

2D and 3D Image Stabilization for Robotic Beating Heart Surgery

Gerhard Kurz and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
gerhard.kurz@kit.edu, uwe.hanebeck@ieee.org

Abstract—Image stabilization is relevant for various industrial and medical applications. In particular, we consider the use of image stabilization in robotic beating heart surgery. A robot, which is remotely controlled by the surgeon, can automatically compensate for the motion of the beating heart. To give the surgeon the illusion of operating on a stationary heart, a stabilized image of the beating heart is shown to the surgeon. Image stabilization cancels the unwanted motion of the heart, but retains changes to color and texture, for example cuts on the heart surface. In this paper, stabilization is first considered as a 2D image transformation problem. Subsequently, it is extended to stabilization of a 3D point cloud or surface. The proposed algorithms are evaluated in both ex-vivo and in-vivo experiments. In the evaluation, the stabilization quality achievable with several common interpolation functions is compared.

Keywords—*interpolation, point cloud, surface reconstruction, coronary artery bypass graft, motion cancellation*

I. INTRODUCTION

Image stabilization has various applications, e.g., stabilizing videos recorded with hand-held cameras or industrial applications that involve handling of deformable objects. More specifically, motion is to be removed, but, changes in appearance or should remain visible. In particular, image stabilization is useful for medical image processing where the natural motion of organs as result of heartbeat and respiration is to be canceled. In the medical context, motion is caused by non-uniform deformation of tissue. Consequently, traditional techniques to remove camera motion (e.g., affine or projective transformations) are insufficient.

A particularly relevant application of image stabilization is beating heart surgery as proposed by Nakamura et al. [1]. Surgery on the heart, for example coronary artery bypass graft (CABG), is commonly performed by a procedure called cardiopulmonary bypass (CPB). The heart is stopped and a heart-lung machine is used for the duration of the operation. This approach poses increased risk for the patient such as anemia and cellular hypoxia [2]. However, performing surgery on the beating heart is very demanding for the surgeon. Therefore, Nakamura et al. suggested to use a teleoperated robot that automatically compensates for the heart motion and to present a stationary picture to the surgeon [1]. The techniques described in this paper can be used to obtain a stationary view of the beating heart.

In previous work, different approaches have been attempted to create a stabilized view of the beating heart. We give an

overview of the most significant contributions in this area. Temporal sub-sampling can be used to only show the heart when it is at the desired position. Experiments with a ECG-triggered stroboscopic light have been performed, but the results were unsatisfactory [3].

Some researchers try to use a global transformation for image stabilization. For example, Stoyanov et al. [4] describe a method that is based on moving a virtual camera in order to compensate for the motion of the heart. Because the heart is affected by non-uniform deformations, global transformations only provide a limited amount of stabilization [5].

To account for the non-uniform deformation of the heart, local transformations have been proposed. In [5], a geometric transformation based on linear interpolation within the triangles of a Delaunay triangulation is presented. A tracking approach based on thin plate splines is proposed in [6]. Thin plate splines describe the deformation of every point on the surface and can also be used for image stabilization.

Because geometric transformations do not necessarily correspond to physically plausible deformations, an image stabilization approach based on a physical model has been suggested in [7]. The physical model describes the heart as an elastically deformable object that is acted upon by outside forces. A refined version that includes automatic adaptation of the model has been published in [8].

Our contribution in this paper consists of the following. We describe a general approach to achieve image stabilization in both 2D and 3D, which is based on interpolation. Furthermore, we compare the 2D and 3D methods as well as several commonly used interpolation functions. We evaluate the different techniques not only ex-vivo with an artificial heart, but also in an in-vivo experiment with a real porcine heart.

II. POINT CORRESPONDENCES

All of the 2D and the 3D approaches presented in this paper rely on a set of known point correspondences between the reference image and the current image. Point correspondences can be obtained by tracking the movement of certain points on the surface. These points can either be natural or artificial landmarks. Various approaches for tracking natural landmarks on the heart surface have been proposed, for example [9], [5], and [10], but regions with insufficient texture and specular reflections make reliable tracking of natural landmarks difficult. This issue can be avoided by using artificial landmarks that are designed to be reliably tracked. To present our method,

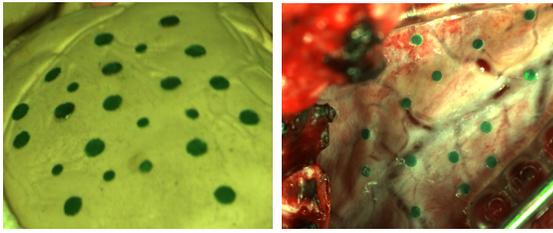


Fig. 1. Heart phantom and porcine heart with artificial landmarks as seen by the camera system.

we only consider artificial landmarks in the experiments (see Fig. 1), but we would like to emphasize that the described methods are not limited to scenarios with artificial landmarks. They could be applied in the presence of natural landmarks just as well, assuming a reliable algorithm for tracking natural landmarks is given. We discuss the tracking of landmarks in more detail in [11].

Tracking can be performed either in 2D or in 3D. If a 3D stabilization is to be used, three-dimensional tracking of the landmarks is necessary. A dense 3D reconstruction of the surface is not required. 2D stabilization can be performed in conjunction with either 2D or 3D tracking schemes. If 3D tracking is used with 2D stabilization, the 3D points are back-projected into the 2D images in order to obtain the required 2D point correspondences.

While it is possible to obtain the coordinates of the landmarks directly from the images, it is beneficial to use a stochastic model to describe their movement. A stochastic model can handle uncertainties as well as noisy measurements and can combine the information from several sensors [12]. Furthermore, it allows tracking even when some or all of the landmarks are occluded. Robust handling of occlusions is of particular importance in a clinical setting, since blood, smoke, or surgical instruments may temporarily occlude some of the landmarks. Models based on Fourier series have been proposed by Richa et al. [13]. Ortmaier et al. have suggested a prediction scheme based on Takens' Theorem [14]. We use a model that takes physical knowledge about the heart into account. Our model is based in a system of partial differential equations that consider the heart as a linear elastic physical body. A more detailed description can be found in [15]. The model provides an estimate of the position of all landmarks at each time step, even when some or all landmarks are currently occluded.

III. 2D STABILIZATION

The general objective of image stabilization is to remove motion from a video stream while preserving changes to color and texture. Changes in illumination and reflections will remain visible, unless further measures are taken to remove them. We avoid the issue of motion blur by assuming the exposure time is sufficiently small such that motion blur is negligible. In situations where this assumption cannot be fulfilled, motion blur may be removed with deconvolution [16].

A reference image from a certain time step is given and information from the current image is transformed to appear in the reference image. This problem formulation is independent of the particular details of the application. Even though we

only consider images obtained with color cameras, image stabilization can also be applied to images obtained by other means, for example X-ray, ultrasound or magnetic resonance imaging.

The goal of 2D image stabilization is finding a mapping

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad [x, y]^T = f(x', y'),$$

which transforms coordinates $[x', y']^T$ in the reference image to coordinates $[x, y]^T$ in the current image. One should be aware that f is not necessarily invertible, so this is not in general equivalent to finding a mapping from the current image to the reference image.

A. Proposed Approach

The marker tracking gives a set of m correspondences

$$\{(x_i, y_i, x'_i, y'_i) \mid 1 \leq i \leq m\}$$

between the current and the reference image. As the image is supposed to be warped to the reference image, the function f should interpolate these corresponding points, i.e., the equation

$$[x_i, y_i]^T = f(x'_i, y'_i), \quad 1 \leq i \leq m$$

should be fulfilled. Thus, the problem of image stabilization is reduced to the problem of scattered data interpolation. In some cases, approximation of the point correspondences according to $[x_i, y_i]^T \approx f(x'_i, y'_i)$, $1 \leq i \leq m$ might be considered instead of exact interpolation.

Some methods for interpolation or approximation are only capable of representing functions $\mathbb{R}^n \rightarrow \mathbb{R}$, but a mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ is required for 2D image stabilization. This can be achieved by considering two separate mappings $x = f_1(x', y')$ and $y = f_2(x', y')$.

B. Approximation and Interpolation Functions

The function f can be chosen from different families of functions. Different types of functions suitable for image warping are reviewed in [17], [18], and [19]. We compare several possible alternatives that are suitable to the problem of image stabilization in a medical context. While there are many different ways to perform interpolation, we restrict ourselves to methods that are fairly common in the medical context, easy to implement and fast enough for real-time applications. Furthermore, we only consider methods that can be used based on point correspondences. Our comparison does not claim to be an exhaustive review of all known interpolation methods.

1) *Affine Transformation*: An affine transformation is given by

$$f(x, y) = \mathbf{A} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \underline{t},$$

where $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ is a non-singular matrix and $\underline{t} \in \mathbb{R}^2$ is a translation vector. Since affine transformations only have six degrees of freedom, in general they only achieve approximation, not interpolation. The parameters \mathbf{A} and \underline{t} can be determined by solving a linear least squares problem.

2) *Piecewise Linear Transformation*: In order to take local differences into account, a Delaunay triangulation of the points $\{(x'_i, y'_i \mid 1 \leq i \leq m)\}$ in the reference image is calculated. Inside each triangle, linear interpolation is performed by considering the plane that is determined by the three corners of the triangle and the corresponding values of f . The resulting surface is C^0 -continuous, but not differentiable. This approach has previously been applied to the problem of image stabilization for the beating heart [5].

3) *B-Splines*: B-Splines are commonly used for interpolation, because they produce a C^2 -continuous surface. They have previously been used for representing a beating heart by Lau et al. [9]. However, applying B-Splines to scattered data interpolation is not straightforward. An algorithm to interpolate scattered points by using multiple levels of B-Splines is presented in [20]. Our implementation is based on this algorithm. An alternative approach to use B-Splines with scattered data can be found in [21].

4) *Radial Basis Functions (RBFs)*: The interpolation function is given by

$$f(x, y) = \sum_{i=1}^m a_i \phi \left(\left\| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix} \right\| \right)$$

where a_1, \dots, a_m are weights and $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is a radial basis function. The weighting coefficients a_1, \dots, a_m can be obtained from the point correspondences by solving a linear system of m equations. Some of the possible choices for the radial basis function ϕ are:

- Locally supported functions, e.g.,

$$\phi(r) = \begin{cases} 1 - \left(\frac{r}{\sigma}\right)^2 \cdot (3 - 2\frac{r}{\sigma}) & r < \sigma \\ 0 & \text{otherwise} \end{cases}$$

with parameter $\sigma > 0$.

- Thin plate splines (TPS) [22], which have been used by [6] for tracking of the beating heart

$$\phi(r) = \begin{cases} r^2 \log r, & r > 0 \\ 0, & r = 0 \end{cases}$$

- Gaussian

$$\phi(r) = e^{-r^2/\sigma^2}$$

with parameter $\sigma > 0$.

The continuity of the interpolation function f is equal to the continuity of $\phi(\|x\|)$. It is common to consider an extended version of f that includes an affine transformation

$$f(x, y) = r_1 + r_2x + r_3y + \sum_{i=1}^m a_i \phi \left(\left\| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix} \right\| \right)$$

and that requires additional constraints

$$\sum_{i=1}^m a_i = 0, \quad \sum_{i=1}^m a_i x_i = 0, \quad \sum_{i=1}^m a_i y_i = 0.$$

An overview of all presented interpolation functions is given in Table I. For each function type, the table describes whether interpolation is achieved (or only approximation), the continuity of the resulting function, and whether or not points have global influence, i.e., whether changes to the value at any given point affect the whole surface.

Function Type	Interpol.	Continuity	Global
affine	no	C^∞	yes
piecewise linear	yes	C^0	no
B-Spline	yes	C^2	yes
RBF (locally supp.)	yes	C^1	no
RBF (TPS)	yes	C^1	yes
RBF (Gaussian)	yes	C^∞	yes

TABLE I. COMPARISON OF APPROXIMATION AND INTERPOLATION FUNCTIONS.

C. Implementation

In general, the transformation does not yield integer coordinates, so bilinear interpolation [23] is used to determine the color of the transformed pixel. More sophisticated approaches such as bicubic interpolation may be used at the cost of increased computation time. In some cases, the transformation might result in coordinates that lie outside the image. These pixels cannot be warped and are colored black. Although most of the proposed interpolation schemes allow for extrapolation, we restrict image stabilization to the convex hull of all reference points, because stabilization quality is low in border areas and it is reasonable to assume that the area of interest is contained in the convex hull of all landmarks.

Input: current image I_c ,
point correspondences $\{(x_i, y_i, x'_i, y'_i) \mid 1 \leq i \leq m\}$
Output: stabilized image I_s
determine f such that $[x_i, y_i]^T = f(x'_i, y'_i), 1 \leq i \leq m$;
for $[x', y']^T \in I_s$ **do**
 $[x, y]^T \leftarrow f(x', y')$;
 if $[x, y]^T$ inside I_c **then**
 $I_s(x', y') \leftarrow \text{bilinearInterpolation}(I_c, x, y)$;
 end
 else
 $I_s(x', y') \leftarrow \text{black}$;
 end
end
Algorithm 1: Algorithm for 2D image stabilization.

IV. 3D STABILIZATION

There are certain limitations associated with 2D stabilization. It does not take advantage of the additional information that is available when 3D tracking is used. Furthermore, 2D stabilization does not consider movement in the third dimension, which means that 3D deformations cannot be described accurately. It is worth noting, however, that there are enhancements of 2D warps that take perspective into account [22]. Rather than trying to use 3D information in a 2D warp, we present a true 3D solution in this section.

A. Proposed Approach

The idea of 3D stabilization is the following: A point $[x', y']^T$ in the reference image is the projection of a point $(X', Y', Z')^T$ on the reference 3D surface. A function

$$h : \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad [X, Y, Z]^T = h(X', Y', Z')$$

then describes a mapping of $[X', Y', Z']^T$ on the reference 3D surface to $[X, Y, Z]^T$ on the current 3D surface. This point $[X, Y, Z]^T$ is projected to $[x, y]^T$ in the current image. Let

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \in \mathbb{R}^{3 \times 4}$$

be the projection matrix of the camera for homogeneous coordinates and $p : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ with

$$p(X, Y, Z) = \frac{\begin{pmatrix} p_{11}X + p_{12}Y + p_{13}Z + p_{14} \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} \end{pmatrix}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

the projection function for non-homogeneous coordinates. For reasons of simplicity, we assume that the lens distortion of the cameras is negligible, so projection can be described by a projection matrix in homogeneous coordinates. This is not a restriction of the described method, which can easily be applied to cameras with lens distortion (like endoscopes) as well. Based on $p(\cdot, \cdot, \cdot)$, the following equations describe the transformation between 2D coordinates $[x, y]^T$ and $[x', y']^T$

$$[x, y]^T = p(X, Y, Z) , \quad (1)$$

$$[X, Y, Z]^T = h(X', Y', Z') , \quad (2)$$

$$[x', y']^T = p(X', Y', Z') . \quad (3)$$

Equation (1) is easy to calculate because p can be deduced from \mathbf{P} , which can be determined with a camera calibration algorithm like [24], and $[X, Y, Z]^T$ can be obtained from (2). Equations (2) and (3) can be solved as discussed in Sec. IV-B and Sec. IV-C.

B. Interpolating the Displacement

The function $h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ in (2) can be obtained similarly to f in the 2D case. If point correspondences

$$\{X_i, Y_i, Z_i, X'_i, Y'_i, Z'_i \mid 1 \leq i \leq m\}$$

are known, h can be calculated as an interpolating function, e.g., as an RBF. If an interpolation method defined on \mathbb{R}^2 is to be used (piecewise linear interpolation, multi-level B-Splines), h can also be defined as a function $h : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ that does not depend on Z' .

Alternatively, the strain h at every point of the surface can be derived from the physical model used for tracking [7]. As the physical model already offers a description of the 3D displacement at each surface point, it is not necessary to interpolate the displacement. The physical model can be adaptively refined in areas where the stabilization error is large. The improved adaptive version of model based stabilization has been published and evaluated in [8].

C. Reconstructing the 3D Surface

In order to calculate the transformation for arbitrary pixels in the current image, it is necessary to reconstruct the 3D surface. Reconstruction of the surface is required even if the physical model from [15] is used, because the physical model only describes the displacement of the surface, not the surface itself.

It is assumed that the surface can be represented by a function

$$s : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad Z' = s(X', Y')$$

in the chosen coordinate system. In the considered application, this assumption is always fulfilled. Even though the heart surface may slightly change during the course of an operation, the surface is always representable by a function, since the coronary artery bypass graft procedure does not involve any deep cuts into the heart surface.

There are some known points $\{X'_i, Y'_i, Z'_i \mid 1 \leq i \leq m\}$ on the surface, so we can use any type of function suitable for $\mathbb{R}^2 \rightarrow \mathbb{R}$ interpolation to obtain the function s that describes the surface.

D. Implementation

Once the surface is known, (3) can be solved. For a pixel with coordinates $(x', y')^T$ it is possible to calculate the line of all 3D points that are projected to this pixel and then obtain $(X', Y', Z')^T$ as the intersection of that line with the surface. However, this can be difficult to accomplish if a complicated function $s(\cdot, \cdot)$ is used to describe the surface. Alternatively, a point cloud of 3D points $(X', Y', s(X', Y'))^T$ can be chosen on the surface and projected into the image according to (1). Points on the surface have to be chosen with sufficient density to cover all image pixels. As long as the reference points do not change, this only needs to be calculated once.

Because $(x', y')^T$ are not integer coordinates in general, once again bilinear interpolation [23] is used when obtaining the color of a pixel. For the same reasons as in the 2D case, we restrict image stabilization to the convex hull of all reference points.

Input: current image I_c , projection function p ,

point correspondences

$$\{(X_i, Y_i, Z_i, X'_i, Y'_i, Z'_i) \mid 1 \leq i \leq m\}$$

Output: stabilized image I_s

determine h such that

$$[X_i, Y_i, Z_i]^T = h(X'_i, Y'_i, Z'_i), 1 \leq i \leq m;$$

determine s such that $Z'_i = s(X'_i, Y'_i), 1 \leq i \leq m;$

$I_s \leftarrow$ black image;

for $[X', Y', s(X', Y')]^T$ on the surface **do**

$$[X, Y, Z]^T \leftarrow h(X', Y', Z');$$

$$[x, y]^T \leftarrow p(X, Y, Z);$$

$$[x', y']^T \leftarrow p(X', Y', Z');$$

if $[x, y]^T$ inside I_c **then**

$$| I_s(x', y') \leftarrow \text{bilinearInterpolation}(I_c, x, y);$$

end

end

Algorithm 2: Algorithm for 3D image stabilization.

E. Applications of 3D Image Stabilization

As presented above, 3D Stabilization uses a 2D image as an input and produces a 2D image as an output. Only the intermediate step, where the interpolation is performed, takes place in 3D. However, this is just a special case of what 3D stabilization is capable of.

In some cases it might be desirable to produce a 3D view of the stabilized image. For example, a surgeon could use a

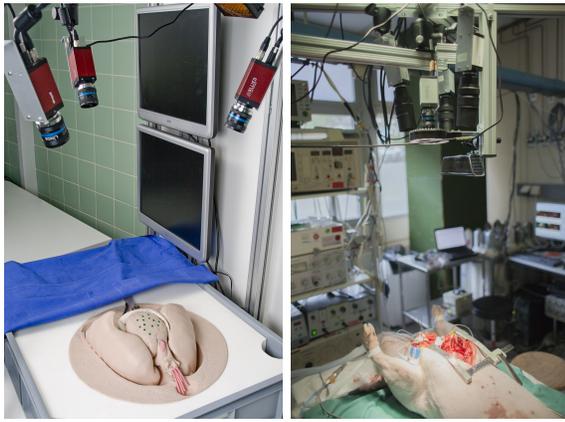


Fig. 4. Heart phantom and porcine heart with camera system.

3D monitor or wear 3D glasses in order to see a stabilized 3D image of the beating heart and take advantage of the additional depth information. This can be easily achieved by projecting the 3D information into two separate images with different projection matrices belonging to virtual cameras. By varying the offset between the virtual cameras, the disparity can be changed and the perception of depth adjusted. Because the heart has a fairly flat surface, its 3D structure can be emphasized by scaling the coordinates along the axis perpendicular to the surface.

Altering the projection matrices also allows creating images from a different perspective. Instead of projecting them to a 2D image, surface points can also be exported as a point cloud or mesh. Because the function describing the 3D surface is continuous, a point cloud or a mesh of arbitrary density can be calculated.

V. EVALUATION

The proposed algorithms have been evaluated both ex-vivo on a heart phantom and in-vivo on a porcine heart. For both experiments, a trinocular stereo camera system (see Fig. 4) consisting of three Pike F-210C cameras [25], each with a resolution of 1920×1080 pixels, has been used. Trinocular stereo camera systems provide slightly better accuracy than binocular stereo camera systems [15], but the discussed algorithms could be applied to a binocular camera system or endoscope as well. Images of the heart phantom and the porcine heart obtained by the camera system are depicted in Fig. 1. The evaluation is based on the same data set as our previous research published in [26]. For the locally supported RBF, the parameter $\sigma = 95$ was chosen and for the Gaussian RBF, the parameter $\sigma = 100$ was used.

The experimental setup for the ex-vivo experiment consists of a pressure-regulated artificial beating heart and the trinocular camera system, which is located approximately 50 cm above the heart. There are 16 large and seven small markers on the surface. In our experiments, only the large markers are considered. A pressure signal with amplitude 100 hPa and frequency 0.7 Hz is used to cause the motion of the artificial heart. For evaluation, an image sequence consisting of 400 frames with a frame rate of 23 fps was recorded. Although this frame rate may not seem very high, it is completely sufficient

to provide a stabilized image that is perceived as smooth by the surgeon. If high-speed cameras are available, the same techniques could be applied to images at a higher frame rate.

The in-vivo experiment was performed at University Hospital Heidelberg. We tried to make the experiment as realistic as possible. To achieve this, the operation setup was not altered except for placing artificial landmarks on the beating heart of a pig. A porcine heart closely resembles the human heart. A trinocular camera system was used to record a sequence of 337 frames at a frame rate of 31 fps. To measure the pressure inside the left ventricle, a cardiac catheter was used. The heart was mechanically stabilized with the commercially available Octopus stabilizer [27], which is commonly used in off-pump surgery on humans as well. A total of 14 markers were placed on the surface. Since the motion of the real heart is affected by breathing and by the motion of all four heart chambers, the physical model was extended with an excitation based on Fourier series.

To analyze the residual motion in the stabilized images, the average difference across all k frames I_1, \dots, I_k to the reference frame $I_{\text{reference}}$ was calculated for each pixel (x, y) and each color channel $c \in \{R, G, B\}$ according to

$$I_{\text{error}}(x, y, c) = \frac{1}{k} \sum_{t=1}^k |I_t(x, y, c) - I_{\text{reference}}(x, y, c)|.$$

For the purpose of this evaluation, only points inside the convex hull of all landmarks are taken into account. The image of average differences I_{error} was converted to grayscale in the range $[0, 1]$ and visualized as a contour plot (Fig. 2 and 3).

The average error e across the entire image summed over all color channels can be obtained by

$$e = \sum_{c \in \{R, G, B\}} \left(\frac{1}{p} \sum_x \sum_y I_{\text{error}}(x, y, c) \right),$$

where p denotes the number of pixels inside the convex hull of all landmarks (see Table II). This evaluation method is based on the assumption that the heart surface does not visibly change during the evaluation. We experimented on sequences that contain occlusions by instruments or blood with promising results, but because of the limitation of our evaluation method, we do not provide a quantitative evaluation of these experiments.

	method	ex-vivo	in-vivo
	unstabilized	0.186	0.133
2D	affine	0.054	0.088
2D	B-Spline	0.037	0.076
2D	piecewise linear	0.039	0.076
2D	RBF (locally supp.)	0.040	0.083
2D	RBF (TPS)	0.038	0.077
2D	RBF (Gaussian)	0.037	0.079
3D	B-Spline	0.042	0.078

TABLE II. AVERAGE ERROR IN THE STABILIZED IMAGES.

As can be seen from the results in Table II, all of the suggested 2D image stabilization methods significantly reduce motion in the image. The affine transformation is clearly

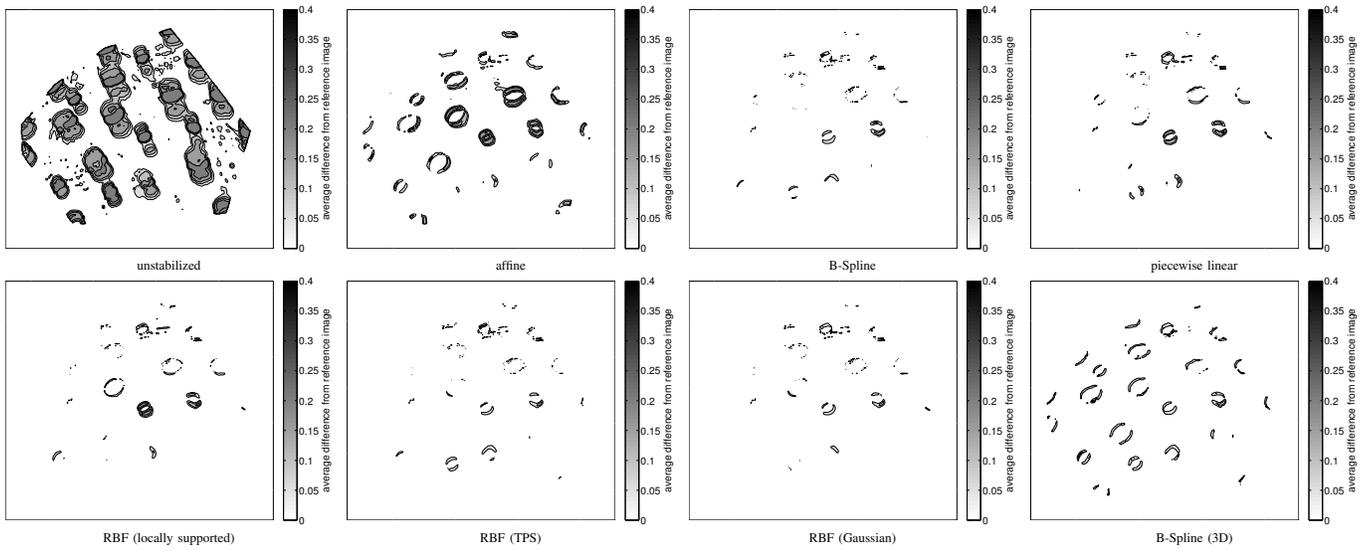


Fig. 2. Average difference to reference frame for ex-vivo experiments.

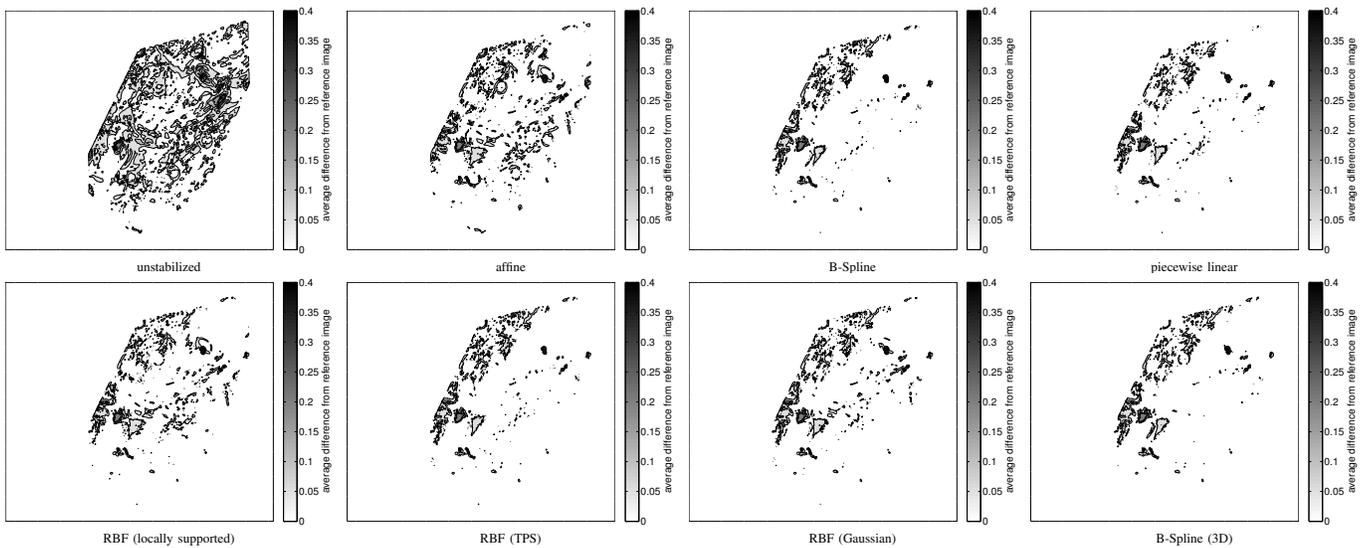


Fig. 3. Average difference to reference frame for in-vivo experiments.

inferior to the other methods, which was to be expected since it does not allow for local differences in the deformation. All other presented methods produce fairly similar results, so the choice of the interpolation function does not seem to have a large impact on the quality of the image stabilization. Our results suggest that the 2D multi-level B-Spline approach is a good choice since it performs well on both in-vivo and ex-vivo experiments.

We also compared the computation time of the different algorithms. All of the 2D algorithms were fast enough for real-time applications. Our non-optimized implementation of the 3D algorithm is currently not fast enough. However, the presented algorithms are easily parallelizable since every pixel is computed independently. Thus, an efficient GPU implementation would certainly allow the 3D algorithm to run in real-time as well.

In our experiments, stabilization in 3D seems to be slightly worse than 2D stabilization, but still produces very good

results. This effect is explained by the fact that 3D stabilization may be affected by additional errors that are introduced as a result of 3D reconstruction (e.g., imprecise camera calibration). However, 3D stabilization offers advantages such as the use of stereoscopic displays that are not possible with 2D stabilization. Depending on the application, these additional possibilities may be more important than a slight improvement in stabilization quality.

VI. CONCLUSIONS

Our experiments show that the presented algorithms can produce a stabilized view of a beating heart with very little residual motion. We show that the non-uniform deformations of the heart cannot be handled well by a simple affine transformation and require the use of an interpolation technique that takes local differences in the transformation into account. While the impact of the chosen interpolation function seems

to be fairly small, 2D stabilization with multi-level B-Splines produced the best results in our experiments.

Although 3D stabilization does not outperform 2D stabilization in terms of stabilization quality in our experiments, it provides a good stabilization quality and offers some additional possibilities such as the use of stereoscopic displays. Consequently, it depends on the intended application whether 2D or 3D stabilization is to be preferred.

All presented stabilization methods rely on point correspondences obtained by precise and robust tracking of landmarks. We show that tracking based on a physical model can be used to reliably obtain these point correspondences both ex-vivo on a heart phantom and in a more demanding in-vivo environment.

ACKNOWLEDGMENT

This work was partially supported by the German Research Foundation (DFG) within the Research Training Group RTG 1126 “*Intelligent Surgery - Development of new computer-based methods for the future working environment in visceral surgery*”.

We thank Evgeniya Ballmann for her work on physics-based motion compensation. Furthermore we thank Szabolcs Páli and Gábor Szabó for their contributions to the in-vivo experiment.

REFERENCES

- [1] Y. Nakamura, K. Kishi, and H. Kawakami, “Heartbeat Synchronization for Robotic Cardiac Surgery,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea, May 2001, pp. 2014–2019.
- [2] W. B. Keeling, M. L. Williams, M. S. Slaughter, Y. Zhao, and J. D. Puskas, “Off-pump and on-pump coronary revascularization in patients with low ejection fraction: a report from the society of thoracic surgeons national database.” *The Annals of Thoracic Surgery*, vol. 96, no. 1, pp. 83–89, July 2013.
- [3] T. J. Gilhuly, S. E. Salcudean, and S. V. Lichtenstein, “Evaluating Optical Stabilization of the Beating Heart,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 4, pp. 133–140, 2003.
- [4] D. Stoyanov and G. Yang, “Stabilization of Image Motion for Robotic Assisted Beating Heart Surgery,” *Medical Image Computing and Computer-Assisted Intervention*, vol. 10, pp. 417–424, 2007.
- [5] M. Gröger and G. Hirzinger, “Image Stabilisation of the Beating Heart by Local Linear Interpolation,” in *Medical Imaging: Visualization, Image-Guided Procedures and Display*, 2006.
- [6] R. Richa, P. Poignet, and C. Liu, “Three-dimensional Motion Tracking for Beating Heart Surgery Using a Thin-plate Spline Deformable Model,” *International Journal of Robotics Research*, vol. 20, no. 2-3, pp. 218–230, Feb. 2010.
- [7] E. Bogatyrenko and U. D. Hanebeck, “Visual Stabilization of Beating Heart Motion by Model-based Transformation of Image Sequences,” in *Proceedings of the 2011 American Control Conference (ACC 2011)*, San Francisco, California, USA, Jun. 2011.
- [8] —, “Adaptive Model-Based Visual Stabilization of Image Sequences Using Feedback,” in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.
- [9] W. W. Lau, N. A. Ramey, J. J. Corso, N. V. Thakor, and G. D. Hager, “Stereo-Based Endoscopic Tracking of Cardiac Surface Deformation,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2004)*, 2004, pp. 494–501.
- [10] A. Noce, J. Triboulet, and P. Poignet, “Efficient Tracking of the Heart Using Texture,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, Lyon, France, Aug. 2007, pp. 4480–4483.
- [11] G. Kurz, M. Baum, and U. D. Hanebeck, “Real-time Kernel-based Multiple Target Tracking for Robotic Beating Heart Surgery (to appear),” in *Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2014)*, A Coruña, Spain, Jun. 2014.
- [12] G. Kurz and U. D. Hanebeck, “Recursive Fusion of Noisy Depth and Position Measurements for Surface Reconstruction,” in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.
- [13] R. Richa, A. P. L. Bó, and P. Poignet, “Beating Heart Motion Prediction for Robust Visual Tracking,” in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4579–4584.
- [14] T. Ortmaier, M. Groeger, D. H. Boehm, V. Falk, and G. Hirzinger, “Motion Estimation in Beating Heart Surgery,” *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 10, pp. 1729–1740, Oct. 2005.
- [15] E. Bogatyrenko, P. Pompey, and U. D. Hanebeck, “Efficient Physics-Based Tracking of Heart Surface Motion for Beating Heart Surgery Robotic Systems,” *International Journal of Computer Assisted Radiology and Surgery (IJCARS 2010)*, vol. 6, no. 3, pp. 387–399, Aug. 2010, doi:10.1007/s11548-010-0517-5. [Online]. Available: <http://dx.doi.org/10.1007/s11548-010-0517-5>
- [16] Y. Yitzhaky, R. Milberg, S. Yohaev, and N. S. Kopeika, “Comparison of Direct Blind Deconvolution Methods for Motion-Blurred Images,” *Applied Optics*, vol. 38, no. 20, pp. 4325–4332, Jul. 1999.
- [17] C. A. Glasbey and K. V. Mardia, “A Review of Image Warping Methods,” *Journal of Applied Statistics*, vol. 25, pp. 155–171, 1998.
- [18] I. Amidror, “Scattered Data Interpolation Methods for Electronic Imaging Systems: A Survey,” *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 157–176, Apr. 2002.
- [19] M. Holden, “A Review of Geometric Transformations for Nonrigid Body Registration,” *IEEE Transactions on Medical Imaging*, vol. 27, no. 1, pp. 111–128, 2008.
- [20] S. Lee, G. Wolberg, and S. Y. Shin, “Scattered data interpolation with multilevel B-splines,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, 1997.
- [21] M. Hansen, R. Larsen, B. Glocker, and N. Navab, “Adaptive parametrization of multivariate B-splines for image registration,” in *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*, 2008.
- [22] A. Bartoli, M. Perriollat, and S. Chambon, “Generalized Thin-Plate Spline Warps,” *International Journal of Computer Vision*, vol. 88, no. 1, pp. 85–110, 2010.
- [23] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.
- [24] T. Svoboda, D. Martinec, and T. Pajdla, “A Convenient Multi-Camera Self-Calibration for Virtual Environments,” *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.
- [25] *PIKE Technical Manual V5.1.2*, Allied Vision Technologies GmbH, Aug. 2012.
- [26] G. Kurz and U. D. Hanebeck, “Image Stabilization with Model-Based Tracking for Beating Heart Surgery,” in *11. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC12)*, Düsseldorf, Germany, Nov. 2012.
- [27] C. Detter, T. Deuse, F. Christ, D. H. Boehm, H. Reichenspurner, and B. Reichart, “Comparison of Two Stabilizer Concepts for Off-Pump Coronary Artery Bypass Grafting,” *The Annals of Thoracic Surgery*, vol. 74, no. 2, pp. 497–501, 2002.