BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 7
Special Issue on Information Fusion

Sofia • 2015 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0086

2D Video Stabilization for Industrial High-Speed Cameras

Atanas Nikolov, Dimo Dimov

Institute of Information and Communication Technologies, BAS, 1113 Sofia, Bulgaria
Emails: a.nikolov@iinf.bas.bg dtdim@iinf.bas.bg

Abstract: *The current research concerns the problem of video stabilization "in a point", which aims to stabilize all video frames according to one chosen reference frame to produce a new video, as by a static camera. Similar task importance relates providing static background in the video sequence that can be usable for correct measurements in the frames when studying dynamic objects in the video. For this aim we propose an efficient combined approach, called "3×3OF9×9". It fuses our the previous development for fast and rigid 2D video stabilization* [*2*] *with the well-known Optical Flow approach, applied by parts via Otsu segmentation, for eliminating the influence of moving objects in the video. The obtained results are compared with those, produced by the commercial software Warp Stabilizer of Adobe-After-Effects CS6.*

Keywords: *Video stabilization, static camera, motion vectors, optical flow, object segmentation.*

1. Introduction

Video stabilization aims at improving qualities of unstable recorded video (amateur or professional one) in a point or on a smooth trajectory. Stabilization in a point means that the stabilization of all video frames is done according to a single reference frame. As a result, a video recorder by a static camera is simulated in this case. The stabilization on a trajectory produces a video, where the camera follows the original operator's movement, but in its smooth variant.

1.1. Video stabilization on a smooth trajectory

In scientific computer vision literature, a great number of software video stabilization methods, classified as 2D or 3D, are well-known. The two-dimensional

methods compute a 2D model of movement (rigid, affine, perspective) between the consecutive frames, after which the model parameters are smoothed in time, and at the end, the geometric transformation from original to smoothed movement is applied [4, 8, 9]. Three-dimensional approaches for a video stabilization use feature points tracking to recover 3D position and orientation of the camera, and then synthesize new frames, located on a smooth trajectory [5].

Lately, the methods, which combine the qualities of 3D methods and speed, and the robustness of 2D methods, turn out quite perspective [3, 6, 12, 13, 14]. All of these methods stabilize on a smooth trajectory. We have presented a more detailed overview on this topic in [2].

1.2. Video stabilization in a point

Video Stabilization in a point is mainly used when the static background is a fundamental requirement for correct measurements of characteristics and properties of studied dynamic objects and/or processes (such as distances, velocities, accelerations, direction and type of movement, elasticity, robustness, etc.), captured by a video camera.

One possible application is to stabilize the video obtained from high-speed industrial cameras, where the necessity of stabilization arises during a handheld video recording or when the foundation for the camera tripod is unstable. The following situations are possible: (1) recording of fast processes in high vibrating (industrial) environment; (2) capturing of a controlled explosion (causing massive disturbances in proximity of the camera); (3) handheld video recording from a vehicle; (4) handheld recording "ad hoc" in "field conditions", etc.

1.3. Video stabilization for high-speed industrial cameras

In the present research, we propose developing and testing of software algorithms for video stabilization of clips, recorded by the high-speed camera[1] of IICT-BAS, bought via AComIn[2] Project. The necessity of video stabilization at this type of cameras arises when the dynamics of environment is not possible to be overcome, i.e., the camera cannot be statically fixed.

Due to the camera specification – a record speed from 50 up to 370 330 frames per 1 s (fps), the video stabilization is carried out in "off-line" mode, i.e., in principle, algorithmic fast response is not looked for here. The most important is the perfect stabilization, especially in the cases of so called "stabilization in a point", i.e., to be accomplished the requirement for complete immobility of the scene in the stabilized video clip.

The challenge here is to reduce the effect of a "sliding scene", which is a typical problem even for the famous commercial algorithms for video stabilization, like Warp Stabilizer of Adobe-After-Effects CS6. But, a dominating requirement in similar algorithms is stabilizing on a smooth trajectory of the camera movement and/or scene objects, where the "sliding scene" effect is not perceived by the user.

[1] http://www.nacinc.com/products/memrecam-high-speed-digital-cameras/hx-6/
[2] http://www.iict.bas.bg/acomin/

Here, in Section 2 we give the necessary background for the combined algorithm, that we propose in Section 3 to solve the highlighted problem. Besides, experimental results are reported in Section 4, before the final conclusion.

2. Description of the proposed separate video stabilization methods

2.1. Dividing the video frames of 3×3 regions

Like in [2], each video frame here is divided in 3×3 uniform regions, which define (equal to 9) the number of the searched motion vectors $\vec{t}_{ij}$ $(i, j = 1, 2, 3)$ between the centers of the 9 sub-images in the current and previous frame (Fig. 1a).

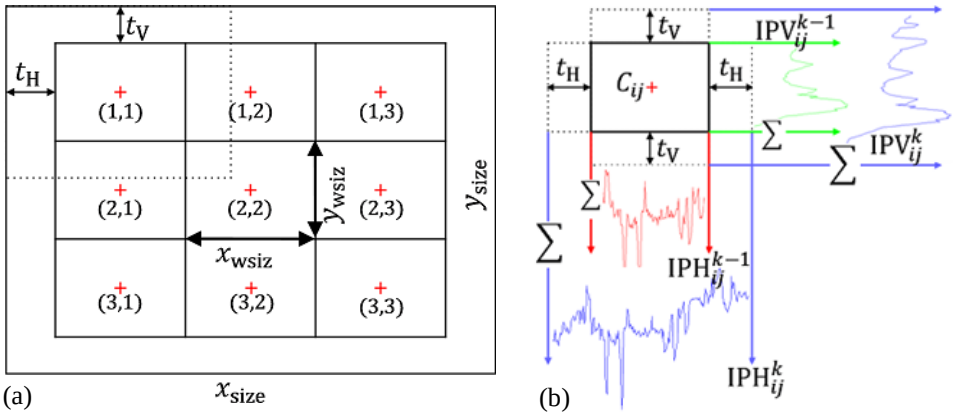


Fig. 1. A base scheme for motion vectors determination: 3×3 frame division (a); horizontal and vertical integral projections (b)

2.1.1. Motion vectors determination

- Horizontal (by columns) and vertical (by rows) integral projections IPH_{ij}^{k} and IPV_{ij}^{k}, for each of the 9th regions with a center C_{ij} for the current (k-th) video frame and the previous one, are determined (Fig. 1b). Here, $(x_{\mathrm{size}}, y_{\mathrm{size}})$ are original frame sizes, and $(x_{\mathrm{wsiz}}, y_{\mathrm{wsiz}})$ are the sizes of the work area C_{ij} from the current frame: $x_{\mathrm{wsiz}} = (x_{\mathrm{size}} - 2t_{\mathrm{H}})/3$ and $y_{\mathrm{wsiz}} = (y_{\mathrm{size}} - 2t_{\mathrm{V}})/3$, where t_{H} and t_{V} are admissible limits for the maximal x and y components of the searched motion vectors $\vec{t}_{ij}$.

- The corresponding motion vectors $\vec{t}_{ij} = (t_{xij}^{(k)}, t_{yij}^{(k)})$ for the current frame are found as the minimum of the Sum of Absolute Difference (SAD) approach:

(1) $$t_{xij}^{(k)} = \mathrm{argmin}_{(\tau)}\{\mathrm{SADH}_{ij}^{k}(\tau)\};$$

$$\mathrm{SADH}_{ij}^{k}(\tau) = \sum_{x=-x_{\mathrm{wsiz}}/2}^{x_{\mathrm{wsiz}}/2} \left|\mathrm{IPH}_{ij}^{k}(x+\tau) - \mathrm{IPH}_{ij}^{k-1}(x)\right|, \quad -t_{\mathrm{H}} < \tau < t_{\mathrm{H}};$$

(1a) $$t_{yij}^{(k)} = \operatorname{argmin}\{\mathrm{SADV}_{ij}^{k}(\tau)\};$$

$$\mathrm{SADV}_{ij}^{k}(\tau) = \sum_{y=-y_{\mathrm{wsiz}}/2}^{y_{\mathrm{wsiz}}/2} \left|\mathrm{IPV}_{ij}^{k}(y+\tau) - \mathrm{IPV}_{ij}^{k-1}(y)\right|, \ -t_{\mathrm{V}} < \tau < t_{\mathrm{V}}\,.$$

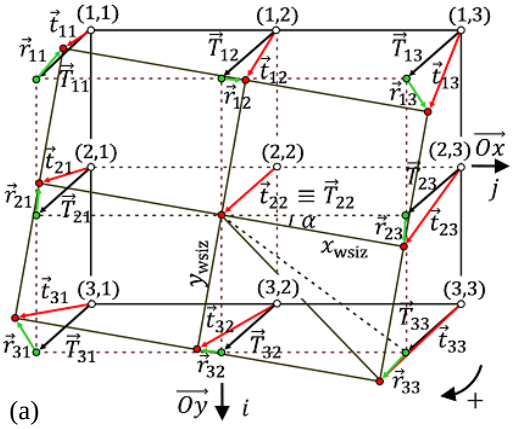
- Our improvement of the upper estimation consists in normalizing the integral projections by "floating" average values, $\mathrm{I\ddot{P}H}_{ij}^{k}$ and $\mathrm{I\ddot{P}V}_{ij}^{k}$. For example, the normalization by horizontals is as follows:

(2) $$\mathrm{I\ddot{P}H}_{ij}^{k}(x,\tau) = \mathrm{IPH}_{ij}^{k}(x) - \widetilde{\mathrm{IPH}}_{ij}^{k}(\tau)\,,$$

$$\widetilde{\mathrm{IPH}}_{ij}^{k}(\tau) = \frac{1}{x_{\mathrm{wsiz}}} \sum_{p=-x_{\mathrm{wsiz}}/2}^{x_{\mathrm{wsiz}}/2} \mathrm{IPH}_{ij}^{k}(\tau+p).$$

2.1.2. Determination of the motion parameters (translation and rotation)

In the proposed basic geometric model, each motion vector $\vec{t}_{ij}$ found by SAD between two frames is decomposed of a sum of two vectors: a translation vector $\vec{T}_{ij}$ and a rotation vector $\vec{r}_{ij}$, i.e., $\vec{t}_{ij} = \vec{T}_{ij} + \vec{r}_{ij}$ (Fig. 2a). The rotation vectors $\vec{r}_{ij}$ are expressed via the reference vector of rotation $\vec{r} = \vec{r}_{23} = (r_{x23},\ r_{y23})$. In this way, a linear system of 18 components (by *Ox* and *Oy*) equations is composed (Fig. 2b), where the unknown parameters are $(T_x,\ T_y,\ r_x\,,\ r_y)$.



(a)

(b)

(i, j)	$\overrightarrow{Ox}$	$\overrightarrow{Oy}$
(1,1)	$T_x - r_x + kr_y = t_{x11}$	$T_y - kr_x - r_y = t_{y11}$
(1,2)	$T_x + kr_y = t_{x12}$	$T_y - kr_x = t_{y12}$
(1,3)	$T_x + r_x + kr_y = t_{x13}$	$T_y - kr_x + r_y = t_{y13}$
(2,1)	$T_x - r_x = t_{x21}$	$T_y - r_y = t_{y21}$
(2,2)	$T_x = t_{x22}$	$T_y = t_{y22}$
(2,3)	$T_x + r_x = t_{x23}$	$T_y + r_y = t_{y23}$
(3,1)	$T_x - r_x - kr_y = t_{x31}$	$T_y + kr_x - r_y = t_{y31}$
(3,2)	$T_x - kr_y = t_{x32}$	$T_y + kr_x = t_{y32}$
(3,3)	$T_x + r_x - kr_y = t_{x33}$	$T_y + kr_x + r_y = t_{y33}$

Fig. 2. A basic geometric model, concerning translation and rotation between two consecutive frames (a); a linear system of 18 component equations for centres (i, j) (b)

For solving this overdetermined linear system (see Fig. 2b), we use the idea for isolating the unknown parameters by writing the system as a matrix equation, whose decision by the Least Squares Method (LSM) looks like:

(3) $$(A^{\mathrm{T}}A)\vec{x} = A^{\mathrm{T}}B \ \Rightarrow\ \vec{x} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}B\,.$$

That equation represents another formulation of the solution [2] (cf. also the similar Equation (8) in next Section 2.3), i.e.,

$$\begin{matrix}\overrightarrow{Ox}\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \overrightarrow{Oy}\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\begin{bmatrix}1 & 0 & -1 & k\\ 1 & 0 & 0 & k\\ 1 & 0 & 1 & k\\ 1 & 0 & -1 & 0\\ 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 0\\ 1 & 0 & -1 & -k\\ 1 & 0 & 0 & -k\\ 1 & 0 & 1 & -k\\ \hdashline 0 & 1 & -k & -1\\ 0 & 1 & -k & 0\\ 0 & 1 & -k & 1\\ 0 & 1 & 0 & -1\\ 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 1\\ 0 & 1 & k & -1\\ 0 & 1 & k & 0\\ 0 & 1 & k & 1\end{bmatrix}\cdot\begin{bmatrix}T_x\\ T_y\\ r_x\\ r_y\end{bmatrix}=\begin{bmatrix}t_{x11}\\ t_{x12}\\ t_{x13}\\ t_{x21}\\ t_{x22}\\ t_{x23}\\ t_{x31}\\ t_{x32}\\ t_{x33}\\ \hdashline t_{y11}\\ t_{y12}\\ t_{y13}\\ t_{y21}\\ t_{y22}\\ t_{y23}\\ t_{y31}\\ t_{y32}\\ t_{y33}\end{bmatrix}\Rightarrow\begin{bmatrix}9 & 0 & 0 & 0\\ 0 & 9 & 0 & 0\\ 0 & 0 & 6+6k^2 & 0\\ 0 & 0 & 0 & 6+6k^2\end{bmatrix}\cdot\begin{bmatrix}T_x\\ T_y\\ r_x\\ r_y\end{bmatrix}=\begin{bmatrix}a\\ b\\ c\\ d\end{bmatrix},$$

$$(A^{\mathrm{T}}A)_{[4\times4]}.x_{[4\times1]}=(A^{\mathrm{T}}B)_{[4\times1]},$$

$$A_{[18\times4]}.x_{[4\times1]}=B_{[18\times1]},$$

where:

(3a)
$$T_x=\frac{a}{9}\,,\quad T_y=\frac{b}{9}\,,\quad r_x=\frac{c}{6+6k^2}\,,\quad r_y=\frac{d}{6+6k^2}\;;$$

$$a=\sum_{i=1,j=1}^{3,3}t_{xij}\;;\;\; b=\sum_{i=1,j=1}^{3,3}t_{yij}\;\;;$$

$$c=k\sum_{j=1}^{3}(t_{y3j}-t_{y1j})+\sum_{i=1}^{3}(t_{xi3}-t_{xi1})\;;$$

$$d=k\sum_{j=1}^{3}(t_{x1j}-t_{x3j})+\sum_{i=1}^{3}(t_{yi3}-t_{yi1})\,;\;\; k=\frac{y_{\mathrm{wsiz}}}{x_{\mathrm{wsiz}}}.$$

The rotation angle α is determined by the components of the rotation vector $\vec{r}$ as $\alpha=2\mathrm{arctg}\,(r_x/r_y)$.

2.2. Dividing the video frames of 9×9 regions

By analogy of 3×3 vector model (see Subsection 2.1.2), a system with 162 component equations (by *Ox* and *Oy*) is composed for the method with 9×9 division, where the unknown parameters are the same as before: (T_x, T_y, r_x, r_y). The equations are composed according to the relation $\vec{t}_{ij}=\vec{T}_{ij}+\vec{r}_{ij}$, where $\vec{t}_{ij}$ are known (estimated by the SAD method) motion vectors between the centers (i,j), $i,\ j=1,\ldots,9$, for each couple of consecutive frames. Similarly, $\vec{T}_{ij}=\vec{T}$, where $\vec{T}=(T_x,\ T_y)$ is the unknown translation vector, while the rotation vectors $\vec{r}_{ij}$ are calculated through the unknown reference vector $\vec{r}=(r_x,\ r_y)=\ \vec{r}_{5,9}$, considering the simple geometric relations, shown in Fig. 3.

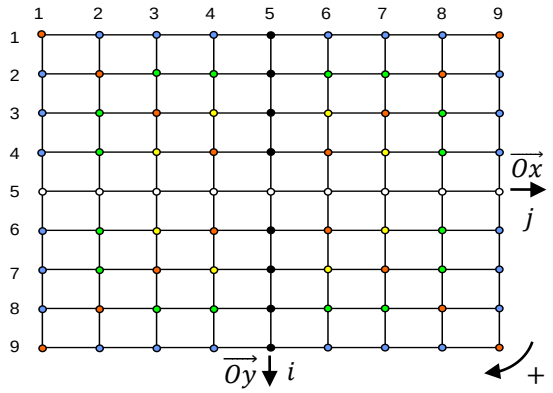


Fig. 3. Frame division of 9×9 regions with the corresponding centre $(i,\ j)$

Thus the unknown $(T_x,\ T_y,\ r_x,\ r_y)$ can be discovered applying LSM on a new overdetermined linear system, similar to this of Equation 3. The obtained solution $(T_x,\ T_y,\ r_x,\ r_y)^{\mathrm{T}} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}B$, is appropriate for real-time implementations:

$$T_x = \frac{a}{81}\,,\quad T_y = \frac{b}{81}\,,\quad r_x = \frac{4c}{135+135k^2}\,,\quad r_y = \frac{4d}{135+135k^2}\,, \tag{4}$$

where:

$$k = \frac{y_{\mathrm{wsiz}}}{x_{\mathrm{wsiz}}};\quad a = \sum_{\mathrm{i=1,j=1}}^{9,9} t_{xij}\,;\quad b = \sum_{\mathrm{i=1,j=1}}^{9,9} t_{yij}\,;$$

$$\begin{aligned} c = {} & k\sum_{j=1}^{9}(t_{y9j} - t_{y1j}) + \frac{3}{4}k\sum_{j=1}^{9}(t_{y8j} - t_{y2j}) + \frac{1}{2}k\sum_{j=1}^{9}(t_{y7j} - t_{y3j}) + \\ & + \frac{1}{4}k\sum_{j=1}^{9}(t_{y6j} - t_{y4j}) + \sum_{i=1}^{9}(t_{xi9} - t_{xi1}) + \frac{3}{4}\sum_{i=1}^{9}(t_{xi8} - t_{xi2}) + \\ & + \frac{1}{2}\sum_{i=1}^{9}(t_{xi7} - t_{xi3}) + \frac{1}{4}\sum_{i=1}^{9}(t_{xi6} - t_{xi4})\,; \end{aligned}$$

$$\begin{aligned} d = {} & k\sum_{j=1}^{9}(t_{x1j} - t_{x9j}) + \frac{3}{4}k\sum_{j=1}^{9}(t_{x2j} - t_{x8j}) + \frac{1}{2}k\sum_{j=1}^{9}(t_{x3j} - t_{x7j}) + \\ & + \frac{1}{4}k\sum_{j=1}^{9}(t_{x4j} - t_{x6j}) + \sum_{i=1}^{9}(t_{yi9} - t_{yi1}) + \frac{3}{4}\sum_{i=1}^{9}(t_{yi8} - t_{yi2}) + \\ & + \frac{1}{2}\sum_{i=1}^{9}(t_{yi7} - t_{yi3}) + \frac{1}{4}\sum_{i=1}^{9}(t_{yi6} - t_{yi4})\,, \end{aligned}$$

and similarly, for the rotation angle α, we have $\alpha = 2\mathrm{arctg}\,(r_x/r_y)$.

2.3. The famous optical flow approach

The goal of the Optical Flow method consists in estimating a 2D vector field of the motion of every pixel between a pair of frames via space-temporal brightness variations (Fig. 4). For determination of the displacement vector $(u,\ v)$ for every pixel from the frames, the following key considerations must be taken into account:

- for *constant brightness*: the projection of a 3D point must look one and the same in all the frames;
- for *temporal consistency*: the movement must be smooth with small displacements between the frames;
- for *spatial coherency*: the movement of a given point must be the same as its neighbours.

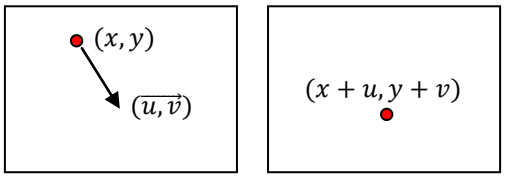


Fig. 4. The optical flow method's idea

The first consideration leads to the equality:

$$I(x+u,\ y+v,\ t+1) = I(x,\ y,\ t). \tag{5}$$

At small displacements, the function I can be linearly approximated in a small neighbourhood of $(x,\ y,\ t)$ via Taylor expansion:

$$I(x+u, y+v, t+1) \approx I(x,y,t) + \frac{\partial I}{\partial x}(x,y,t).u + \frac{\partial I}{\partial y}(x,y,t).v + \frac{\partial I}{\partial t}(x,y,t). \tag{6}$$

Replacing (6) in (5), the basic Optical Flow equation is obtained:

$$I(x,y,t) + I_x.u + I_y.v + I_t - I(x,y,t) = 0 \Leftrightarrow I_x.u + I_y.v + I_t = 0 \Leftrightarrow \tag{7}$$
$$\Leftrightarrow \nabla I[u\ v]^{\mathrm{T}} + I_t = 0,$$

where I_x and I_y are the spatial derivatives at time t; I_t is the time derivative at $(x,\ y,\ t)$, i.e., $I_t = I(x,\ y,\ t+1) - I(x,\ y,\ t)$; $\nabla I = [I_x\ I_y]$ is the gradient vector in the point $(x,\ y,\ t)$.

For exact solving of the basic equation, at least two linear independent equations are necessary. Thus, based on the *temporal consistency* constraint, Lukas and Kanade [7] offer consideration of a neighbourhood of 5×5 pixels around each point:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \Rightarrow \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}. \tag{8}$$

Thus, the solution of the over determined system $A\vec{x} = b$ can be found in a LSM sense via solving the matrix equation $(A^{\mathrm{T}}A)\vec{x} = A^{\mathrm{T}}b \Rightarrow \vec{x} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$.

This matrix equation has been offered for first time by Lukas and Kanade, which solution is possible when: (1) $A^{\mathrm{T}}A$ is an invertible matrix; (2) the eigenvalues λ_1 and λ_2 of $A^{\mathrm{T}}A$ should not be too small, i.e., $\det(A^{\mathrm{T}}A) = \lambda_1\lambda_2 \neq 0$.

The Lukas and Kanade's method in this form is applicable only for small displacements between the frames (up to 1 pixel). For overcoming this limitation, the iterative approach for estimating the optical flow in a Gaussian pyramid of images is used [1].

3. Description of the proposed combined approach "3×3OF9×9" for video stabilization

For solving the task of video stabilization in a point, the methods described in Section 2 could not cope alone with satisfactory precision. Their shortcomings can be marked as follows:

- The 3×3 approach is more sensitive to errors, because of the small number used for motion vectors, via which the searched translation and rotation are determined;
- The 9×9 approach is comparatively steady to wrong estimated motion vectors via SAD correlation, because of the bigger number of region centres, which are uniformly distributed in each frame. But, like the 3×3 method, it suffers from the integer value of the motion vectors (at pure translation) via SAD, which leads to shaking within the limits of one pixel (this is especially noticeable at stabilization in a point, but not on a smooth trajectory).
- The Optical Flow method in its most lightweight (i.e., fast and enough imprecise) version [11] works well only for comparatively small displacements between the frames. It is sensitive to uneven changes of illumination in the images (e.g., at presence of shadows), as well as to sharp changes in the border of objects of different depth.

For getting over the shortcomings of the above three approaches, when consider them separately, an original combined approach "3×3OF9×9" for video stabilization in a point is presented hereinafter. It consists of two consecutive stages of stabilization.

3.1. First stage of stabilization

1) A coarse stabilization according to a chosen reference frame is accomplished via our previous method with 3×3-division of frames [2]. Thus, the bigger displacements between the frames can be compensated fast and efficiently;

2) In this way the stabilized frames, the known method of Optical Flow [1] is used for determination the motion vectors with sub-pixel accuracy for every pixel in the compared pair of respective sub-frames;

3) A 9×9-extension of our 3×3-method is applied to the corresponding 9×9 centres in the Optical Flow vector field for improving the coarse stabilization of the 3×3 method.

Using only the first stage of stabilization in a point leads to the typical effect of a "sliding scene". This effect is due to accumulating a sub-pixel error in the global trajectory (from displacements between consecutive pair of frames) according to the chosen reference frame. Additionally, the existence of a massive dynamic object, as well as objects of different depths, also leads to big errors in estimated displacements.

In this way, this first stage of stabilization can be used independently for precise 2D rigid stabilization (translation + rotation) on a smooth trajectory. For precise extra stabilization "in a point", the next stage will be necessary.

3.2. Second stage of stabilization

For reduction of the accumulated error in the global reference trajectory that often leads to the effect of a "sliding scene", the following strategy is applied:

1) On the frames coarsely stabilized at the first stage, a binary mask for isolating the influence of the dynamic objects is applied (Fig. 5).

2) Motion vectors between enough distant frames (e.g., about a 100 frames interval) are calculated, so that the error from the accumulated trajectory to be distributed among the intermediate frames in the interval. Thus, the accumulated global trajectory error is corrected every time in the chosen interval of frames.

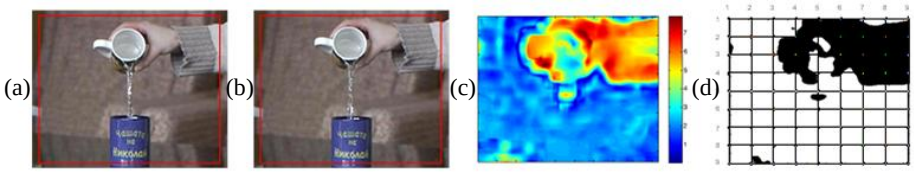


Fig. 5. Elimination of the moving objects influence: first control frame (a); second control frame (b); map of magnitudes of optical flow vectors (c); Otsu binarization [10], applied to the extended 9×9 motion vectors to improve the precision of the static background stabilization (d)

4. Experimental results

The experiments of stabilization for a static camera are held on two types of videos:

- A video clip, downloaded from the producer's webpage of the high-speed camera NAC Memrecam HX-6, containing a record of the controlled explosion, where a large dynamic object is present. The video has 1280×720 resolution and a service speed of 25 fps.
- Clips recorded by hand with simulated shaking, containing a record of water/drops in result from pouring in a cup (Fig. 7). The clips have 640×480 resolution and speed of 50, 100, and 500 fps.

4.1. Experiment with a controlled explosion

The presence of a large moving object in the scene causes the algorithms (of fixed division scheme, 3×3 and/or 9×9) to follow the object's movement. This effect is expressed in the characteristic 'sliding of the scene', i.e., the scene background "moves" in the object opposite direction (Fig. 6b). For this reason, we use stabilization on the trajectory of the predominant object (Fig. 6c), thus, ignoring the sliding effect and achieving the desirable static background.

Fig. 6. Stabilization of a video clip with controlled explosion: first frame (a); last frame after stabilization in a point (BAD) (b); last frame after stabilization on a smooth trajectory (OK) (c)

4.2. Experiment with pouring of water

At this experiment, for the three video clips with water pouring, which are with different speed and amplitudes of shaking, the "sliding scene" effect is also present (Fig. 7), but in much smaller limits than the effect at controlled explosion. This allows for applying our strategy for reducing the sliding error (Fig. 8) via isolating (segmenting) the moving objects (Section 3.2).

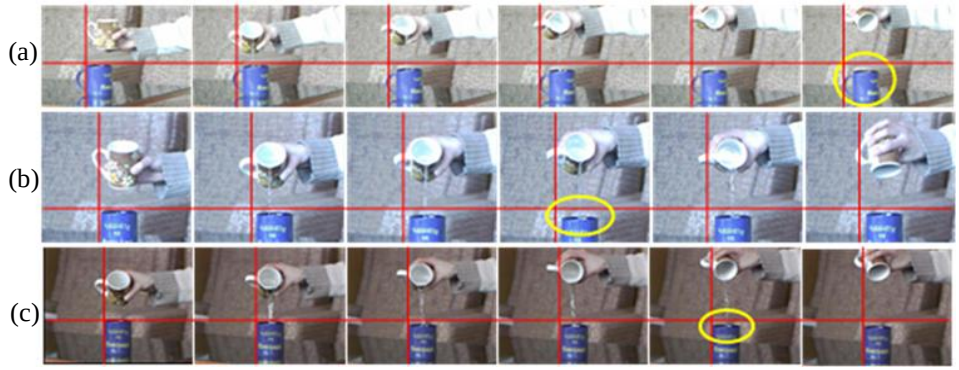


Fig. 7. "Stabilized frames" in a point of the 1st stage of stabilization: 50 fps (a); 100 fps (b); 500 fps (c)

The errors from the 1st stage of stabilization are greatly reduced at the 2nd stage (Fig. 8), as now they are less than 1% from the frames size (Table 1).

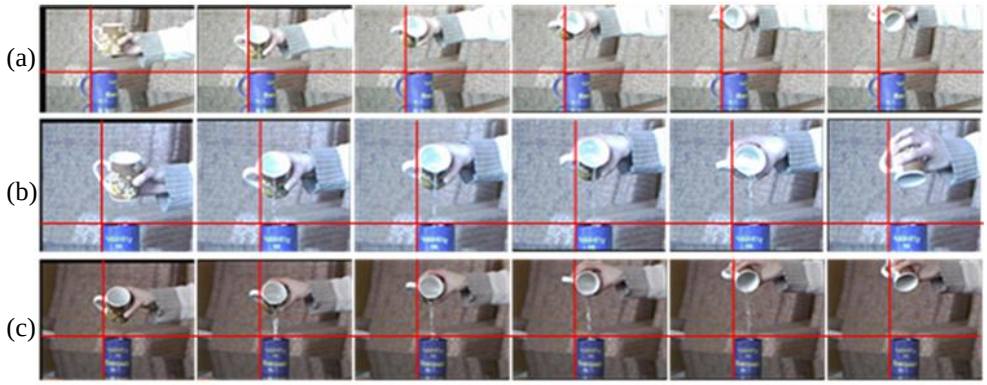


Fig. 8. Stabilized frames in a point of the second stage of stabilization: 50 fps (a); 100 fps (b); 500 fps (c)

In Table 1 the maximal error on *Ox* and *Oy* directions for the corresponding speed of water video clips at both stages of stabilization are given.

Table 1. Errors from stabilization in a static background after the 1st and the 2nd stage of stabilization

Resolution 640×480	**Video 1 (50 fps)**		**Video 2 (100 fps)**		**Video 3 (500 fps)**	
	1st stage	2nd stage	1st stage	2nd stage	1st stage	2nd stage
max *X* error	3.44%	0.16%	1.09%	0.47%	1.09%	0.16%
max *Y* error	5.00%	1.04%	3.75%	0.63%	4.17%	0.42%

4.3. Comparison with Adobe-After-Effects (AAE) CS6

In Fig. 9 a visual comparison (our stabilization results vs. AAE CS6) for stabilization in a point for the water video clip with 500 fps speed is done. In this experiment the effect of slowly floating scene is evident at AAE's result, although they claim to have an option for stabilizing in a static background.

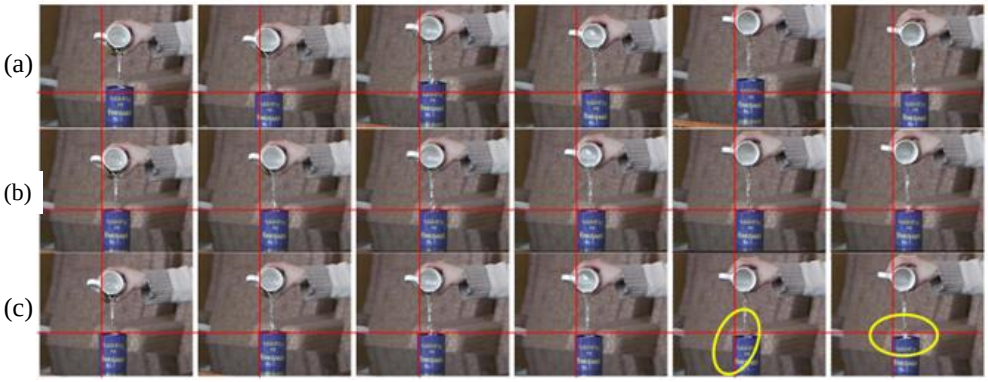


Fig. 9. Our result for stabilization in a static background vs. the result from Adobe-After-Affects CS6: original video frames (a); our stabilized result via 3×3OF9×9 method (b); AAE stabilized result (c)

5. Conclusion

A combined approach (called "3×3OF9×9") for video stabilization with detection and elimination of moving objects of reasonable size in the scene is proposed. The method is distinguished by speed and precision of the stabilization. It is intended for video experiments (e.g., with a high-speed camera), where the static background is usually required. The obtained experimental results can be evaluated as quite good, according to the requirement for a static background of the scene or for a static predominant object in the scene, which allows correct measurements in video frames. The specific effect of the "sliding scene" has been reduced less than 1% for a frame size of 640×480 pixels, and the pure execution time of the algorithm remains about 30 fps. The future efforts will be directed to more efficient segmentation of big moving objects in the proposed method context.

***Acknowledgements*:** This research is supported by AComIn Project "Advanced Computing for Innovation" of IICT-BAS (2012-2016), Grant 316087, funded by FP7 Capacity Programme (**www.iict.bas.bg/acomin/**).

References

1. B o u g u e t, J. Pyramidal Implementation of the Lucas-Kanade Feature Tracker: Description of the Algorithm. Technical Report, Open CV Document, Intel Microprocessor Res. Labs, 2000.
2. D i m o v, D., A. N i k o l o v. Real Time Video Stabilization for Handheld Devices. – In: Proc. of CompSysTech'14, Ruse, Bulgaria, ACM ICPS, Vol. **883**, 2014, pp. 124-133.

3. G o l d s t e i n, A., R. F a t t a l. Video Stabilization Using Epipolar Geometry. – ACM Transactions on Graphics, Vol. **31**, 2012, No 5, pp. 126:1-126:10.
4. G r u n d m a n n, M., V. K w a t r a, I. E s s a. Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. – In: Conference on CVPR, 2011, pp. 225-232.
5. L i u, F., M. G l e i c h e r, H. J i n, A. A g a r w a l a. Content-Preserving Warps for 3D Video Stabilization. – ACM Transactions on Graphics, Vol. **28**, 2009, Issue 3, pp. 44:1-44:9.
6. L i u, F., M. G l e i c h e r, J. W a n g, H. J i n, A. A g a r w a l a. Subspace Video Stabilization. – ACM Transactions on Graphics, Vol. **30**, 2011, Issue 1, pp. 4:1-4:10.
7. L u c a s, B., T. K a n a d e. An Iterative Image Registration Technique with an Application to Stereo Vision. – In: Proc. of Int. Joint Conf. on Artificial Intelligence, 1981, pp. 674-679.
8. M a t s u s h i t a, Y., E. O f e k, W. G e, X. T a n g, H. S h u m. Full-Frame Video Stabilization with Motion Inpainting.– IEEE Trans. on PAMI, Vol. **28**, 2006, No 7, pp. 1150-1163.
9. M o r i m o t o, C., R. C h e l l a p p a. Evaluation of Image Stabilization Algorithms. – In: DARPA Image Understanding Workshop DARPA97, 1997, pp. 295-302.
10. O t s u, N. A Threshold Selection Method from Gray-Level Histograms. – IEEE Transactions on Systems, Man, and Cybernetics, Vol. **9**, 1979, Issue 1, pp. 62-66.
11. Piotr's Computer Vision Matlab Toolbox. **http://vision.ucsd.edu/~pdollar/toolbox/doc/**
12. S h u a i c h e n g, L., L. Y u a n, P. T a n, J. S u n. Bundled Camera Paths for Video Stabilization. – ACM Transactions on Graphics, Vol. **32**, 2013, No 4, Article 78. 10 p.
13. S h u a i c h e n g, L., L. Y u a n, P. T a n, J. S u n. SteadyFlow: Spatially Smooth Optical Flow for Video Stabilization. – In: IEEE Conf. on CVPR, June 2014, pp. 4209-4216.
14. W a n g, Y.-S., F. L i u, P.-S. H s u, T.-Y. L e e. Spatially and Temporally Optimized Video Stabilization. – IEEE Trans. on Visualiz. and Comp. Graphics, Vol. **19**, 2013, Issue 8, pp. 1354-1361.