Computer Physics Communications 227 (2018) 8–16

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A high-speed tracking algorithm for dense granular media

Mauricio Cerda [a,b,*], Cristóbal A. Navarro [c], Juan Silva [d,f], Scott R. Waitukaitis [e], Nicolás Mujica [f], Nancy Hitschfeld [d]

[a] Anatomy and Developmental Biology Program, Institute of Biomedical Sciences, Facultad de Medicina, Universidad de Chile, PO Box 70031, Santiago, Chile
[b] Biomedical Neuroscience Institute, Independencia 1027, Santiago, Chile
[c] Instituto de Informática, Facultad de Ciencias de la Ingeniería, Universidad Austral de Chile, General Lagos 2086, Valdivia, Chile
[d] Departamento de Ciencias de la Computación, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Avenida Beauchef 851, Santiago, Chile
[e] Leiden Institute of Physics, Leiden University, Niels Bohrweg 2, 2333 CA Leiden, Netherlands
[f] Departamento de Física, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Avenida Blanco Encalda 2008, Santiago, Chile

ARTICLE INFO

Article history:
Received 13 September 2017
Received in revised form 15 January 2018
Accepted 10 February 2018
Available online 16 February 2018

Keywords:
Particle tracking
Peak detection
GPU computing
Granular media

ABSTRACT

Many fields of study, including medical imaging, granular physics, colloidal physics, and active matter, require the precise identification and tracking of particle-like objects in images. While many algorithms exist to track particles in diffuse conditions, these often perform poorly when particles are densely packed together—as in, for example, solid-like systems of granular materials. Incorrect particle identification can have significant effects on the calculation of physical quantities, which makes the development of more precise and faster tracking algorithms a worthwhile endeavor. In this work, we present a new tracking algorithm to identify particles in dense systems that is both highly accurate and fast. We demonstrate the efficacy of our approach by analyzing images of dense, solid-state granular media, where we achieve an identification error of 5% in the worst evaluated cases. Going further, we propose a parallelization strategy for our algorithm using a GPU, which results in a speedup of up to 10× when compared to a sequential CPU implementation in C and up to 40× when compared to the reference MATLAB library widely used for particle tracking. Our results extend the capabilities of state-of-the-art particle tracking methods by allowing fast, high-fidelity detection in dense media at high resolutions.

1. Introduction

Extracting quantitative information from image data is at the heart of scientific fields ranging from biology [1,2] to physics [3,4]. Often, the first step in image analysis is the identification of objects of interest, *i.e.* "particles". As examples, one can look to the identification of stars in telescope images, the tracking of individual cells in a biomedical experiment, or the tracking of grains of sand or other media in table top physical systems. While many algorithms exist to identify particles that are diffusely distributed throughout an image, e.g. stars in a telescope image, there are few algorithms that are able to reliably identify particles that are in close contact, as would be encountered in a dense population of living cells or a closely-packed colloidal system.

One particularly interesting case, which has been used to some degree as a model system, is the tracking used to study the dynamics of confined, quasi-two-dimensional granular systems. The typical experimental setup consists of granular particles placed in a shallow, enclosed container that is vibrated vertically. Measurements are performed by taking images or videos with a camera from above. Physically, this system is of interest because it undergoes state phase transitions (*e.g.* liquid to solid) when energy is injected into it during vibration [5,6]. The confined geometry has a great advantage because it permits the observation of both individual trajectories and collective behavior, which enables one to study both the microscopic and macroscopic dynamics. Computationally, this system creates a complex and challenging tracking situation because following particles through the phase transition requires particle identification in both diffuse and dense conditions.

From a computational point of view, this kind of identification and tracking analysis is performed with two main post-processing strategies: (i) particle-image velocimetry (PIV) and (ii) particle tracking (PT) [7]. PIV has the advantage that it is capable of extracting motion vector fields from images without requiring the identification of each particle. This is achieved by looking at the correlation of small windows of the field of view from one image to the next. For computer vision, this kind of method corresponds to the so called "optical flow" family of algorithms [7]. The main disadvantage of PIV methods is that the information about individual particles is lost, thus statistical measurements such as

* Corresponding author at: Anatomy and Developmental Biology Program, Institute of Biomedical Sciences, Facultad de Medicina, Universidad de Chile, PO Box 70031, Santiago, Chile.
E-mail address: mauriciocerda@med.uchile.cl (M. Cerda).

https://doi.org/10.1016/j.cpc.2018.02.010
0010-4655/

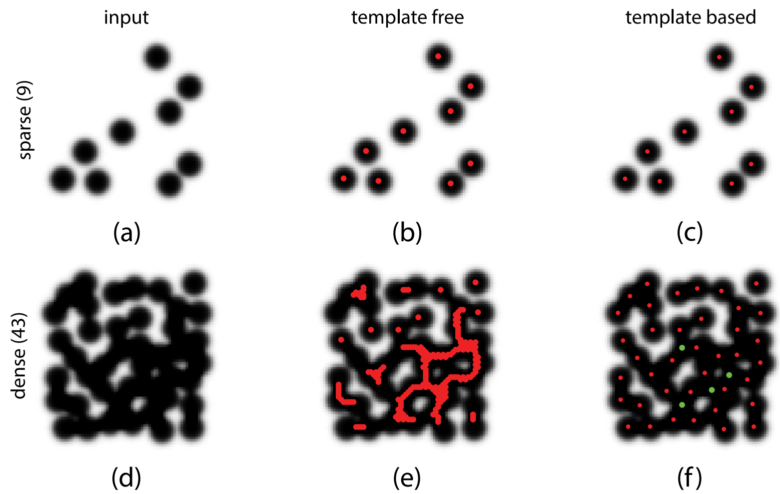

Fig. 1. Comparison of PT segmentation. The two scenarios: sparse (upper row), and dense (lower row) of PT are shown using synthetic images. The input column shows the granular media input image. Red dots show particle detection result for template free (column 2) and template based algorithm (column 3). Green dots show missing particle for the template based algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

velocity profiles are only approximated [8,9]. On the other hand PT methods have the advantage of extracting information at the single object level, being a direct and more precise way to look at the microscopic details [10]. However, PT has the disadvantages that it comes at significant computational cost and fails in situations where particles are so closely packed together that individuals are difficult to detect.

This work proposes an automatic PT method that can handle in-contact media as well as accelerate its performance by offloading its most computationally-intensive tasks to a Graphics Processing Unit (GPU). GPU computing is a useful tool for the field of computational physics as it can produce up to an order of magnitude of speedup on data-parallel problems when compared to a CPU [11,12]. We illustrate the strategy used to accelerate the algorithm with GPU computing and benchmark its effectiveness through analysis of synthetic images that imitate the well-studied yet computationally challenging physical system that was first studied by Olafesn and Urbach [13]: quasi-two-dimensional grains undergoing a phase transition from a diffuse-gas like state to a dense solid-like state. This system serves as an ideal candidate for this purpose as its transitions from a relatively easy tracking problem in the gas state, where particles are fully separated, to a nearly intractable one in the solid state, where particles clump together to create large borderless units. For a review of the physics behind this system see [14]. The proposed strategy allows us to handle experiments that are dense, long in duration, and recorded at high resolution all in a single computer. The paper is organized by first presenting a review of common PT methods in Section 2. Section 3 presents the new algorithm to handle in-contact particles, and Section 4 presents the GPU implementation of the proposed algorithm. Finally, in Section 5 we detail the results and in Section 6 we summarize the main findings.

2. Related work

Particle tracking problems have been extensively addressed in the literature and are typically separated into three stages: segmentation, correspondence, and parameter extraction [7]. The first stage aims to identify the objects in the image, from simple dot-like features to complex shapes. The correspondence stage attempts to match identified objects from one frame (or set of frames) to the next. At the end the set of identified trajectories are studied, where the parameter extraction stage obtains descriptors such as trajectory length, mean speed, directionality, and also domain-specific descriptors such as mean squared displacement, bond-orientational parameter, structure factor, velocity correlations and so forth. In the two dimensional granular system we work with here, correspondence is simple as data is obtained from controlled conditions with regard to lighting and camera parameters, thus a closest neighbor criterion has been reported as satisfactory [15]. For this reason, we focus our efforts on the segmentation stage, which has a higher degree of difficulty.

In the case of the system we study, the segmentation stage is a straightforward process if objects are sparse and have high contrast and uniform shape, such as those shown in the upper row of Fig. 1. This task becomes much more difficult in dense configurations, e.g. as shown in the lower row of Fig. 1. Within this context, several segmentation methods have been propose. These can be classified in as those that involve a template model and those that do not, which we detail in the following subsections.

2.1. Template Free PT methods (TFPT)

Perhaps the simplest strategies for performing image segmentation are the template free PT methods (TFPT), which are based on the detection of local intensity maxima (or minima) of the image. To illustrate the key idea, a practical realization of the algorithm is shown in Fig. 1b, e. The procedure has 3 steps: filtering (Gaussian filter with parameters $\sigma = 2.5$), a regional maximum operation (*imregionalmax* MATLAB function), and finally a step to reduce each object found to a single location. By construction, the location of a particle's center is limited to a precision of 1 pixel. The main drawback of this approach is that when particles are in-contact, their centers are no longer local maxima/minima, as shown in Fig. 1e. This causes most TFPT methods to systematically overdetect particles in high-density regions, as well as miss particles in other regions.

For an image of N pixels TFPT methods have a computational complexity of $O(N)$, where N is the number of image pixels [16]. Thus, for a movie of T frames, the overall complexity is $O(TN)$, which makes the algorithm fast and suitable for parallel architectures. The main issue with TFPT methods is that they fail in dense scenarios. The difficulty in these cases is that particles can be in contact, thus there is no measurable decrease in the image intensity between neighboring peaks. Another issue is that, due to the lack of geometrical constraints, there is no built-in crosscheck for the minimum spacing between particles, although this can be added as a post-processing step.