

Non-Linear Beam Model for Tracking Large Deformations

Slobodan Ilić
Deutsche Telekom Laboratories
TUB, Germany
Slobodan.Ilic@tu-berlin.de

Pascal Fua*
Computer Vision Laboratory
EPFL, Switzerland
Pascal.Fua@epfl.ch

Abstract

In this paper we investigate physics-based plane beam model, frequently used in mechanical and civil engineering, to track large non-linear deformations in images. Such models do not only contribute to robust and precise tracking, in the presence of clutter and partial occlusions, but also allow to compute the forces that produce observed deformations. We verify the correctness of the recovered forces by using them in a simulation and compare the results to the original image displacements. We apply this method to track deformations of the pole vault, the rat whiskers and the car antenna.

1. Introduction

Physics-based models have demonstrated their effectiveness in the computer vision problems since their introduction twenty years ago [7, 2, 16, 3]. However, they typically rely on simplifying assumptions to yield easy to minimize energy functions and ignore the complex non-linearities that are inherent to large deformations. In this paper, we use the case of beam structures to demonstrate that replacing the simplified models by a fully non-linear physics-based one, which approximates the physical behavior more closely, yields two benefits:

- We can track large deformations more robustly in the presence of clutter and partial occlusions.
- As a byproduct of the computation, we obtain an estimate of the forces that were applied to the beam to produce the observed deformations.

The latter is novel because the so-called *image forces* that are used by traditional snake-like models do not correspond to the physical forces that are applied to the material. The

forces we extract are, at the moment of equilibrium, opposite to the model internal forces. The model internal forces are the forces that balance and react to external loads applied to a body. It is also of great interest to engineers, who want to understand the loads that the objects undergo, to compare simulated behavior against reality and to perfect their design.

More specifically, we developed a 2D model-based tracking algorithm based on Timoshenko's beam theory of structural mechanics [17], which accounts for large displacements and rotations and is widely accepted as physically valid. Because, we work with images, we use the *planar beam* theory to describe our model. The planar beam deforms in 3D in some plane under the influence of forces applied in that plane. As shown in Fig. 1, our algorithm can effectively handle very large displacements and estimate the forces being applied to the model to produce them. Our algorithm is guided by the image forces. Since the image forces are proportional to the square distance of the model to the image data, they are usually not sufficient to deform the model of known material properties, so that it fits the data immediately. The image forces only move the model toward the image observations. To fit the model to the data we repeat Newton minimization in several steps. We stop when the distance of the model to the data of two consecutive Newton minimizations is smaller than some given precision.

In effect, we are performing an *inverse simulation*: we observe the model deformation using monocular uncalibrated camera and recover the beam deformation in the image plane. Using physically valid beam model and computed image displacements we can compute the internal forces which body uses to resist the external loads. Taking the negative of these forces we can reproduce the observed displacement. Fig. 1 depicts these forces at nodes of the beam. These forces, since computed using 2D image displacements, are not those applied to the model in 3D. However, since 2D and 3D displacements are related by a projection, the forces we recover are also related to the forces applied to the model in 3D by a projection. For that reason

*The rat whiskers images shown in the paper were obtained at Harvard University's School of Engineering and Applied Sciences by Robert A. Jenks.

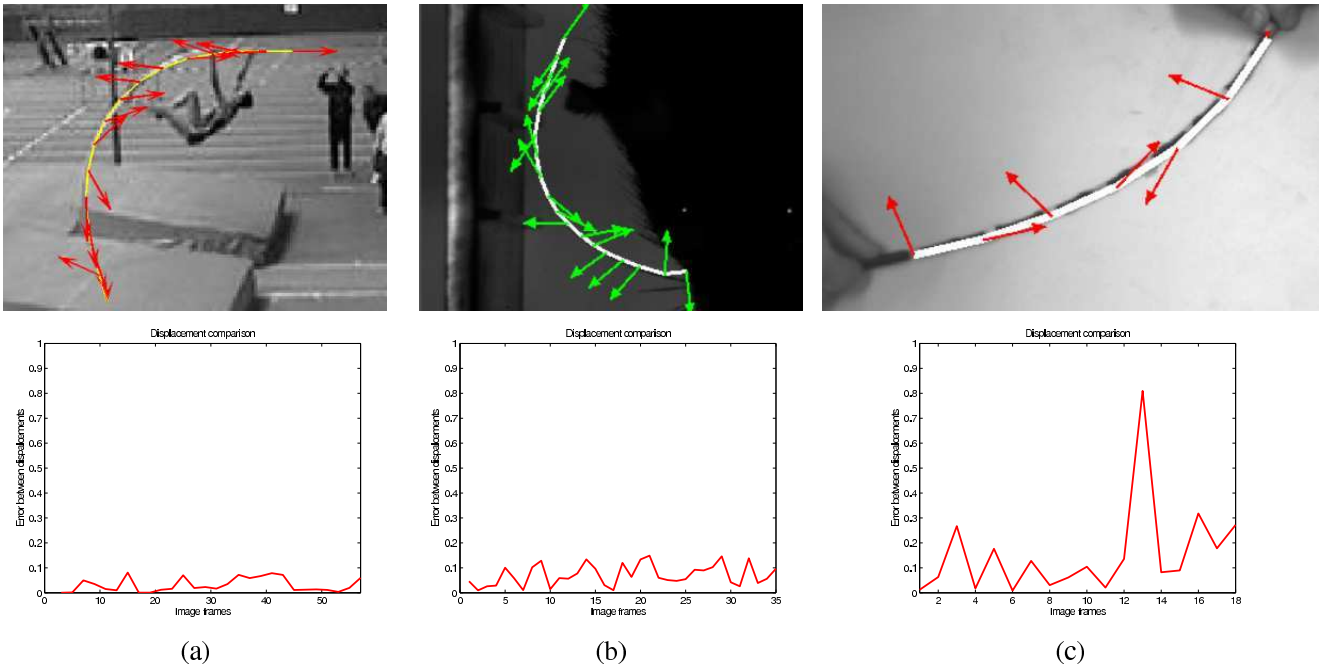


Figure 1. Examples of recovering beam like object deformations.(a) Pole vault. (b) Rat whiskers tracked while touching the walls of the maze. (c) Car antenna deformation. Top row: Magnified frames from three different tracking sequences with recovered model shown in yellow/white and the force resultants of the axial and transversal forces shown as red/green arrows. They indicate only directions but no intensity. Bottom row: Corresponding mean square error between displacement vectors obtained from fitting the models to the images and the simulation where extracted forces were applied. Note the sequences of (a) and (b) are relatively small resolution while the sequence (c) is high resolution, thus produces higher error in pixels.

we call the recovered forces *projective forces* and they are consistent with the forces in 3D up to the scale factor. We outline the proof relating projective and the real forces.

To verify that the recovered forces are correct we perform the direct FEM simulation with them. We let the forces act on the model and compute the displacement. We compare this displacement and the one obtained by our fitting algorithm. The error between those displacements is, in worst case, smaller than one pixel as shown in the bottom row of Fig. 1.

In the remainder of the paper we give a brief overview of the known physic-based techniques, then introduce nonlinear Timoshenko planar beam model, describe our algorithm, present the results and in the appendix develop additional TB equations.

2. Related work

Recovering model deformations from images requires deformable models to constrain the search space and make the problem tractable. In recent decade deformable models were exploited in Computer Graphics [6], Medical Imaging [10] and Computer Vision [11]. There are several important categories of physically based models: mass-spring models [8], finite element models (FEM) [5, 15], snake like models [7, 16, 3] and models obtained from FEM, to reduce

number of dofs, by modal analysis [14, 2, 4, 12].

In this paper we are particularly interested in physically based models, especially those based on FEM. FEM are known to be precise and to produce physically valid deformations. However, because of their mathematical complexity the FEM were mainly developed for small linear deformations [13] where the model stiffness matrix is constant. In case of large deformations the stiffness matrix and the applied forces become the function of the displacement. Such non-linear FEM were used by [18] to recover material parameters from images of highly elastic shell like models. The model deformation was measured from 3D model scan and then given to the Finite Element Analysis (FEA) software. Then FEA software does inverse FEM. The simulation was continuously repeated with changed material elasticity until the measured displacement was obtained. In this paper we developed nonlinear FEM model for large geometric deformations. We used such model to recover deformations of beam structures from images automatically through optimization. Then, thanks to the physically valid mathematical model based on nonlinear FEM, we computed forces which caused this deformation.

In computer vision snake like models, based on the continuous energy function were extensively used. The original ones [7] were 2D and have been shown to be effective for 2D deformable surface registration [1]. They were soon

adapted for 3D surface modeling purposes by using deformable superquadrics [16, 3], triangulated surfaces [2], or thin-plate splines [10]. In this framework, modeling generic structures often requires many degrees of freedom that are coupled by regularization terms. In practice, this coupling implicitly reduces the number of degrees of freedom, which makes these models robust to noise and is one of the reasons for their immense popularity. This reduction can also be explicitly achieved via modal analysis [14, 2, 4]. We compare our true physics-based beam model to 2D elastic beam represented as polygonal line whose internal energy was continuous and resembles the one of the classical snakes.

3. Plane beam model

The beam represents the most common structural component in civil and mechanical engineering. A beam is a rod-like—meaning that one dimension is considerably larger than the other two—structural element that resists transversal loading applied between its supports. Generic beams can be used as isolated structures or combined to form *frame* structures that are common in buildings and can be subjected to a wide variety of loads in arbitrary directions. Because, we work in the 2D image plane, we will use the simpler *planar beams* that deforms in one 3D plane under the influence of forces applied in that plane.

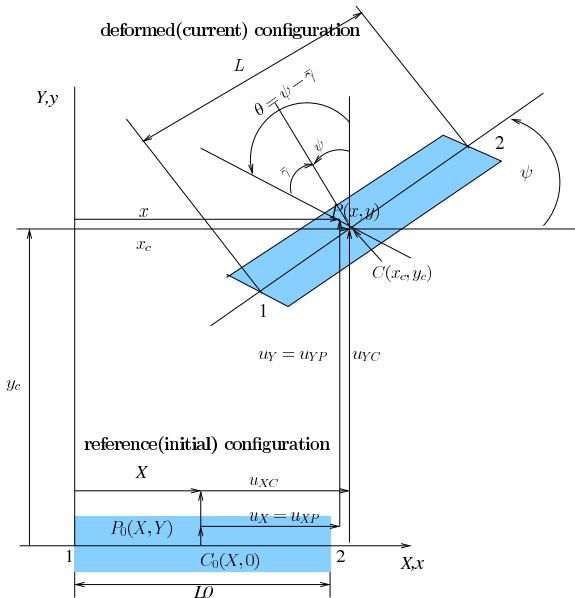


Figure 2. Plane Timoshenko beam kinematics notation.

3.1. Mathematical model description

Euler-Bernoulli beam (EB) theory is widely used for modeling small deformations. Timoshenko beam (TB) theory [17] extended EB equations to account for nonlinear

effect such as shear. Both theories rely on the description of the cross-section behavior expressed in terms of quantities measured at the longitudinal axis. More precisely, plane beam element kinematics is defined by 3 dof per node being the axial displacement $u_X(X)$, the transverse displacement $u_Y(X)$ and the cross-section rotation $\theta(X)$, where X is longitudinal coordinate in the reference configuration as shown in Fig 2. The EB theory assumes that the cross-section remains normal to the deformed longitudinal axis, while the TB removes normality constraint by introducing shear deformation. In addition, both theories neglect changes in dimensions of the cross sections as the beam deforms. TB theory can account for geometrically nonlinear behavior due to large displacements and rotations. Though more complex TB appear to be more efficient in the context of FEM computation.

The beam is divided in a number of segments (finite elements). The beam elements are straight and have two nodes as shown in Fig. 2. We collect all nodal dofs into the system vector of dofs which we name a state vector:

$$\mathbf{u} = [u_{X1} \quad u_{Y1} \quad \theta_1 \quad \dots \quad u_{Xn} \quad u_{Yn} \quad \theta_n]^T. \quad (1)$$

We assume to know model material properties such as E Young modulus of elasticity, and G is shear modulus. The material remains linearly elastic. The forces acting on the beam nodes are collected to form the force vector which we also recover:

$$\mathbf{f} = [f_{X1} \quad f_{Y1} \quad f_{\theta_1} \quad \dots \quad f_{Xn} \quad f_{Yn} \quad f_{\theta_n}]^T. \quad (2)$$

with n being the total number of nodes of the beam model.

The strain is a measure of the change of the object shape, in this case the length, before and after the deformation caused by some applied load. The stress is the internal distribution of force per unit area that balances and reacts to external loads applied to a body. In the case of the TB we have three different stress components per beam element: e axial strain measuring the beam relative extension, γ shear strain measuring the relative angular change between any two lines in a body before and after the deformation and κ measuring the curvature change. We can collect them into a generalized beam strain vector:

$$\mathbf{h}^T = [e_1 \quad \gamma_1 \quad \kappa_1 \quad \dots \quad e_{n-1} \quad \gamma_{n-1} \quad \kappa_{n-1}]. \quad (3)$$

The stress resultants in the TB define the axial force N , transverse force V and bending moment M per unit area of the cross-section. They are collected into the generalized stress-resultant:

$$\mathbf{z} = [N_1 \quad V_1 \quad M_1 \quad \dots \quad N_{n-1} \quad V_{n-1} \quad M_{n-1}] \quad (4)$$

where $n - 1$ corresponds to the number of the beam elements.

The internal model strain energy along the beam can be expressed as the length integral:

$$U = \int_L \mathbf{z}^T \mathbf{h} dX \quad (5)$$

where L is the beam length. The internal force vector can be obtained by taking the first variation of the strain energy with respect to the nodal displacement:

$$\mathbf{p} = \frac{\partial U}{\partial \mathbf{u}} = \int_L \mathbf{B}^T(\mathbf{u}) \mathbf{z} dX. \quad (6)$$

This expression we evaluate by one point Gauss integration. \mathbf{B} is strain-displacement matrix derived in Appendix. Finally, the first variation of the internal force defines the tangent stiffness matrix:

$$\mathbf{K}_T = \frac{\partial \mathbf{p}}{\partial \mathbf{u}} = \int_L (\mathbf{B}^T \frac{\partial \mathbf{z}}{\partial \mathbf{u}} + \frac{\partial \mathbf{B}}{\partial \mathbf{u}} \mathbf{z}) dX = (\mathbf{K}_M + \mathbf{K}_G). \quad (7)$$

where \mathbf{K}_M is material stiffness and \mathbf{K}_G is geometric stiffness. The material stiffness is constant and identical to the linear stiffness matrix of the C^1 Euler-Bernoulli beam. The geometric stiffness comes the variation of \mathbf{B} while stress resultants are kept fixed and carries the beam nonlinearity responsible for large geometric deformations. More detailed derivation of the TB element quantities is given in the Appendix.

4. Model fitting

The general approach in mechanical simulations is that some external load \mathbf{f} is applied to the model and the displacement \mathbf{u} is computed. This can be done through energy minimization, where the total potential energy Π of the system is computed as the sum of the system internal strain energy U and the energy caused by the external load P . The minimum of the energy in respect to the displacement \mathbf{u} can be found by derivation:

$$\frac{\partial \Pi}{\partial \mathbf{u}} = \frac{\partial U}{\partial \mathbf{u}} + \frac{\partial P}{\partial \mathbf{u}} \Rightarrow \mathbf{r}(\mathbf{u}) = \mathbf{p}(\mathbf{u}) + \mathbf{f} = \mathbf{0} \quad (8)$$

The $\mathbf{r}(\mathbf{u})$ is a force residual, $\mathbf{p}(\mathbf{u})$ is internal force of Eq. 6 and \mathbf{f} is the external load. This is a nonlinear system of equations and is usually solved by using Newton-Raphson method. It is incremental method and at each iteration we solve for the displacement update $d\mathbf{u}$ by solving linear system $\mathbf{K}_T d\mathbf{u} = -(\mathbf{f} + \mathbf{p}(\mathbf{u}))$.

In classical mechanical simulation the external forces \mathbf{f} are a priori given. In our case we do not know them. Here, we propose a method to compute the model displacements \mathbf{u} from the images and the forces \mathbf{f} that caused the deformation. Solving for both simultaneously is an ill-posed problem, so we find the solution in two steps. First we compute the displacements by fitting the model to the images.

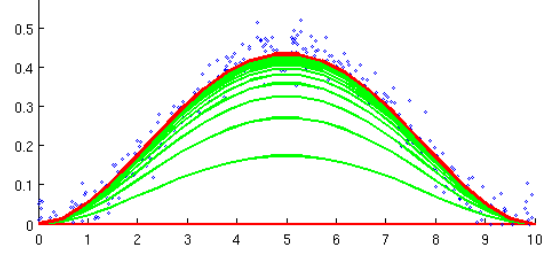


Figure 3. Synthetic example of fitting the plane beam, initially aligned along the x-axis, to the synthetic image data shown as blue circles. The intermediate steps shown in green are the output of number of repeated Newton optimizations driven by the image forces.

Then we compute the internal force for that displacement in respect to the reference configuration. In the equilibrium the force residual is zero and the external forces become $\mathbf{p}(\mathbf{u}) = -\mathbf{f}$, as it can be derived from Eq. 8.

To compute model displacements we also solve for Eq. 8, using *image forces*. We create a vector of image observations $\mathbf{F}(\mathbf{u}) = [d_1(\mathbf{u}) \ d_2(\mathbf{u}) \ \dots \ d_N(\mathbf{u})]^T$ where $d_i(\mathbf{u})$ are distances of the image observations from the beam segments. We use the edges in the image, obtained using Canny edge detector, to be our observations. In practice we sample every beam segment and then search for the multiple observations in the direction of the beam normal. The external image energy becomes $P_I = \frac{1}{2} \mathbf{F}^T(\mathbf{u}) \mathbf{F}(\mathbf{u})$. The image forces are obtained as a derivative of the energy in respect to the displacement $\mathbf{f}_I = \nabla \mathbf{F}^T(\mathbf{u}) \mathbf{F}(\mathbf{u})$. The force residual of Eq. 8 becomes: $\mathbf{r}(\mathbf{u}) = \mathbf{p}(\mathbf{u}) + \mathbf{f}_I(\mathbf{u}) = \mathbf{0}$. We derive the displacement increment of Newton-Raphson method by developing the residual in a Taylor series around the current displacement \mathbf{u} as follows:

$$\begin{aligned} \mathbf{r}(\mathbf{u} + d\mathbf{u}) &= \mathbf{p}(\mathbf{u} + d\mathbf{u}) + \mathbf{f}_I(\mathbf{u} + d\mathbf{u}) = \mathbf{0} \\ \nabla \mathbf{F}^T \mathbf{F} + (\nabla^2 \mathbf{F}^T \mathbf{F} + \nabla \mathbf{F}^T \nabla \mathbf{F}) d\mathbf{u} + \mathbf{p}(\mathbf{u}) + \mathbf{K}_T d\mathbf{u} &= \mathbf{0} \\ (\mathbf{K}_T + \nabla^2 \mathbf{F}^T \mathbf{F} + \nabla \mathbf{F}^T \nabla \mathbf{F}) d\mathbf{u} &= -(\nabla \mathbf{F}^T \mathbf{F} + \mathbf{p}(\mathbf{u})) \\ &\approx (\mathbf{K}_T + \nabla \mathbf{F}^T \nabla \mathbf{F}) d\mathbf{u} = -(\nabla \mathbf{F}^T \mathbf{F} + \mathbf{p}(\mathbf{u})) \end{aligned} \quad (9)$$

We neglect the second order term $\nabla^2 \mathbf{F}^T \mathbf{F}$ of Eq. 9. To make it more robust we use Tukey robust estimator ρ of [9]. This is simply done by weighting the residuals $d_i(\mathbf{u})$ of the image observation vector $\mathbf{F}(\mathbf{u})$ at each Newton iteration: Each $d_i(\mathbf{u})$ is replaced by $d_i^w(\mathbf{u}) = w_i d_i(\mathbf{u})$ such that: $(d_i^w)^2(\mathbf{u}) = (w_i)^2 d_i^2(\mathbf{u})$, therefore the weight is chosen to be: $w_i = \rho(d_i(\mathbf{u}))^{1/2} / d_i(\mathbf{u})$. We then create a weighting matrix $\mathbf{W} = \text{diag}(\dots, w_i, \dots)$. We then solve in each Newton step:

$$(\mathbf{K}_T + \nabla \mathbf{F}^T \mathbf{W} \nabla \mathbf{F}) d\mathbf{u} = -(\nabla \mathbf{F}^T \mathbf{W} \mathbf{F} + \mathbf{p}(\mathbf{u})) \quad (10)$$

By solving the Eq. 9 we compute the displacement caused by the image forces. Since the image forces are propor-

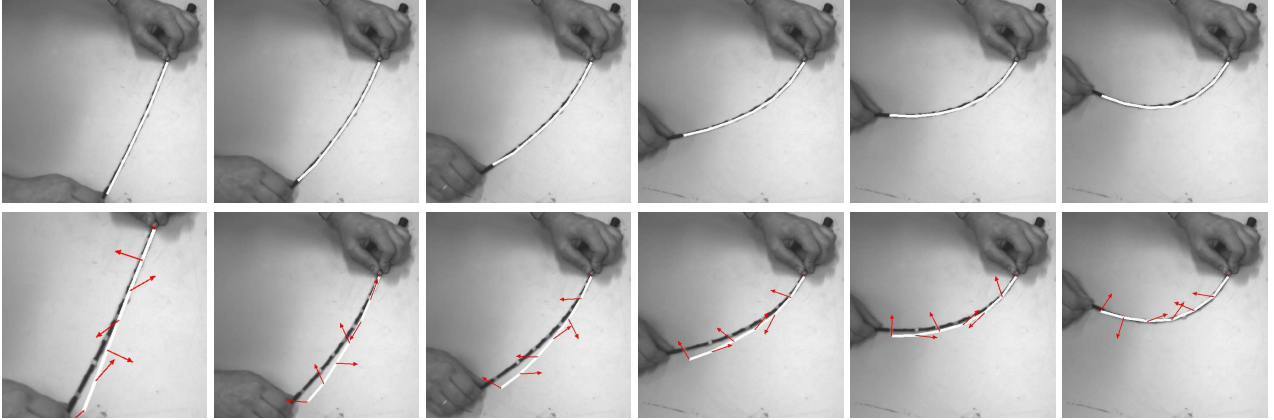


Figure 4. The deformation of the car antenna. Top row: Selected frames from the tracking sequence with the recovered model shown in white. Bottom row: The force resultants of the axial and transversal forces are shown as red arrows. In the bottom row the models are shown in the frame prior to fitted position of the top row.

tional to the square distance of the model to the image edge points, they are not sufficient to deform the model so that it fits the data. They only move the model toward the image observations. To obtain the exact fit of the model to the data we repeatedly fit the model to the data performing Newton method in several steps. We stop when the distance of the model to the data of two consecutive Newton minimizations becomes smaller than some given precision. The optimization algorithm is illustrated on the synthetic example of Fig. 3.

We obtain the total displacement \mathbf{u}_T as a sum of all intermediate displacements. The force producing this deformation is computed as $\mathbf{f}_T = -\mathbf{p}(\mathbf{u}_T)$. Note that the forces are computed between consecutive frames or even between selected non-consecutive frames. The forces we compute are proven to produce exact displacements when applied to the model in a simulation. We call those forces *projective* since they are computed from the image displacements. The image displacements are related to the real 3D displacements by the projective transformation $\mathbf{u} = \mathbf{Q}\mathbf{v}$, where \mathbf{Q} is projection matrix, \mathbf{v} displacement in 3D and \mathbf{u} displacement in the image. If we replace this into Eq. 6 we get the *projective* forces to be $\mathbf{f} = -\mathbf{p}$ and \mathbf{p} is related to 3D internal forces \mathbf{p}_{3d} as follows:

$$\mathbf{p} = - \int_L \mathbf{B}^T(\mathbf{Q}\mathbf{v})z d\mathbf{X} \approx \mathbf{Q} \int_L \mathbf{B}^T(\mathbf{v})z d\mathbf{X} = \mathbf{Q}\mathbf{p}_{3d}. \quad (11)$$

The computational complexity of our method is quadratic and corresponds to the complexity of the Newton minimization. It converges fast to the solutions thanks to the good initial guess which relies on the previous position. In practice, the algorithm requires only several repetitions of Newton optimization per image frame.

5. Results

We run our experiments on three different cases: the deformation of a car antenna, pole vault and the deformation of the rat whiskers. In all three cases we recovered the forces applied to it. For the pole vault and rat whiskers, we demonstrate that our approach performs better than the one where snake-like energy function was used. However, this approach appeared to be effective in the case of car antenna, which is not surprising considering relatively easy background.

5.1. Car antenna

In this example we track the car antenna deforming in the plane of the table as shown in Fig. 4. The straight beam model in arbitrary position was drawn on the first frame for the initialization Fig. 4(a). In the top row we show some characteristic frames of the tracking sequence with the recovered model shown in white. In the bottom row the force resultants of the axial and transversal forces are shown as red arrows. They indicate only directions but no intensity. Note that there is also a force moment responsible for bending and cannot be shown.

In practice the beam model is divided into 7 equal segments representing finite elements. There are 8 nodes with 3 dofs per node which is 24 dofs in total. The sequence is 72 frames long and we choose every 5th frame for fitting. Since our model accounts for large deflections we are allowed to use non-consecutive frames.

To verify that the recovered forces, acting on the model to cause deformation between chosen frames, are correct we perform the direct FEM simulation with them. We let the forces act on the model and compute the displacement. We compare this displacement and the one obtained by our fitting algorithm. The error between those displacements for each of 18th images used for tracking is shown in bottom

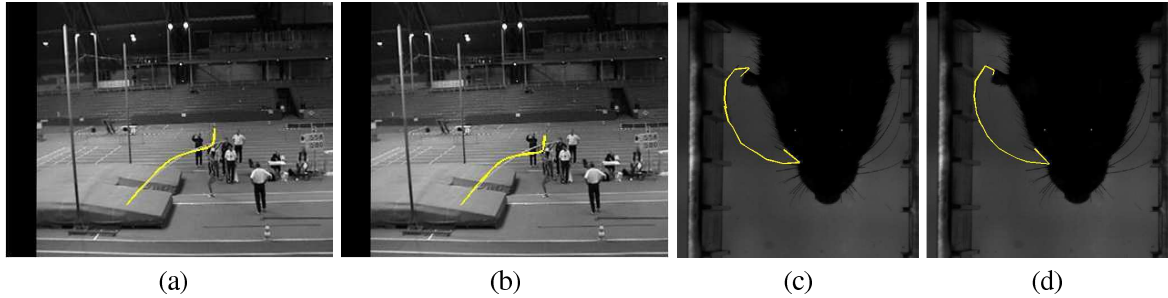


Figure 5. Failure examples using alternative snake-like energy function with different energy weighting coefficients: w_s for stretching, w_b for bending. (a,b) The pole vault fails in 3rd frame with the energy coefficient being $w_b = w_s = 1000$ and $w_b = 1000, w_s = 10$ respectively. (c,d) The rat whisker fails in the 10th frame because of the occluded ear, and the energy coefficients are also $w_b = w_s = 1000$ and $w_b = 1000, w_s = 10$ respectively.

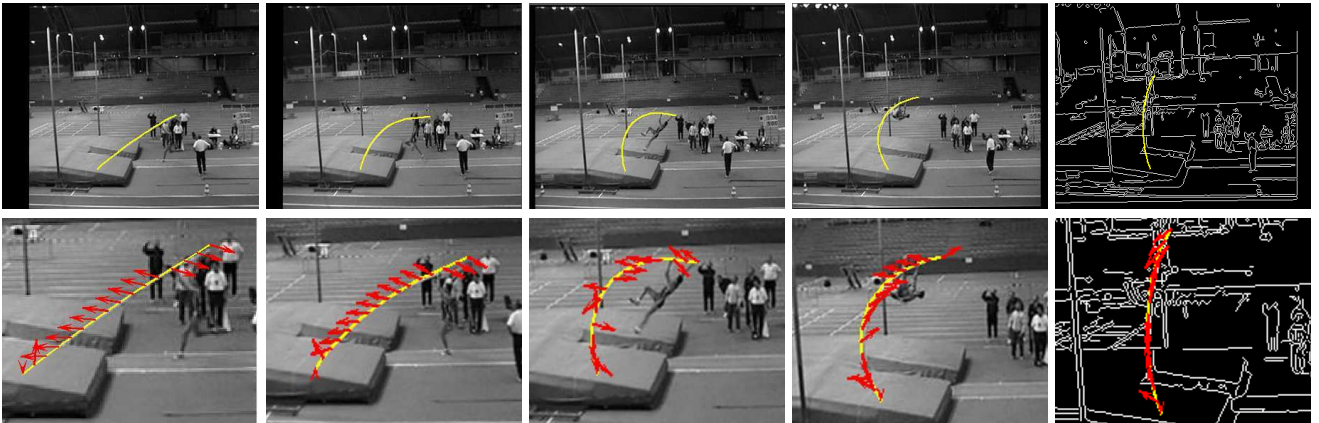


Figure 6. The deformation of the pole in a pole vault. The top row shows selected frames from the tracking sequence with the recovered model shown in yellow. In the bottom row the force resultants of the axial and transversal forces are shown as red arrows in the magnified images. In this case the models are again shown in the frame prior to fitted position of the top row. The provided videos correspond to this description.

row of Fig. 1(c). The image resolution is 1024×768 , so the errors below 1 pixel indicate high tracking precision.

5.2. Pole vault

In a pole vault, the coaches are curious to find out which force are applied to the pole at every moment of the jump. This may indicate why some athletes perform better than the others. The original video is extremely challenging because of the moving camera, cluttered background and partial occlusions. In order to make this example tractable we first pre-process the sequence, thus we warp all the frames to one reference frame using robustly estimated homography. Then we compute the point in the image where the pole is always stuck to the ground by intersecting two straight pole positions of the first and the last frame. In our fitting we impose a boundary condition on axial and transverse displacements of this point. Note that the lower part of the pole is mainly occluded by the mattress landing surface.

We initially built the snake like energy function of the pole. It consists of the stretching energy and the bending energy terms. The total model energy E_D is given as $E_D = w_{bend}E_{bend} + w_{ext}E_{ext}$, where E_{bend} penalizes curvature

and E_{ext} extension. The image energy is identical to P_I the one we defined in Section 4. We equally performed robust weighting of the image observations and fitted the model to the data with different settings of energy weights as can be seen in Fig. 5. We depict the 3rd frame of the video sequence when the tracking actually starts failing.

The pole was divided in 15 segments having in total 48 dofs. Every third frame was used for tracking. In Fig. 6 we show the pole deformation. The images are warped version of the originals using homography. Note that the tracking is robust in spite to the clutter and the partial occlusion. The actual edge image are depicted for the last tracking frame of Fig 6.

5.3. Rat whiskers

An interesting problem appeared in neurology where the behavior of the rats moving in the maze is observed with very high frame rate camera. The rats touch the walls of the maze with their whiskers and depending of the force of the touch certain receptors in the brain are activated. The neurologist want to know what is the intensity of the forces the rats are experiencing with their whiskers. In contract to the

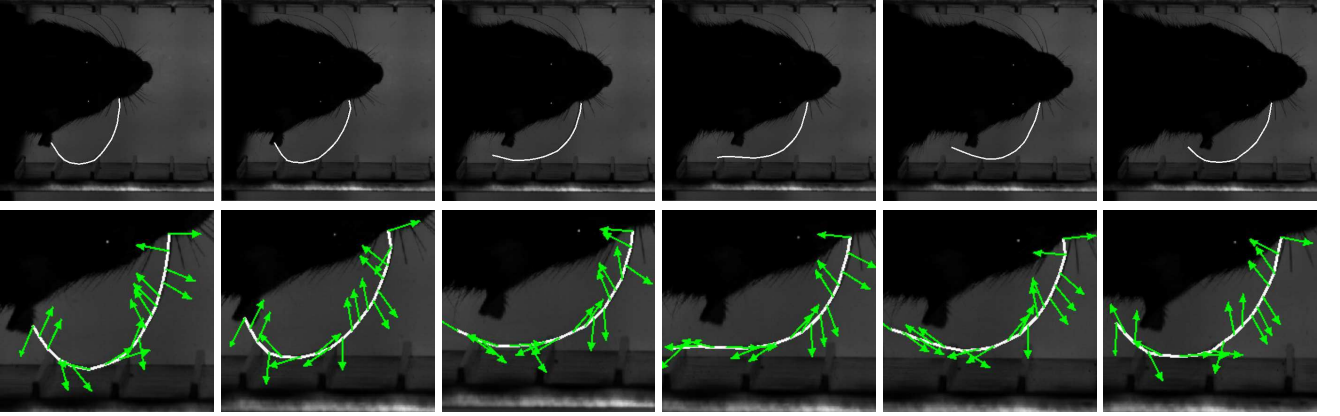


Figure 7. The deformation of the rat whisker. The top row shows selected frames from the tracking sequence with the recovered model shown in white. In the bottom row the force resultants of the axial and transversal forces are shown as green arrows.

other two examples, we tracked whisker of the rat without imposing any boundary conditions on the beam movements. We also started with the curved beam obtained by drawing the polygonal line of 17 segments in the first frame of the tracking sequence. In this way we produced the beam segments to be of different lengths which is different from the previous examples. In spite of this new settings the tracking performed very well as can be seen in Fig. 7. Every second frame was used for tracking. Note in the second column of Fig. 7 the whisker was partially occluded by the ear which was efficiently handled with our beam model.

6. Conclusion

In this paper we investigated physics-based plane beam model to track large non-linear deformations in images. We compared them to the snake-like approach which uses simplifying physical assumptions to create the model energy, similar to most physically based models used in computer vision. These approaches ignore the complex nonlinearities that are inherent to large deformations. We discovered that using our model, which approximates the physical behavior more closely, contributed to robust tracking of large deformations in the presence of clutter and partial occlusions and allowed us to obtain an estimate of the forces that were applied to the beam to produce the observed deformations. In the future work we plan to further investigate more complex 3D spatial beam and shell structures.

APPENDIX

We consider straight two node TB element shown in Fig. 2. It is deforming in XY-plane and, for simplicity, it is initially aligned with the X-axis. We later relax this assumption and consider the beam to be arbitrarily positioned. The undeformed initial configuration is referred as *reference* and the deformed one as *current* configuration. The beam rotation is defined by the angle ψ being equal to the rotation of

the cross-section when it remains normal to the beam longitudinal axis. The angle $\bar{\gamma}$ is a shear angle for which the cross section deviates from its normal position defined by the angle ψ . The total rotation of a beam cross section becomes $\theta = \psi - \bar{\gamma}$ which is exactly rotational dof.

We consider the motion of the point $P_0(X, Y)$ in reference configuration to the point $P(x, y)$ in the current configuration. We keep the assumption that the cross section dimensions do not change and that the shear rotation is small $\bar{\gamma} \ll 1$ so that $\cos \bar{\gamma} \approx 1$. The Lagrangian representation of the motion relating points $P_0(X, Y)$ and $P(x, y)$ is then given by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X + u_X - Y \sin \theta \\ u_Y + Y \cos \theta \end{bmatrix} \quad (\text{A-1})$$

The displacement can be then represented by a vector $w = [u_X(X) \ u_Y(X) \ \theta(X)]$. In the FEM formulation for 2-node C^0 element it is natural to express the displacements and rotation functions of w as linear combination of node displacements:

$$\mathbf{w} = \begin{bmatrix} \mathcal{N}_1 & 0 & 0 & \mathcal{N}_2 & 0 & 0 \\ 0 & \mathcal{N}_1 & 0 & 0 & \mathcal{N}_2 & 0 \\ 0 & 0 & \mathcal{N}_1 & 0 & 0 & \mathcal{N}_2 \end{bmatrix} \begin{bmatrix} u_{X1} \\ u_{Y1} \\ \theta_1 \\ u_{X2} \\ u_{Y2} \\ \theta_2 \end{bmatrix} = \mathbf{N}\mathbf{u}. \quad (\text{A-2})$$

where $\mathcal{N}_1 = \frac{1}{2}(1-\xi)$, $\mathcal{N}_2 = \frac{1}{2}(1+\xi)$ are the linear element shape functions.

The deformation gradient of motion can be derived from Eq. A-1 to be:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} \end{bmatrix} \quad (\text{A-3})$$

from which Green-Lagrange(GL) strain tensor describing the model strain becomes:

$$\mathbf{e} = \begin{bmatrix} e_{XX} & e_{XY} \\ e_{YX} & e_{YY} \end{bmatrix} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (\text{A-4})$$

After the derivation, the only nonzero elements are axial strain e_{XX} and the shear strain $2e_{XY} = e_{XY} + e_{YX}$. We can finally express strain vector as:

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} e_{XX} \\ 2e_{XY} \end{bmatrix} = \begin{bmatrix} e - Y\kappa \\ \gamma \end{bmatrix} \quad (\text{A-5})$$

where three strain quantities introduced above are e axial strain, γ shear strain and κ curvature. These can be collected in the element strain vector \mathbf{h}_e^T . Because of the assumed linear variations in X of $u_X(X)$, $u_Y(Y)$ and $\theta(X)$, e and γ depend on θ and κ is constant over the element. e and γ can be expressed in a geometrically invariant form which used for further development:

$$e = \frac{L}{L_0} \cos \bar{\gamma}, \gamma = \frac{L}{L_0} \sin \bar{\gamma}. \quad (\text{A-6})$$

where $\bar{\gamma} = \theta - \psi$. These geometrically invariant strain quantities can be used for the beam in arbitrary reference configuration. The variations of δe , $\delta \gamma$ and $\delta \kappa$ with respect to the nodal displacement variations are required for derivations of strain-displacement relation $\delta \mathbf{h} = \mathbf{B} \delta \mathbf{u}$. To form strain-displacement matrix \mathbf{B} we take partial derivatives of e , γ and κ with respect to the node displacements and collect them into matrix \mathbf{B} :

$$\mathbf{B} = \begin{bmatrix} -c\omega & -s\omega & L_0 \mathcal{N}_1 \gamma & c\omega & s\omega & L_0 \mathcal{N}_2 \gamma \\ s\omega & -c\omega & -L_0 \mathcal{N}_1 (1+e) & -s\omega & c\omega & -L_0 \mathcal{N}_2 (1+e) \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A-7})$$

where $\omega = \theta + \psi$, $c\omega = \cos \omega$ and $s\omega = \sin \omega$.

We assume that the material is homogeneous and isotropic. The only nonzero second Piola-Kirchhoff stresses are axial stress s_{XX} and s_{XY} . They are related to the Green-Lagrange strain by the linear elastic equation:

$$\mathbf{s} = [s_{XX} \quad s_{XY}] = [s_1^0 \quad s_2^0] + \begin{bmatrix} E & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \mathbf{s}^0 + \mathbf{E} \mathbf{e} \quad (\text{A-8})$$

where E is a Young modulus and G is the shear modulus. We introduce pre-stress resultants N^0 , V^0 and M^0 which define the axial forces, transverse shear forces and bending moments respectively, in the reference configuration. We also define the stress resultants in the current configuration to be $N = N^0 + EA_0 e$, $V = V^0 + GA_0 \gamma$ and $M = M^0 + EI_0 \kappa$, and collect them into stress-resultant element vector $\mathbf{z}_e = [N \quad V \quad M]$. The derived quantities are used to derive strain energy of Eq. 5 and internal forces of Eq. 6.

References

- [1] A. Bartoli and A. Zisserman. Direct Estimation of Non-Rigid Registration. In *British Machine Vision Conference*, Kingston, UK, September 2004. 2
- [2] L. Cohen, , and I. Cohen. Deformable models for 3-d medical images using finite elements and balloons. In *Conference on Computer Vision and Pattern Recognition*, pages 592–598, 1992. 1, 2, 3
- [3] D. T. D. Metaxas. Constrained deformable superquadrics and nonrigid motion tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993. 1, 2, 3
- [4] H. Delingette, M. Hebert, and K. Ikeuchi. Deformable surfaces: A free-form shape representation. In *SPIE Geometric Methods in Computer Vision*, volume 1570, pages 21–30, 1991. 2, 3
- [5] I. Essa, S. Sclaroff, and A. Pentland. *Physically-based modeling for graphics and vision*. In Ralph Martin, editor, *Directions in Geometric Computing*. Information Geometers, U.K., 1993. 2
- [6] S. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Lab., Cambridge, MA, 1997. 2
- [7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 1, 2
- [8] Y. Lee, D. Terzopoulos, and K. Walters. Realistic modeling for facial animation. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62, New York, NY, USA, 1995. ACM Press. 2
- [9] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, October 2005. 4
- [10] T. McInerney and D. Terzopoulos. Deformable models in medical images analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996. 2, 3
- [11] D. Metaxas. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging*. Kluwer Academic Publishers, Norwell, MA, USA, 1996. 2
- [12] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11), November 1996. 2
- [13] J. F. O'Brien, P. R. Cook, and G. Essl. Synthesizing sounds from physically based motion. In E. Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 529–536, 2001. 2
- [14] A. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, March 1990. 2, 3
- [15] S. Sclaroff and A. P. Pentland. Physically-based combinations of views: Representing rigid and nonrigid motion. Technical Report 1994-016, 1994. 2
- [16] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:703–714, 1991. 1, 2, 3
- [17] S. Timoshenko and G. MacCullough. *Elements of Strength in Materials*. van Nostrand, New York, 3rd edn., 1949. 1, 3
- [18] L. Tsap, D. Goldgof, and S. Sarkar. Fusion of physically-based registration and deformation modeling for nonrigid motion analysis, 2001. 2