Binding Vision to Physics Based Simulation: The Case Study of a Bouncing Ball

Nikolaos Kyriazis[1,2]
http://www.ics.forth.gr/~kyriazis

Iason Oikonomidis[1,2]
http://www.ics.forth.gr/~oikonom

Antonis A. Argyros[1,2]
http://www.ics.forth.gr/~argyros

[1] Computational Vision and Robotics Lab., Institute of Computer Science, FORTH

[2] Computer Science Department, University of Crete, Greece

Abstract

A dynamic scene and, therefore, its visual observations are invariably determined by the laws of physics. We demonstrate an illustrative case where *physical explanation*, as a vision prior, is not a commodity but a necessity. By considering the problem of ball motion estimation we show how physics-based simulation in conjunction with visual processes can lead to the reduction of the visual input required to infer physical attributes of the observed world. Even further, we show that the proposed methodology manages to reveal certain physical attributes of the observed scene that are difficult or even impossible to extract by other means. A series of experiments on synthetic data as well as experiments with image sequences of an actual ball, support the validity of the proposed approach. The use of generic tools and the top-down nature of the proposed approach make it general enough to be a likely candidate for handling even more complex problems in larger contexts.

1 Introduction

Computer vision is concerned with the understanding of the physical world through the analysis of its image(s). Such an understanding may be defined at various levels of abstraction. Whatever the level of abstraction may be, this understanding is always associated with a context, i.e. an assumption of a generative process that produces the observations. It is convenient to think about such a context as a set of rules that transform some initial conditions into images. In this work, we are interested in deriving a *physically plausible explanation* of a dynamic scene. Thus, the respective rules governing the generative process are the *laws of physics*.

We argue that by exploiting this type of context as a prior, we can derive very useful information for a dynamic scene, that is difficult or even impossible to derive by other means. Consider for example the testbed scenario according to which we are interested in estimating the state of a uniformly colored bouncing ball through its observation by a single or by multiple calibrated cameras (Fig. 1). By employing standard computer vision techniques, accounting for the position of the ball at each time step is not trivial. The possibly inadequate acquisition frame rate may lead to aliasing and the possibly large shutter time may lead to
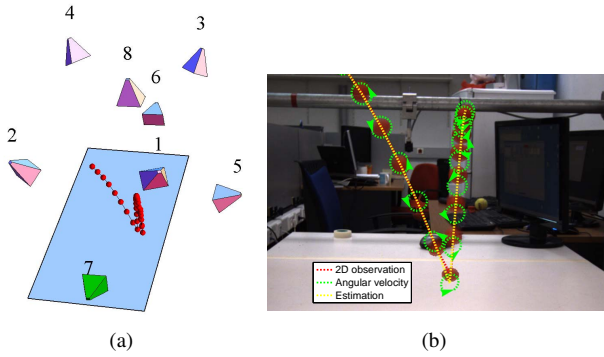
Figure 1: A ball is thrown towards a table with a high back-spin. By incorporating physics-based simulation, we infer the ball's 3D trajectory (a), and its linear and angular velocities *from a single camera* (cam. 7). The proposed method identifies that a back-spin is the cause of the reduction of the outgoing angle of the bounce. The green ellipses in (b) are projections of an equator of the ball and the arrows represent the direction of the estimated angular speed.

motion blur. On top of the above mentioned difficulties, for some aspects of the state of the ball (i.e., its orientation and/or angular velocity) there is no direct evidence, whatsoever. The problem becomes even more challenging when we are interested in solving the above problems based on single-camera observations and/or when, due to occlusions, the available set of visual observations becomes even more limited.

We show that through the direct incorporation of explicit physics we are able to tackle these challenges. We demonstrate how hidden variables like the position of the ball when it is occluded, its orientation and angular velocities, can be estimated. We highlight that physics provide a strong prior, which permits the successful treatment of these challenges, even for the case of single camera 2D observations that may be incomplete due to occlusions. The incorporation of physics is performed in a clean, top-down fashion that could be generalized and scaled towards solving larger problems in different contexts.

The proposed framework becomes possible because of the evolution of optimization methods, the advancement of physics-based simulation and the availability of substantial computational power. Powerful optimization techniques enable efficient optimization of hard, multi-dimensional problems [16]. Physics simulation has advanced to a point where computational demands can be efficiently handled, realism is a common denominator in most physics simulators and the extension of simulators is easy due to their carefully designed software architectures. Moreover, parallel multicore technologies like contemporary CPUs and GPUs allow for the computation of thousands of simulations per second. Although not exploited in this work, the latter further extends our method's potential.

2   Relevant work

In the past, several researchers have stressed the benefits stemming from the consideration of physics as integral part of computer vision processes. Although beyond the scope of this work, we mention approaches [12, 17, 19, 25] that exploit the physical nature of light to process images and estimate or predict otherwise unaccountable information.

The prevalent case study of employing physics in vision is the problem of 3D human

tracking. The dynamics of the human body can be exploited towards the formation of strong priors. Popović and Witkin [21] rectified 3D motion capture data to make them compliant to physical constraints. Vondrak *et al*. [24] fused motion planning, contact dynamics and a ground assumption to track humans from multiple cameras. Brubaker *et al*. [6, 7, 9] employed realistic metaphors of the lower body dynamics to estimate and predict walking. Going further, they incorporated a friction model for a ground that affords human motion upon it [8].

There are also approaches that reflect physics implicitly or metaphorically. Brand *et al*. [4, 5] exploited the physical notion of causality in order to perform qualitative reasoning in computer vision problems. Delamarre [14] assigned a physical behaviour to a contour model that drove the optimization process of recovering it. Chen *et al*. [11] were able to track a basketball, while in the air and despite occlusions, by assuming the parabolic nature of free flight. Papadourakis and Argyros [20] identified the physical notion of object permanence as the ambiguity resolver for the case of multiple objects tracking. Sethi and Roy-Chowdhury [22] gave physical substance to image features and used methods, usually employed in physics, in order to model activities in image sequences.

This work is most closely related to the works in [3, 15, 18]. Metaxas and Terzopoulos [18] defined a continuous Kalman filter that was able to track a deformable object. This was achieved by a detailed motion model that was tightly coupled to the optimization process. Although interesting, the extensibility of their approach is hindered by this tight coupling. Bhat *et al*. [3] performed 3D tracking of an object by searching over parameterized experiments that optimally project back to an image sequence. However, the shape of the object and the restriction that it is tracked while in flight does not expose the full potential of employing physics. Finally, Duff and Wyatt [15] used physical simulation and search heuristics to track a fast moving ball, despite occlusions. They reasoned upon the ball's 2D position but they did not consider the 3D case, or the hidden variables of ball orientation and angular velocity.

Despite the significant amount of existing work, no existing study demonstrates the full potential of binding vision to physics-based simulation. We try to fill this gap by proposing a method that is generic, top-down, simulation based and incorporates realistic simulation of physics. As a result, and to the best of our knowledge, the proposed method is the first to consider physical properties that can be estimated through physics-based simulation, even in the case of single camera observations and severe occlusions.

3 Methodology

Let a colored ball be thrown on a table so that it bounces for several times and then rests. The 2D image position of the ball can be easily recovered for every time step and for every camera that views it for the case of moderate velocities. Accurate recovery is problematic for the case of larger velocities and especially around bounces, due to blurring and aliasing (see Fig. 1). These problems hinder a bottom-up resolution of the problem, but, as it will be shown, they do not prevent a top-down approach from being effective.

We consider the *physical explanation e* of the bouncing of the observed ball. We assume that certain scene properties (mass, inertia, collision properties) and initial conditions (position and velocities of the throw), together with the laws of physics, generate a 3D trajectory which optimally projects back to all cameras and matches the observations o. We define an objective function S that quantifies the discrepancy between the actual observations and the
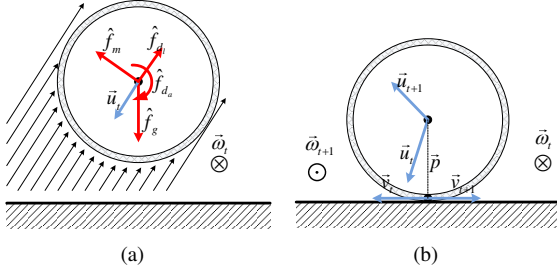
(a)   (b)

Figure 2: The two phases of the bouncing ball (a) flight and (b) bounce. Red arrows represent impulses and blue arrows represent velocities. Angular velocities are perpendicular to the image plane. Black arrows represent the air flow with respect to the flying ball.

camera back-projection of a simulated parameterized ball throwing experiment. The latter can be sub-sampled to match the acquisition rate of the actual camera set. This also accounts for the aliasing effects of the acquisition process.

Since whatever is observed must be *physically plausible*, the *physical explanation e* is the minimizer parameter vector *x* of this objective function. In notation:

$$e = \arg\min_{x} S(o,x) \quad \text{where} \quad S(o,x) = \text{BackProjectionError}(o, \text{Simulation}(x)) \quad . \quad (1)$$

Our method receives 2D or 3D trajectories that represent the course of a bouncing ball and outputs the parameters of a simulated experiment that optimally matches the observations. In the next sections we describe in detail the parameter space of the simulations (search space for *e*), the simulation procedure as well as the minimization process.

3.1   Physics of the bouncing ball

In order to account for the dynamics of the trajectory of a bouncing ball we explicitly reason upon an idealized, yet sufficient, physics model. We identify two alternating phases, namely ball flying and ball bouncing. The two phases are detailed in Fig. 2. Since we consider average effects over generally small time intervals, we discuss impulses rather than forces. For a time interval dt and a function of force $\vec{f}$ over that interval, the respective impulse is defined as $\hat{f} = \int \vec{f} dt$.

During its flight (Fig. 2(a)), the ball undergoes velocity changes that are inflicted by the gravitational attraction and the resistance of the air. Gravity constantly exerts a downwards impulse of $\hat{f}_g = m \cdot \vec{g}$, where m is the mass of the ball and $\vec{g}$ is the gravitational acceleration. Given enough air resistance, at each time step t, the linear velocity $\vec{u}_t$ and angular velocity $\vec{\omega}_t$ are decreased in magnitude due to friction (linear damping d_l and angular damping d_a). Also, an impulse $\hat{f}_m = K \cdot (\vec{u}_t \times \vec{\omega}_t)$ that is perpendicular to the linear velocity and the axis of angular velocity, makes the ball travel in a curved trajectory [1]. For every part of the flight the standard equations of motion hold and suffice in order to predict the state of the ball.

At every bounce (Fig. 2(b)), a portion of the ball's vertical energy is lost according to an elasticity factor $\beta \in [0,1]$. An amount of dynamic friction redistributes energy between its linear and angular motion in the horizontal, according to a friction factor $\alpha \in [-1,1]$. The friction model adopted here is an extension of [2] to the 3D case and identifies friction as the reason that scales the total linear velocity $\vec{v}_{t+1}$ of the contact point. This modeling

accounts for a great variety of frictions. For example, both the glass ball (no friction) and the super ball (extreme friction) can be modeled for $\alpha = 1$ and $\alpha = -1$, respectively. The vertical axis of rotation has no contribution to the horizontal contact. Moreover, the impulse which changes the horizontal linear momentum is also the negative impulse that changes the horizontal angular momentum. All the aforementioned constraints define a system of equations:

$$
\begin{array}{rcl}
S_y \vec{u}_{t+1} & = & -\beta S_y \vec{u}_t \\
S_y \vec{\omega}_{t+1} & = & S_y \vec{\omega}_t \\
S_{xz} \vec{v}_{t+1} & = & \alpha S_{xz} \vec{v}_t \\
m \cdot \vec{p} \times S_{xz} (\vec{v}_{t+1} - \vec{v}_t) & = & -I \cdot S_{xz} (\vec{\omega}_{t+1} - \vec{\omega}_t)
\end{array}
\quad \text{with} \quad
\begin{array}{l}
S_y = Diag([0,1,0]) \\
S_{xz} = Diag([1,0,1]) \\
\vec{v}_k = \vec{u}_k + \vec{p} \times \vec{\omega}_k
\end{array}
\quad (2)
$$

These equations linearly relate the pre-bounce velocities $\vec{u}_t, \vec{\omega}_t$ to post-bounce velocities $\vec{u}_{t+1}, \vec{\omega}_{t+1}$. Solving this system for time $t+1$ yields the post-bouncing state of the ball.

3.2 Physics-based simulation

Ubiquitous physics simulators are already able to account for the most part of the presented physics modeling. They also ease the incorporation of more detailed models via modular architectures that are carefully designed for that purpose. In our scenario we augment such a simulator by incorporating the effects of $\hat{f}_g$ and $\hat{f}_m$ and by adjusting the collision module so that it also accounts for the exchange of horizontal energy. The vertical is already in agreement with our equations. The parameter vector of a simulation is $(m, I, \beta, \alpha, d_l, d_a, K, \vec{s}_0, \vec{u}_0, \vec{\omega}_0, T)$. The state of the ball $\left(\vec{l}_t, \vec{q}_t, \vec{u}_t, \vec{\omega}_t \right)$ is the result of the invocation of the simulator for a given parameter vector, where t is a time step of the whole duration T and $\vec{l}_t, \vec{q}_t, \vec{u}_t, \vec{\omega}_t$ represent position, orientation, linear and angular velocity (all in 3D space).

3.3 Optimization through Differential Evolution (DE)

Differential Evolution (DE) [13, 23] is an evolutionary optimization method. It depends on only a few parameters that have an intuitive explanation and exhibits remarkable performance in difficult problems of large dimensionality. DE effectively handles real-valued multidimensional, potentially non-linear and non-differentiable objective functions. It performs optimization by evolving a set of N_H hypotheses H_g and dimensionality D. Being evolutionary, DE is defined via its mutation, crossover and selection mechanisms that are applied at every generation g.

During mutation, every hypothesis $h_{g,i} \in H_g$ becomes a linear combination of three randomly selected, pairwise different hypotheses of the previous generation $g - 1$.

$$ h_{g,i} = h_{g-1,r_1} + F \left(h_{g-1,r_2} - h_{g-1,r_3} \right), r_j \sim U \left(0, |H_g| \right) \wedge r_k \neq r_l \forall k, l. \quad (3) $$

Mutation is controlled by the differentiation factor $F \in [0,2]$. Each mutated hypothesis $h_{g,i}$ is then combined with $h_{g-1,i}$ in order to produce a replacement candidate $\hat{h}_{g,i}$ in the crossover phase.

$$ \hat{h}_{g,i}(j) = \begin{cases} h_{g,i}(j), & r_j \leq CR \vee j = idx_{g,i} \\ h_{g-1,i}(j), & \text{otherwise} \end{cases}, j = 1, \ldots, D, r_j \sim U(0,1), idx_{g,i} \sim U(1,D) $$

$$ (4) $$

In Eq. (4), $h_{g,i}(j)$ denotes the j-th component of the i-th hypothesis in the g-th generation. The crossover constant CR controls the combination of individual parameters of $h_{g,i}$ and $h_{g-1,i}$. A random parameter index $idx_{g,i}$ is preselected in order to ensure that at least one parameter of the mutated vector will survive the crossover.

Finally, in the selection phase, the replacement candidate actually replaces the original one in the next generation, if it scores better in the objective function.

The original algorithm is parallel, in the sense that two consecutive generations are two distinct sets. We consider a serial variant, where the two generations are mixed. This means that a mutation may be based on already mutated vectors in the same generation. We have experimentally observed this mixing to add quicker reflexes to the algorithm, leading to faster convergence. We also consider a dithering parameter δ that modulates F at each generation. Dithering improves convergence and helps in avoiding local optima [10]. The DE variant of our choice appears with the coding DE/rand/1/bin in [23]. The input of DE is a real-valued objective function f (in our case, BackProjectionError in Eq. (1)), the number of generations N_G, hypotheses per generation N_H and constants F, CR, δ. The output of DE is the real-valued parameter vector that optimizes f. In all experiments we used the following parameterization for DE: $(N_G, N_H, F, CR, \delta) = (300, 72, 0.9, 0.9, 1.5)$.

4   Experimental results

A series of experiments were conducted to assess our method's ability to account for 3D and 2D observations of a bouncing ball. From an implementation point of view, we used the DE implementation of the SwarmOps[1] library, the Newton Game Dynamics[2] simulator and the MATLAB[3] platform for the rest of the logic.

4.1   Results on synthesized image sequences

A first series of experiments were carried out to assess the capability of the proposed method to come up with physically plausible explanations of various simulated ball throws, performed in different world contexts and initial conditions. We distinguished the parameters representing scene properties $(m, I, \beta, \alpha, d_l, d_a, K)$ and those representing initial conditions $(\vec{s}_0, \vec{u}_0, \vec{\omega}_0)$. We generated 3 random scene property parameter vectors and 3 random initial condition vectors. We then considered all possible combinations resulting in a total of 9 experiments. The experiment parameterizations generated 9 ball 3D trajectories for a time duration of $T = 4s$, each. Each 3D trajectory was considered in conjunction with 6 levels of Gaussian noise at each of the 3 spatial dimensions, separately and with variances $0m$, $0.03m, 0.05m, 0.1m$, $0.2m$ and $0.5m$, respectively. For each set of parameters, 20 repetitions were executed. This protocol led to a total of $3 \times 3 \times 6 \times 20 = 1080$ experiments accounting for various world properties, initial conditions and amounts of noise. For each experiment, the physics-based simulator produced a ground truth 3D trajectory of the ball and the proposed method was employed to provide a physical explanation of it. We evaluated the optimization accuracy by measuring, for each experiment, the average of the Euclidean distances between corresponding points of the simulated and the recovered 3D ball trajectories. The results presented in Fig. 3(a) show that the proposed method is able to perform

---

[1]http://www.hvass-labs.org/projects/swarmops/c/
[2]http://www.newtongamedynamics.com
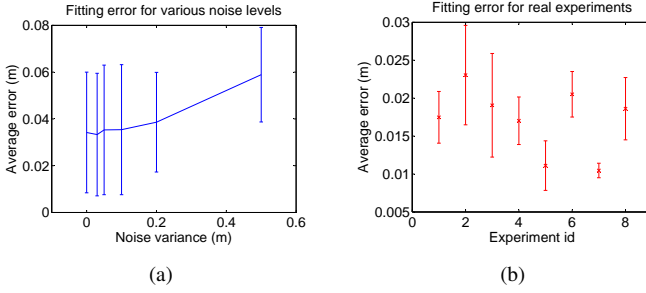[3]http://www.mathworks.com/products/matlab/

Figure 3: The mean values and standard deviations for the errors on the experiments with (a) synthetic and (b) real observations.
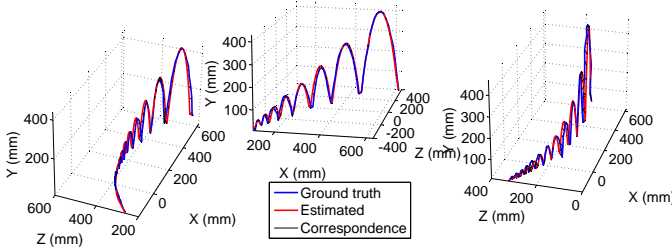

Figure 4: Examples of actual and estimated 3D trajectories. The illustrated trajectories have high left, low left and high right curvature, respectively. The respective average trajectory point estimation errors were $1.38cm$, $0.75cm$ and $0.83cm$.

well even under severe Gaussian noise. This is because by conception, the method allows for physically plausible solutions, only. Thus, observations that are heavily contaminated by this type of noise cannot distract the estimation towards physically implausible solutions.

4.2 Multiview estimation of 3D trajectories

Experiments analogous to those of Sec. 4.1 were also performed in the real world, i.e., using multicamera observations of an actual bouncing ball. For these experiments, we employed the setup that is illustrated in Fig. 1(a). It consists of a $2 \times 1m^2$ hard table, a red table tennis ball of radius $2cm$, and 8 synchronized and calibrated Flea2 PointGrey cameras. All cameras provided images at a resolution of 1280×960 and at an acquisition rate of $30fps$. Processing was performed on a workstation that has an Intel Core i7 950 CPU @ $3.07Ghz$ and $6GB$ of RAM. All computations were performed on a single thread of a single core of the CPU.

As a first step, the ball was detected in every frame for all sequences (all cameras inclusive). We applied color thresholding to isolate red areas in every frame. We then filtered each extracted connected blob based on its shape, to ensure high confidence detection and excluded partially occluded and/or significantly blurred detections. 3D ball positions were then estimated through multiview 3D reconstruction of the ball centroids.

We conducted several ball throwing experiments in our physical setup. We selected 8 of them according to our empirical criterion of diversity. Each one was input to our method 20 times. The optimization process of Sec. 4.1 was employed, where synthetic data were

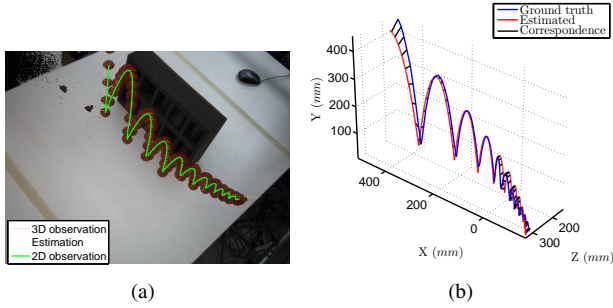(a)   (b)

Figure 5: Estimation of the 3D trajectory of a ball from single camera 2D observations (camera 3) and the assumption of a given physical world.

replaced by real data. The obtained results are presented in Fig. 3(b). Some examples of actual trajectories and the respective estimates are shown in Fig. 4. As it can be verified, the proposed method faithfully reproduces the actual 3D observations.

4.3   Single view estimation of 3D trajectories

We are interested in estimating the ball 3D trajectory by a single camera. Without any physics-based prior information and for the case of a ball of known size, single view ball 3D localization depends on the ability to accurately estimate the ball's projected shape and size. In practice, this is problematic due to acquisition and processing artifacts, which lead to errors in depth estimation that are difficult to treat in a bottom-up fashion. However, by modeling the physics of the process, we are able to infer depth from a more reliable source, the 2D trajectory of the ball on the image plane of a single camera. To demonstrate this, we optimized S for 2D observations of a single camera and the non trivial cases of non-planar (due to spin) trajectories. During optimization, the simulator generated 3D data from which 2D reference trajectories were produced by means of projection. The back-projection error, i.e., the average Euclidean distance between back-projected and observed 2D positions of the ball was guiding the optimization process. An exemplar 3D estimation is illustrated in Fig. 5. It can be verified that the estimation from a single camera is almost indistinguishable from the ground truth.

Interestingly, no post-processing is required to enforce the plausibility of the solution because implausible hypotheses are not considered at all. Even more importantly, even though 3D estimation from 2D trajectories relies on the knowledge of the respective ball heights, we do not account for this knowledge explicitly. 3D reconstruction comes effortlessly, as a byproduct of physics-based simulation. Another interesting observation is that the points at which bounces are observable suffer from aliasing. However, since we also sub-sample the simulator's behavior at real acquisition rate, we also account for this type of aliasing.

4.4   Seeing through walls with a single camera

We also tested our method's effectiveness under considerable lack of constraints, i.e., in the case of partial observations due to occlusions. We recorded ball throws that were largely invisible to camera 6 due to a purposefully placed large obstacle (see Fig. 6(a)). Based on this partial evidence, the proposed method estimated the 3D trajectory of the ball from the
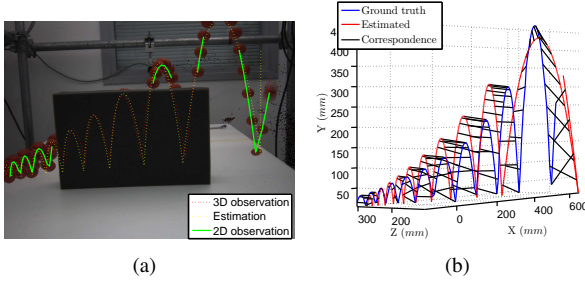
(a)   (b)

Figure 6: Single view estimation of the 3D trajectory of a ball from partial 2D observations (camera 6) and the assumption of a given physical world.

(single) view of camera 6. The ball was still visible to some of the rest of the cameras. This information was only used to estimate a kind of ground truth for the 3D trajectory of the ball. Figure 6 shows the actual observations, the ball trajectory as this was estimated by camera 6 and the ground truth as this was measured by the rest of the cameras. The estimation of the 3D trajectory (see Fig. 6(b)) is not that accurate due to the lack of enough constraints. Still, it is quite satisfactory given the fact that it has been obtained through single camera observation and in the presence of occlusions.

4.5   Inferring angular velocity

The ball's angular velocity cannot be estimated by any direct vision method in any of the considered experiments. However, evidence regarding this parameter is encapsulated in the overall dynamic behavior of the ball. By seeking for a physically plausible explanation of the observed scene, the proposed approach reveals, as a byproduct, information regarding the hidden variable of angular velocity.

We performed a series of ball throws with high back-spin, so that the ball resists its original tendency to move forward (Fig.1). We then optimized S for the resulting 2D observations of camera 7. An exemplar bounce is shown in Fig. 1(b). The proposed method inferred the 3D state of the ball accurately (once more, ground truth and estimated ball positions are indistinguishable). Moreover, as demonstrated in Fig. 1(b), we were able to compute a qualitative measure of the ball's angular velocity.

5   Summary

We presented a method that interprets a dynamic scene by binding vision to physics based simulation. We combined a powerful optimization method and a detailed physics model of a bouncing ball in order to track the latter in challenging scenarios. We experimentally demonstrated that accounting for physics does not simply constitute yet another complementary source of information but rather, a strong prior that permits the treatment of underconstrained vision problems. In fact, we demonstrated that by incorporating physics, we may require less cameras/observations to obtain the same type of information or even gain access to information that is otherwise "invisible" to a vision system. Given the continuous advancement in optimization techniques [16], simulation tools and computational power,

we believe that the proposed method holds great potential towards addressing problems of greater dimensionality and complexity.

Acknowledgments

References

[1] A. Armenti. *The Physics of Sports*. Copernicus Books, 1992.

[2] P.J. Aston and R. Shail. The Dynamics of a Bouncing Superball with Spin. *Dynamical Systems*, 22(3):291–322, 2007.

[3] K. Bhat, S. Seitz, J. Popović, and P. Khosla. Computing the Physical Parameters of Rigid-body Motion from Video. In *ECCV 2002*, pages 551–565. Springer, 2002.

[4] M. Brand. Seeing physics, or: Physics is for prediction. In *Proceedings of the Workshop on Physics-based Modeling in Computer Vision: June 18-19, 1995, Cambridge, Massachusetts*, page 144. IEEE Computer Society, 1995.

[5] M. Brand. Physics-Based Visual Understanding. *Computer Vision and Image Understanding*, 65(2):192–205, 1997.

[6] M Brubaker and Leonid Sigal. Physics-based Human Motion Modeling for People Tracking: A Short Tutorial. *Image (Rochester, N.Y.)*, pages 1–48, 2009.

[7] MA Brubaker and DJ Fleet. The Kneed Walker for Human Pose Tracking. In *CVPR 2008*, pages 1–8. IEEE, 2008.

[8] M.A. Brubaker, L. Sigal, and D.J. Fleet. Estimating Contact Dynamics. In *ICCV 2009*, pages 2389–2396. IEEE, 2009.

[9] M.A. Brubaker, D.J. Fleet, and A. Hertzmann. Physics-based Person Tracking Using the Anthropomorphic Walker. *International Journal of Computer Vision*, 87(1):140–155, 2010.

[10] U.K. Chakraborty. *Advances in Differential Evolution*. Springer Publishing Company, Incorporated, 2008.

[11] H.T. Chen, M.C. Tien, Y.W. Chen, W.J. Tsai, and S.Y. Lee. Physics-based Ball Tracking and 3D Trajectory Reconstruction with Applications to Shooting Location Estimation in Basketball Video. *Journal of Visual Communication and Image Representation*, 20 (3):204–216, 2009.

[12] O. Cossairt, S. Nayar, and R. Ramamoorthi. Light Field Transfer: Global Illumination Between Real and Synthetic Objects. In *ACM SIGGRAPH 2008*, pages 1–6. ACM, 2008.

[13] S. Das, A. Abraham, and A. Konar. Particle Swarm Optimization and Differential Evolution Algorithms: technical analysis, applications and hybridization perspectives. *Advances of Computational Intelligence in Industrial Systems*, pages 1–38, 2008.

[14] Q. Delamarre and O. Faugeras. 3D Articulated Models and Multiview Tracking with Physical Forces. *Computer Vision and Image Understanding*, 81(3):328–357, 2001.

[15] D.J. Duff, J. Wyatt, and R. Stolkin. Motion Estimation using Physical Simulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1511–1517. IEEE, 2010.

[16] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing Results of 31 Algorithms from the Black-box Optimization Benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Comp on Genetic and Evolutionary Computation*, pages 1689–1696. ACM, 2010.

[17] I. Ihrke, K.N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich. Transparent and Specular Object Reconstruction. In *Computer Graphics Forum*. Wiley Online Library, 2010.

[18] D. Metaxas and D. Terzopoulos. Shape and Nonrigid Motion Estimation through Physics-based Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[19] S.K. Nayar and S.G. Narasimhan. Vision in Bad Weather. In *ICCV 1999*, volume 2, pages 820–827. IEEE, 1999.

[20] V. Papadourakis and A. Argyros. Multiple Objects Tracking in the Presence of Long-term Occlusions. *Computer Vision and Image Understanding*, 114(7):835–846, 2010.

[21] Z. Popović and A. Witkin. Physically Based Motion Transformation. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 11–20. ACM Press/Addison-Wesley Publishing Co., 1999.

[22] R.J. Sethi and A.K. Roy-Chowdhury. Physics-based Activity Modelling in Phase Space. In *Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 170–177. ACM, 2010.

[23] R. Storn and K. Price. Differential Evolution–A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4): 341–359, 1997.

[24] M. Vondrak, L. Sigal, and OC Jenkins. Physical Simulation for Probabilistic Motion Tracking. In *CVPR 2008*, pages 1–8. IEEE, 2008.

[25] T.E. Zickler, P.N. Belhumeur, and D.J. Kriegman. Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction. In *International Journal of Computer Vision*, volume 49, pages 215–227. Springer, 2002.