Towards Simulated Reality

Alberto Badias, Iciar Alfaro, David Gonzalez, Elias Cueto
Universidad de Zaragoza, Spain
{abadias, iciar, gonzal, ecueto}@unizar.es

Francisco Chinesta
Arts et Métiers, ParisTech, France
francisco.chinesta@ensam.eu

Abstract

We present a new framework to introduce physical computing in mixed reality. We use model order reduction techniques to allow the system to understand the physical phenomena that occur in its environment in real time. It then can give valuable information to the user about object internal properties such as stress or strain, thus leading to a human augmented intelligence system. We include several examples that cover different branches of computational mechanics, such as solid deformations with nonlinear materials, general contact between physical and virtual objects or fluid mechanics.

1. Introduction

Augmented reality (AR) is a well known concept nowadays, since it is undergoing a great revolution. Probably because it has managed to cross the limits of research and be part of the development lines of some companies that already offer AR applications to users. Although there is still much to be done, some concepts that until recently were thought as futuristic, are getting closer us. One example is the approach of *Spatial AI* [5], where a new system of visual perception is proposed, understood as a mixture between geometric measures, semantic perception and machine learning. Keeping this idea in mind, we want to add our small contribution from the machine learning side, but from a branch that may be far from the computer vision community—we are not talking about neural networks—. We propose the use of model order reduction (MOR) techniques to reduce the complexity of the equations that best explain the physical changes that the objects that surround us experience. And finally create a system able to explain us these physical changes by means of a visual device.

Some real time physical engines are already available through video game platforms such as *Unity* or *Unreal Engine*, among others. These are great simplifications of physical phenomena to work in real time frequencies, which translates into 30-60 fps, usually. But these simplifications often involve simple and linear models, which are not always able to describe reality as we would like. Therefore, we believe that we must create a new framework to introduce high fidelity simulations that better describe the behavior of the objects. Since this is not cheap, computationally speaking, we use MOR methods to reduce the model complexity and be able to work in real time frequencies with a minimum, controlled, loss of accuracy.

In this work we demonstrate the power of merging MOR methods with visual perception systems [2], applying them to different physical problems of distinct nature. In addition, since it is a method for understanding physical phenomena it is totally independent of the image capturing system. We have tested our method with monocular and stereo cameras, estimating the behavior of both rigid and deformable objects. Table 1 shows a summary of the examples we provide.

Monocular			Stereo
Rigid Body (Static Scene)	Deformable (Non-static Scene[1])		Deformable (Non-static Scene)
Car Aerodynamics	2D	3D	Stanford Objects
	Beam	Rubber Piece	
LLE	PGD	POD	sPGD

Table 1. Summary of the examples we provide in this paper, classified according to (from up to down) input device, objects being tracked, particular example and MOR method used. Go to Fig. 1 or section 5 for more details.

2. Dimensionality Reduction

In order to achieve the standard video frequency it is required to reduce the complexity of the original physical equations. We use model order reduction methods, which are based on the projection of the original solution of dimension N_h on a space of reduced dimensionality N, being $N \ll N_h$.

Assuming a partial differential equation governs our problem and it depends on a vector of parameters $\boldsymbol{\mu} \in \mathcal{P}$,

[1] Known as a *Dynamic Scene* in computer vision, in computational mechanics it may be confusing to talk about dynamics without inertial terms or mass acceleration.

Figure 1. Examples introducing computational mechanics in mixed reality. From left to right: car arodynamics, foam beam, rubber piece and Stanford objects.

where $\mathcal{P}$ is the set of all possible values of the parameters being a compact subset of $\mathbb{R}^P$, the manifold $\mathcal{M}$ contains all of the solutions $\boldsymbol{u}(\boldsymbol{\mu})$.

$$\mathcal{M} = \varphi(\mathcal{P}) = \{\boldsymbol{u}(\boldsymbol{\mu}) \in V : \boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P\},$$

where φ is the solution map and V is a suitable Hilbert space. The solution map φ is defined as

$$\varphi : \mathcal{P} \rightarrow V, \qquad \boldsymbol{\mu} \mapsto \boldsymbol{u}(\boldsymbol{\mu}).$$

In a deformable solid problem we call $\boldsymbol{u}(\boldsymbol{\mu})$ to the 3D displacements, but it can be referred to any other physical quantity. We apply a suitable discretization to the original solution to obtain the high-fidelity solution set $\mathcal{M}_h$, also known as the discrete manifold. We assume that $\mathcal{M}_h$ is so close to $\mathcal{M}$ that no differences appear in discretization points. The discretized solution of the equation is $\boldsymbol{u_h}(\boldsymbol{\mu})$ and it belongs to a finite-dimensional subspace V_h of dimension N_h.

$$\mathcal{M}_h = \varphi_h(\mathcal{P}) = \{\boldsymbol{u_h}(\boldsymbol{\mu}) \in V_h : \boldsymbol{\mu} \in \mathcal{P}\} \subset V_h.$$

After applying the approximation to $u(\mu)$ we obtain also the discrete solution map

$$\varphi_h : \mathcal{P} \rightarrow V_h, \qquad \boldsymbol{\mu} \mapsto \boldsymbol{u_h}(\boldsymbol{\mu}).$$

There are many MOR methods to reduce the dimensionality of a stated problem to obtain a reduced basis that governs the problem considering only the latent variables (more information in section 3). The reduced solution belongs to a low-dimensional subspace $V_N \subset V_h$ of dimension $N \ll N_h$ where the construction of the basis of V_N is generated from a set of N functions called the reduced basis functions.

$$V_N = \text{span}\{\zeta_1, \ldots, \zeta_N\}.$$

A set of coefficients that multiply those functions are needed to solve the high-fidelity problem. Here different methods have been developed, and for example, in the Proper Generalized Decomposition (PGD) [4] case, we suppose the solution in a variables separation form so we have a product of univariate functions depending only on one parameter. Other methods project the differential operator to the reduced space, like Proper Orthogonal Decomposition (POD)[3]. We explain the basic differences between these methods in next section.

3. Model Order Reduction Methods

These methods can be classified into *a priori* and *a posteriori* methods. *A priori* methods try to solve the differential equation directly in the reduced space, being PGD an example. *A posteriori* methods try to project the solution once it has been obtained on the original space, and a standard example is POD, based on the algebraic decomposition method PCA (Principal Component Analysis)[7]. Table 2 summarizes some details about MOR methods, and section 5 shows which methods we used in our examples.

A priori	A posteriori
PGD[1]	POD[3], Reduced Basis[11], Locally Linear Embedding (LLE)[12]
Pros (PGD): Solving directly in the reduced space. Non-orthogonality restriction in reduction.	Pros (POD): Optimality (2^{nd} Order tensors). Independent of the data source (non intrusive).
Cons (PGD): Non-optimal (convergence dependent). Knowledge of the governing equation.	Cons (POD): Precompute all of the solutions (curse of dimensionality).

Table 2. Some properties of model order reduction methods.

4. Data Assimilation Process

The data assimilation process is crucial for estimating the model parameter values $\boldsymbol{\mu}$ by solving the inverse problem. We define the following functional

$$\mathcal{J}(\boldsymbol{\mu}) = \sum_{j=1}^{\mathtt{n}_{\text{corr}}} \left(\boldsymbol{u}_{\text{pix}}^{\text{meas}}(\boldsymbol{x}_j) - \Pi_t\left(\boldsymbol{u}^{\text{MOR}}(\boldsymbol{x}_j, \boldsymbol{\mu})\right)\right)^2, \quad (1)$$

being $\mathtt{n}_{\text{corr}}$ the number of correspondences, $\boldsymbol{u}_{\text{pix}}^{\text{meas}}$ the 2D pixel coordinates measured in the image, $\boldsymbol{u}^{\text{MOR}}(\boldsymbol{x}, \boldsymbol{\mu})$ the 3D position predicted by the reduced-order parametric solution and Π_t the projection to the image plane at any instant t (frame). We estimate the set of values that minimize the projection error $\hat{\boldsymbol{\mu}} = \text{argmin}_{\boldsymbol{\mu}} \mathcal{J}(\boldsymbol{\mu})$ by using the well-known Levenberg-Marquardt algorithm [8] [9]

$$\begin{aligned}&[\boldsymbol{J}^T\boldsymbol{J} + \lambda\,\text{diag}(\boldsymbol{J}^T\boldsymbol{J})]\,\delta \\ &\qquad = \boldsymbol{J}^T\left[\boldsymbol{u}_{\text{pix}}^{\text{meas}}(\boldsymbol{x}_j) - \Pi_t\left(\boldsymbol{u}^{\text{MOR}}(\boldsymbol{x}_j, \boldsymbol{\mu})\right)\right],\end{aligned}$$

where $\boldsymbol{J}$ is the Jacobian matrix, defined as

$$\boldsymbol{J} = \frac{\partial \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}}. \quad (2)$$

Equation 2 can be expressed using a separate representation, being a great advantage of some of the MOR methods, like PGD, allowing to express the jacobian as

$$\boldsymbol{J}_k = \sum_{i=1}^{N} \boldsymbol{F}_i^1(\mu_1)\cdot\ldots\cdot\frac{\partial \boldsymbol{F}_i^k(\mu_k)}{\mu_k}\cdot\ldots\cdot\boldsymbol{F}_i^{\mathtt{n}_{\text{param}}}(\mu_{\mathtt{n}_{\text{param}}}).$$

We want to emphasize the power of this property, as it means we can precompute the derivative function of each parameter offline. These derivative functions are usually known as *sensitivities* of the solution. We can interpret them as the local variation of the function $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\mu})$ with respect to changes in a parameter μ_k. It avoids the exploration of all of the parametric space in search of derivatives, applying directly the differentiation on the separated vectors, involving a great computational reduction.

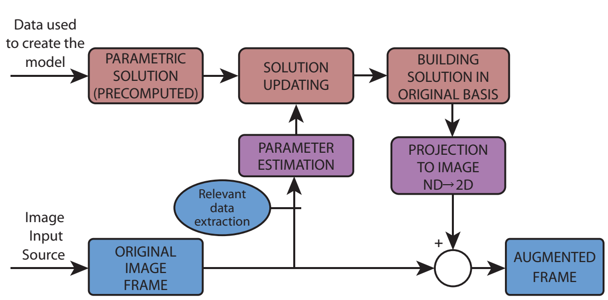

Figure 2. Assimilation process to update the parametric model using data coming from images to produce the augmented frame.

5. Examples

Four examples (all working at 30 fps) are provided to show how high-fidelity models can explain the real physics.

5.1. Car aerodynamics

The main goal is to use manifold learning techniques to *learn vehicle aerodynamics*. We precomputed the aerodynamics of a set of 80 cars coming from different segments and we applied Locally Linear Embedding, a non-linear dimensionality reduction technique. An L_2-norm error in the velocity field was estimated in 5 cars with a coarse mesh, ranging from 1.02% for a Hummer model to 0.56% for a classic sedan car. We use standard ORB-SLAM[10] implementation to build the map and a simple self-developed method based in plane detection and principal directions to estimate the car localization and shape. We can plot the estimated velocity field of a new LLE interpolated car in real time.

5.2. Foam Beam

Using simple homographies with fiducial markers and knowing the material properties, we were able to estimate the position of the load applied to a beam made in foam. We used a non-linear material behavior (Saint-Venant-Kirchhoff hyperelasticity) to describe the physics of the material deformation. PGD method was applied here to compute the displacements $\boldsymbol{u}(\boldsymbol{x}, s)$ in a separate way, being $\boldsymbol{x}$ the 3D space vector and s the 1D position of the load.

5.3. Rubber Piece

Using a modified version of ORB-SLAM to estimate mechanical deformations we were able to track the 3D movements of any ORB feature captured from the image that best fits to our precomputed model. We first solve the mechanical problem using a standard solver, project it using POD method (12.36% of memory cost with 0.005 mm of mean error) and in the on-line step we estimate the object deformation using ORB features, after applying a registration step based in the Iterative Closest Point [13] method. The great advantage here is that we can estimate the whole stress field of the object in real time, being able to predict the point where a piece may start failing.

5.4. Stanford Objects

General contact between virtual objects (Stanford bunny and dragon) and any object of the real world has been performed here. We precomputed the displacements of the objects when forces are applied in the contour, and using sparse Proper Generalized Decomposition (sPGD) [6] we can generalize to other contact points. We used the *Zed Mini* device from *Stereolabs* to directly estimate the depth to perform contact tasks. Neo-hook non-linear material was used to model the object deformations.

References

[1] Amine Ammar, Béchir Mokdad, Francisco Chinesta, and Roland Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 139(3):153–176, 2006. 2

[2] Alberto Badías, Icíar Alfaro, David González, Francisco Chinesta, and Elías Cueto. Reduced order modeling for physically-based augmented reality. *Computer Methods in Applied Mechanics and Engineering*, 341:53–70, 2018. 1

[3] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993. 2

[4] Francisco Chinesta, Pierre Ladeveze, and Elías Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395, 2011. 2

[5] Andrew J Davison. Futuremapping: The computational structure of spatial ai systems. *arXiv preprint arXiv:1803.11288*, 2018. 1

[6] Rubén Ibáñez, Emmanuelle Abisset-Chavanne, Amine Ammar, David González, Elías Cueto, Antonio Huerta, Jean Louis Duval, and Francisco Chinesta. A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. *Complexity*, 2018, 2018. 3

[7] Ian Jolliffe. *Principal component analysis*. Springer, 2011. 2

[8] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 3

[9] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 3

[10] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 3

[11] Janet S Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):777–786, 1989. 2

[12] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000. 2

[13] Radu B Rusu and S Cousins. Point cloud library (pcl). pages 1–4, 2011. 3