Computer Physics Communications 185 (2014) 268–273

Contents lists available at ScienceDirect

Computer Physics Communications

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

"Red Tweezers": Fast, customisable hologram generation for optical tweezers☆

Richard W. Bowman[a,b,*], Graham M. Gibson[a], Anna Linnenberger[c], David B. Phillips[d], James A. Grieve[d], David M. Carberry[d], Steven Serati[c], Mervyn J. Miles[d], Miles J. Padgett[a]

[a] SUPA, School of Physics & Astronomy, University of Glasgow, G12 8QQ, UK

[b] Department of Physics, University of Cambridge, CB3 0HE, UK

[c] Boulder Nonlinear Systems, 450 Courtney Way, Lafayette, CO 80026, USA

[d] HH Wills Physics Laboratory, University of Bristol, BS8 1TL, UK

ARTICLE INFO

Article history:
Received 10 July 2013
Received in revised form
5 August 2013
Accepted 8 August 2013
Available online 21 August 2013

Keywords:
Optical tweezers
Hologram
OpenGL
GPGPU
Spatial light modulator
LabVIEW
Microscopy

ABSTRACT

Holographic Optical Tweezers (HOT) are a versatile way of manipulating microscopic particles in 3D. However, their ease of use has been hampered by the computational load of calculating the holograms, resulting in an unresponsive system. We present a program for generating these holograms on a consumer Graphics Processing Unit (GPU), coupled to an easy-to-use interface in LabVIEW (National Instruments). This enables a HOT system to be set up without writing any additional code, as well as providing a platform enabling the fast generation of other holograms. The GPU engine calculates holograms over 300 times faster than the same algorithm running on a quad core CPU. The hologram algorithm can be altered on-the-fly without recompiling the program, allowing it to be used to control Spatial Light Modulators in any situation where the hologram can be calculated in a single pass. The interface has also been rewritten to take advantage of new features in LabVIEW 2010. It is designed to be easily modified and extended to integrate with hardware other than our own.

Program summary

Program title: Red Tweezers.

Catalogue identifier: AEQH_v1_0.

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEQH_v1_0.html.

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland.

Licensing provisions: GNU General Public License.

No. of lines in distributed program, including test data, etc.: 79147.

No. of bytes in distributed program, including test data, etc.: 11130332.

Distribution format: tar.gz.

Programming language: LabVIEW 2010, C++, OpenGL Shader Language.

Computer: Intel-based personal computer, nVidia or AMD graphics card supporting OpenGL 2.4.

Operating system: Microsoft Windows XP or later.

Has the code been vectorised or parallelised?: Designed for GPUs.

RAM: 2 Gb (highly dependent on video camera).

Classification: 18.

External routines: OpenGL, National Instruments Vision Development Module.

Nature of problem:
This program controls a holographic optical tweezers instrument, including GPU-accelerated rendering of holograms, monitoring the video feed and presenting a user-friendly interface to manipulate particles.

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author at: Department of Physics, University of Cambridge, CB3 0HE, UK.
E-mail address: richard.bowman@cantab.net (R.W. Bowman).

http://dx.doi.org/10.1016/j.cpc.2013.08.008

Solution method:
An extendable LabVIEW user interface, including a plugin architecture, is implemented to provide user-friendly control. The program also contains a component that accelerates the necessary digital holography step by rendering patterns using OpenGL shaders.

Restrictions:
The rendering engine is single-pass, i.e. iterative Fourier transform algorithms are not accelerated on the graphics card.

Additional comments:
The rendering engine is a separately-compiled executable controlled via UDP and can be used for other purposes. It allows simple OpenGL shaders to be used to render functions, without writing boilerplate code.

Running time:
This instrument control program is intended to run for as long as the experiment requires, over days if necessary. It can be re-started without losing most of its state information.

1. Introduction

Optical tweezers is a technique in which micron-sized particles are held and manipulated in a laser focus [1], and has become a versatile tool for measurement of tiny forces from, for example, single biomolecules [2,3]. Various strategies for creating multiple, steerable optical traps are commonly employed [4], including the use of acousto-optic deflectors and beamsplitters [5–7]. Holographic optical tweezers (HOT) systems use a programmable spatial light modulator (SLM) to split, steer and shape the laser beam [8–11] resulting in multiple traps which can be independently controlled in 3D. A typical HOT optical set-up is shown in Fig. 1.

SLMs are an increasingly popular tool for controlling and shaping light beams, used in many areas of physics such as laser marking [12], two-photon polymerisation [13] and HOT. Most currently available SLMs use a nematic liquid crystal layer to alter the optical path length of light reflected from the device. This is usually placed on top of a pixellated silicon backplane (and optionally a dielectric mirror) such that the electric field can be set independently for each pixel of the device, and we can thus impart an arbitrary phase function to the reflected beam.

Modulating the phase of an incident laser beam provides an unprecedented degree of control over optical fields. However, the calculation of holograms for these devices is often slow, which is a particular problem when some degree of interactivity is desirable, such as in optical micromanipulation. The massively parallel architecture of graphics processing units (GPUs) has been exploited to speed up the calculation of holograms by several orders of magnitude [14–18]. Our software allows LabVIEW control programs to take advantage of this speed improvement without having to write low-level OpenGL code. Modification of the algorithm used to calculate the holograms has hitherto required recompilation of the hologram generator; however our system allows the GPU code to be recompiled on-the-fly. This enables the use of a text string containing a single OpenGL shader function within LabVIEW (or another high-level control language such as Python or Matlab) to modify or replace the code running on the graphics card without an extra compilation step.

A useful laboratory tool must, in addition to its core function, provide an effective means of control. To this end, we have endeavoured to produce an intuitive, easy-to-use front-end in LabVIEW. It is our hope that this interface will also serve as a framework to customise the control logic for different applications, through providing LabVIEW subVIs to handle common tasks such as setting the locations of traps, rendering them to the SLM, and converting between SLM and camera coordinate systems. It is our hope that this will complement free software already available for simulating [19] and analysing [20] the data recorded by optical tweezers systems. It would also be possible to use our interactive interface with other freely available hologram generation code, for example the implementation of the weighted Gerchberg–Saxton algorithm by Bianchi et al. [18] or "HOTlab" by Persson et al. [17]. This last package communicates with the SLM through a PCIe card and also has its own LabVIEW interface.

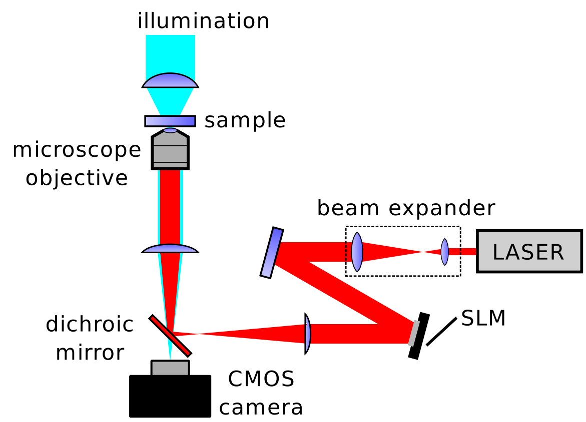

Fig. 1. A schematic of our holographic optical tweezers system.

2. Holographic optical tweezers control

In order for an HOT system to perform as a useful tool, particularly for the non-specialist, it is necessary to provide a simple, intuitive interface. By using interface elements users are already familiar with, for example clicking optical traps to select them and dragging to move them, we hope to make the technique more accessible and soften the learning curve for new users. By splitting up the available controls into tabs, and providing a plug-in area to add more controls, the program is broken into manageable chunks which can be easily modified to customise its behaviour.

Our interface centres around a live image, acquired from a camera on the tweezers system, with a tabbed toolbox at the side containing extra controls as shown in Fig. 2. Markers representing optical traps can be added to the image by double-clicking, or by using the controls in the toolbox. Once trap markers are placed, the LabVIEW program communicates the desired trap coordinates to the OpenGL "hologram engine", which renders a hologram to produce the correct intensity pattern. Markers can be selected by clicking them, and once selected can be dragged with the mouse. Multiple markers can be selected by holding the shift key. It is also possible to scale and rotate (in 2D) by holding the shift key while dragging. The selection is used by a number of operations, such as