented using different optical flow methods [25–34]. In its simplest form, the optical flow method assumes that it is sufficient to find only two parameters of the translation vector for every pixel.

Let $I_{(x,y,t)}$ be a turbulent source image frame and $\bar{I}_{x,y,\hat{t}\in\Omega}$ be a reference image, the vectorial difference between the pixel's location in the original image and its location in the reference image be the motion vector $[\Delta x,\ \Delta y] = [x - \bar{x}, y - \bar{y}]$. For the subsequent processing stages, the translation vector is presented in polar coordinates, hence magnitude and angle $\{M_{(x,y,t)}, \theta_{(x,y,t)}\}$ of the motion vector. Having the motion field, one can discriminate real motion from turbulent one through a statistical analysis of the Magnitude $\{M_{(x,y,t)}\}$ and Angle $\{\theta_{(x,y,t)}\}$ components of the motion field.

3.2.1 *Real motion versus turbulence caused motion discrimination through magnitude's distribution cluster analysis*

Cluster analysis of the Magnitude distribution function for all (x, y), in a particular frame t, allows separating two types of motion amplitudes: small and irregular and large and regular. The first is associated with small movements caused by turbulence. The latter corresponds to movements caused by real motion. At the result, each pixel in the frame is assigned with a certainty grade between 0 and 1. This "Magnitude Driven Mask" ($\mathrm{MDM}_{(x,y,t)}$) characterizes magnitude based likelihood that particular pixels belong to objects in a real motion. Figure 3 presents a graph of the certainty as a function of the Motion vectors' magnitudes. Small Motion Vectors' magnitudes correspond to turbulent motion, while large magnitudes correspond to real-motion. The intermediate levels comprise motion vectors' magnitude upon which concise decision can not be made. The magnitudes' thresholds presented as $\boldsymbol{T_L}$ and $\boldsymbol{T_H}$ are application dependent and can be set by the user. In some applications, where the conditions are maintained, the system can set those thresholds automatically. Figure 5d presents the MDM extracted from the frame presented in Fig. 5a. Pixels where real-motion was detected are marked in white. In this example $\boldsymbol{T_L}$ and $\boldsymbol{T_H}$ were set empirically and are 2 and 4 pixels, respectively.

3.2.2 *Real motion versus turbulence caused motion discrimination through motion field's angle distribution*

Pixel's motion discrimination through angle distribution is achieved by means of statistical filtering of the angle component motion field. For each pixel, its neighborhood's angle's standard deviation is computed. Turbulent motion has chaotic directions. Therefore, a motion field, in a small spatial neighborhood, distorted by turbulence, has considerably large angular standard deviation. Real motion, on the other hand, has strong regularity in its direction and therefore its angles' standard deviation value over a local neighborhood will be relatively small. Homogeneous background areas contain no motion. Therefore the standard deviation of the zero motion vectors will be zero as well. The neighborhood size, in which the pixel's angular standard deviation is computed, should be large enough to make angle based discrimination of turbulent from real motion possible, and as small as possible to meet the terms of real-time computing. In experiments with real database it was found that neighborhood's size of 11 × 11–15 × 15 present a reasonable compromise.

As it is illustrated in the graph presented in Fig. 4, each pixel is assigned with "Angle Driven Mask" ($\mathrm{ADM}_{(x,y,t)}$), which presents an angle distribution based likelihood that this pixel belongs to an object in a real motion. Both turbulent and background areas should be regarded as stable. This means that real moving objects have a bounded angular standard deviation. $\boldsymbol{T_L}$ and $\boldsymbol{T_H}$ are the decision

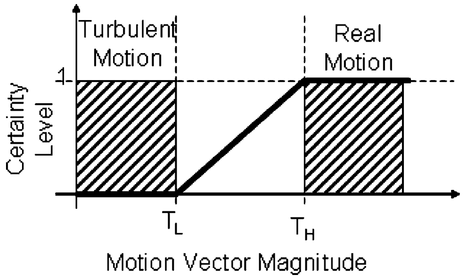


Fig. 3 Magnitude Driven Mask (MDM). MDM certainty level as a function of the motion vector's magnitude

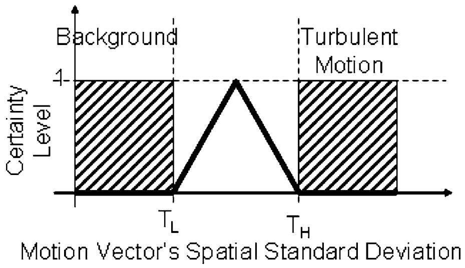


Fig. 4 Angle Driven Mask (ADM). ADM certainty level as a function of the motion vector's local spatial standard deviation

boundaries. Pixels with angular standard deviation smaller than $\boldsymbol{T_L}$ or higher than $\boldsymbol{T_H}$ are regarded as stationary. Those values are set by the observer.

Figure 5e presents the ADM extracted from the frame shown in (a), with low standard deviation values displayed with brighter pixels. One can see that background areas are tagged in white, hence contain low standard deviation. The signs on the road, which suffer most from turbulent in the acquired scene, have darker values, hence, high angular standard deviations. $\boldsymbol{T_L}$ and $\boldsymbol{T_H}$ were empirically assigned with $\frac{\pi}{3}$ and $\frac{\pi}{6}$, respectively.

3.2.3 Real motion separation mask

Having both the $\mathrm{MDM}_{(x,y,t)}$ and $\mathrm{ADM}_{(x,y,t)}$, a combined Real Motion Separation Mask ($\mathrm{RMSM}_{(x,y,t)}$) is formed as following:

$$\mathrm{RMSM}_{(x,y,t)} = \begin{cases} \mathrm{ADM}_{(x,y,t)}, & \left|\mathrm{ADM}_{(x,y,t)} - \frac{1}{2}\right| > \left|\mathrm{MDM}_{(x,y,t)} - \frac{1}{2}\right| \\ \mathrm{MDM}_{(x,y,t)}, & \text{otherwise} \end{cases} \tag{5}$$

The MDM and ADM are certainty measures ranging from 0 (turbulent motion) to 1 (real motion). Equation (7) implies that the ADM measure is more concise than the MDM when the term $\left|\mathrm{ADM}_{(x,y,t)} - \frac{1}{2}\right|$ has a higher value than $\left|\mathrm{MDM}_{(x,y,t)} - \frac{1}{2}\right|$. In this case the ADM measure will be used; otherwise the MDM value will be applied.

4 Generation of stable frames

Based on the notations derived in the previous sections, the output frame $F_{(x,y,t)}$ is given by

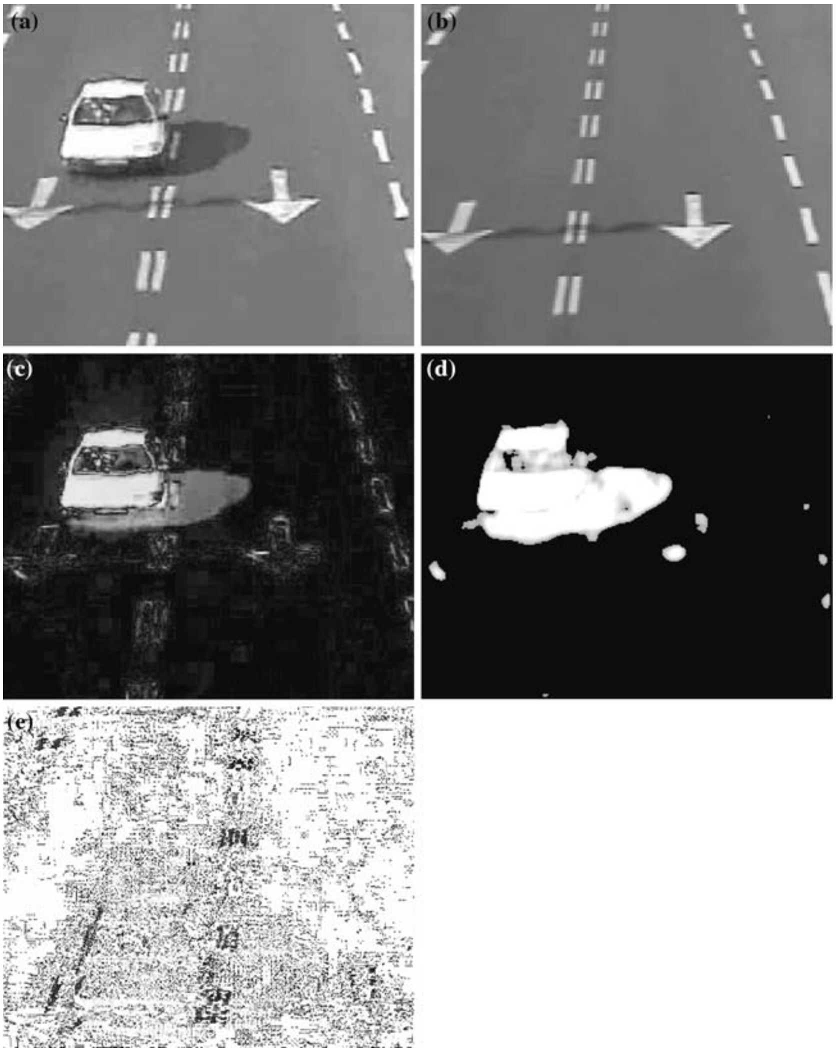


Fig. 5 Magnitude Driven Mask (MDM) and Angle Driven Mask (ADM). **a** presents a single frame extracted from real-life turbulent degraded sequence. **b** is the estimated stable scene. **c**, **d** and **e** present real-motion extraction by applying DFM, MDM and ADM, respectively

$$F_{(x,y,t)} = \bar{I}_{x,y,t} \cdot \left\{1 - \left[\mathrm{RMSM}_{(x,y,t)} \cdot \left(1 - \mathrm{DFM}_{x,y,t}\right)\right]\right\} + I_{(x,y,t)} \cdot \left[\mathrm{RMSM}_{(x,y,t)} \cdot \left(1 - \mathrm{DFM}_{x,y,t}\right)\right] \tag{6}$$

where "·" denotes element-wise matrix multiplication, $\bar{I}_{(x,y,t)}$ is an estimate of the stable component of the frame, DFM is the mask derived from the distance from the temporal median measure, as described in Sect. 3.1 and RMSM is the Real Motion Separation Mask detailed in Sect. 3.2.3.

5 Simulation and results

For evaluating the suggested method, a testing visual database has been composed that is freely accessible at [23]. The database contains several typical thermal and visual real-life video sequences as well as synthetic ones. The synthetic sequences are generated from real-life sequences containing real-movement with no turbulent motion. Turbulent-like degradations are induced using computer software. The process of generating images with turbulence-like distortions is outlined in Sect. 5.1 and the image stabilization algorithm results are detailed in Sect. 5.2. Throughout this section, the dense-map was computed using the Horn and Schunck optical flow method [27].

5.1 Turbulence simulation software

According the theory outlined in Sect. 1, the turbulence results in spatially and temporally correlated random geometrical distortions of the acquired image. In order to generate local correlated shifts for each pixel in the input image, a band limited correlated distortions field is generated. Figure 6 illustrates the process. For each direction, X and Y, a 3D random white noise 'cube' is first created. Then, a 3D Low-Pass filter is applied to the generated noise to produce spatial–temporal band-limited noise that defines a displacement field for each spatial coordinate. For shifting each pixel according to the displacement field, sliding window discrete sinc-interpolation algorithm [35] is applied.

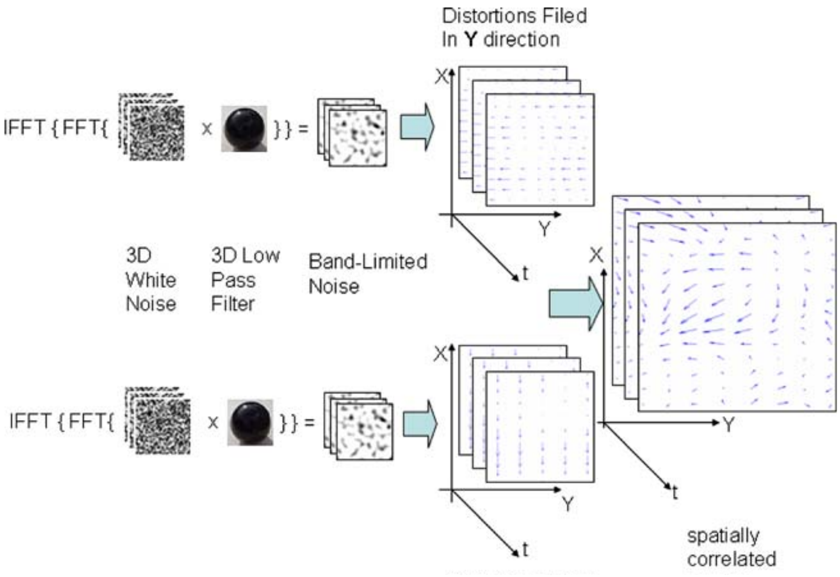
Figure 7 shows simulation results. Figure 7a is a frame extracted from a sequence containing real motion, with no turbulent degradation distortions present, while Fig. 7b is the corresponding frame from the sequence after applying turbulence-like degradation.

5.2 Results

5.2.1 *Turbulence compensation—quantitative evaluation*

As described earlier, the synthesized test sequence is a turbulent free video stream with real motion in which a turbulent like degradation is induced. The compensation process aims at removing the turbulent motion while retaining real moving objects. Figure 7 illustrates the compensation results achieved on the test sequence. Figure 7a shows a frame of the initial non-distorted sequence. Figure 7c shows a compensated corresponding frame of synthetic Fig. 7b. Comparing Fig. 7a, b and c, one can see that while the vehicle, as a real moving object, is retained, the turbulence distortions visible, for instance, on road markings in Fig. 7b of the background are compensated.

Fig. 6 Flow chart of the algorithm for generation of spatially and temporally correlated motion fields



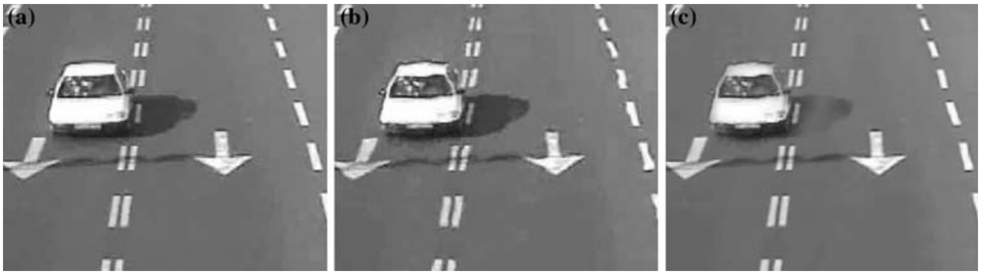


Fig. 7 Turbulence simulation and compensation. **a** Original sequence with real-motion and no turbulent motion; **b** the same frame with turbulence-like distortions induced; **c** the same frame after the turbulence compensation process

Quantitatively, the turbulence compensation quality can be evaluated using mean squared (MSE) compensation error defined by Eq. 7, and by the peak signal to compensation error ratio (PSNR), as defined by Eq. 8.

$$\mathrm{MSE} = \frac{1}{N_x N_y} \sum_{x,y} \| I_O(x,y) - I_C(x,y) \|^2 \qquad (7)$$

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{(\max\{I\})^2}{\mathrm{MSE}} \right) \qquad (8)$$

where $I_O(x,y)$ is the original image and $I_C(x,y)$ is the image after inducing and compensating turbulence degradations, N_x and N_y are dimensions of images in pixels and max $\{I\}$ is the maximum value can be assigned to a pixel, hence 255.

In the given example, the PSNR gives a high grade of 34.9 dB. Typical values for the PSNR in images are between 30 and 40 dB, where higher values than 34 are consider as good quality [36].

Yet another measure of the turbulence compensation quality is the ratio of compensation error energy to the original image energy.

The error image's energy to the original image's energy ratio is given by

$$\frac{\sum_{x,y} [I_O(x,y) - I_C(x,y)]^2}{\sum_{x,y} I_O^2(x,y)} \qquad (9)$$

For the given example, this ratio is 0.04.

5.2.2 *Turbulence compensation—qualitative evaluation*

In order to verify the method's real-life applicability, turbulent degraded sequences which were acquired by operational long range observation systems were processed by the system.

Figures 8 and 9 illustrate the results obtained with real-life video sequences. Figures 8a and 9a are both frames extracted from real life turbulent degraded sequences (see [23]). Corresponding figures (b) show the stable scene estimation computed by element-wise temporal median over of 117 frames. Figure (c) display the Distance From Median masks (DFM) and (d) display the Real Motion Separation Masks (RMSM). Comparing them, one can notice how the real motion extraction process is refined. Background areas which were tagged as real-motion by the DFM in (c) are removed from the RMSM in (d). The stable output of the non-turbulent background and unaffected vehicles with real moving objects are given in, Figs. 8e and 9e.

5.3 Real-time applicability

Since the number of operations, needed to complete the task is platform independent, while execution time is resources and platform dependent, the real-time applicability is evaluated through the number of operations. A tool, which helps examining if the number of operations for an algorithm complies with real-time constrains on a specific machine, can be downloaded from the author's site [23]. The analysis of the number of calculations in the following sections is given in terms of operations per pixel.

5.3.1 *Stable scene estimation computation*

The stable scene estimation is computed by applying pixel-wise, temporal median filtering. For computing the median over N samples, a sorting operation is needed. Optimized sorting operation requires $O(N\log(N))$ operations per pixel, where N, following the notations in Sect. 2, is the window's size along the temporal axis. For each window position, a recursive median algorithm requires, neglecting the initialization process, two operations per pixel for updating the temporal histogram (***a-***), and three operations modifying ***b-***, ***c-*** and ***d-*** parameters. Additional two operations per pixel are needed for computing Eqs. (1) and (2). This will allow determining if the median value has changed. If Eq. (1) is not fulfilled, than the new median value is lower than the older one. The algorithm slides leftward on the temporal histogram to the adjacent bin, when the number of pixels in the new pointed bin is

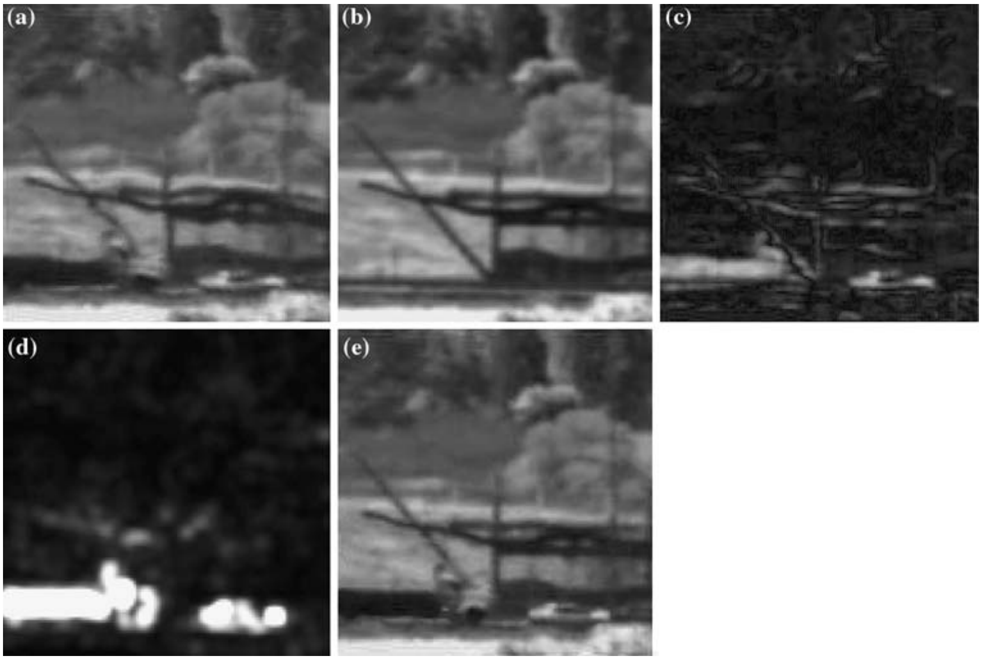


Fig. 8 Real-time turbulence compensation—trucks sequence. **a** Real-life atmospheric turbulence degraded image; **b** the stable scene estimation; **c** DFM; **d** RMSM; **e** the output image

subtracted from the ***d***-parameter. The sliding process is held till Eq. (1) is met. If Eq. (2) condition is not satisfied, the same process is held till its condition is met. In the latter case the sliding is rightward and for each shift the number of pixels in the bin is added to ***d***- parameter. In the most probable case, when the median does not change the number of computation needed is 7. When the median does change, for each shift, three additional operations are required. When the change in the median value is small this will not introduce significant additional computational load. When the change is large, hence substantial number of shifts, the process will introduce a major computational complexity which will be resulted in the lower processing throughput. The maximum number of shifts is 255.

5.3.2 Real motion extraction

5.3.2.1 Phase I This phase consists of a simple subtraction of the median from the input frame and applying threshold on the result. The number of operations per pixel is 2.

5.3.2.2 Phase II Optical Flow methods require the minimization of an energy functional [25–34]. In order to solve these large sparse systems numerically, classical iterative methods such as the Gauss–Seidel algorithm are commonly used [27]. While they are simple to implement, their convergence is not very fast, and often thousands of iterations may be necessary to get sufficiently close to the minimum of the energy functional. This is the reason why optical flow methods are too slow for time-critical applications.

As derived from the theory, the turbulence distortions are modeled by spatial, and temporal, random local shifts. This means that the motion of a pixel should be derived from its local neighborhood in the acquired and reference frames, rather than the entire image. While the general purpose of optical flow methods is computing the entire dense-map, the paper suggests, for the sake of reducing the computation load, to apply the optical flow computations to only certain pixels rather than to the entire image. As pixels tagged as stationary by phase I are not processed by phase II, the number of pixels processed by this phase will be small enough to allow real-time applicability. If the number of pixels processed in this phase is too big so the computational complexity exceeds the processing machine capabilities, the estimation of phase I is used solely. The magnitude and angles computation and thresholding are regarded as part of the computations needed for optical flow.

5.3.3 Generation of the output frames

According to Eq. 6, the generation of the stable frames requires eight operations per pixel.

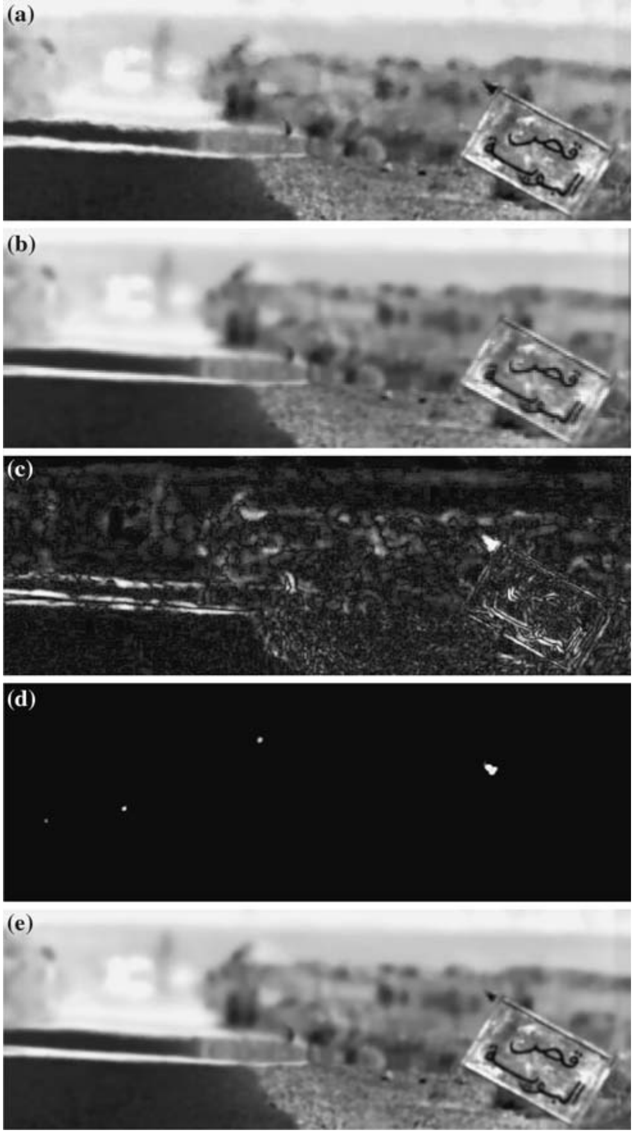


Fig. 9 Real-time turbulence compensation—bird sequence. **a** real-life atmospheric turbulence degraded image; **b** the stable scene estimation; **c** DFM; **d** RMSM; **e** the output image

5.3.4 Total computational complexity

Table 1 shows the number of operations per pixel for each task of the algorithm's tasks. As the number of pixels processed by phase II of the real motion extraction is content dependant, the maximum and minimum number of operations is indicated.

The acquisition system, acquires interlaced 4CIF format images of 704 × 576 pixels [37]. The method disclosed here eliminates the need for computing the entire dense-map, applying a hierarchical motion segmentation technique. If the number of pixels processed on the second stage of the real-motion extraction is less than 10% of the total number of pixels in the frame then the performance is equivalent to processing 25 frames per second on a standard 3.0 GHz PC. Further improvement to processing time can be achieved by utilizing the interlaced property of the incoming video. This is achieved by processing the odd and even fields individually on multi-processor/multi-core architectures.

Table 1 Number of Operations per Task

Task	Maximum	Minimum
Median	~770	7
Real motion extraction—phase I	2	2
Real motion extraction—phase II	~170	0
Generation of the output frames	8	8
Total	~950	17

6 Conclusion

The paper presents an algorithm for real-time compensating atmospheric turbulence distortions in video sequences using standard hardware, while keeping the real moving objects in the video unharmed. The algorithm is based on three building blocks: (1) estimation of the stable scene, (2) real motion extraction, and (3) generation of stabilized frames. To preserve real motion in the scene, moving objects are located and the compensation for the distortion of the turbulence is applied only to the static areas of images. To this goal, for each pixel in the incoming frame it is decided whether it is of a moving or a stationary object. A hierarchical two-stage decision making mechanism is suggested to this goal.

At the first stage, the absolute difference of the pixel's gray-level value and the temporal median is used to generate the motion extraction mask. This stage is computationally light and it allows to extracts most of stationary areas. The second stage improves accuracy of separating moving objects by more computationally complex algorithms. At this stage, optical flow computation is used for motion segmentation. Discriminating real motion is achieved through statistical analysis of the magnitude and angle of the motion field elements, which result in the Magnitude driven (MDM) and angle driven (ADM) motion separation masks that are combined using fuzzy logic operations to form the RMSM. The second stage exploits the turbulence motion characteristics, which are modeled by spatial and temporal random shifts and compute pixels' shifts in local neighborhoods. This allows computing pixels' shifts without the need to extract the entire scene motion field.

Finally, all areas in the incoming frame, which were tagged as stationary by the DFM and RMSM, are replaced with an estimation of the stationary scene. Note that real motion extraction, used in this paper for turbulence distortions compensation, can be used for obtaining any physical measure derived from the motion, such as speed and trajectory analysis.

The method was tested on several turbulent degraded synthetic as well as real-life videos. The restored videos exhibit excellent stability for stationary objects and yet retain the moving objects unharmed and easier to visually detect and track in a stable background. Those video are freely accessible on line [23].

The described algorithm was implemented on a standard desktop PC with 3 GHz processor and 1 GB of RAM on which it was running in real-time with the speed of 25 frames per second on 4CIF format images of 704 × 576 pixels.

Acknowledgments The authors appreciate the contribution to this research of Ofer Ben-Zvi and Alon Shtern, Faculty of Engineering, Tel-Aviv University, for their useful suggestions and their help with the C++ implementation of the algorithm. The video database was acquired with the kind help of Elbit Systems Electro-Optics—ELOP Ltd, Israel and the Israeli Army R&D branch.

References

1. Roggermann, M.C., Welsh, B.: Imaging Through Turbulence, Chap. 3, CRC Press, USA, pp. 57–115 (1996)
2. Farmer, W.M.: The Atmospheric Filter, vol. I Sources. JCD Publishing, (2001)
3. Welsh, B.M., Gardner, C.S.: Performance analysis of adaptive optics systems using slope sensors. J. Opt. Soc. Am. A **6**:1913–1923 (1989)
4. Thorpe, G., Lambert, A., Fraser, D.: Atmospheric turbulence visualization through image time-sequence registration. In: Proceedings International Conference on Pattern Recognition, vol. 2, pp. 1768–1770. IEEE Computer Society, Brisbane (1998)
5. Ellerbroek, B.: First-order performance evaluation of adaptive-optics systems for atmospheric-turbulence compensation in extended-field-of-view astronomical telescopes. J. Opt. Soc. Am. A, **11**(2):783 (1994)
6. Farmer, W.M.: The Atmospheric Filter, vol. II Sources. JCD Publishing, Winter Park (2001)
7. Sheppard, D.G., Hunt, B.R., Marcellin, M.W.: Iterative multi-frame super-resolution algorithms for atmospheric turbulence-degraded imagery. J. Opt. Soc. Am. A, **15**(4):972–992 (1998)
8. Sadot, D., Kopeika, N.: Imaging through the atmosphere: practical instrumentation-based theory and verification of aerosol modulation transfer function. J. Opt. Soc. Am. A, **10**(1):172 (1993)
9. Cohen, B., Avrin, V., Belitsky, M., Dinstein, I.: Generation of a restored image from a videosequence recorded under turbulence effects. Opt. Eng. **36**(12):3312–3317 (1997)
10. Frieden, B.R.: Turbulent image reconstruction using object power spectrum information. Opt. Commun. **109**(3–4): 227–230 (1994)
11. Wang, Y., Frieden, B.R.: Minimum entropy-neural network approach to turbulent-image reconstruction. Appl. Opt. **34**(26):5938–5944 (1995)
12. Mohammed, A.T., Burge, R.E.: Short-exposure turbulent image reconstructions. J. Phys. D Appl. Phys. **21**(7):1067–1077 (1988)
13. David, H.F., Joseph, W.M., Mark J.T.S.: Suppression of atmospheric turbulence in video using an adaptive control grid interpolation approach. In: Wee, S.J., Apostolopoulos J.G. (eds.) IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP01), vol. 3, pp. 1881–1884. IEEE International Society, Salt Lake City (2001)
14. van der Elst, H., van Schalkwyk, J.J.D.: Modelling and restoring images distorted by atmospheric turbulence. South African Symposium on Communications and Signal Processing (COMSIG-94), pp. 162–167. IEEE International Society, Stellenbosch (1994)
15. Gepshtein, Sh., Shteinman, A., Fishbain, B., Yaroslavsky, L.: Restoration of atmospheric turbulent video containing real motion using elastic image registration. In: The 2004 European Signal Processing Conference (EUSIPCO-2004), John Platt, pp. 477–480. Vienna, Austria, (2004)
16. Yaroslavsky, L., Fishbain, B., Shteinman, A., Gepshtein, Sh.: Processing and fusion of thermal and video sequences for terrestrial long range observation systems. In: Johan, S (ed.) The 7th International Conference on Information Fusion, pp. 848–855, International Society of Information Fusion, Stockholm, Sweden, June 2004
17. Yaroslavsky, L.P., Fishbain, B., Ideses, I., Slasky, D., Hadas, Z.: Simple methods for real-time stabilization of turbulent video. In: Calvo, M.L., Pavlov, A.V., Jahns, J. (eds.) Proceeding of ICO topical meeting on optoinformatics/information photonics, ITMO, pp. 138–140. St Petersburg, Russia (2006)
18. Fishbain, B., Yaroslavsky, L.P., Ideses, I.A., Shtern, A., Ben-Zvi, O.: Real-time stabilization of long-range observation system turbulent video. In: Proceedings of Real-Time Image Processing/Electronic Imaging 2007, SPIE Vol. 6496, San-Jose, CA, USA, 28 January–1 February 2007
19. Yaroslavsky, L.P.: Digital Holography and Digital Image Processing. Kluwer, Boston (2003)
20. Bondeau, C., Bourennane, E.: Restoration of images degraded by the atmospheric turbulence. In: Proceedings of the 4th International Conference on Signal Processing (ICSP), vol.2, pp. 1056–1059. Beijing China (1998)
21. Glick, Y., Baram, A., Loebenstein, H.M., Azar Z:. Restoration of turbulence-degraded images by the most-common method. Appl. Opt. **30**(27):3924–3929 (1991)
22. Cheung, S.C., Kamath, C.: Robust techniques for background subtraction in urban traffic video. Video Communications and Image Processing, SPIE Electronic Imaging, San Jose (2004)
23. http://www.eng.tau.ac.il/~barak/RealTimeTurbulenceCompensation
24. Huang, T.S., Yang, G.J., Yang, G.Y.: A Fast Two-Dimensional Median Filtering Algorithm. IEEE Trans. Acoust. Speech Signal Process. ASSP-27, 13 (1979)
25. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. J Imaging Vis. **14**(3), 245–255 (2001)
26. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings Of the 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674–679. Vancouver BC (1981)
27. Horn, B., Schunck, B.: Determining optical flow. Art. Intell. **17**, 185–203 (1981)
28. Mitiche, A., Bouthemy, P.: Computation and analysis of image motion: a synopsis of current problems and methods. Int. J. Comput. Vis. **19**(1), 29–55 (1996)
29. Barron, L.J., Fleet, D.J., Beachemin, S.S.: Performance of optical flow techniques. Int. J. Comput. Vis. **12**(1), 43–77 (1994)
30. Alvarez, L., Weickert, J., Sanchez, J.: Reliable estimation of dense optical flow fields with large displacement. Int. J. Comput. Vis. **39**(1), 41–56 (2000)

31. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation based on Theory for Wrapping. European Conference on Computer Vision (ECCV04), Vaclav Hlavac, vol. 4, pp. 25–36. Springer LNCS, Prague Czech Republic (2004)
32. Ben-Ari, R., Sochen, N.: A general framework for regularization in PDE based computation of optical flow via embedded maps and minimal surfaces. In: Proceedings of IEEE Computer Vision and Pattern Recognition Conference, Dan Huttenlocher and David Forsyth, pp. 529–536. IEEE Computer Society, New-York (2006)
33. Nagel, H.H., Enkelman, W.: An investigation of smoothness constrains for estimation of displacement vector fields from image sequences. IEEE Trans. Patt. Anal. Mach. Intell. **8**(5), 565–593 (1986)
34. Deriche, R., Kornprobst, P., Aubert, G.: Optical flow estimation while preserving its discontinuities: a variational approach. Lecture Notes in Computer Science, vol. 1035. pp. 71–80 (1996)
35. Yaroslavsky, L.P., Eden, M.: Fundamental of Digital Optics. Birkhäuser, Boston (1996)
36. Stefan W.: Digital Video Quality—Vision Models and Metrics. Wiley, New York (2005)
37. Recommendation H.261 of the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Line Transmission of Non-Telephone Signals—Video Codec for Audiovisual Services at p x 64 kb (1993)

Author Biographies

Barak Fishbain is a Ph.D. student in the school of Electrical Engineering in Tel-Aviv University, researching video enhancement through super resolution and motion estimation algorithms for traffic monitoring and remote sensing applications videos. Barak holds an M.Sc degree in Electrical Engineering from Tel-Aviv University (2004) and a B.Sc degree in Electrical Engineering from the Technion, Israel's institute of Technology (1998).

Leonid P. Yaroslavsky MS, (Summa cum laude, 1961), Ph.D. (1968), Dr. Sc.-Phys. Math (1982). Till 1995, he had headed a Laboratory of Digital Optics at the Institute for Information Transmission Problems, Russian Academy of Sciences. Beginning 1995, he is Professor at Department of Interdisciplinary Studies, Faculty of Engineering, Tel Aviv University. He is an author of several books and more then 100 papers on digital image processing and digital holography. Fellow of Optical Society of America.

Ianir A. Ideses is a Ph.D. student in the school of Electrical Engineering in Tel-Aviv University, researching 3D visualization, synthesis and compression. Ianir holds an M.Sc degree in Electrical Engineering from Tel-Aviv University (Magna cum laude, 2004) and a B.Sc degree in Electrical Engineering from the Technion, Israel's institute of Technology (1998).