Computer Science, Technology and Applications

Computer Vision and Simulation

Methods, Applications and Technology

Sherri Alexander
Editor

NOVA

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

COMPUTER VISION AND SIMULATION

METHODS, APPLICATIONS AND TECHNOLOGY

Computer Science, Technology and Applications

Additional books in this series can be found on Nova's website under the Series tab.

Additional e-books in this series can be found on Nova's website under the e-book tab.

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

COMPUTER VISION AND SIMULATION

METHODS, APPLICATIONS AND TECHNOLOGY

SHERRI ALEXANDER
EDITOR

nova
publishers
New York

NOTICE TO THE READER

The Publisher has taken reasonable care in the preparation of this book, but makes no expressed or implied warranty of any kind and assumes no responsibility for any errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of information contained in this book. The Publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or in part, from the readers' use of, or reliance upon, this material. Any parts of this book based on government reports are so indicated and copyright is claimed for those parts to the extent applicable to compilations of such works.

Independent verification should be sought for any data, advice or recommendations contained in this book. In addition, no responsibility is assumed by the publisher for any injury and/or damage to persons or property arising from any methods, products, instructions, ideas or otherwise contained in this publication.

This publication is designed to provide accurate and authoritative information with regard to the subject matter covered herein. It is sold with the clear understanding that the Publisher is not engaged in rendering legal or any other professional services. If legal or any other expert assistance is required, the services of a competent person should be sought. FROM A DECLARATION OF PARTICIPANTS JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Additional color graphics may be available in the e-book version of this book.

Library of Congress Cataloging-in-Publication Data
Names: Alexander, Sherri, editor.
Title: Computer vision and simulation: methods, applications and technology / editor, Sherri Alexander.
Description: Hauppauge, New York: Nova Science Publishers, Inc., [2016] |
Series: Computer science, technology and applications | Includes index.
Identifiers: LCCN 2016029819 (print) | LCCN 2016032704 (ebook) | ISBN 9781634857901 (hardcover) | ISBN 9781634858038 ()
Subjects: LCSH: Computer vision. | Computer simulation.
Classification: LCC TA1634 .C64895 2016 (print) | LCC TA1634 (ebook) | DDC 006.3/7--dc23
LC record available at https://lccn.loc.gov/2016029819

Published by Nova Science Publishers, Inc. † New York

CONTENTS

PREFACE

This book provides new research on computer vision and simulation. Chapter One studies and compares the representation capability of several different layers in convolutional neural network (CNN) showing that they contain more accurate information about the face image than to believe. Chapter Two finds, empirically, the best methods for describing a given texture using an ensemble to harness the discriminative power of different texture approaches. Chapter Three provides a computer study of the interaction of mercury with graphene. Chapter Four discusses the influence of yttrium(III) ion on calcium(II) and zinc(II) biospeciation in human blood plasma by computer simulation. Chapter Five reviews the simualation of diffraction gratings in the Fresnel diffraction regime using the *ab-initio* iterative Fresnel Integral Method (IFIM). Chapter Six introduces an example of a simple visual feedback control system of a mobile robot with an axis-symmetric shape for mechatronics education.

Chapter 1 - Most of recent advances in the field of face recognition are related to the use of a convolutional neural network (CNN) and the availability of very large scale training datasets. Unfortunately, large scale public datasets are not available to most of the research community, which therefore can hardly compare with big companies. To overcome this drawback, in this work the authors suggest to use an already trained CNN and the authors perform a study in order to evaluate the representation capability of its layers. Most of previous face recognition approaches based on deep learning use a CNN self-trained on a very large training set, taking one on the last intermediate layer as a representation and adding a classification layer trained over a set of known face identities to generalize the recognition capability of the CNN to a set of identities outside the training set. The idea is that the representation capabilities of the last one of two layers of a deep trained CNN is higher than traditional handcrafted features. In this work, starting from a CNN trained for face recognition, the authors study and compare the representation capability of several different layers in CNNs (not only the last ones) showing that they contain more accurate information about the face image than to believe. The proposed system extracts learned features from different layers of a CNN and uses them as a feature vector for a general purpose classifier. Moreover, the authors study the independence of the different sets of features used and between learned and handcrafted features, showing that they can be exploited to design an effective ensemble.

The proposed approach gains noticeable performance both in the FERET datasets, with the highest performance rates published in the literature, and the Labeled Faces in the Wild (LFW) dataset where it achieves good results. The MATLAB source of the authors' best ensemble approach will be freely available at https://www.dei.unipd.it/node/2357 "+Pattern Recognition and Ensemble Classifiers"

Chapter 2 - The goal of this chapter is to find empirically the best methods for describing a given texture using an ensemble to harness the discriminative power of different texture approaches. The authors begin the authors' investigation by comparing the performance of a large number of different texture descriptors and their fusions. The best fusion approach is then tested across a diverse set of databases and compared with some of the best performing approaches proposed in the literature. Whenever possible the original code of each approach is used on the datasets for fair comparison. Both stand-alone and ensembles of texture descriptors are investigated. In addition, some tests based on deep learning features are reported. The support vector machine is tested as a stand-alone classifier and as the base classifier in ensembles. Extensive experiments conducted on benchmark databases spanning several domains show that the authors' proposed approach outperforms recent state-of-the-art approaches. The proposed tool is available at (https://www.dei.unipd.it/node/2357 + Pattern Recognition and Ensemble Classifiers).

Chapter 3 - The contamination of natural waters and the lower atmosphere by heavy metal ions creates a serious ecological problem. Mercury is one of the most toxic heavy metals, because it is not biodegradable. The authors have studied the physical properties of mercury films on partially hydrogenated imperfect graphene by means of molecular dynamics at 300 K. Films prepared on the basis of three various types of the atomic interaction potential for mercury and other constant interaction potentials are considered. It is shown that the one most promising is the Schwerdtfeger potential function, at which mercury atoms do not fall into the divacancies present on graphene and atom packing with the lowest energy are realized in a liquid film and the film gradually fold into a drop. Another computer experiment has been employed to study rapid heating of a mercury film on graphene containing Stone–Wales defects. Hydrogenated edges of a graphene sheet withstand heating by 800 K. As the film contracts into a droplet, the horizontal component of the self-diffusion coefficient of Hg atoms monotonically decreases, while the vertical component passes through a deep minimum, which reflects the onset of droplet rising over the substrate. Formation of the droplet leads to a decrease in the blunt contact angle. Temperature–related changes in graphene manifest themselves as a rise in the intensity of additional peaks in the angular distribution of the closest neighbors, oscillatory pattern of the stress acting in its plane, and an almost linear growth of roughness. Molecular dynamics simulation of the bombardment of a target with a Xe_{13} cluster beam at energies of 5–30 eV and incidence angles of 0°–60° aiming to remove a mercury film from partially hydrogenated imperfect graphene has been performed. The graphene is completely cleaned of mercury at a cluster energy of $E_{Xe} \geq 15$ eV. Mercury is removed from the graphene film via sputtering of single atoms and droplet detachment. A stress in graphene resulting from forces normal to the sheet plane is noticeably higher than that due to forces acting in its plane. Bombardment at an angle of incidence of 45° is more efficient than that at incidence of 0° and 60° and leads to lower graphene roughness. Thus, mercury can be removed from graphene by heating or bombarding with heavy noble gas clusters.

Chapter 4 - The effect of yttrium(III) ion on calcium(II) and zinc(II) speciation in human blood plasma was studied by computer simulation using the program Hyss2009. Calcium-hydrogen carbonate $[CaHCO_3]^+$ and ternary zinc-cysteinate-citrate $[ZnCysCit]^{-3}$ complexes are predominant species of Ca(II) and Zn(II) ions in normal human blood plasma. Exogenously introduced yttrium(III) ion can compete with Ca(II) and Zn(II) ions for low molecular mass (LMM) ligands in blood plasma, thus influencing their biospeciation. The results showed that at the normal blood yttrium concentration all the Y(III) species are soluble and no precipitate appear. However, at total Y(III) concentration higher than 1×10^{-6} $molL^{-1}$, the insoluble species become dominant ($Y_2(CO_3)_2$ and YPO_4). At this concentration level of Y(III) the distribution of Ca(II) and Zn(II) species does not change appreciably. If the total concentration of Y(III) is higher than 1×10^{-3} $molL^{-1}$ its influence on biodistribution on Ca(II) and Zn(II) ions is significant. The concentration of free calcium ion increase from 79% to 86% and decreases [$CaHCO_3$] percentage. With further increasing of yttrium concentration (5×10^{-2} $molL^{-1}$), [$CaHCO_3$] disappear and dominant species is free calcium ion, whit redistribution of zinc species. Main species ZnCysCit (~38%) becomes minor species (<1%), while $ZnCys_2$ (~35%) and ZnCysHis (~20%) become major zinc species.

Chapter 5 - Computer-based virtual experiments and simulations in all branches of physical sciences and engineering has attracted wide spread interest among the researchers from all parts of the scientific world due to its multifaceted applications and versatility. Computer simulation of diffraction phenomena, including simulation of diffraction gratings, has widespread applications, since diffraction gratings, especially amplitude diffraction gratings, are used extensively in spectrographs and spectrometers. Usually, these are used in the Fraunhofer (far-field) regime. In this Chapter, the authors have used the *ab-initio* Iterative Fresnel Integral Method (IFIM) for the complete simulation of the near-field Fresnel diffraction images from any amplitude diffraction grating. The simulations can be performed in any PC in a reasonable amount of time and are executed in the MATLAB language. Complete explanations of the computational method, as applied to the diffraction gratings, are described, along with the simulation algorithms. Comparison of the simulated results with certain situations, which can be described by analytical equations, is made. The agreement confirms the correctness of the present simulation methods that will pave the way for future studies. The authors finally mention some extensions of the N-stilt problem, namely the application to tilted and rotating gratings and multi-wavelength illuminations.

Chapter 6 - Recently, visual feedback control system is becoming more attractive for mechatronics education due to the development of RGB-D cameras such as Kinect and Xtion. In this paper, an example of a simple visual feedback control system of a mobile robot with an axis-symmetric shape is introduced for mechatronics education which has to be demonstrated within a time limit of a lecture. Positions of a robot in image plane and projected plane can be calculated by referring to RGB stream and depth stream obtained from Xtion camera, respectively. As the first simple exercise, a virtual barrier fence is designed, so that a mobile robot can move within the virtual fence even without a real one. In addition, if a mobile robot has an axis-symmetric shape, e.g., circle, from the top view, it is difficult for a vision system to identify the orientation of the robot in the coordinate system. Another exercise is introduced to deal with an orientation following control using a forward direction vector. The forward direction vector can be calculated from point cloud data obtained by making the robot move forward for a short distance, e.g., 30 mm, every dynamic sampling period. The effectiveness and usability of the presented work is demonstrated experimentally.

In: Computer Vision and Simulation
Editor: Sherri Alexander
ISBN: 978-1-63485-790-1
© 2016 Nova Science Publishers, Inc.

Chapter 1

DEEP FEATURES COMBINED WITH HAND-CRAFTED FEATURES FOR FACE RECOGNITION

***Alessandra Lumini*[1,*], *Loris Nanni*[2] *and Stefano Ghidoni*[2]**
[1]DISI, Università di Bologna, Cesena, Italy
[2]DEI, University of Padua, Padua, Italy

ABSTRACT

Most of recent advances in the field of face recognition are related to the use of a convolutional neural network (CNN) and the availability of very large scale training datasets. Unfortunately, large scale public datasets are not available to most of the research community, which therefore can hardly compare with big companies. To overcome this drawback, in this work we suggest to use an already trained CNN and we perform a study in order to evaluate the representation capability of its layers. Most of previous face recognition approaches based on deep learning use a CNN self-trained on a very large training set, taking one on the last intermediate layer as a representation and adding a classification layer trained over a set of known face identities to generalize the recognition capability of the CNN to a set of identities outside the training set. The idea is that the representation capabilities of the last one

[*] Corresponding author: E-Mail: alessandra.lumini@unibo.it.

of two layers of a deep trained CNN is higher than traditional handcrafted features. In this work, starting from a CNN trained for face recognition, we study and compare the representation capability of several different layers in CNNs (not only the last ones) showing that they contain more accurate information about the face image than to believe. The proposed system extracts learned features from different layers of a CNN and uses them as a feature vector for a general purpose classifier. Moreover, we study the independence of the different sets of features used and between learned and handcrafted features, showing that they can be exploited to design an effective ensemble.

The proposed approach gains noticeable performance both in the FERET datasets, with the highest performance rates published in the literature, and the Labeled Faces in the Wild (LFW) dataset where it achieves good results. The MATLAB source of our best ensemble approach will be freely available at https://www.dei.unipd.it/node/2357 "+Pattern Recognition and Ensemble Classifiers"

Keywords: face recognition, similarity metric learning, deep learning, shallow descriptors

1. Introduction

Face recognition has been an area of intense research since the 1960s (Zhou et al. 2014), and the great interest about it is justified by the growing number of applications ranging from biometric security to criminal identification, from access management to human machine interaction, from photo album management in social networks to digital entertainment. Many innovative applications making use of this technology are continuously being developed at a rapid pace. Such applications can be categorized into three classes according to the goal of the face recognition task: face verification, where the aim the authentication of an individual to assert his/her identity; face identification, where the aim is to find a correspondence in a database of faces; face tagging, where the aim is to label face images based on identification. The face recognition problem consists in comparing two images of faces and determining whether both images frame the same person or not. The typical face recognition pipeline consists of four steps: face detection, face alignment, feature extraction and classification, where feature extraction is the crucial step. Most conventional face recognition techniques based on hand-crafted features such as Local Binary Patterns (LBP) (Ahonen et al. 2006), Local Phase Quantization (LPQ) (Chan et al. 2013) or Patterns of Oriented Edge

Magnitudes (POEM) (Vu 2013)(Nanni et al. 2013) perform well when facial images are captured in optimal (controlled) conditions; but their performance is quickly degraded when facial images are captured in the wild. Unfortunately, faces appearing in most applications, like social networks and digital entertainment, are acquired in uncontrolled conditions: they usually exhibit dramatic pose, expression and illumination variations and often a low image quality. The main difficulty of face identification consists in separating the specific features carrying information on the identity from the huge mass of features expressing other characteristics. It is still an open problem to find an ideal feature set for face recognition, robust under any acquisition setup. In the last few years, a new class of methods has been proposed, based on convolutional neural networks (CNN), and referred to as "*deep methods*" in opposition to "*shallow methods*" which are based on hand-crafted features. Deep methods learn their features during a training phase, and the set of learned features are more robust than handcrafted ones in detecting complex intra-personal variations.

Deep learning is a real breakthrough in the field of face recognition: a CNN model can not only characterize large data variations but also learn a compact and discriminative feature representation that can be generalized to dataset that were not involved in training. Deep learning has a great advantage over shallow methods in both identification and recognition. The deep features learned by trained CNN models are highly discriminative in performing large-scale face identification.

The first precursor paper in this area was proposed in 2005 (Chopra et al. 2005) and employed a convolutional neural network to learn a metric between face images. The deep learning approach was so dominant that after a decade of study, researchers (Taigman et al. 2014) have finally closed the "gap to human-level performance in face verification": DeepFace has achieved 97.25% accuracy on the LFW dataset, which is very close to human level accuracy (97.53%) using an ensemble of CNNs to find a good numerical representation of the face. Afterwards, many other deep learning approaches (Lu and Tang 2014)(Sun et al. 2014b)(Sun et al. 2014a)(Sun et al. 2015) have significantly outperformed previous shallow methods. For example, the approach based on Gaussian Processes and multi-source training sets in (Lu and Tang 2014) has achieved 98.52% accuracy on the LFW dataset, which is better than human performance.

Even if the LFW dataset is the de-facto benchmark for face recognition in-the-wild, some researchers have pointed out (Zhou et al. 2015) some limitations existing relationship between big training set and recognition performance. During the history of LFW benchmark, the largest performance improvements have been gained the last few years by deep learning techniques trained from huge datasets (from ~10000 samples in (Cao et al. 2013) to ~4,000,000 images in (Taigman et al. 2014)). The best performance using a training set of less than 10,000 images with deep learning was lower than 85%. Since Deep Learning approaches require millions of images for training, their results on benchmarks cannot be directly compared with approaches obtained using a testing protocol based on a few training samples.

The approach presented in this paper can be referred to as a "*transfer learning*" method. Unlike *shallow approaches*, it is not based on a representation of the face image by means of handcrafted descriptors only. It is also different from deep methods, since it is not based on a supervised deep neural networks specifically trained for this face recognition problem, i.e., to minimize the distance between features of the same identity while simultaneously decreasing intra-personal variations.

The system presented here is an evolution of the approach in (Lumini et al. 2016) where preliminary results about transfer learning were discussed as a direction for future research. The approach presented in (Lumini et al. 2016) is a *shallow* method based on a combination of handcrafted local image descriptors. The system is based on a combination of different preprocessing techniques and of several handcrafted feature extractors: then, similarly to this work, the classification is performed by an ensemble of classifiers. Moreover, in (Lumini et al. 2016) preliminary results about the combination of "learned" and "handcrafted" features were discussed.

In this paper we further evaluate the idea of performing "*transfer learning*" from an already trained CNN, analyzing the layer of the network which is most suited for face representation. The feature extraction step, obtained by convolving the input face image with a CNN and extracting the response of several different layers, is inserted into a well-tested framework consisting in face detection and cropping, frontalization, feature extraction and classification. The components used in each step have been already tested and tuned in (Lumini et al. 2016) and demonstrated good performance both in the FERET and LFW datasets. In this work we test the proposed system using different set of "learned" features, which have been obtained from the internal representation of a deep method, specifically a Convolutional Neural Network (CNN) trained for the face recognition problem.

The resulting fusion with the handcrafted features proposed in (Lumini et al. 2016) obtains, to the best of our knowledge, the highest mean accuracy ratings on the FERET datasets and very good results on the LFW dataset.

2. The Proposed Approach

The method proposed in this work is an evolution of the approach presented in (Lumini et al. 2016) where learned features are employed instead of handcrafted descriptors. The general schema of the approach is shown in Figure 1 and consists of the following steps:

- *Face detection and crop*: once the precise position of the face image is detected according to the approach in (Hassner et al. 2015a) the resulting face is tight cropped and aligned according to eye position;
- *Frontalization:* recent experiments (Lumini et al. 2016) demonstrated the importance of frontalization for precise face recognition also in presence of pose changes; in this work the approach proposed in (Hassner et al. 2015a) is used to synthesize frontal views of faces from the detected face;
- *Feature extraction:* feature extraction is performed using learned features obtained taking the response to the input face image of one intermediate layer of a CNN. Several experiments are reported to evaluate the best combination of layers;
- *Feature Transformation*: before classification the dimensionality of each descriptor is reduced via Principal Component Analysis (PCA) (Duda et al. 2000);
- *Classification:* a general-purpose classifier is trained on each reduced feature vector. The final decision is then determined according to the sum rule by summing up the scores/similarity values (SIM_i) obtained from each classifier. In this work, the simple angle distance is used in the FERET datasets, where the aim is identification. SML classifier (Cao et al. 2013) is used on the LFW dataset, where the aim is to verify a given match.

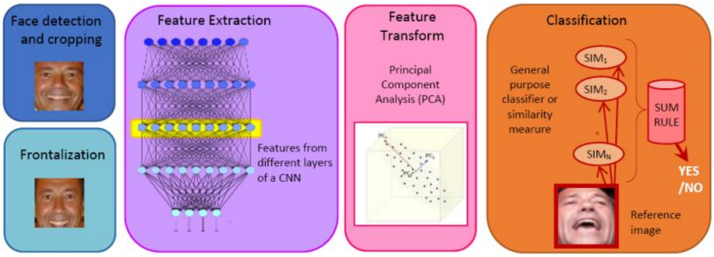

Face detection and cropping
Frontalization
Feature Extraction
Features from different layers of a CNN
Feature Transform
Principal Component Analysis (PCA)
Classification
General purpose classifier or similarity mearure
SIM$_1$
SIM$_2$
SIM$_N$
SUM RULE
YES /NO
Reference image

Figure 1. Schema of the proposed face recognition ensemble.

2.1. Hard Frontalization (HF)

Hard Frontalization (HF) (Hassner et al. 2015b) is a technique that uses a unique 3D geometrical shape to obtain a frontal view from a set of face pictures acquired in the wild from different angles. This is in contrast to the other approaches that employ a different 3D shape for each person considered. While the last approach aims at a more accurate reconstruction, the first one assumes that the difference between a "standard" 3D shape and the real one can be neglected for the purpose of frontalization.

Images acquired in the wild are first processed to extract faces. Each face is rescaled to a standard size, and a set of 49 facial features are detected in the sample. Such features are exploited to estimate the 3×4 projection matrix *P* describing the camera pose under which the face was framed, i.e., the rototranslation between the camera and the framed face. The *P* matrix describes the geometrical association between each pixel and the portion of the face it represents. In other words, by knowing *P* it is possible to understand which part of the face is represented by each pixel found in the image, and it is also possible to project such pixel onto the standard 3D face model considered in the HF algorithm, thus associating a color to each 3D location. The model itself is supposed to have a plane of symmetry, therefore some parts of the model that are not seen are filled with the color of the symmetrical locations; for example, if the left half of the mouth is seen in the image, the right half will be completed exploiting the symmetry.

The HF algorithm generates the frontalized view starting from the textured 3D model described above following a four step process. In the first step, a frontal synthetic view of the model is obtained by projecting the 3D model using a camera matrix whose rotation matrix and translation vector define a frontal projection that is used as a reference coordinate system. The second step generates the frontal pose synthesis by projecting the facial features from the 3D model onto such reference system. Step 3 deals with visibility estimation, which depends on the projection of the reference 3D model onto the given view. Finally, in step 4 the detection problems introduced by conditional soft-symmetry are tackled; this is done based on a standard representation based on LBP (Local Binary Pattern) and an SVM classifier.

2.2. Feature Extraction

In this paper we extract “learned” features from a CNN already trained for face recognition. However, we do not rely only on the data provided by the last layer of the CNN, as it is usually the case. Rather, we consider the information provided by other layers throughout the network (deep layers). The layers of a CNN can be considered as a set of features that are automatically learned during the training phase, and whose characteristics depend on the depth of the layer itself. Those layers that are closer to the input data, process information coming from a small neighborhood of pixels, and extract low-level, local features. Conversely, layers that are far from input data are made of nodes getting input that has passed through several processing steps, and depends on a larger set of pixels: this leads to the conclusion that such layers provide as output high-level, global features. The transition from local to global features can be seen as a gradual process that is the consequence of the peculiar scheme of the connections among nodes in a CNN, that makes it particularly suited for processing 2D data, as it is the case of images.

The features extracted from different stages of a CNN are used in the same way as it usually happens with hand-crafted features: the feature vectors become the input of a classifier – a set of SVMs in our case – that is trained to solve the face classification problem. This structure, composed of a previously trained classifier connected to a second classification stage for changing the problem to be solved, is known as transfer learning. It is particularly convenient when dealing with Deep Neural Networks (DNNs), because it allows to skip the training phase of such networks, which is computationally very intensive and requires a huge number of samples. Instead, the training of a set of SVMs requires smaller datasets and reduced computational effort.

The transfer learning scheme described above has been applied considering different combinations for the layers of the CNNs to be used as features. This enables to investigate the representation capabilities at various depths of a convolutional network, and the dependencies among their information representation. In this study, the CNN presented in (Parkhi et al. 2015) was considered: it is a VGG- Very-Deep-16 CNN architecture whose models, trained on a very large face collection, are freely available for downloadin

2.3. Feature Transform

The features described above represent a high amount of data that could cause the system to fall into the curse of dimensionality. To cope with this problem, dimensionality reduction methods have been applied. The best performance was achieved using PCA, Principal Component Analysis (Duda et al. 2000), which is a common approach. This technique generates a projection of the original space onto a reduced number of directions in order to maximize the variance of the projected vectors. In our experiments, the orthogonal basis used for projecting the features expresses 99% of the input variance. When the classifier is chosen to be an SML, the first 300 components are selected, as suggested in (Lumini et al. 2016).

2.4. Classification

The descriptors previously detailed are processed using a separate distance function or classifier (depending on the problem being addressed) for each feature. Such functions are then combined by sum rule to obtain the final classification output. This technique was selected because it does not require a deep analysis of the uncertainty space of the ensemble classifiers, as it was performed in (Fernández-Martínez and Cernea 2015).

The similarity function chosen for comparing faces in the experiments on identification (run on the FERET datasets) are angle distance. The angle distance α between two vectors v_1 and v_2 is evaluated as:

$$\alpha = \sin^{-1} \frac{v_1 \times v_2}{\|v_1\| \|v_2\|},$$

and represents the size of the angle defined by the two directions defined by the vectors. A different function is used for the experiments aiming at verifying given matches, run on the LFW dataset. In this case, a general purpose binary classifier – Similarity Metric Learning or SML (Cao et al. 2013) in our experiments – is used to distinguish between good and bad match. SML is a novel regularization framework proposed for learning similarity metrics for unconstrained face verification. The similarity function between the images x_i, x_j is defined as:

$$f_{M,G}(x_i, x_j) = s_G(x_i, x_j) - d_M(x_i, x_j)$$

where $s_G(x_i, x_j)$ and $d_M(x_i, x_j)$ are a weighed similarity and a weighed distance, respectively. The weight matrices G and M are learned from the training set with the goal of being robust to large intra-personal variations.

3. EXPERIMENTS

3.1. Datasets

The performance of the proposed approach was assessed on the FERET (Phillips et al. 2000) and LFW (Huang et al. 2007) datasets. The FERET dataset was collected in the context of FacE REcognition Technology (FERET) program; it is made of five datasets acquired in different time periods, under different weather conditions – the gallery set Fa (1196 images), and four datasets used for testing:

- Fb: 1195 samples acquired in the same day as Fa, using the same camera and under similar lighting conditions;
- Fc: 194 samples taken in the same day as Fa, but with a different camera and under different lighting conditions;
- Dup1: 722 samples acquired within one year since the acquisition of Fa;
- Dup2: 234 samples acquired more than one year after the acquisition of Fa.

FERET proposes a standard evaluation protocol that requires each test image to be compared against all the images in the gallery set. In our experiments we modified the images by aligning all the faces using the true eye positions and cropping the images to a fixed size of 110×110 pixels. Some samples from the FERET databases are reported in Figure 2.

Some samples from the FERET databases are reported in Figure 2.

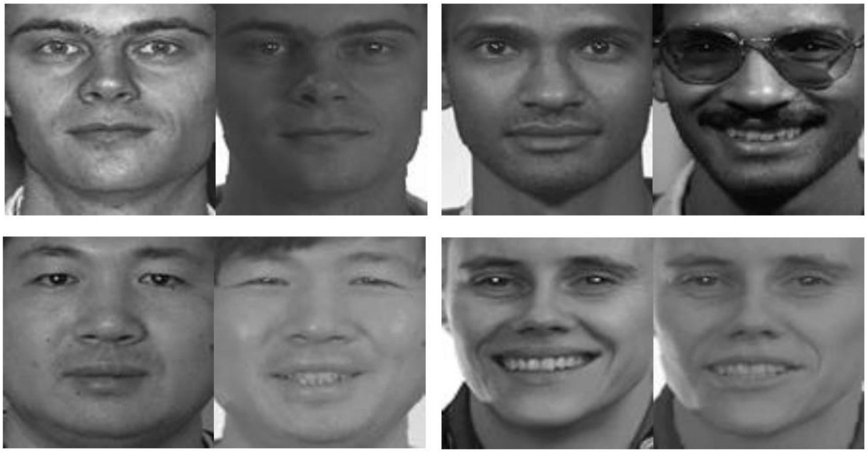


Figure 2. Samples from the FERET database.

The LFW (Labeled Faces in the Wild) database addresses the problem of unconstrained face recognition. It is made of more than 13000 internet images of 5749 celebrities, 1680 of which appear in two or more images. It represents a very difficult testbed because images were acquired in a totally uncontrolled way, and there is no control over imaging system, lighting, image quality and appearance of the subjects, since images of the same person at different ages are present. The database is divided into two views: the first one shall be used for training and testing (supports 10-fold validation), while view 2 is meant for benchmarking. Our experiments are designed to follow the testing protocol and dataset subdivision guidelines proposed by the authors of the database.

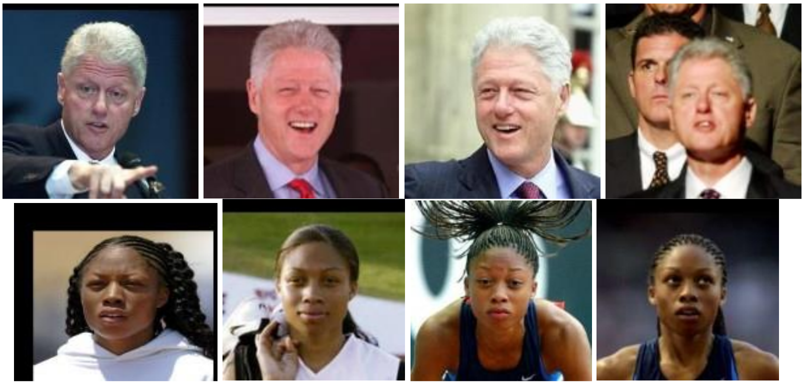


Figure 3. Samples from the LFW database.

Table 1. Accuracy obtained by our ensemble as a variation of the CNN layers for tight cropped images

Layers		**FERET recognition rate**				**LFW accuracy**
Level	**Dimensionality**	**Fb**	**Fc**	**Dup1**	**Dup2**	
30	100352	98.66	**100**	84.21	86.32	92.70
31	100352	99.33	98.97	90.3	89.74	93.07
32	25088	99.16	99.48	90.58	89.74	92.12
33	4096	98.66	98.97	91	**91.88**	92.15
34	4096	98.66	98.97	90.72	**91.88**	93.00
35	4096	98.66	98.97	89.89	**91.88**	92.82
36	4096	98.66	**100**	89.61	91.45	92.88
37	2622	97.49	98.97	87.67	87.18	92.30
[33 34]	8192	**98.74**	99.48	**90.86**	91.45	**93.43**
[36 37] (Lumini et al. 2016)	6718	98.33	99.48	89.06	91.03	93.22

The performance of our algorithms was measured by means of recognition rate for the FERET dataset, and accuracy for the LFW dataset, defined as the ratio between correct classification results (true positives and true negatives) and the total population. Some samples from the LFW database are reported in Figure 3.

Some samples from the LFW database are reported in Figure 3.

3.2. Results

The first experiment is aimed at evaluating the importance of the frontalization step: starting from results published in (Lumini et al. 2016), where the CNN outputs of the 37th and 36th fully-connected layers were used for describing the images, we evaluate the recognition performance on the LFW dataset.

The recognition accuracy reported using frontalization is 93.22, while it drops to 92.82 without frontalization. Therefore, all the following experiments will be carried out maintaining the frontalization step.

The second experiment is aimed at comparing the representation capabilities of several different layers of the VGG-Very-Deep-16 CNN proposed in (Parkhi et al. 2015). In the first column of Table 1 the layers used for representation purposes are reported (the presence of two numbers denotes the concatenation of two layers); the second column is the dimensionality of the feature vector, and the remaining columns report the recognition accuracies. The last row of Table 1 also reports the result of the fusion of the last two layers, already published in (Lumini et al. 2016). The experiments are carried out using the complete approach described in Figure 1 (including the steps of frontalization and feature transformation). The classifiers used in these experiments are SML for LFW and the angle distance for the FERET datasets. The best result is obtained when the CNN outputs of the 33th and 34th layers are used for describing the images.

All the results reported in Table 1 were obtained using a tight cropping, i.e., cropping faces so that the background is minimally involved in classification. For the sake of completeness we also report in Table 2 performance obtained using larger images (in Figure 4 an example of different cropping sizes is reported). Large crop include portions of hairs and clothes that can be considered as “soft biometrics”, useful to improve the recognition rate. In order to confirm this hypothesis we also tested the performance obtained by a large crop image where the face was removed (see Figure 2.d) obtaining a surprisingly high accuracy of 75.52%.

The third experiment, reported in Table 3, is a comparison with the state-of-art for both the FERET and LFW datasets for methods not based on outside training data. In Table 3 the best approach tested in this work is denoted by HERE (i.e., the one reported in last line of Table 1 and related to tiny cropped images and to the layers [33 34]). The last four rows of Table 3 report the weighed fusion of HERE and some of the best shallow methods proposed in the literature. Examining Table 3, it is clear that the system performance has significantly increased in the last few years: the proposed system gains very good performance in both the datasets. The fusion between the “learned features” proposed in this work and the handcrafted features of (Lumini et al. 2016) further improves both approaches obtaining one of the best recognition performance ever published for the FERET databases, and very valuable results in LFW too. When different approaches are combined, before the fusions, their scores are normalized to have zero mean and standard deviation 1. The methods 2 × X + Y means that the methods are combined with weighted sum rule where the weight of X is 2.

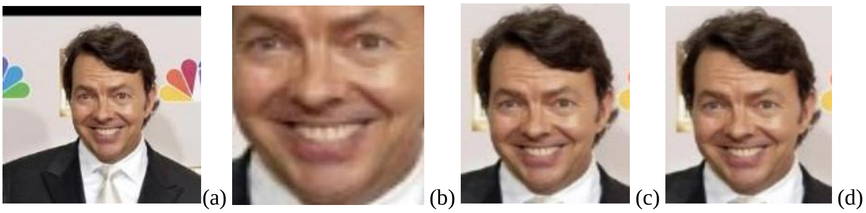


Figure 4. Sample of different cropping from the LFW database: (a) original image, (b) tight crop (c) large crop, (d) face removed.

Table 2. Accuracy obtained by our ensemble as a variation of the CNN layers for large cropped images

Layers		**FERET recognition rate**				**LFW accuracy**
Level	**Dimensionality**	**Fb**	**Fc**	**Dup1**	**Dup2**	
30	100352	98.83	98.97	69.81	60.68	93.87
31	100352	99.33	97.94	89.75	85.47	95.53
32	25088	99.67	98.97	91.27	87.61	94.72
33	4096	**100**	**99.48**	94.6	**94.02**	96.03
34	4096	**100**	98.97	94.6	93.59	96.95
35	4096	99.92	98.97	93.21	91.88	96.40
36	4096	99.92	**99.48**	92.94	92.74	**96.93**
37	2622	99.75	**99.48**	90.03	88.46	96.50
[33 34]	8192	**100**	**99.48**	**95.01**	**94.02**	96.85
[36 37] (Lumini et al. 2016)	6718	99.92	**99.48**	92.11	91.88	96.75

Our deep transfer learning approach does not achieve performance comparable with the state of the art of deep learning methods (e.g., see (Zhou et al. 2015)). However, we use the CNN only for extracting the features from the images, and only the standard training set of LFW is used for training SML. Interestingly, in the FERET dataset (high quality frontal images) the hand crafted features work better than the features extracted by CNN, but their fusion nevertheless permits to boost the performance.

Table 3. Comparison among the proposed ensemble with the state-of-the-arts approaches

Methods		FERET recognition rate					LFW accuracy
Reference	**Year**	**Fb**	**Fc**	**Dup1**	**Dup2**	**Avg**	
(Ahonen et al. 2004)	2004	93.0	51.0	61.0	50.0	63.8	---
(Zhang et al. 2005)	2005	94.0	97.0	68.0	58.0	79.2	---
(Deng et al. 2005)	2005	96.3	99.5	78.8	77.8	88.1	---
(Zhang et al. 2007)	2007	97.6	99.0	77.7	76.1	87.6	---
(Tan and Triggs 2007)	2007	98.0	98.0	90.0	85.0	92.8	---
(Xie et al. 2010)	2010	99.0	99.0	94.0	93.0	96.3	---
(Yang et al. 2012)	2012	99.7	99.5	93.6	91.5	96.07	---
(Vu 2013)	2013	99.7	100	94.9	94.0	97.2	86.2
(Nanni et al. 2013)	2013	98.7	100	94.6	93.6	96.7	76.9
(Chai et al. 2014)	2014	99.9	100	95.7	93.1	97.17	---
(Nowak and Jurie 2007)	2007	---	---	---	---	---	73.9
(Wolf et al. 2008)	2008	---	---	---	---	---	78.5
(Pinto et al. 2009)	2009	---	---	---	---	---	79.35
(Li et al. 2013)	2013	---	---	---	---	---	84.08
(Arashloo and Kittler 2013)	2013	---	---	---	---	---	79.08
(Simonyan et al. 2013)	2013	---	---	---	---	---	87.47
(Cao et al. 2013)	2013	---	---	---	---	---	88.5[1]
(Li and Hua 2015)	2015	---	---	---	---	---	88.97
(Arashloo and Kittler 2014)	2015	---	---	---	---	---	95.89
(Li et al. 2015)	2015	---	---	---	---	---	91.10
(Juefei-Xu et al. 2015)	2015	---	---	---	---	---	87.55
(Lumini et al. 2016)	2016	99.2	100	94.6	94.0	97.0	91.7
HERE	-	98.74	99.48	90.86	91.45	95.13	93.43
HERE+(Lumini et al. 2016)	-	99.58	100	97.37	96.15	98.27	93.32
2×HERE+(Lumini et al. 2016)	-	99.58	100	97.78	97.44	98.7	93.65
HERE + (Lumini et al. 2016) + (Cao et al. 2013)	-	---	---	---	---	---	93.97
2 × HERE+(Lumini et al. 2016)+(Cao et al. 2013)	-	---	---	---	---	---	94.08

CONCLUSION

In this work we studied the representation capability of convolution neural networks using intermediate layers of an already trained CNN for extracting

[1] Obtained using the source code shared by the authors of (Cao et al. 2013) and the testing protocol described in this work (which is slightly different from the one used in (Cao et al. 2013)).

features for the face recognition problem. Our experiments, carried out considering two of the most used benchmark databases in this field, show that not only the last two layers, but also several different internal layers in CNNs contain accurate information about the face image.

The proposed approach gains noticeable performance both in the FERET dataset, with the highest performance rates published in the literature, and the Labeled Faces in the Wild (LFW) dataset, where it achieves good results.

In the LFW dataset the approach proposed here, combined with the method in (Lumini et al. 2016), obtains a 93.65% accuracy, which can be further improved to 94.08% considering also the method in (Cao et al. 2013). The only approach which outperforms our method, without using outside training data, is (Arashloo and Kittler 2014) with 95.89% accuracy, but the authors do not share their source code, therefore results are not easily reproducible.

Another important aspect to be analyzed in face recognition approaches is the dimension of the cropping for the face image: our experiments demonstrate that there is a noticeable performance gap between loosely cropped and tightly cropped images. In this work we observe that even if a tight crop produces a performance drop, it is more fair for pure face recognition, since it allows to base the recognition task only on the face region, discarding possible information in the contours. Unfortunately, results reported in the literature using the LFW or FERET benchmark do not always clearly explain which kind of crop was used, therefore a fair comparison is not possible.

In the future, we plan to experiment CNN not specifically trained for the face recognition task (i.e., object recognition, scene classification, etc.) in order to evaluate the degree of independence of such sets of features and their ability to work with different classification problems.

REFERENCES

Ahonen, T., et al., (2004). Face Recognition with Local Binary Patterns. In *European Conference on Computer Vision*., pp. 469–481.

Ahonen, T., Hadid, A. and Pietikainen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), pp. 2037–2041.

Arashloo, S. R. and Kittler, J. (2014). Class-Specific Kernel Fusion of Multiple Descriptors for Face Verification Using Multiscale Binarised

Statistical Image Features. *Information Forensics and Security, IEEE Transactions on*, 9(12), pp.2100–2109.

Arashloo, S. R. and Kittler, J. (2013). Efficient processing of mrfs for unconstrained-pose face recognition. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on.*, pp. 1–8.

Cao, Q., Ying, Y. and Li, P. (2013). Similarity Metric Learning for Face Recognition. *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2408–2415. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6751410.

Chai, Z., et al., (2014). Gabor ordinal measures for face recognition. *IEEE Transactions on Information Forensics and Security*, 9(1), pp. 14–26.

Chan, C. H., et al., (2013). Multiscale local phase quantization for robust component-based face recognition using kernel fusion of multiple descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(5), pp. 1164–1177.

Chopra, S., Hadsell, R. and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, pp. 539–546.

Deng, W., Hu, J. and Guo, J. (2005). Gabor-eigen-whiten-cosine: a robust scheme for face recognition. In *Analysis and Modelling of Faces and Gestures.*, Springer, pp. 336–349.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000). *Pattern Classification*,

Fernández-Martínez, J. L. and Cernea, A. (2015). Exploring the uncertainty space of ensemble classifiers in face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *29*(03), p. 1556002.

Fernández-Martínez, J. L. and Cernea, A. (2014). *Numerical Analysis and Comparison of Spectral Decomposition Methods in Biometric Applications*, Available at: http://www.worldscientific.com/doi/abs/10.1142/S0218001414560011.

Hassner, T., et al., (2015a). Effective Face Frontalization in Unconstrained Images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).*

Hassner, T., et al., (2015b). Effective Face Frontalization in Unconstrained Images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Available at:\url{http://www.openu.ac.il/home/hassner/ projects/frontalize}.

Huang, G., et al., (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts Amherst Technical Report 07*, 49(07-49), pp. 1–11. Available at: http:// citeseerx.ist.psu.edu/viewdoc/download? doi= 10.1.1. 122.8268&amp;rep=rep1&amp;type=pdf\nhttps://www.cs.umass.edu/~el m/papers/lfw.pdf.

Juefei-Xu, F., Luu, K. and Savvides, M. (2015). Spartans: Single-sample Periocular-based Alignment-robust Recognition Technique Applied to Non-frontal Scenarios. *Image Processing, 24*(12).

Li, H., et al., (2015). Eigen-pep for video face recognition. In *Computer Vision--ACCV 2014*., Springer, pp. 17–33.

Li, H., et al., (2013). Probabilistic elastic matching for pose variant face verification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3499–3506.

Li, H. and Hua, G. (2015). Hierarchical-PEP Model for Real-world Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*., pp. 4055–4064.

Lu, C. and Tang, X. (2014). Surpassing Human-Level Face Verification Performance on LFW with GaussianFace. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*., pp. 3811–3819. Available at: http://arxiv.org/abs/1404.3840.

Lumini, A., Nanni, L. and Braham, S. (2016). Ensemble of texture descriptors and classifiers for face recognition. Applied Computing and Informatics, to appear 2016 doi:10.1016/j.aci.2016.04.001.

Nanni, L., et al., (2013). Ensemble of Patterns of Oriented Edge Magnitudes Descriptors For Face Recognition. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV).*

Nowak, E. and Jurie, F. (2007). Learning visual similarity measures for comparing never seen objects. In *IEEE Conference on Computer Vision and Pattern Recognition*., pp. 1–8.

Parkhi, O. M., et al., (2015). Deep face recognition. *Proceedings of the British Machine Vision*.

Phillips, P. J., et al., (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22(10), pp. 1090–1104. Available at: http:// ieeexplore. ieee.org/lpdocs/epic03/wrapper.htm?arnumber=879790.

Pinto, N., DiCarlo, J. J. and Cox, D. D. (2009). How far can you get with a modern face recognition test set using only simple features? *2009 IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, pp. 2591–2598.

Simonyan, K., et al., (2013). Fisher Vector Faces in the Wild. In *Procedings of the British Machine Vision Conference 2013*. pp. 8.1–8.11. Available at: http://www.bmva.org/bmvc/2013/Papers/paper0008/index.html.

Sun, Y., Wang, X. and Tang, X. (2014a). Deep Learning Face Representation by Joint Identification-Verification. In NIPS. pp. 1–9. Available at: http://arxiv.org/abs/1406.4773.

Sun, Y., Wang, X. and Tang, X. (2014b). Deep Learning Face Representation from Predicting 10,000 Classes. In CVPR. pp. 1–9. Available at: http://arxiv.org/abs/1406.4773.

Sun, Y., Wang, X. and Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. Cvpr.

Taigman, Y., et al., (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 8. Available at: http://www.cs.tau.ac.il/~wolf/papers/deepface_11_01_2013.pdf.

Tan, X. and Triggs, W. (2007). Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition. *Analysis and Modeling of Faces and Gestures*, 4778, pp. 235–249. Available at: http://eprints.pascal-network.org/ archive/ 00003663/.

Vu, N. S. (2013). Exploring patterns of gradient orientations and magnitudes for face recognition. *IEEE Transactions on Information Forensics and Security*, *8*(2), pp. 295–304.

Wolf, L., Hassner, T. and Taigman, Y. (2008). Descriptor based methods in the wild. *Science*, 6, pp. 1–14. Available at: http://hal.inria.fr/inria-00326729/\
nhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.150.2875&amp;rep=rep1&amp;type=pdf.

Xie, S., et al., (2010). Fusing local patterns of gabor magnitude and phase for face recognition. *IEEE Transactions on Image Processing*, *19*(5), pp. 1349–1361.

Yang, M., et al., (2012). Monogenic Binary Coding: An Efficient Local Feature Extraction Approach to Face Recognition. *Information Forensics and Security, IEEE Transactions on*, *7*(6), pp. 1738–1751.

Zhang, B., et al., (2007). Histogram of Gabor phase patterns (HGPP): A novel object representation approach for face recognition. *IEEE Transactions on Image Processing*, *16*(1), pp. 57–68.

Zhang, W., et al., (2005). Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A novel non-statistical model for face representation and recognition. *Proceedings of the IEEE International Conference on Computer Vision, I*, pp. 786–791.

Zhou, E., Cao, Z. and Yin, Q. (2015). Naive-Deep Face Recognition: Touching the Limit of LFW Benchmark or Not? *CoRR*, abs/1501.0. Available at: http://arxiv.org/abs/1501.04690.

Zhou, H., et al., (2014). Recent advances on singlemodal and multimodal face recognition: A survey. *IEEE Transactions on Human-Machine Systems*, *44*(6), pp. 701–716.

In: Computer Vision and Simulation
Editor: Sherri Alexander

ISBN: 978-1-63485-790-1

Chapter 2

REVIEW ON TEXTURE DESCRIPTORS FOR IMAGE CLASSIFCATION

Loris Nanni[1,*,†]***, Michelangelo Paci***[2,†]***,***
Florentino Luciano Caetano dos Santos[2,†]***,***
Sheryl Brahnam[3,†] ***and Jari Hyttinen***[2,†]
[1]Department of Information Engineering,
University of Padua, Padua, Italy
[2]Department of Electronics and Communications Engineering,
Tampere University of Technology, BioMediTech, Tampere, Finland
[3]Computer Information Systems, Missouri State University,
National, Springfield, MO, US

ABSTRACT

The goal of this chapter is to find empirically the best methods for describing a given texture using an ensemble to harness the discriminative power of different texture approaches. We begin our investigation by comparing the performance of a large number of different texture descriptors and their fusions. The best fusion approach is then tested across a diverse set of databases and compared with some of the best performing approaches proposed in the literature. Whenever possible the original code of each approach is used on the datasets for fair

[*] Corresponding author: E-mail: loris.nanni@unipd.it (LN).
[†] Equal contribution.

comparison. Both stand-alone and ensembles of texture descriptors are investigated. In addition, some tests based on deep learning features are reported. The support vector machine is tested as a stand-alone classifier and as the base classifier in ensembles. Extensive experiments conducted on benchmark databases spanning several domains show that our proposed approach outperforms recent state-of-the-art approaches. The proposed tool is available at (https://www.dei.unipd.it/node/2357 + Pattern Recognition and Ensemble Classifiers).

Keywords: texture descriptors, ensemble, local binary patterns, deep learning, support vector machines

1. Introduction

Texture analysis is a branch of computer vision that aims to solve problems related to texture (e.g., classification, segmentation, etc.). The idea is to extract a compact representation of the texture information included in an image so that it can be processed through mathematical and logical algorithms. Applications requiring an understanding of texture cover a wide spectrum: medical [1–3] and biological [4–7] image processing, material sciences [8], food quality assessment [9,10], and even music classification [11]. Yet, despite the fact that the word *texture* has been in common use in the field of computer vision for the last fifty years, the term is still quite ambiguous; there is currently no clear and generally agreed upon definition of the term. Originally, the word *texture* was adopted from textiles, where it refers to the weave or structure of various threads (whether loose or tight, even or mixed) [12,13]. In the computer vision literature, many attempts have been made to capture and formalize the idea of texture [14–18]. Despite differences in approaches, two main ideas about texture have emerged from the literature: (i) texture involves the occurrence of repetitive patterns characterized by the same size in the textured area and (ii) the patterns' non-random spatial organization in a region is larger than the patterns' size [19]. Texture analysis has proven to be an incredibly versatile discipline.

A number of important approaches have been developed to tackle texture-based segmentation and classification. To analyze a texture's spatial distribution, statistical approaches derive a set of local statistical features from the distribution of the grey values at each point in the image [20]. First order statistics (such as mean, variance, skewness, kurtosis, etc.) estimate properties of individual pixels, consequently ignoring the spatial relations between neighboring pixels. Increasing the number of pixels for the computation of the local features allows higher-order statistics to be used. The most common example of second order statistical features was presented by Haralick et al. in 1973 [21], where the spatial relationships between adjacent pixels were summarized in the grey level co-occurrence matrix (according to a specific orientation and distance between pixels) and from which a set of fourteen features was extracted.

Spectral approaches exploit image filtering to extract textural information from the image spectra. Fourier transform is useful for describing the orientation of bidimensional periodic patterns, but it lacks information about the patterns spatial localization. An improved description of the spatial localization is offered by Gabor filters [22], especially when gathered in a bank of filters including different scales and orientations. A similar approach exploits the wavelet transform, which still allows for a multiscale description of the image; this approach offers a large variety of wavelets to choose from to meet the needs of specific applications [23].

But the real Copernican Revolution happened with the introduction of neighborhood based texture descriptors, i.e., operators that assign a label to each pixel of the image according to the information gathered from a neighborhood of that pixel. The most seminal example is the Local Binary Pattern (LBP) [24], which computes labels according to the binarization of the difference between the grey intensities of the neighboring pixels and the central pixel in the neighborhood. The simplicity and effectiveness of the LBP formulation generated a plethora (of which the following list is a simplified and incomplete representation) of LBP-based texture descriptors. These are based on changes in (i) the coding which was extended to ternary and quinary coding [6, 25], (ii) the neighborhood shape, considering other geometric loci instead of the original circular neighborhood [26], (iii) the area of the image from which the textural information is extracted (e.g., considering textures from the edges in the image) [27] or even (iv) by pushing LBP to a totally new level, i.e., by not considering anymore the spatial relationships between pixels but rather the spatial relationships to LBP labels in the image, such as in the Rotation Invariant Co-occurrence among Adjacent Local Binary Pattern (RIC-LBP) [28].

Recently [26, 29], the effectiveness of preprocessing has been demonstrated for texture-based image classification. Preprocessing not only reduces noise and enhances particular characteristics of the image but it also augments the dataset under investigation and enhances the classification performance. The simplest approaches consist in rotating and mirroring the image [30–32]. However, the very same techniques introduced as spectral approaches mentioned above have proved their utility in generating new views of a given image, e.g., by mean of Gabor filters or wavelet transform [26].

Finally, it has been theoretically and empirically demonstrated that combining different texture descriptors (possibly combining them as well with some preprocessing procedures) allows for further improvements of the performance of classification techniques [26, 33, 34]. This is due to the different information that different descriptors extract from the image. For example, LBP informs on the spatial organization of the gray tones, Local Phase Quantization (LPQ) moves the analysis in the Fourier domain, exploiting the invariance of the phase of the image 2D Fourier transform, RIC-LBP informs on the spatial organization of whole patterns, and so on. Moreover, different kinds of features not strictly related to textures, e.g., morphological characteristics or colors, can also be effectively combined with texture information.

This chapter aims to fulfill two distinct goals:

1. Providing a survey of state-of-the-art texture descriptors that have most-widely been used during the last decade;
2. Providing one or more sets of texture descriptors, not tailored to a specific problem (e.g., classification of subcellular parts or crowd identification in images from security cameras), that perform well on a large variety of datasets. This was accomplished by testing different state-of-the-art descriptors on several different datasets.

2. Methods

In this chapter, we present different methods for building ensembles that enhance the classification of image datasets. All approaches consist in augmenting the feature set obtained by means of state-of-the-art texture descriptors that describe each image in the dataset.

In this section, we describe the following: i) the stand-alone descriptors tested in this work (see Table 1), ii) ensembles of variants of the Local Binary Pattern Histogram Fourier, iii) the preprocessing techniques used in this work, iv) methods for splitting an image into different images for building an ensemble, v) an ensemble of different Binarized Statistical Image Features sets, and vi) an ensemble obtained using deep transfer learning for applying non-handcrafted features. Regarding ensembles, whenever a given ensemble is built up with *x* descriptors, the scores are summed and normalized by dividing the sum by *x* (this is useful when different ensembles are combined). Note as well that before each fusion the scores of each descriptor are normalized to mean 0 and standard deviation 1.

2.1. Descriptors

In Table 1 we list the stand-alone descriptors that are tested in this work. The most important of these are described below.

Table 1. Texture descriptors and their parameter sets. This table is partly reproduced from [26] under the Creative Commons Attribution (CC BY) license

Acronym	Descriptor and parameters	Ref
LHF	Multi-scale *LBP* Histogram Fourier features with 2 (*radius, neighboring points*) configurations: (1,8) and (2,16).	[35]
LPQ	Multi-scale Local Phase Quantization with *radius* ∈ {3, 5}.	[36]
HOG	Histogram of Oriented Gradients with 30 cells (5 by 6).	[37]
LBP	Multi-scale Uniform *LBP* with 2 (*radius*, *neighboring points*) configurations: (1,8) and (2,16).	[24]
LTP	Multi-scale Uniform *LTP* with 2 (*radius, neighboring points*) configurations: (1,8) and (2,16).	[33]
MOR	Strandmark Morphological Features.	[38]
LCP	Multi-scale Linear Configuration model with 2 (*radius*, *neighboring points*) configurations: (1,8) and (2,16).	[39]
NTL	Multi-scale Noise Tolerant *LBP* with 2 (*radius*, *neighboring points*) configurations: (1,8) and (2,16).	[40]
DEN	Multi-scale Densely Sampled Complete *LBP* histogram with 2 (*radius*, *neighboring points*) configurations: (1,8) and (2,16).	[41]
CLBP	Completed *LBP* with 2 (*radius*, *neighboring points*) configurations: (1,8) and (2,16).	[42]
RICLBP	Multi-scale Rotation Invariant Co-occurrence of Adjacent LBP with *radius* ∈{1, 2, 4}.	[28]
WLD	Weber Law Descriptor.	[43]
HASC	Heterogeneous Auto-Similarities of Characteristics	[44]
Gab	The mean-squared energy and the mean amplitude were calculated from 5 different scale levels and 14 different orientations. In this way, a feature vector of size 5×14×2 is obtained.	[45]

Table 1. (Continued)

Acronym	Descriptor and parameters	Ref
RLBP	Rotated Local Binary Pattern with 2 (*radius, neighboring points*) configurations: (1,8) and (2,16).	[46]
MRE	Default parameter settings in the original journal paper were used: *radius* ∈ {2, 4, 6, 8} and a point set of 8. The image are resized to a fixed size (height of 150 pixel) before feature extraction.	[47]
MSJ	Using the default parameters for the code the authors shared with us.	[48]
LTDP	Local Directional Texture Pattern with two configurations: *mask size* ∈ {3, 5}.	[49]
LAP	Laplacian Features	[50]
HA	Average energy of the three high-frequency components is calculated up to the tenth level decomposition using both the scaling and the wavelet functions of the Haar wavelet.	[51]
DB4	Average energy of the three high-frequency components is calculated up to the tenth level decomposition using both the scaling and the wavelet functions of the DB4 wavelet.	[51]
COI	Average energy of the three high-frequency components is calculated up to the tenth level decomposition using both the scaling and the wavelet functions of the Coif2 wavelet.	[51]
FDCT	Fast Discrete Curvelet Transform via wedge wrapping.	[52]
DISC	Discriminative completed Local Binary Pattern, with the best approach used and labeled dis(S+M), as in the original paper; the number of neighborhoods is 8 and the *radius* ∈ {1, 3, 5}.	[53]
GO	Gaussians of Local Descriptors. Here we train a different support vector machine from each region of the spatial pyramid and combine them by sum rule. We also use one level spatial pyramid decomposition. The decomposition consists of the entire image, followed by level one, where the image is subdivided into four quadrants	[54]

2.2. Local Binary/Ternary Coding

The ternary coding aims i) to overcome some limitations of LBP [24], such as the high sensitivity to noise in near-uniform regions, and ii) to introduce a higher level of granularity, which allows the descriptor to catch a greater number of textural features [6]. The traditional LBP codes, as formulated in [24], are represented by the following equation:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(x)2^{p},$$

where $x = q_p - q_c$ is the difference between the intensity levels of the neighboring pixels (q_p) and the central (q_c) within a circular neighborhood of radius R and P neighboring pixels, while s(x) is the simple binary coding

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, otherwise \end{cases}$$

Consequently, each digit of a LBP code is assigned or 0 or 1, thus producing codes ranging in [0, 2^P-1].

The first step to obtain ternary descriptors consists in extending *s(x)*. In [25] and [6], two non-binary codings *s(x)* were proposed: ternary coding, which generates LTP and quinary coding which generates LQP. The ternary coding encodes the difference *x* with three values, by means of threshold τ:

$$s(x) = \begin{cases} 1, x \geq \tau \\ 0, -\tau \leq x < \tau \\ -1 \; otherwise. \end{cases}$$

The quinary coding further extends the non-binarization by using a couple of thresholds (τ_1, τ_2), where $\tau_1 < \tau_2$ encodes *x* with five different values:

$$s(x) = \begin{cases} 2, x \geq \tau_2 \\ 1, \tau_1 \leq x < \tau_2 \\ 0, -\tau_1 \leq x < \tau_1 \\ -1 \; -\tau_2 \leq x < -\tau_1 \\ -2 \; otherwise. \end{cases}$$

The higher verbosity of the ternary coding (3^P LTP possible codes compared to 2^P LBP codes for a *P-pixel* neighborhood) is then compensated by splitting the LTP histograms into binary subhistograms, which are then concatenated. Each LTP code is divided into a positive and negative binary pattern according to the sign of its components.

The multi-threshold approach can be extended to all those texture descriptors which exploit a binarization, e.g., LPQ. In [6], we extended the simple binary quantizer of LPQ [36] with its multi-ternary version (MLPQ), which was computed using $\tau \in \{0.2, 0.4, 0.6, 0.8, 1\}$.

In this chapter different configurations of LPQ are examined. We combine sets of LPQ extracted by varying the parameters: r (the neighborhood sizes and with $r \in \{1, 3, 5\}$), a (the scalar frequency and with $a \in \{0.8, 1, 1.2, 1.4, 1.6\}$), and ρ (the correlation coefficient between adjacent pixel values and with $\rho \in \{0.75, 0.95, 1.15, 1.35, 1.55, 1.75, 1.95\}$). This is the same set as proposed [55] coupled with the ternary encoding. Each extracted descriptor is used to train a different support vector machine (SVM), thereby avoiding the curse of dimensionality.

2.3. Ensemble of Local Binary Pattern Histogram Fourier

The Ensemble of Local Binary Pattern Histogram Fourier (MLHF) is an ensemble of variants of the LBP Histogram Fourier [33]. Seven descriptors (each trained by an SVM and with SVM scores summed and normalized by dividing the sum by seven) are used in building the ensemble.

1. FF (the original method): from each discrete Fourier transform (DFT), the first half of coefficients are retained.
2. DC: from each discrete Cosine transform (DCT), the first half of coefficients are retained.
3. WFF: the histogram is decomposed by Daubechies wavelet before DFT, and then FF is performed.
4. WDC: the histogram is decomposed by Daubechies wavelet before DCT, and then DC is performed.
5. AWFF: the histogram is decomposed by Daubechies wavelet before DFT (retaining all coefficients).
6. AWDC: the histogram is decomposed by Daubechies wavelet before DCT (retaining all coefficients).
7. AH: all bins of the histogram are retained.

As another variation (labeled MLHF_t), we test the same ensemble starting from the local ternary bins instead of the local binary bins.

2.4. Binarized Statistical Image Features

The canonical Binarized Statistical Image Features (BSIF) descriptor consists in assigning an n-bit label to each pixel of an image by a set (n) linear filters, thus projecting local image patches (size *l x l* pixels) onto a subspace. The *n*-bit label can be determined by binarizing as so:

$$s = \boldsymbol{Wx},$$

where $\boldsymbol{x}$ is the l^2 *x 1* vector notation of the *l x l* neighborhood and $\boldsymbol{W}$ is a *n x* l^2 matrix containing the compilation of the filters' vector notations. In detail, the *i*-th digit of *s* is a function of the *i*-th linear filter $\boldsymbol{w_i}$, and it is expressed as

$$s_i = \boldsymbol{w}_i^T \boldsymbol{x}.$$

Thus, each bit of the BSIF code can be obtained as

$$b_i = \begin{cases} 1, if\ s_i > 0 \\ 0, if\ s_i \leq 0 \end{cases}$$

The set of filters $\boldsymbol{w_i}$ is created by maximizing the statistical independence of the filter responses $\boldsymbol{s_i}$ on a set of patches from natural images by independent component analysis [56]. To improve the descriptive power of the canonical BSIF, it was expanded by varying the parameters of filter size (SIZE_BSIF, *size* ∈ {3, 5, 7, 9, 11}) and the threshold used for binarizing (FULL_BSIF, *th* ∈ {-9, -6, -3, 0, 3, 6, 9}). In total, this approach produces 35 possible (*size*, *th*) combinations, each one used to train a different SVM that were all combined by sum rule (we label this approach FBSIF).

2.5. Deep Transfer Learning Features

This is a set of features obtained through a remapping of a given deep Convolutional Neural Network (CNN) trained to solve one problem (e.g., a specific image problem, say, the classification of houses) so that it solves an entirely different problem (e.g., the classification of horses).

Abstractly, let $\boldsymbol{A} = \{a_0, a_1, \ldots, a_N\}$ be a set of labels representing some classification problem Q_1 into which a given input $\boldsymbol{x}$ should be classified, and let $\boldsymbol{y} = f_A(\boldsymbol{x})$ be the classification function implemented by the deep CNN. The deep learner takes the input vector $\boldsymbol{x}$ and provides an output vector $\boldsymbol{y}$ of length *N* that contains the scores assigned to all the possible outcomes considered in the classification problem. The features obtained through deep transfer learning take the input $\boldsymbol{x}$ of an entirely different problem, say Q_n, and sends input to several deep CNNS that have been trained to solve a set of very different classification problems, e.g.,, $Q_{1,}$ $Q_{2,}$ Q_3, each characterized by a different number of labels (***M***, ***N***, ***P***, respectively). These three deep networks provide three output vectors: $\{a_0, \ldots, a_M\}$, $\{b_0, \ldots, b_N\}$, $\{c_0, \ldots, c_P\}$. With deep learning transfer, these outputs subsequently become the inputs to three SVMs that are trained to provide a result that solves problem Q_n, and the outputs of the three SVMs are combined using the sum rule for a final decision. In other words, the SVMs learn the correlations between the problems $Q_{1,}$ $Q_{2,}$ Q_3 as they pertain to problem Q_n. In this double-stage classification pipeline, the deep CNNs implement models that have already been trained for different problems (stage 1). This is computationally a very expensive process. The SVMs in stage 2 learn the correlations between problems: a process that is computationally far less expensive.

A major assumption behind this approach is the existence of a certain degree of similarity between the different classification problems. In other words, it is assumed that a class a_i belonging to classification problem Q_n will have some degree of similarity with another class b_j belonging to classification problem Q_m. A certain amount of similarity is likely to exist with image problems. Taken alone, however, the degree of similarity between problems will probably be limited and lacking in discriminant power. It is necessary, therefore, to consider multiple classification problems simultaneously, allowing each problem to contribute its own degree of similarity to the problem at hand. Thus, a large number of output classes given in the different problems is key to the success of this approach.

However, the large number of resulting features (partially correlated) provided by each CNN (which are the inputs of each SVM), coupled with the relatively small number of training samples that are commonly available in the datasets of the new problems (in the order of 500-2000 images), is a major source of performance degradation due to the curse of dimensionality. This problem can be resolved using a random subspace (RS) ensemble [57, 58] in place of a stand-alone classifier. A RS ensemble reduces the number of input values by considering randomly drawn subsets of the input features. The RS ensembles used in this work are composed of 50 SVMs combined by sum rule and trained using a random subset of 50% of the given input values.

All input images are preprocessed before being sent to the CNNs. Because it is necessary to have a fixed size for the input data, images are resized to have the same number of rows and columns. The average training image is also subtracted from each image before processing as suggested in [59] to reduce the outlier affect. The trained CNN models used in this work are those available with the MatConvNet[1] toolbox [59]. These models result in 1000 categories.

In the following tests, we report two methods:

- DpSA: a single feature set is extracted using the imagenet-vgg-verydeep-19 model;
- DpEN: a set of features are extracted using the following models: imagenet-vgg-verydeep-19; imagenet-vgg-verydeep-16; imagenet-vgg-f; imagenet-vgg-m; imagenet-vgg-s; imagenet-vgg-m-2048; imagenet-vgg-m-1024; imagenet-vgg-m-128 pretrained; imagenet-caffe-ref; imagenet-caffe-alex.

2.6. Supervised Local Quinary Pattern

Proposed in [60], the goal of Supervised Local Quinary Pattern (SLQ) is to enhance performance by selecting a set of rotation invariant bins to train an RS of SVM.

This seven-step approach is accomplished as follows.

- Step 1: extract the rotation invariant bins (labeled SET-A) using LQP (with τ_1 = 3 and $\tau 2$ = 7).
- Step 2: retain in SET-A, 250 bins with highest variance.
- Step 3: select from SET-A a random subset of 125 features.
- Step 4: reduce the set of 125 features obtained in step 3 to 45 using principal component analysis (PCA) followed by neighborhood preserving embedding (NPE).
- Step 5: train and test a SVM using the features extracted in Step 4.
- Step 6: repeat steps 2–4 for a total of 50 times.
- Step 7: obtain a set of class similarity measures (labeled SCORE-A) by combining the 50 classifier results using the sum rule.

It will be noticed that in step 1 we use a variance selection process. We then select the histogram bins with the highest variance in the training data. The NPE feature transform is used in step 4 as a bin selector. PCA is used first

[1] The models are available at: http://www.vlfeat.org/matconvnet/pretrained/.

to reduce computation time; since 99.999% of the variance in the data is retained, little information is lost.

Finally, we combine scores obtained by SLQ with SVM trained using LTP based on uniform bins (as described in Table 1), before the fusion the scores of SLQ and LTP are normalized to mean 0 and standard deviation 1; we label this approach SLQT.

2.7. Preprocessing

The ensembling approach detailed in this section uses preprocessing before feature extraction to augment the features describing an image [61]. The general procedure consists in applying a specific preprocessing procedure to an image to obtain a new set of images, each of which is then processed by texture descriptors. For each descriptor, classification is performed separately with SVM.

We tested the following preprocessing methods: wavelet decomposition, Gaussian scale-space representation, gradient image, and an image enhancer.

2.7.1. Wavelet

A wavelet transform is used to obtain four new images from the original image: *viz.* its approximation and three new images containing the horizontal, vertical, and diagonal details of the original image. Using wavelets [62] for 2D image decomposition requires a 2D scaling function $\varphi(x,y)$ and three 2D wavelets functions, $\psi^i(x,y)$, where $i = \{H, V, D\}$ represents the three possible directions: horizontal, vertical, and diagonal.

The scaled and translated basis functions are defined as:

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\varphi\left(2^j x - m, 2^j y - n\right),$$
$$\psi^i_{j,m,n}(x,y) = 2^{j/2}\psi^i\left(2^j x - m, 2^j y - n\right), i = \{H, V, D\}.$$

For an arbitrary initial scalej_0, the three discrete wavelet transform functions W^H, W^V, and W^D of a $M \times N$ function $f(x,y)$ are formulated as:

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\varphi_{j_0,m,n}(x,y),$$
$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\psi^i_{j,m,n}(x,y), i = \{H, V, D\},$$

The $W_{\varphi}(j_0, m, n)$ coefficients represent an approximation of $f(x, y)$ at the j_0 scale. The $W_{\psi}^{i}(j, m, n)$ coefficients represent the three directional details at scales higher than j_0.

In our tests, the Daubechies wavelet family (*Wa*) with four vanishing moments was used.

2.7.2. Gaussian Scale-Space Representation

Gaussian Scale-Space Representation (MRS) smooths the original image by means of a 2D symmetric Gaussian low-pass filter. According to the number of different kernel sizes used for MRS, a more or less comprehensive multiscale representation of the original image is obtained. In the following, we used two kernel sizes, 3 and 5 pixels, thereby obtaining two new smoothed images for each original image.

2.7.3. Gradient Image

Gradient is an image operation that is commonly used to detect edges. Gradient Image (GR) calculates the magnitude of the gradients of each pixel in the x- and y-direction based on its neighbors. The processed image at coordinates (x, y) is given by the magnitude of the gradient vector at the same coordinates. Thus, each pixel of a gradient image measures the change in intensity in a given direction of the corresponding point in the original image.

2.7.4. Image Enhancement

In this work we apply a novel Exposure-Based Sub-Image Histogram Equalization (ENH) method proposed in [63] that enhances the contrast of low exposure grayscale images. Exposure thresholds are computed to divide the original image into subimages of different intensity levels and the histogram is cropped using a threshold value (the average number of gray level occurrences) to control the enhancement rate. The individual histograms of the subimages are then equalized independently, with all subimages finally being integrated into a single image, which is used for analysis.

2.8. Region-Based Descriptors

As the name suggests, the region-based descriptors are descriptors whose function is to extract information from the various “subcomponents” (*Edge*, *Saliency*, *Difference of Gaussians*, and *Wavelet*) that constitute an image [64].

2.8.1. Edge

In [65], it was hypothesized that, when we attentively observe an image, the most likely perceived locations present the highest spatial frequency edge information. Another approach [27] proposed the edge-based LBP variant Edge, which is an interesting approach since it uses the original image edge information to segment the LBP from the same image.

The following steps formulate the Edge descriptor:

1. Obtain the LBP image (*LBPI*) from the original image;
2. Use the Sobel filter to obtain the original image edge information. This produces two binary edge maps: *E*, where edge pixels are coded as 1 and the background as 0, and *NE*, the inverse of *E*, i.e., where the background is coded as 1 and the edge pixels 0;
3. Compute the two histograms (*hE* for edge pixels and *hNE* for non-edge pixels) after segmenting the *LBPI* with *NE* and *E*;
4. Retrieve the final histogram using weighted concatenation, fusing the *hNE* and *hE*:

$$H = (w_E \times H_E, w_{NE} \times H_{NE}), w_E > w_{NE,}$$

where w_E and w_{NE} are empirically determined weights.

Our approach does not merge the two histograms into a single feature vector but rather consists in training two different SVMs with *hE* and *hNE*, the results of which are then fused by sum rule.

Using the same approach as in Edge, other methods are applied for map extraction in order to obtain different histograms. Specifically, the following steps required are:

1. Apply the chosen descriptors to the texture image to get the labeled image *DescI*;
2. Compute two maps, Map^+ and Map^-, according to Edge, Saliency, Difference of Gaussians or Wavelet (details are provided below);
3. Compute the two histograms, H^+ and H^- by combining *DescI* with Map^+ and Map^-, respectively;
4. Train two separate SVMS with H^+ and H^- and combine results by sum rule.

2.8.2. Saliency

The method used for extracting the saliency map was taken from Hou et al. [66]. The first step in the methodology is to obtain the image signature, defined as:

$$ImageSignature(\boldsymbol{x}) = sign\big(DCT(\boldsymbol{x})\big),$$

with *sign()* as the sign operator and *DCT()* as the Discrete Cosine Transform. In [66] the authors demonstrate that by using the inverse procedure it is possible to approximate the support of the foreground of an image by the reconstructed image $\overline{\boldsymbol{x}}$:

$$\overline{\boldsymbol{x}} = IDCT\big(ImageSignature(\boldsymbol{x})\big).$$

From this transformation we can obtain the saliency map $\boldsymbol{m}$ as

$$\boldsymbol{m} = g * (\overline{\boldsymbol{x}}\; o\; \overline{\boldsymbol{x}}\,),$$

where g is a Gaussian kernel (standard deviation of the Gaussian kernel was set to 2) and *o* is the entrywise matrix product operator. The value g blurs the noise due to by the sign quantization. This approach is particularly useful when the foreground of the reconstructed image is clearly contrasted with the background. Using this methodology, two regions are extracted that present higher or lower saliency values compared with a prefixed threshold. Using two thresholds, 0.5 and 0.7, we obtain two saliency maps and four histograms for each image.

2.8.3. Wavelet

Wavelet decomposition was performed using Daubechies wavelet considering the horizontal, vertical and diagonal coefficients. Each matrix was normalized to the original image size, and the mean value was taken from each. This mean value is used to segment the original image into two regions. This produces six histograms, extracted from each image.

2.8.4. Difference of Gaussians

Difference of Gaussians (DO) computes two maps from a given image: *Map*$^+$ and *Map*$^-$, where *Map*$^+$ corresponds to the "positive" side of the image edges and *Map*$^-$ corresponds to the "negative" side of the image edges.

Textural information is extracted from these two maps, and a Gaussian low-pass filter (with *size* = 5 and *sigma* = [1, 4]) is applied.

3. DATASETS

Since the aim of this chapter is to present ensembling techniques that improve the classification power of state-of-the-art texture descriptors that are not tailored to a specific project but rather offer a more general approach, we test our ensembles on a wide variety of image datasets, ranging from biomedical images to images of insects and paintings. In this section each dataset is briefly presented with the most significant information about each dataset, e.g., the number of classes, reported in Table 2. Unless specified otherwise, each dataset is classified using a 5-fold cross-validation protocol. The area under the ROC curve (AUC) is the performance indicator since it provides a better overview of classification results. AUC is a scalar measure that can be interpreted as the probability that the classifier will assign a higher score to a randomly picked positive sample than to a randomly picked negative sample [67]. In the multi-class problem, AUC is calculated using the one-versus-all approach (i.e., a given class is considered “positive” while all the other classes are considered “negative”) and the average AUC is reported.

Here is the list of datasets, along with a brief description, that were used in our experiments:

- **PS**: the PAP SMEAR dataset contains 917 images acquired during Pap tests to identify cervical cancer diagnosis [68].
- **VI**: the VIRUS dataset contains 1500 images of viruses, divided into 10 classes. Images were acquired by negative stain transmission electron microscopy [69]. VI also includes the masks for background removal, which are not used in our tests.
- **CH**: the CHINESE HAMSTER OVARY CELLS dataset contains 327 fluorescent microscopy images, divided into 5 classes [70].
- **SM**: the SMOKE dataset contains 1383 images of smoke, divided into 2 classes used for a challenge of intelligent video surveillance systems [71]. The same training and testing sets proposed in [71] are used here.
- **HI**: the HISTOPATHOLOGY dataset contains 2828 images of connective, epithelial, muscular, and nervous tissue classes [72].

- **BR**: the BREAST CANCER dataset contains 1394 images divided into the control, malignant cancer, and benign cancer classes [73].
- **PR**: the DNA dataset contains 329 proteins, divided into the DNA-binding and non-DNA-binding classes [74].
- **HE**: the 2D HELA dataset contains 862 images of HeLa cells acquired by fluorescence microscope and divided into 10 classes [70].
- **LO**: the LOCATE ENDOGENOUS dataset contains 502 images of mouse sub-cellular images showing endogenous proteins or specific organelle features [75]. The images are unevenly divided into 10 classes.
- **TR**: the LOCATE TRANSFECTED dataset contains 553 mouse sub-cellular images showing fluorescence-tagged or epitope-tagged proteins transiently expressed in specific organelles [75]. The images are unevenly divided into 11 classes.
- **PI**: the HOLY BIBLE dataset contains images extracted from digitalized pages of ancient editions of the Holy Bible (1450 - 1471 A.D.). The images are divided into 13 classes [76].
- **RN**: the FLY CELL dataset contains 200 images of fly cells acquired by fluorescence microscopy and divided into 10 classes.
- **PA**: the PAINTING dataset contains 2338 paintings by 50 artists divided into 13 painting styles [77]. The same training and testing sets proposed in [77] are used here.
- **LE**: the BRAZILIAN FLORA dataset contains 400 images of several species of Brazilian flora evenly divided into 20 classes [78]. Each image was manually split into three windows of size 128×128 pixels leading to 1200 texture samples.
- **IS**: the ISMIR 2004 GENRE CLASSIFICATION dataset contains 1458 music pieces assigned to six different genres: classical, electronic, jazz/blues, metal/punk, rock/pop, and world. It is one of the most widely used datasets in music information retrieval research. The audio signal is converted into a spectrogram image (x axis: time; y axis: frequencies) from which texture features are extracted [11].
- **KU**: the BUTTERFLY dataset contains 140 butterfly images divided into 14 classes of different butterfly species of Styridae family. Each image was cropped to a 256 × 256 pixel image before processing [79]. Each image was split into two non-overlapping equal regions (an upper and lower region). For each region a set of descriptors were extracted.

Table 2. Datasets used in this paper

Dataset	#Classes	#Samples	Sample size	Available at
PS	2	917	Variable	http://labs.fme.aegean.gr/decision/downloads
VI	15	1500	41 × 41	http://www.cb.uu.se/~gustaf/virustexture
CH	5	327	512 × 382	http://ome.grc.nia.nih.gov/iicbu2008/hela/index.html#cho
SM	2	2868	100 × 100	http://staff.ustc.edu.cn/~yfn/vsd.html
HI	4	2828	Variable	http://www.informed.unal.edu.co/histologyDS
BR	2	584	Variable	Upon request to ge.braz@gmail.com
PR	2	349	Variable	Upon request to loris.nanni@unipd.it
HE	10	862	512 × 382	http://ome.grc.nia.nih.gov/iicbu2008/hela/index.html
LO	10	502	768 × 512	http://locate.imb.uq.edu.au/downloads.shtml
TR	11	553	768 × 512	http://locate.imb.uq.edu.au/downloads.shtml
PI	13	903	Variable	http://imagelab.ing.unimo.it/files/bible_dataset.zip
RN	10	200	1024 × 1024	http://ome.grc.nia.nih.gov/iicbu2008/rnai/index.html
PA	13	2338	Variable	http://www.cat.uab.cat/~joost/painting91.html
LE	20	1200	128 × 128	Upon request to bruno@ifsc.usp.br
IS	6	1424	513 × 800	Upon request to loris.nanni@unipd.it
KU	14	140	128 × 256	Upon request to yilmazkaya1977@gmail.com
FM	10	1000	192 × 256	http://people.csail.mit.edu/celiu/CVPR2010/FMD/
SR	15	4485	Variable	http://www-cvr.ai.uiuc.edu/ponce_grp/data/
FR	102	8148	Variable	http://www.robots.ox.ac.uk/~vgg/data/flowers/102/

- **FM**: the FLICKR MATERIAL dataset contains 1000 images of 10 different materials classes: fabric, foliage, glass, leather, metal, paper, plastic, stone, water, and wood. For each class, 50 images are close-up views, and the other 50 are views at object-scale. The location of the object is defined in each image by a binary, manually-labeled mask. Here we considered only the foreground image and use the same training and testing sets suggested in [80].
- **SR**: the SCENE dataset contains 8100 images divided into 15 classes scene [81]. The testing protocol requires five experiments, each using 100 randomly selected images per class for training; the remaining images are used for testing. Each image was split into four non-overlapping equal regions and a central region half the size of the original image. For each region a set of descriptors were extracted.
- **FR**: the OXFORD FLOWERS 102 dataset contains 8189 images divided into 102 categories, containing 40 to 250 images each [82]. Twenty images were selected for each class [82].

4. Experimental Results

The aim of the first experiment, reported in Table 2, was to compare the different stand-alone descriptors that were listed and described in section 3 and reported in Table 1.

We compared all the descriptors reported in Table 3 using the Wilcoxon signed rank test (p-value 0.1). The best performing approaches are the following:

- LTP outperforms 20 other approaches.
- DEN outperforms18 other approaches.
- CLBP outperforms18 other approaches.
- RIC outperforms18 other approaches.

In the last two rows of Table 3, we reported two ensembles of classifiers that significantly boosts the performance of the stand-alone methods:

- F1 is the sum rule among LTP, GO and RIC.
- F2 is the sum rule among LTP, GO, RIC, MOR, HOG and CLBP.

Table 3. Comparison among state-of-the-art approaches

	PS	VI	CH	SM	HI	BR	PR	HE	LO	TR	PI	RN	PA	LE	IS	KU	FM	SR	FL
LHF	86.4	92.5	97.9	99.7	84.7	91.1	82.5	95.9	97.1	97.6	85.5	92.4	79.1	96.5	90.4	89.5	76.5	96.4	86.4
LPQ	90.3	94.9	99.2	99.8	91.3	95.6	86.1	97.5	97.6	97.6	90.7	95.3	88.3	98.9	91.9	91.1	78.2	97.8	90.8
HOG	90.2	95.0	99.2	93.8	92.0	95.7	86.2	97.2	97.6	97.7	92.8	95.2	79.2	90.9	90.4	91.9	78.3	93.0	90.7
LBP	90.1	92.1	99.4	99.7	89.1	93.7	80.1	98.0	98.6	98.5	91.7	94.7	85.3	98.3	92.2	89.9	76.8	97.1	89.3
LTP	91.4	93.4	99.9	99.7	91.5	96.9	89.6	98.1	99.4	99.2	92.8	96.9	89.0	97.9	92.3	91.5	79.0	97.6	89.8
MOR	91.1	96.5	99.9	99.3	89.9	89.1	87.6	97.8	99.3	99.4	87.5	97.1	83.8	95.4	88.1	89.6	73.5	95.4	87.2
LCP	77.7	87.3	98.8	97.1	81.9	93.1	80.0	96.1	98.9	96.8	83.3	94.4	77.9	93.3	90.4	82.7	71.5	94.6	78.9
NTL	83.2	91.6	99.3	99.5	84.6	82.1	70.9	96.5	96.3	97.5	93.7	88.6	93.7	86.7	79.6	93.1	76.1	93.8	83.7
DEN	85.7	95.2	99.7	99.8	91.8	93.6	87.9	98.0	98.8	98.3	92.2	95.3	89.5	98.2	91.7	90.9	78.9	98.2	88.3
CLBP	88.0	95.6	99.3	99.9	92.4	95.3	83.9	98.7	98.4	98.6	91.7	94.9	89.2	98.1	92.7	89.9	80.1	98.0	90.0
RIC	91.8	97.6	99.2	99.8	90.2	92.9	88.6	97.3	99.0	98.8	90.8	96.6	86.7	97.4	88.6	89.3	80.3	97.0	92.3
WLD	79.5	88.0	99.9	98.4	87.2	93.0	85.4	94.1	97.9	98.8	87.0	97.4	85.9	96.5	90.4	91.0	73.8	96.0	86.8
HASC	90.1	94.2	99.7	99.8	87.2	90.6	88.9	97.2	99.0	99.3	88.0	96.7	85.6	98.1	93.2	89.8	74.0	96.0	86.1
GAB	90.0	91.5	99.3	98.1	83.0	91.0	83.4	95.0	98.0	98.4	89.4	96.7	79.4	95.4	91.6	91.6	66.2	94.5	82.3
RLBP	87.4	94.2	99.0	99.8	89.4	90.3	81.3	97.6	99.1	99.4	85.5	95.2	85.9	97.4	89.2	88.4	80.4	95.9	90.1
MRE	87.5	98.0	98.6	99.8	91.9	78.0	86.4	98.2	98.4	97.5	91.7	90.2	85.9	97.3	88.5	89.3	76.2	97.1	92.0
MSJ	85.7	91.0	98.9	99.1	85.0	91.0	84.8	96.0	98.9	98.2	85.4	96.8	84.6	95.4	91.8	81.4	78.5	92.5	59.3
LDTP	87.8	88.1	99.6	99.0	86.4	90.3	84.4	97.1	98.6	98.4	86.1	96.4	82.8	95.9	90.4	89.4	66.7	96.3	85.8
LAP	81.4	89.1	99.3	99.8	83.5	91.5	77.2	97.4	99.3	98.7	86.3	92.9	81.2	92.2	89.7	89.5	77.2	96.7	88.8
HA	82.9	90.0	96.6	97.8	76.9	79.6	74.2	95.4	98.6	98.2	84.7	91.2	73.4	94.5	90.4	89.8	65.4	95.4	80.7
DB4	83.6	91.7	98.7	97.7	78.5	82.8	78.0	95.9	96.9	97.7	79.7	91.9	72.5	91.6	90.3	89.3	63.8	95.8	80.0
COI	85.0	92.5	98.8	97.1	77.4	81.0	75.5	96.4	97.3	98.0	82.1	91.1	74.0	92.6	91.2	90.0	64.2	95.2	81.1
FDCT	85.4	91.2	99.1	96.8	82.2	91.8	84.7	95.2	98.3	98.5	84.3	96.8	83.0	95.8	92.0	89.4	70.9	93.3	79.4
DISC	87.9	97.0	99.2	99.8	90.3	79.3	86.5	97.1	97.7	51.6	88.9	94.3	87.9	98.0	89.5	88.4	78.2	96.5	91.9
GO	89.1	95.3	99.1	99.8	87.4	80.1	91.2	98.2	98.4	96.1	98.7	90.2	88.4	97.0	93.3	92.6	75.4	98.2	94.4
F1	94.6	98.9	99.9	99.9	92.9	95.2	94.2	99.4	99.9	99.7	98.3	97.5	93.2	98.6	94.3	91.2	84.5	98.9	96.1
F2	96.0	98.3	99.9	99.9	94.5	96.7	94.5	99.5	99.8	99.8	98.0	98.6	94.2	98.8	95.3	91.9	85.5	99.1	95.8

Table 4. Ensemble of descriptors

	PS	VI	CH	SM	HI	BR	PR	HE	LO	TR	PI	RN	PA	LE	IS	KU	FM	SR	FL
LPQ	90.3	94.9	99.2	99.8	91.3	95.6	86.1	97.5	97.6	97.6	90.7	95.3	88.3	98.9	91.9	91.1	78.2	97.8	90.8
MLPQ	91.8	97.6	100	99.3	93.1	97.0	94.7	99.2	99.8	99.6	95.3	98.4	92.1	97.7	95.6	90.8	81.5	98.4	94.0
DpSA	84.6	95.0	98.2	97.0	84.0	77.6	74.0	96.7	98.2	95.4	83.4	87.7	74.5	94.6	85.7	88.1	61.7	95.2	86.6
DpEN	91.4	97.9	99.9	99.2	90.5	85.7	90.5	98.8	99.1	98.7	91.4	92.1	79.7	96.0	91.8	90.4	69.9	97.6	86.7
LHF	86.4	92.5	97.9	99.7	84.7	91.1	82.5	95.9	97.1	97.6	85.5	92.4	79.1	96.5	90.4	89.5	76.5	96.4	86.4
MLHF	91.3	93.9	99.2	99.7	90.1	94.6	85.4	97.7	98.2	98.4	90.9	95.1	85.1	98.0	90.3	90.0	78.9	97.1	89.1
MLHF_t	91.5	93.3	99.6	99.5	90.5	96.7	86.5	97.3	99.2	98.7	91.8	97.2	87.2	97.5	91.7	91.0	80.6	97.2	89.4
BSIF	87.1	91.2	99.3	99.8	91.0	94.8	89.2	97.2	98.7	98.6	93.5	93.5	87.0	97.8	92.6	92.5	76.0	96.5	91.5
FBSIF	91.4	97.0	99.9	99.9	94.0	96.7	91.9	99.2	99.8	99.8	95.7	98.2	93.2	99.1	94.8	90.6	79.8	98.6	94.4
LTP	91.4	93.4	99.9	99.7	91.5	96.9	89.6	98.1	99.4	99.2	92.8	96.9	89.0	97.9	92.3	91.5	79.0	97.6	89.8
SLQ	85.2	95.3	100	99.7	92.2	95.5	79.9	97.8	99.2	99.7	83.2	97.2	88.2	97.4	91.3	88.9	82.0	96.4	92.0
SLQT	90.1	95.7	100	99.8	93.2	97.6	87.7	98.8	99.7	99.7	91.2	97.7	90.7	98.4	93.1	90.6	82.9	98.0	92.4
F2	96.0	98.3	99.9	99.9	94.5	96.7	94.5	99.5	99.8	99.8	98.0	98.6	94.2	98.8	95.3	91.9	85.5	99.1	95.8
F3	96.5	98.5	100	99.9	94.9	97.1	95.0	99.6	99.9	99.8	98.2	98.8	94.8	98.9	95.9	90.6	85.5	99.2	95.9

Table 5. Preprocessing validation

	PS	VI	CH	SM	HI	BR	PR	HE	LO	TR	PI	RN	PA	LE	IS	KU	FM	SR	FL
RIC	91.8	97.6	99.2	99.8	90.2	92.9	88.6	97.3	99.0	98.8	90.8	96.6	86.7	97.4	88.6	89.3	80.3	97.0	92.3
RICep	91.8	98.2	99.6	99.7	92.8	96.6	90.9	98.3	99.5	99.5	92.1	96.2	58.9	98.1	90.7	88.7	83.0	98.2	92.3
LPQ	90.3	94.9	99.2	99.8	91.3	95.6	86.1	97.5	97.6	97.6	90.7	95.3	88.3	98.9	91.9	91.1	78.2	97.8	90.8
LPQep	91.6	96.7	99.5	99.7	92.8	96.8	90.2	98.0	99.0	98.9	93.3	96.1	91.4	99.2	93.0	90.7	81.2	98.6	91.8
HASC	90.1	94.2	99.7	99.8	87.2	90.6	88.9	97.2	99.0	99.3	88.0	96.7	85.6	98.1	93.2	89.8	74.0	96.0	86.1
HASCep	92.8	96.7	99.9	99.9	89.7	94.8	90.5	98.6	99.5	99.8	91.5	98.2	88.7	98.9	95.3	91.2	77.6	97.5	88.6
GAB	90.0	91.5	99.3	98.1	83.0	91.0	83.4	95.0	98.0	98.4	89.4	96.7	79.4	95.4	91.6	91.6	66.2	94.5	82.3
GABep	92.6	94.6	99.7	99.1	86.9	94.2	88.8	96.5	99.2	99.5	91.8	97.1	82.2	97.1	93.8	91.6	70.5	96.1	84.0
CLBP	88.0	95.6	99.3	99.9	92.4	95.3	83.9	98.7	98.4	98.6	91.7	94.9	89.2	98.1	92.7	89.9	80.1	98.0	90.0
CLBPep	90.7	97.1	99.9	99.7	93.1	95.9	91.4	98.2	99.0	99.1	92.8	94.9	91.9	98.4	93.6	90.3	81.9	98.7	91.0
LTP	91.4	93.4	99.9	99.7	91.5	96.9	89.6	98.1	99.4	99.2	92.8	96.9	89.0	97.9	92.3	91.5	79.0	97.6	89.8
LTPep	91.0	95.8	99.9	99.8	93.0	97.7	91.1	98.6	99.8	99.6	94.4	97.7	91.5	97.9	93.4	89.3	81.6	98.5	91.6

Table 6. Region-based validation

	PS	VI	CH	SM	HI	BR	PR	HE	LO	TR	PI	RN	HP	PA	LE	IS	KU	FM	SR	FL
RIC	91.8	97.6	99.2	99.8	90.2	92.9	88.6	97.3	99.0	98.8	90.8	96.6	93.5	86.7	97.4	88.6	89.3	80.3	97.0	92.3
RICer	92.6	97.8	99.8	99.9	91.9	94.5	89.9	98.5	99.5	99.2	93.5	97.0	94.2	89.5	97.7	91.4	89.7	82.5	97.4	93.8
LTP	91.4	93.4	99.9	99.7	91.5	96.9	89.6	98.1	99.4	99.2	92.8	96.9	88.8	89.0	97.9	92.3	91.5	79.0	97.6	89.8
LTPer	91.3	94.5	100	99.7	92.4	97.4	92.8	98.8	99.7	99.6	93.6	97.3	91.6	90.8	97.9	93.6	91.5	77.5	97.9	91.5
LPQ	90.3	94.9	99.2	99.8	91.3	95.6	86.1	97.5	97.6	97.6	90.7	95.3	90.9	88.3	98.9	91.9	91.1	78.2	97.8	90.8
LPQer	90.7	95.3	99.7	99.9	92.3	97.0	89.8	98.3	98.6	98.7	93.4	95.3	92.3	89.8	98.9	94.0	91.3	80.5	98.1	92.3

In the following experiment (see Table 4), we reported the performance of different fusions among descriptors. Clearly, the ensemble versions of a descriptor improved the stand-alone versions, i.e., MLPQ outperformed LPQ, DpEN outperformed DpSA, and so on. Among the methods reported in Table 4, the best performance was obtained by MLPQ, which outperformed LTP (the best stand-alone descriptor) with a p-value of 0.01. We tried to improve the ensemble F2 reported in Table 4 by combining it with methods reported in Table 4; the highest performance was obtained by combining the descriptors that belong to F2 with MLPQ, DpEN, and FBSIF using the sum rule (we label this ensemble F3). Notice that before each fusion the scores of each descriptor were normalized to mean 0 and standard deviation 1.

In Tables 5 and 6, we compared standard texture descriptors with their ensembles built using the preprocessing and the region-based methods detailed in sections 2.7 and 2.8. To reduce the computation time, we run this experiment only with a subset of the whole set of texture descriptors. Given a descriptor named *X*, its ensemble version based on preprocessing methods was labeled *Xep*, while its ensemble version using the region-based approach was labeled *Xer*. As expected, the ensembles clearly outperformed the stand-alone versions of each descriptor.

The ensemble *Xep* was given by weighted sum rule among SVM, trained using the descriptor extracted from the original image (with weight 4), and the SVMs trained on the images obtained by the four preprocessing methods presented in section 2.7 (each with weight 1).

The ensemble *Xer* was given by sum rule among SVM trained using the descriptor extracted from the original image and the SVMs trained using the histograms built by the methods Edge, Saliency, and Wavelet. Whenever a given method built *x* histograms, the scores were summed and normalized by dividing the sum by *x*.

We attempted to improve F3 by combining it with the approaches reported in Tables 5 and 6 but found no improvement with a p-value of 0.1.

CONCLUSION

In this chapter, we attempt to determine empirically, using a large set of benchmark databases representing a broad range of problems, the best ensemble method for extracting features from an image using a set of texture descriptors. Both stand-alone and ensembles of descriptors are compared, along with many stand-alone and state-of-the-art ensemble approaches proposed in the literature.

Our experimental results lead us to propose an ensemble of descriptors that works extremely well across the entire set of tested datasets. Experiments clearly demonstrate that our proposed ensemble F3 gets very good results, even when we compare it with systems that were built ad-hoc for a given image classification problem, such as the following:

- In [77] the dataset PA was proposed and an ensemble of different descriptors (along with bag-of-feature approaches and color descriptors) obtained an accuracy of 62.2%, while F3 obtains an accuracy of 66.3%.
- In [11] a set of texture descriptors obtained an accuracy of 81.6% in the IS dataset, whereas F3 obtains an accuracy of 86.3%.

In the future, we plan on improving our ensemble approaches by combining them with color-based descriptors and approaches based on bag-of-features (e.g., SIFT).

The MATLAB source code to replicate our experiments will be made available at (https://www.dei.unipd.it/node/2357 +Pattern Recognition and Ensemble Classifiers).

ACKNOWLEDGMENTS

The authors would like to thank the researchers who shared their MATLAB code with us and the CSC - IT Centre for Science in Finland for their generous computational resources.

REFERENCES

[1] Wu CM, Chen YC, et al. Texture features for classification of ultrasonic liver images. *IEEE Trans Med Imaging*. 1992;11(2):141–52.

[2] Castellano G, Bonilha L, et al. Texture analysis of medical images. Clin Radiol. 2004;59(12):1061–9.

[3] Nanni L, Lumini A, et al. Local binary patterns variants as texture descriptors for medical image analysis. *Artif Intell Med.* 2010;49(2):117–25.

[4] Vu N, Nguyen T, et al. Improving texture categorization with biologically-inspired filtering. Image Vis Comput. 2014;32(6-7):424–36.

[5] Nanni L, Paci M, et al. Virus image classification using different texture descriptors. *The 14th International Conference on Bioinformatics and Computational Biology* (BIOCOMP'13). 2013. p. 56–61.

[6] Paci M, Nanni L, et al. Non-Binary Coding for Texture Descriptors in Sub-Cellular and Stem Cell Image Classification. *Curr Bioinform.* 2013;8(2):208–19.

[7] Thibault G, Angulo J, et al. Advanced statistical matrices for texture characterization: Application to cell classification. *IEEE Trans Biomed Eng.* 2014;61(3):630–7.

[8] Fongaro L, Lin Ho DM, et al. Application of the angle measure technique as image texture analysis method for the identification of uranium ore concentrate samples: New perspective in nuclear forensics. Talanta. Elsevier; 2016;152:463–74.

[9] Mendoza F, Dejmek P, et al. Colour and image texture analysis in classification of commercial potato chips. *Food Res Int.* 2007;40(9):1146–54.

[10] Pieniazek F, Sancho A, et al. Texture and Color Analysis of Lentils and Rice for Instant Meal Using Image Processing Techniques. *J Food Process Preserv*. 2016;n/a – n/a.

[11] Nanni L, Costa YMG, et al. Combining visual and acoustic features for music genre classification. *Expert Syst Appl*. Elsevier Ltd; 2016;45: 108–17.

[12] Kerman J. *Listen.* St Martins Pr; 1980.

[13] Hay GJ, Niemann KO, et al. An object-specific image-texture analysis of H-resolution forest imagery. *Remote Sens Environ*. 1996;55(2):108–22.

[14] Tamura H, Mori S. Textural features corresponding to visual perception. *Syst Man*. 1978;75(6):460–73.

[15] Sklansky J. Image segmentation and feature extraction. *IEEE Trans Syst Man Cybern*. 1978;75(4):237–47.

[16] Haralick R. Statistical and structural approaches to texture. *Proc IEEE.* 1979;67(5):786–804.

[17] Hawkins JK. Textural Properties for Pattern Recognition. In: Lipkin B, Rosenfeld A, editors. *Picture Processing and Psychopictorics Psychopictorics.* New York: Academic Press; 1969. p. 347–70.

[18] Tuceryan M. Texture Analysis. In: Chen CH, Pau LF, Wang PSP, editors. The Handbook of Pattern Recognition and Computer Vision. 2nd ed. World Scientific Publiching Co.; 1998. p. 207–48.

[19] Paci M, Nanni L, et al. An ensemble of classifiers based on different texture descriptors for texture classification. *J King Saud Univ - Sci.* King Saud University; 2013;25(3):235–44.

[20] Srinivasan G, Shobha G. Statistical texture analysis. *Proc world Acad* 2008;36(December):1264–9.

[21] Haralick R, Dinstein, et al. Textural features for image classification. *IEEE Trans Syst Man Cybern.* 1973;SMC-3:610–21.

[22] Manthalkar R, Biswas P., et al. Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognit Lett.* 2003;24(12):2061–8.

[23] Materka A, Strzelecki M. Texture Analysis Methods – A Review. *Methods.* 1998;11:1–33.

[24] Ojala T, Pietikäinen M, et al. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Trans Pattern Anal Mach Intell. Los Alamitos, CA, USA: *IEEE Computer Society;* 2002;24(7):971–87.

[25] Tan X, Triggs B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Process IEEE Trans*. 2010;19(6):1635–50.

[26] Nanni L, Paci M, et al. Texture Descriptors Ensembles Enable Image-Based Classification of Maturation of Human Stem Cell-Derived Retinal Pigmented Epithelium. *PLoS One*. 2016;11(2):e0149399.

[27] Abdesselam A. Improving Local Binary Patterns Techniques by Using Edge Information. *Lect Notes Softw Eng*. 2013;1(4):360–3.

[28] Nosaka R, Fukui K. HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns. Pattern Recognit. Elsevier; 2014;47(7):2428–36.

[29] Foggia P, Percannella G, et al. Pattern recognition in stained HEp-2 cells: Where are we now? *Pattern Recognit.* 2014;47(7):2305–14.

[30] Manivannan S, Li W, et al. HEp-2 Cell Classification Using Multi-resolution Local Patterns and Ensemble SVMs. 2014 1st Work Pattern Recognit Tech Indirect Immunofluoresc Images. *Ieee;* 2014;37–40.

[31] Gao Z, Zhang J, et al. HEp-2 Cell Image Classification with Convolutional Neural Networks. 2014 1st Work Pattern Recognit Tech Indirect Immunofluoresc Images. *Ieee;* 2014;24–8.

[32] Codrescu C. Quadratic Recurrent Finite Impulse Response MLP for Indirect Immunofluorescence Image Recognition. 2014 1st Work Pattern Recognit Tech Indirect Immunofluoresc Images. *Ieee*; 2014;(1):49–52.

[33] Nanni L, Brahnam S, et al. Combining different local binary pattern variants to boost performance. Expert Syst Appl. 2011;38(5):6209–16.

[34] Nanni L, Lumini A. A reliable method for cell phenotype image classification. *Artif Intell Med.* 2008;43(2):87–97.

[35] Zhao G, Ahonen T, et al. Rotation-invariant image and video description with local binary pattern features. *IEEE Trans Image Process*. 2012;21(4):1465–77.

[36] Ojansivu V, Heikkilä J. Blur Insensitive Texture Classification Using Local Phase Quantization. Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2008. p. 236–43.

[37] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05). *IEEE;* 2005. p. 886–93.

[38] Strandmark P, Ulen J, et al. HEp-2 staining pattern classification. Pattern Recognition (ICPR), 2012 21st International Conference on. 2012. p. 33–6.

[39] Guo Y, Zhao G, et al. Texture Classification using a Linear Configuration Model based Descriptor. Procedings of the British Machine Vision Conference 2011. British Machine Vision Association; 2011. p. 119.1–119.10.

[40] Fathi A, Naghsh-Nilchi AR. Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern Recognit Lett. Elsevier B*.V.; 2012;33(9):1093–100.

[41] Ylioinas J, Hadid A, et al. Efficient image appearance description using dense sampling based local binary patterns. Computer Vision – ACCV 2012 Lecture Notes in Computer Science. 2013. p. 375–88.

[42] Guo Z, Zhang L, et al. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Trans Image Process*. 2010;16(6):1657–63.

[43] Chen J, Shan S, et al. WLD: a robust local image descriptor. *IEEE Trans Pattern Anal Mach Intell.* 2010;32(9):1705–20.

[44] San Biagio M, Crocco M, et al. Heterogeneous Auto-similarities of Characteristics (HASC): Exploiting Relational Information for

Classification. Computer Vision (ICCV), 2013 *IEEE International Conference* on. 2013. p. 809–16.

[45] Fogel I, Sagi D. Gabor filters as texture discriminator. *Biol Cybern*. 1989;61(2):103–13.

[46] Mehta R, Egiazarian K. Dominant Rotated Local Binary Patterns (DRLBP) for texture classification. *Pattern Recognit Lett. Elsevier B.V.;* 2016;71:16–22.

[47] Liu L, Lao S, et al. Median Robust Extended Local Binary Pattern for Texture Classification. IEEE Trans Image Process. 2016;25(3):1368–81.

[48] Qi X, Shen L, et al. Globally rotation invariant multi-scale co-occurrence local binary pattern. Image Vis Comput. Elsevier B.V.; 2015;43:16–26.

[49] Ramírez Rivera A, Rojas Castillo J, et al. Local Directional Texture Pattern image descriptor. *Pattern Recognit Lett*. 2015;51:94–100.

[50] Xu Y, Huang S, et al. Scale-space texture description on SIFT-like textons. *Comput Vis Image Underst. Elsevier Inc.*; 2012;116(9):999–1013.

[51] Huang K, Murphy RF. Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. *BMC Bioinformatics*. 2004;5(78).

[52] Candès E, Demanet L, et al. Fast Discrete Curvelet Transforms. *Multiscale Model Simul.* 2006;5(3):861–99.

[53] Guo Y, Zhao G, et al. Discriminative features for texture description. *Pattern Recognit*. 2012;45(10):3834–43.

[54] Serra G, Grana C, et al. GOLD: Gaussians of Local Descriptors for image representation. *Comput Vis Image Underst. Elsevier Inc.*; 2015;134:22–32.

[55] Nanni L, Brahnam S, et al. Ensemble of Local Phase Quantization Variants with Ternary Encoding. In: Brahnam S, Jain LC, Nanni L, Lumini A, editors. Local binary patterns: New variants and new applications. Springer; 2014. p. 177–88.

[56] Kannala J, Rahtu E. Bsif: Binarized statistical image features. Pattern Recognit (ICPR), 2012 21st Int Conf. 2012;1363–6.

[57] Bryll R, Gutierrez-osuna R, et al. Attribute bagging : improving accuracy of classiÿer ensembles by using random feature subsets. *Pattern Recognit.* 2003;36:1291–302.

[58] Ho TK. The random subspace method for constructing decision forests. *IEEE Trans Pattern Anal Mach Intell.* 1998;20(8):832–44.

[59] Jia Y, Shelhamer E, et al. Caffe: Convolutional Architecture for Fast Feature Embedding. *Proc ACM Int Conf Multimed*. 2014;675–8.

[60] Nanni L, Lumini A, et al. Survey on LBP based texture descriptors for image classification. *Expert Syst Appl. Elsevier Ltd*; 2012;39(3):3634–41.

[61] Lumini A, Nanni L, et al. Multilayer descriptors for medical image classification. *Comput Biol Med.* 2016;72:239–47.

[62] Mallat S. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Anal Mach Intell IEEE Trans*. 1989;11(7):674–93.

[63] Singh K, Kapoor R. Image enhancement using Exposure based Sub Image Histogram Equalization. *Pattern Recognit Lett. Elsevier B.V.*; 2014;36(1):10–4.

[64] Nanni L, Brahnam S, et al. Region-based approaches and descriptors extracted from the co-occurrence matrix. *Int J Latest Res Sci Technol.* 2014;3(6):192–200.

[65] Baddeley RJ, Tatler BW. High frequency edges (but not contrast) predict where we fixate: A Bayesian system identification analysis. *Vision Res.* 2006;46(18):2824–33.

[66] Hou X, Harel J, et al. Image Signature: Highlighting Sparse Salient Regions. *IEEE Trans Pattern Anal Mach Intell.* 2011;34(1):194–201.

[67] Fawcett T. ROC Graphs: Notes and Practical Considerations for Researchers. Tech report, Palo Alto, USA HP Lab. 2004;

[68] Jantzen J, Norup J, et al. Pap-smear benchmark data for pattern classification. Nature inspired Smart Information Systems (NiSIS), EU co-ordination action Albufeira, Portugal: *NiSIS*. 2005. p. 1–9.

[69] Kylberg G, Uppström M, et al. Virus texture analysis using local binary patterns and radial density profiles. 18th Iberoamerican Congress on Pattern Recognition (CIARP). Martin S, Kim S-W; 2011. p. 573–80.

[70] Boland M V, Murphy RF. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics.* 2001;17(12):1213–23.

[71] Yuan F. Video-based smoke detection with histogram sequence of LBP and LBPV pyramids. *Fire Saf J. Elsevier*; 2011;46(3):132–9.

[72] Cruz-Roa A, Caicedo JC, et al. Visual pattern mining in histology image collections using bag of features. *Artif Intell Med. Elsevier B.V.;* 2011;52(2):91–106.

[73] Braz Junior G, Cardoso de Paiva A, et al. Classification of breast tissues using Moran’s index and Geary's coefficient as texture signatures and SVM. Comput Biol Med. Elsevier; 2009;39(12):1063–72.

[74] Nanni L, Shi J-Y, et al. Protein classification using texture descriptors extracted from the protein backbone image. *J Theor Biol. Elsevier*; 2010;264(3):1024–32.

[75] Hamilton N, Pantelic R, et al. Fast automated cell phenotype image classification. *BMC Bioinformatics*. 2007;8(1):110.

[76] Borghesani D, Grana C, et al. Miniature illustrations retrieval and innovative interaction for digital illuminated manuscripts. *Multimed Syst.* 2014;20:65–79.

[77] Khan FS, Beigpour S, et al. Painting-91: A large scale database for computational painting categorization. *Mach Vis Appl*. 2014;25(6):1385–97.

[78] Casanova D, de Mesquita Sá Junior JJ, et al. Plant leaf identification using Gabor wavelets. *Int J Imaging Syst Technol*. 2009;19(1):236–43.

[79] Kaya Y, Kayci L. Application of artificial neural network for automatic detection of butterfly species using color and texture features. *Vis Comput.* 2013;30(1):71–9.

[80] Sharan L, Rosenholtz R, et al. Material perception: What can you see in a brief glance*? J Vis*. 2014;14(9).

[81] Oliva A, Hospital W, et al. Modeling the Shape of the Scene: *A Holistic Representation of the Spatial Envelope*. 2001;42(3):145–75.

[82] Nilsback ME, Zisserman A. Automated flower classification over a large number of classes. *Proc - 6th Indian Conf Comput Vision, Graph Image Process ICVGIP* 2008. 2008;722–9.

In: Computer Vision and Simulation
Editor: Sherri Alexander

ISBN: 978-1-63485-790-1

Chapter 3

COMPUTER STUDY OF THE INTERACTION OF MERCURY WITH GRAPHENE

Alexander Y. Galashev[*]
Institute of High-Temperature Electrochemistry, Ural Branch,
Russian Academy of Sciences, Yekaterinburg, Russia

ABSTRACT

The contamination of natural waters and the lower atmosphere by heavy metal ions creates a serious ecological problem. Mercury is one of the most toxic heavy metals, because it is not biodegradable. We have studied the physical properties of mercury films on partially hydrogenated imperfect graphene by means of molecular dynamics at 300 K. Films prepared on the basis of three various types of the atomic interaction potential for mercury and other constant interaction potentials are considered. It is shown that the one most promising is the Schwerdtfeger potential function, at which mercury atoms do not fall into the divacancies present on graphene and atom packing with the lowest energy are realized in a liquid film and the film gradually fold into a drop. Another computer experiment has been employed to study rapid heating of a mercury film on graphene containing Stone–Wales defects. Hydrogenated edges of a graphene sheet withstand heating by 800 K. As the film contracts into a droplet, the horizontal component of the self-diffusion coefficient of Hg atoms monotonically decreases, while the

[*] E-mail address: alexander-galashev@yandex.ru.

vertical component passes through a deep minimum, which reflects the onset of droplet rising over the substrate. Formation of the droplet leads to a decrease in the blunt contact angle. Temperature–related changes in graphene manifest themselves as a rise in the intensity of additional peaks in the angular distribution of the closest neighbors, oscillatory pattern of the stress acting in its plane, and an almost linear growth of roughness. Molecular dynamics simulation of the bombardment of a target with a Xe_{13} cluster beam at energies of 5–30 eV and incidence angles of 0°–60° aiming to remove a mercury film from partially hydrogenated imperfect graphene has been performed. The graphene is completely cleaned of mercury at a cluster energy of $E_{Xe} \geq 15$ eV. Mercury is removed from the graphene film via sputtering of single atoms and droplet detachment. A stress in graphene resulting from forces normal to the sheet plane is noticeably higher than that due to forces acting in its plane. Bombardment at an angle of incidence of 45° is more efficient than that at incidence of 0° and 60° and leads to lower graphene roughness. Thus, mercury can be removed from graphene by heating or bombarding with heavy noble gas clusters.

1. Introduction

Environmental pollution with heavy metals is a global problem [1, 2], due to its detrimental consequences for health [3]. Composite membranes based on graphene for the accumulation of mercury have were proposed in [4]. The structure and physical properties of liquid mercury–graphene interfaces remain unstudied; meanwhile, the prospect of using graphene as a filter demands their study. Liquids with isotropic pair interactions encounter vibrational interface structures at temperatures close to melting point T_m, providing that the T_m / T_c ratio (where T_c is the critical temperature) is low. The melting point of bulk mercury T_m = 234 K. Cold liquid metals such as Hg and Ga have low T_m / T_c = 0.13 and 0.15, respectively.

Molecular dynamic (MD) modeling reveals their nonmonotonous density profiles near the liquid mercury/vapor interface [5]. On the other hand, the interface range has a non-zero thickness that depends on temperature. Calculating the properties of a Hg liquid–vapor interface with clear allowance for the dependence of the potential on density did not result in satisfactory agreement with the experimental data on ionic and electron density distributions orthogonal to the surface or on the reflection coefficient [6].

Theoretical study of liquids of the Hg type requires knowledge of the effective atomic potential, which allows correct predictions of the liquid/vapor phase diagram in temperature–density coordinates. Out of all the proposed potential functions for mercury, it is difficult to choose one on whose basis the structure of liquid mercury on graphene can be reproduced satisfactorily.

The structure of solid metals in contact with graphene (e.g., deposited copper films) has been studied more thoroughly [7].

Mercury absorption from smoke fumes has been studied with the use of X–ray absorption fine structure (XAFS) spectroscopy [8, 9]. XAFS spectra suggested that there is chemisorption of Hg on activated carbon. These data gave grounds to think that adsorption took place via halides, sulfides, and oxygen anions present on an activated–carbon surface. Moreover, chlorinated and bromated activated carbon was revealed with the use of X–ray absorption spectroscopy and X–ray photoelectron spectroscopy after exposure of carbon samples in smoke fumes containing Hg in an amount of 204 μg/m^3 [10]. Mercury was not found on the surface of activated carbon; however, $\mathrm{Hg-Br}$ and $\mathrm{Hg-Cl}$ complexes were present. This fact underlay the assumption that sites containing Cl and Br were formed on a carbon surface prior to the capture of Hg.

The mechanism of mercury binding by activated–carbon–based sorbents was studied in [11]. It was shown that, at low Hg concentrations, it was difficult to distinguish between the mechanisms of oxidation and adsorption. The difference between them gradually grew with Hg concentration and enhancement of $\mathrm{Hg-Hg}$ interaction. However, because of the close values of the bond energies in HgO, $\mathrm{Hg_2Br_2}$, and $\mathrm{HgBr_2}$, these surface–bound compounds were, as a rule, indistinguishable by photoelectron spectroscopy.

Liquid mercury does not wet graphite. Indeed, on highly ordered pyrolitic graphite, fresh mercury droplets have a contact angle of 152.5° [12]. As do any other liquid metals with surface tensions γ higher than 0.18 N/m, mercury does not wet carbon nanotubes [13]. The surface tension of mercury is 0.46 N/m. Nevertheless, wetting and filling of internal cavities of carbon nanotubes with mercury take place due to electrowetting [14]. The effect of electrostatic interactions on the sorption of hydrocarbons by water droplets ($\gamma_{\mathrm{H_2O}}$ = 0.0729 N/m) was shown in [15]. The mercury contact angle linearly increases with the curvature of carbon nanotube walls. Therefore, the internal surface of a nanotube has a higher phobicity with respect to mercury than the planar surface of graphene has [16]. Graphene wetting with mercury has not been studied.

Mercury is the only one of the most abundant metals that remains liquid at room temperature. The study of the adsorption of mercury at activated carbon was, as a rule, carried out experimentally. There is a limited number of theoretical studies concerning this theme.

Steckel [17] has investigated the interaction between elemental mercury and a single benzene ring in order to explain the mechanism through which elemental mercury is bound with carbon. Padak et al. [18] investigated the effect of different surface functional groups and halogens present on the surface of activated carbon on the adsorption of elemental mercury. It has been established that the addition of halogen atoms strengthens the adsorption of mercury. In [19], Padak and Wilcox have demonstrated a thermodynamic approach to the examination of the mechanism of binding of mercury and its capture in the form of HgCl and HgCl_2 on the surface of activated carbon. The energies of different possible surface complexes have been determined. In the presence of chlorine, the mercury atoms are strongly coupled to the surface. In the case of dissociative adsorption, Hg can undergo desorption, while HgCl remains on the surface. The compound HgCl_2 was not found on the stable carbon surface [20]. Understanding of the mechanism of the adsorption of mercury at activated carbon is important for the development of efficient technologies for capturing mercury.

Mercury is one of the most toxic heavy metals, and its presence is due to a combination of natural processes (volcanic activity, erosion of the mercury–containing sediments) and anthropogenic activity (extraction of minerals, pollution from the leather–dressing production and metallization of objects). Adsorption is considered to be one of the most efficient and economical methods of removing mercury from wastewater and air.

Recently, graphene membranes have begun to be used in filters for separation of trace amounts of undesirable impurities [21–23]. Repeated use of graphene in filters requires its nondestructive purification from adsorbed substances. Graphene may be purified from metals by irradiating with cluster beams of noble gases [24–29] or heating [30–33]. However, heating is reasonable to be used, when a metal has rather low boiling temperature T_b. Mercury seems to be a possible candidate for the use of this procedure. As a rule, ideal graphene is not destroyed upon heating to the boiling temperature of many metals, such as Al, Ni, or Cu, although its edges are damaged [30–32]. Graphene edges may be reinforced by hydrogenation. Graphene treated in this way withstands cluster bombardment even at a beam energy of 30 eV [34]. It is unclear how graphene with a high concentration of Stone–Wales defects will behave, because these defects are formed before its melting [35].

The aim of this work is to choose an effective pair potential that allows reproduction of the structural, thermodynamic and kinetic properties of mercury films deposited on graphene in a molecular dynamics model. This work also studies the morphology and variations in physical properties (induced by rapid–heating) of mercury film on graphene with hydrogenated edges and high concentration of Stone–Wales defects. This investigation is also aimed at studying the possibility of removing adsorbed mercury from graphene via bombardment by xenon clusters.

2. Molecular Dynamic Model

The interatomic interactions in graphene are represented by the many–body Tersoff potential [36]. The energy of pairwise interaction of atoms *i* and *j* taking into account the influence of other atoms (many–body effects) is written as

$$V_{ij} = f_C(r_{ij})\left[A\exp(-\lambda^{(1)}r_{ij}) - Bb_{ij}\exp(-\lambda^{(2)}r_{ij})\right], \tag{1}$$

$$f_C(r_{ij}) = \begin{cases} 1, \\ \frac{1}{2} + \frac{1}{2}\cos\left[\pi\left(r_{ij} - R^{(1)}\right)/(R^{(2)} - R^{(1)})\right], \\ 0 \end{cases} \tag{2}$$

$$\begin{matrix} r_{ij} < R^{(1)} \\ R^{(1)} < r_{ij} < R^{(2)}, \\ r_{ij} > R^{(2)} \end{matrix}$$

Here, r_{ij} is the spacing between the atoms i and j and the parameters A and B assign the energy characteristics of the repulsion and attraction. The many–body parameter of the bond order b_{ij} describes how the binding energy (attractive part V_{ij} of the bond) is formed during a local atomic arrangement due to the presence of other neighboring atoms. The function f_C decreases from 1 to 0 in the region of $R^{(1)} \le r_{ij} \le R^{(2)}$. The parameters $R^{(1)}$ and $R^{(2)}$ were selected so as to include into the consideration only nearest neighbors. The potential energy is a many–body function of the positions of atoms i, j, and k and is determined by the parameters

$$b_{ij} = (1+\beta^{n}\xi_{ij}^{n_i})^{-1/(2n)}, \tag{3}$$

$$\xi_{ij} = \sum_{k\neq i,j} f_C(r_{ij})g(\theta_{ijk}), \tag{4}$$

$$g(\theta_{ijk}) = 1+\frac{c^2}{d^2}-\frac{c^2}{\left[d^2+(h-\cos\theta_{ijk})^2\right]}, \tag{5}$$

where the parameters n, n_i, and β assign the binding force depending on the environment. The effective coordination number ξ_{ij} determines the average number of nearest neighbors with taking into account not only the distances between them, but also the bond angles θ_{ijk}. The summing up in expression (4) is conducted over all k first–order neighbors not equal to i and j. These neighbors are selected for each i-j pair and are defined at each time moment; $g(\theta_{ijk})$ is the function of the angle between $\mathbf{r}_{ij}$ and $\mathbf{r}_{ik}$ where $\mathbf{r}_{ij}$ is the vector drawn from the point of the location of the atom i to the point where the atom j is located. The parameter d assigns the width of the sharp maximum in the $g(\theta_{ijk})$ angular dependence, the parameter c assigns the height of this peak, and the function $g(\theta_{ijk})$ has a minimum at $h = \cos(\theta)$. All parameters of the potential were selected so as to match the theoretical and experimental data (energy of cohesion, lattice parameters, bulk moduli) for real and hypothetical graphite and diamond.

Because of the insufficiently precise determination of the force characteristics that control the C-C bonds, the Tersoff potential does not have a barrier to the rotation about the single bond. The inadequacy of the semiempirical Tersoff potential is revealed when studying the dynamic properties of graphite; it manifests in the rotation of the entire fragment to be simulated and can be corrected due by adding a torsion–like term [37]. The parameters of this potential were refined via fitting to the observed properties (standard deviations for the vibration frequencies) of graphite and diamond. The new analytical form of the potential of local torsion is given in [38]. The use in this work of weighting functions for the bond orders ensures a smooth removal of the energy of torsion connected with the dihedral angle upon any sequential break of bonds [38]. The distance $R^{(2)}$ of the covalent binding in the original Tersoff potential was limited to the value of 0.21 nm. The simulation of graphene with this potential led not only to an uncontrollable rotation, but also to the cracking of the graphene sheet [39, 40]. Therefore, we increased the value of $R^{(2)}$ to 0.23 nm and also included an additional weak attraction at $r >$ 0.23 nm assigned by the Lennard–Jones (LJ) potential with the parameters used in [38].

The modeling of mercury adsorption on surface requires exact potentials of $\mathrm{Hg}-\mathrm{Hg}$ and $\mathrm{Hg}-$ substrate interaction. Potentials presented as pair interactions are usually used to describe adsorption. The Lennard–Jones potential is the simplest of these. The parameters of this potential were chosen according to the data on the viscosity of gaseous mercury [41]. The Silver–Goldman potential (SG) is adjusted to ab initio data and provides good agreement with experimental data on spectroscopic constants [42]. The SG potential is based on the Hartree–Fock model of dispersion,

$$V_{SG}(r) = \exp(\alpha - \beta r - \gamma r^2) - f_c(r)\left(\sum_{n=3}^{5} \frac{C_{2n}}{r^{2n}}\right), \tag{6}$$

where

$$f_c(r) = \exp\left[-(1.28 r_c / r - 1)^2\right], \; r < 1.28 r_c$$
$$= 1.0, \qquad r \geq 1.28 r_c \tag{7}$$

The parameters of this potential are given in [42].

The pair potential that was utilized mainly for the description of Hg-Hg interactions was proposed in [43] in the following form:

$$V_{Sch}(r)=\sum_{j=3}^{9} a_{2j}^{*} r^{-2j}. \tag{8}$$

The authors of [43] corrected the original Schwerdtfeger (SCH) potential [44] for mercury dimer by scaling distances using the coefficient λ = 1.167. The parameters a_{2j}^{*} represented in [43] correspond to the density of liquid Hg at T = 300 K. The Hg-C and Xe-Xe interactions were assigned by a Lennard–Jones potential with the parameters established in [45, 46]. The interaction between Xe atoms and the atoms of the target (Hg and C) was assigned by a purely repulsive Ziegler–Biersack–Littmark (ZBL) potential as follows [47]:

$$\Phi=Z_i Z_j \frac{e^2}{r}\left\{0.1818\exp\left(-3.2\frac{r}{a}\right)+0.5099\exp\left(-0.9423\frac{r}{a}\right)+\right.$$
$$\left.+0.2802\exp\left(-0.4029\frac{r}{a}\right)+0.02817\exp\left(-0.2016\frac{r}{a}\right)\right\}, \tag{9}$$

where Z_i and Z_j are the atomic numbers of the atoms i and j; e is the elementary electric charge; r is the interatomic distance; and the parameter a is determined by the expression

$$a=0.8854a_0\left(Z_i^{0.23}+Z_j^{0.23}\right)^{-1}, \tag{10}$$

where a_0 is the Bohr radius.

We disregard the weak attraction between the atoms of Xe and Hg and also between Xe and C, since the primary purpose of this investigation is the examination of the transfer of energy and momentum, rather than of the chemical bonding [48].

The choice of interaction potential between atoms of mercury placed on graphene was carried out in the presence of divacancies in the substrate. Divacancies are one of the most widespread defects in graphene. The presence of such defects remarkably enhances the coupling of graphene with a deposited metal. In present model, nine divacancies were formed nearly uniformly on a graphene sheet. The hydrogenation of graphene results in slight surface ribbing, which also increases the linkage between metal and graphene. Preliminary partial hydrogenation strengthened the graphene edges and stabilized divacancies. A hydrogen atom was effectively added to each boundary C atom (including those in the vicinities of divacancies). More specifically, an entire CH group was considered in considering interactions, rather than individual H atoms. This group interacted with C atoms, other CH groups [49], and Hg atoms [42] through the LJ potential. Fourteen CH groups were arranged along the perimeter of each divacancy. Each group was described according to monoatomic scheme in [49]. This general scheme was designed in developing translated force fields used to predict the thermodynamic properties of complex molecules [50]. Similar hydrogenation was employed to strengthen graphene edges in cases of modeling of the mercury heating on graphene and the Xe cluster bombardment of the "mercury–graphene" target.

Stone–Wales defects along with divacancies are ones of the most widespread defects in graphene. Each Stone–Wales defect is a combination of two contiguous, five, and seven–membered rings. When heating or bombarding was investigated the graphene sheet that used to deposit Hg had six such defects approximately uniformly distributed over its surface.

A film of mercury on graphene was formed in a separate molecular dynamic calculation in two stages. At the first stage, the Hg atoms were placed above the centers of nonadjacent cells of graphene in such a way that the interatomic distance between Hg and C atoms be equal to 2.30 Å, calculated according to the density–functional method [11]. On top of this loose film consisting of 49 mercury atoms, 51 additional Hg atoms were deposited randomly. Then, the system, which consists of 100 atoms of Hg and 406 atoms of C, was brought to equilibrium in the MD calculation with a duration of 1 million time steps (Δt = 0.2 fs). For the numerical solution of the equations of motion, the Verlet algorithm was used [51]. The thus–obtained target was then bombarded with icosahedral Xe_{13} clusters. Five starting points for the positioning of the centers of Xe_{13} clusters were located uniformly along a line parallel to the oy axis (the armchair direction). This line was placed either along the left–hand edge of the graphene sheet (upon the vertical bombardment) or with an additional displacement to the left from it (upon the inclined bombardment), and was lifted to a height of 1.5 nm in the direction of the oz axis. The interval equal to the length of the graphene sheet in the direction of the *ox* axis (the zigzag direction) was divided into five equal sections with a length $L_i = L_x / 5$. At the beginning of every subsequent cycle of cluster impacts, the line of the starting points of the Xe_{13} clusters was advanced a distance L_i horizontally. As a result, the surface of the film approximated by the plane was covered with 25 evenly distributed points at which the cluster impacts were aimed. Each series included 5 cycles, or 25 impacts. At the starting point, all atoms of the Xe_{13} cluster were given the same velocity in the direction of bombardment. The clusters were sent off in turn toward the target. The lifetime (determined by the sum of the time of flight and time of interaction with the target) of each cluster was limited to 8 ps. After this time, the Xe atoms of the destroyed cluster were excluded from the consideration and a new Xe_{13} cluster began moving from another initial point. The cycle of bombardment by five clusters took 40 ps, while the series of five cycles took 0.2 ns and the entire time of bombardment (five series) took 1 ns. The clusters used for bombardments had kinetic energies of 5, 10, 15, 20, and 30 eV; the angles of incidence were 0°, 45°, and 60°.

The impact of a cluster on the surface was accompanied by heating the system. The moderate removal of the heat released from the system was performed according to the Berendsen scheme with a coupling time constant τ_c = 4 fs [52]. The forced reduction in the temperature was conducted via the scaling of velocities v at each time step as follows:

$$v' = \lambda v, \quad \lambda = \left[1 + \frac{\Delta t}{\tau_c}\left(\frac{T_0}{T} - 1\right)\right]^{1/2}, \qquad (11)$$

where v' and v is the new and current value of velocity, respectively, λ is the scaling factor, T_0 is the assigned temperature (300 K), and T is the current temperature.

The density profile of the metallic film was calculated as follows:

$$\rho(z) = \frac{n(z)\sigma_{\mathrm{Hg}}^3}{\Delta h S_{xy} N_s}, \tag{12}$$

where $n(z)$ is the number of Hg atoms in the layer parallel to the plane of the graphene, σ_{Hg} is the effective diameter of the Hg atom, Δh is the width of the layer, S_{xy} is the area of the graphene surface, and N_s is the number of tests.

In order to calculate contact angle θ between a droplet (film) surface and graphene, the largest horizontal cross–sectional area of a droplet was divided into three regions: (1) a circle with a constant area, which determines the region of the contact with graphene, (2) a ring comprising the projections of neighbors closest to region (1), and (3) an analogous ring used to reveal the external atoms of the droplet. Mercury atoms closest to the graphene surface were located in regions 1 and 2. Parameters of the procedure used for determining angles θ were selected empirically. The averaging over the sizes of the rings and heights (or the number of selected Hg atoms), at which Hg atoms were located in regions 2 and 3 yielded the average values of the horizontal and vertical coordinates used to find $\tan\theta$. The determination of angle θ required averaging over time as well.

The self–diffusion coefficient was defined through the mean square of the displacement $\left\langle [\Delta \mathbf{r}(t)]^2 \right\rangle$ of the system consisting of N atoms of Hg as follows:

$$D = D_{xy} + D_z = \frac{1}{2\Gamma} \lim_{\tau \to \infty} \frac{1}{\tau} \left\langle [\Delta \mathbf{r}(t)]^2 \right\rangle_p . \tag{13}$$

Here, $\Gamma = 3$ is the dimensionality of space; $\langle ... \rangle$ means averaging over p, where p is the number of time intervals (with the initial time t_0) for the determination of $[\Delta \mathbf{r}(t)]^2 = \frac{1}{N}\sum_{j=1}^{N}[\mathbf{r}_j(t) - \mathbf{r}_j(t_0)]^2$ and $\mathbf{r}_j$ is the radius vector of the atom j. Averaging is performed over five time dependences, each calculated in an interval τ = 200 ps.

To calculate stresses that appear in graphene, the graphene sheet was divided into surface elements. The stresses $\sigma_{u\alpha}(l)$ that appear under the action of the forces of direction α ($= x$, y, z) are calculated on each element with the order number l that has the orientation u. In these calculations, products of the projections of the velocities of atoms and the projections of the forces f_{ij}^{α} that act on the lth element from the other atoms with the fulfillment of corresponding conditions are used as follows [53, 54]:

$$\sigma_{u\alpha}(l) = \left\langle \sum_{i}^{k} \frac{1}{\Omega}\left(m v_u^i v_\alpha^i\right)\right\rangle + \frac{1}{S_l}\left\langle \sum_{i}^{k} \sum_{j \neq i}^{(u_i \le u, u_j \ge u)} \left(f_{ij}^{\alpha}\right)\right\rangle. \qquad (14)$$

Here, k is the number of atoms on the element l, Ω is the volume per atom, m is the mass of an atom, v_α^i is the α projection of the velocity of atom i, and S_l is the area of the element l. The conditions for summation over j in the last sum in expression (14) are given in the lower and upper indices of the sum, the force that appears upon the interaction of atoms i and j goes through the lth element, and u_i is the current coordinate of the atom i (u can take values x, y, z). In the case when $u = z$, u represents the average level (height) of atoms C in graphene.

The graphene sheet had dimensions of 3.4 × 2.8 nm and contained 406 atoms. Each element l separated on this sheet and elongated along the axis oy (perpendicular to the zigzag direction of graphene) contained 14 C atoms and had an area of 0.68 nm^2. Specifically, this layout corresponds to the data shown in Figure 17.

The total stresses that act in the plane of graphene were determined by summing the corresponding elementary stresses as follows:

$$\sigma_{u\alpha} = \sum_{l=1}^{N_l} \sigma_{u\alpha}(l), \tag{15}$$

where N_l is the number of surface elements.

The roughness of the surface (or the arithmetic mean deviation of the profile) was calculated as

$$R_a = \frac{1}{N_g} \sum_{i=1}^{N_g} \left| z_i - \bar{z} \right|, \tag{16}$$

where N_g is the number of sites (atoms) on the surface of the graphene sheet, z_i is the level of atom i, $\bar{z}$ is the level of the graphene surface, and the levels z_i and $\bar{z}$ are determined at the same time moment.

The total energy of a free one–sheeted graphene obtained at T = 300 K is equal to –7.02 eV, which is in agreement with the quantum–mechanical calculation (–6.98 eV) [55]. The value of the isochoric heat capacity of liquid mercury at this temperature (28.4 J/(mol K) calculated in the MD model agrees with the experimental value of 26.9 J/(mol K).

3. Check of the Hg–Hg Interaction Potential for Mercury Deposited on Graphene

Energy $U_{\mathrm{Hg-Hg}}$ of $\mathrm{Hg-Hg}$ interaction in the film, which was set after equilibrating the system with the LJ potential, was one-third that of the bond energy in a $\mathrm{Hg_2}$ dimer, determined with the same potential [42]. The energies $U_{\mathrm{Hg-Hg}}$ for three potentials with energies of mercury–graphene interaction $U_{\mathrm{Hg-C}}$ are given in the Table 1. It can be seen the highest absolute values of energy $U_{\mathrm{Hg-Hg}}$ were obtained for the Sch potential; the lowest, for the SG potential. On the other hand, the best cohesion between mercury and graphene was provided by the LJ potential; the worst, by the Sch potential.

Table 1. Energies $U_{\mathrm{Hg-Hg}}$ and $U_{\mathrm{Hg-C}}$ of a liquid mercury film on graphene for three potentials

Energy	Potentials		
	LJ	*Sch*	*SG*
$U_{\mathrm{Hg-Hg}}$, eV	-0.0236	-0.0280	-0.0011
$U_{\mathrm{Hg-C}}$, eV	-0.0154	-0.0121	-0.0148

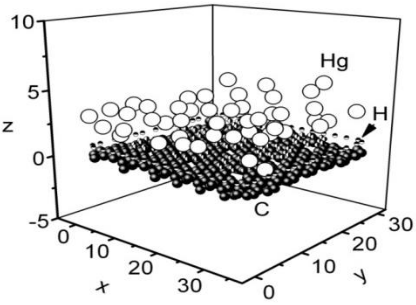


Figure 1. Configuration of a mercury film on a modified graphene system, obtained at the moment of 200 ps. The positions of H atoms correspond to the coordinates of CH-groups reduced to one point at the initial moment in time.

Using the LJ, Sch, and SG potentials for mercury yielded metal films of various structures on graphene. The LJ potential yields a denser packing of Hg atoms, while the SG potential yields more loose and uniform packing. There is a tendency toward the vaporization of atoms at temperatures as low as 300 K for the SG potential. The configuration of the H–graphene–Hg–film system obtained with the Sch potential is given in Figure 1 for the moment of 200 ps. At this time, graphene had a slight ribbing that could be detected from the deviation of boundary C atoms from the even row of H atoms built along the initial coordinates of the CH-groups. The Hg film was in this case quite uniform. However, it did not spread over the entire graphene surface; rather, it gathered into an elongated drop that was flattened to graphene. None of the Hg atoms spilled onto the other side of graphene through a divacancy, though several metal atoms did get stuck in defects. The movement of Hg atoms to the other side of the graphene was observed for two other potentials, though these spills were less than 0.08 nm long. The Sch potential was the one that was best from the viewpoint of retaining atoms on graphene.

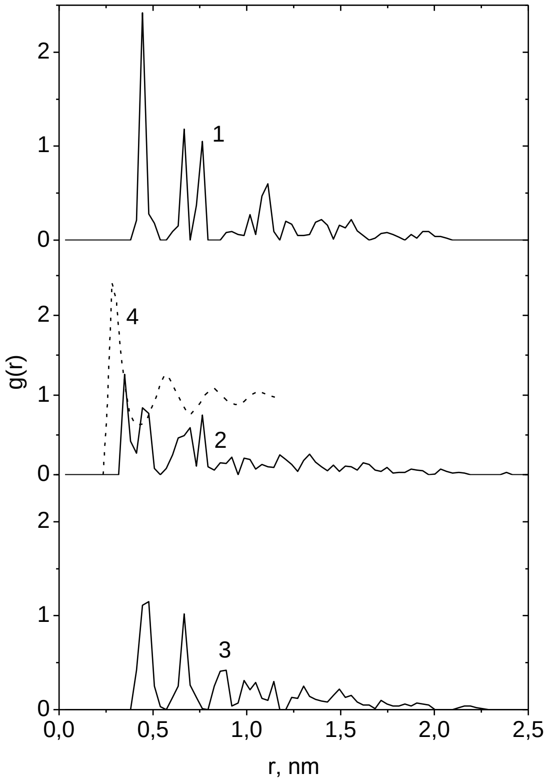


Figure 2. Radial distribution functions of mercury films on graphene, obtained with the atomic interaction potentials (*1*) LJ, (*2*) Sch, (*3*) SG; (*4*) *g(r)* of bulk liquid mercury (MD calculations) [56].

Due to the thinness of the film, its z-profile of density was determined quite roughly and revealed no oscillations for the three types of potentials. However, the distribution of Hg atoms over the graphene surface was neither homogeneous nor uniform for the considered cases. The greatest tendency toward the formation of dense clusters in a film was characteristic for the system created using the LJ potential for mercury (Figure 2), where the first three peaks of function $g(r)$ were the highest and well resolved. A tighter and more compact structure was characteristic of the film obtained using the Sch potential: only the first four peaks of function $g(r)$ were clearly resolved. In this case, however, the Hg film was also shown to be very loose, so the first peak of function $g(r)$ shifted ~0.07 nm away from the position of the corresponding peak of function $g(r)$ for bulk liquid mercury [56]. The four first peaks of the film were distributed between the positions of the first and third peaks of function $g(r)$ for liquid mercury. The radial distribution function for the film obtained with the SG potential had the greatest (~0.17 nm, relative to the position of peak of $g(r)_{liquid}$) shift of the first peak. The emergence of the second peak $g(r)$ of this film only slightly anticipated the position of the third peak of this function for the Hg film formed using the Sch potential.

The specificity of the geometry of system requires individual consideration of horizontal and vertical mobility of Hg atoms. The behavior of the horizontal D_{xy} and vertical D_z components of the self-diffusion coefficient of Hg when calculating at the time intervals p of 200 ps with different atomic interaction potentials for mercury is shown in Figure 3. Component D_{xy} grows only up to p = 2. At subsequent time intervals, D_{xy} usually stabilized or fell inconsiderably.

This behavior of D_{xy} was due mainly to the initial sealing of the Hg film and the subsequent retention of its density. The highest values of D_{xy} were obtained using the SG potential, while the lowest values were obtained with the LJ potential. The Sch potential produced most stable values of D_{xy} at p ≥ 2. In addition, these values did not differ appreciably from the D_{xy} value obtained with the LJ potential. The vertical component of the mobility of Hg atoms behaved differently for all considered potentials. In all cases, the D_z value grew nonmonotonously along with p. Finally, the maximum value of D_z was reached with the LJ potential; the minimum value, with the SG potential. In this context, the situation is inverse to the behavior of component D_{xy} at high p. Another feature of dependence $D_z(n)$ was determined by the tendency toward vaporization from the film of Hg atoms with each potential. For all three of our model potentials, self-diffusion coefficients were obtained that were lower than the experimental value of D (15.9 × 10^{-11} m^2/s at T = 298 K) for liquid mercury [57]. Somewhat better agreement with the calculated values of $D = D_{xy} + D_z$ was achieved when D was determined via nonelastic neutron scattering on liquid mercury (14.3 × 10^{-11} m^2/s at = 297 K) [58].

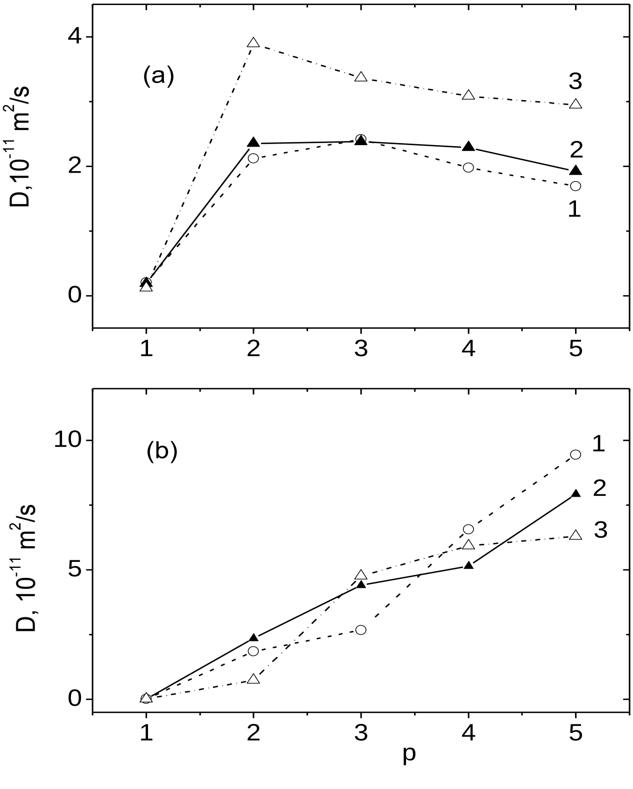


Figure 3. (a) Horizontal and (b) vertical components of the mobility coefficients of Hg atoms in mercury films on graphene, obtained using the atomic potentials (*1*) LJ, (*2*) Sch, (*3*) SG; *p* is the number of the interval in which coefficients D_{xy} and D_z were determined.

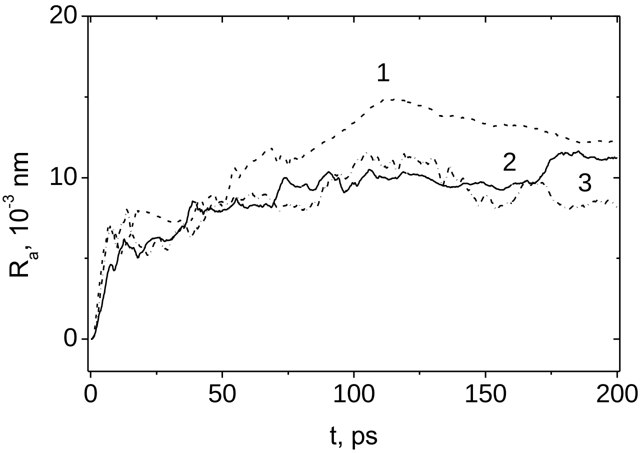


Figure 4. Evolution of the roughness of graphene coated by mercury films, obtained using the atomic potentials (*1*) LJ, (*2*) Sch, and (*3*) SG.

Like hydrogenation, a mercury film that forms on graphene affects its 3D structure (i.e., its roughness R_a). In calculations, the R_a value increases for Hg films that form with all three potentials (Figure 4). The highest R_a values are characteristic of graphene with a metal film obtained via Lennard Jones interaction. The Hg films created with the Sch and SG potentials have similar R_a values throughout all calculations. At the final step of calculation, however, the R_a value for the Hg film formed as the result of using the SG potential becomes lower.

4. Mercury Droplet Formation on a Graphene Surface

Results present in this section, obtained using the Sch potential. Taking into account the value of the time step, the calculation time, and the addend for the increase in the temperature, it is easy to show that the average rate of the system heating is ~10^{11} K/s. Under these conditions of incomplete structural relaxation of the system, it may be superheated. In the case of metals, the superheating is aggravated by the effect of the electron subsystem, which stabilizes the condensed state. Variations accompanying the heating of a mercury film on graphene are illustrated in Figure 5. The liquid metal film begins to partly separate out of graphene already at T = 300 K.

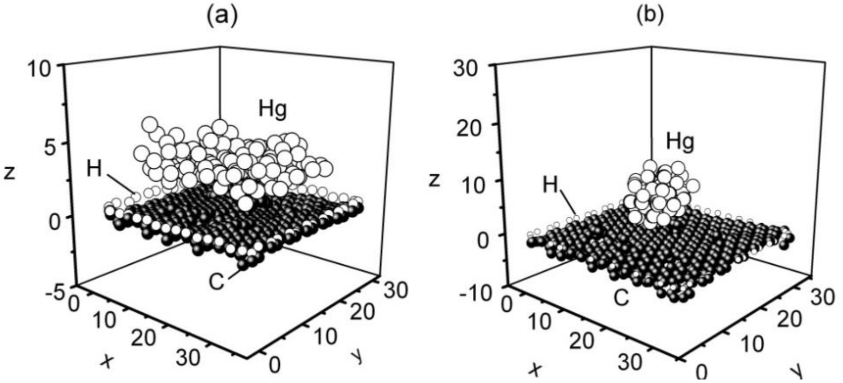


Figure 5. Configurations of the “Hg film on partly hydrogenated defective graphene” system resulting from stepwise heating at temperatures of (a) 300 and (b) 1100 K. Coordinates of atoms are given in angstroms.

This is reflected in the rise of the film edges over graphene and film thickening. The atoms of the central region of the bent Hg film are more strongly bonded to the substrate and have average minimum distance (created by 12–18 Hg atoms) = 0.28 nm. At 600 K, the Hg film is completely transformed into a droplet contacting with graphene. In this case, average distance $\bar{r}_{\mathrm{C-Hg}}^{\mathrm{min}}$ increases to 0.34 nm. A further increase in the temperature leads to a higher rise of the majority of the droplet mass over the graphene surface. For example, at 1100 K, $\bar{r}_{\mathrm{C-Hg}}^{\mathrm{min}}$ = 0.47 nm.

As the Hg film contracts into the droplet, horizontal component D_{xy} of the mobility coefficient of mercury atoms decreases, while vertical component D_z passes through a minimum at 600 K (Figure 6). The smooth decrease in D_{xy} characterizes the rolling of the film into a dense droplet. The behavior of component D_z indicates that the process of droplet formation ends at T = 600 K, and, upon a further increase in the temperature, the vertical mobility is somewhat enhanced because of a slight increase in the distance between the droplet and the graphene surface.

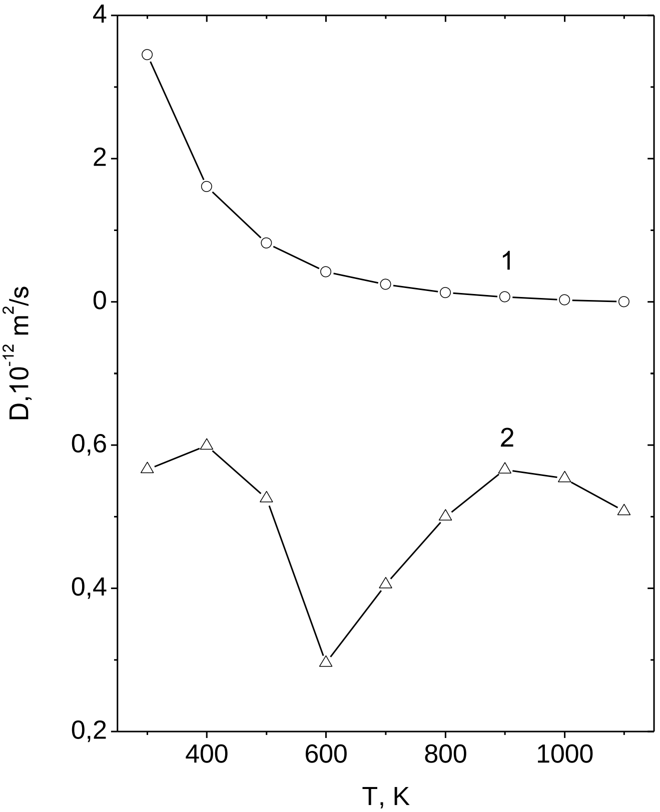


Figure 6. Temperature dependences of the (*1*) horizontal and (*2*) vertical components of the mobility coefficient for Hg atoms.

The extent of the transformation of the vibrational spectra of Hg atoms with the temperature increasing from 300 to 1100 K is illustrated in Figure 7. At T = 300 K, the spectrum of the horizontal vibrations is characterized by strong bursts diminishing with frequency. At 1100 K, the asymptotic of this spectrum remains unchanged, but the intensity of the decreasing peaks drops by six or seven times. The intensity of the vertical vibration spectrum gradually decreases down to disappearance at frequencies $\omega \geq 9.1 \times 10^{12}$ s^{-1} irrespective of the temperature of mercury. However, as the temperature increases, the small–scale vibrations imposed onto the spectrum pattern are smoothed out. The vertical vibration spectrum is wider than the spectrum of horizontal vibrations of Hg atoms.

Vertical (scanned along the *oz* axis) density profiles $\rho(z)$ of mercury at 300 and 600 K are presented in Figure 8. The narrow $\rho(z)$ profile measured at T = 300 K has two sharp peaks, which suggest a predominantly two–layer arrangement of Hg atoms on graphene. However, at T = 600 K, the density profile widens and shifts upward. The low intensity of the $\rho(z)$ spectrum at the edges and the higher density of the intense peaks in the middle of the spectrum characterize the appearance of a spherelike formation, i.e., a droplet with a layered structure, which is evident from the large number of narrow peaks in the $\rho(z)$ spectrum. The very close arrangement of a number of these peaks indicates the irregularity of the formed structure.

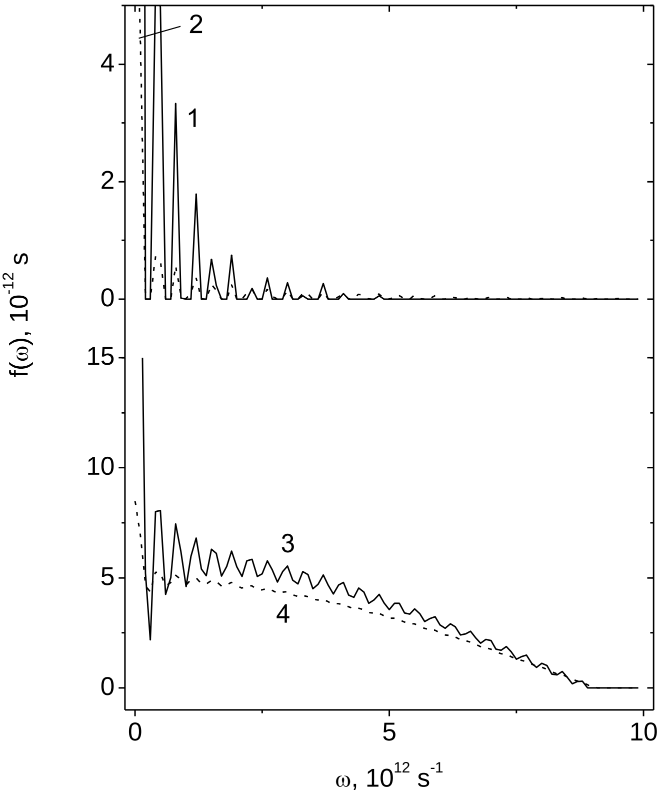


Figure 7. Frequency dependences of the (*1*, *2*) horizontal and (*3*, *4*) vertical components of phonon spectrum of liquid mercury on graphene measured at different temperatures: (*1*, *3*) 300 and (*2*, *4*) 1100 K.

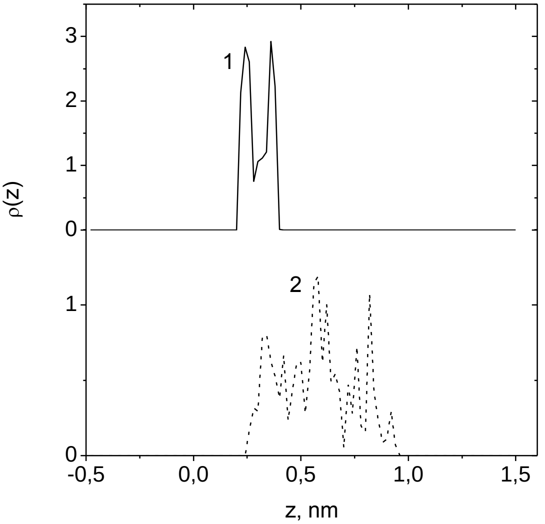


Figure 8. Vertical density profiles for liquid mercury on graphene at different temperatures: (*1*) 300 and (*2*) 600 K.

The $g(r)$ radial distribution functions (Figure 9) plotted for the Hg atom nearest to the center of mass of liquid mercury also indicate the formation of a more compact structure at T = 1100 K than that at an initial temperature of 300 K. The $g(r)$ function reflects the spherically averaged structure of liquid mercury, including that in the horizontal plane, while the $\rho(z)$ function does not do so. A reduction in the number of peaks in the $g(r)$ function at T = 1100 K suggests the formation of an irregular compact structure, in which the distances to the first– and second–order neighbors are estimated to be r_1 = 0.29 nm and r_2 = 0.48–0.57 nm, respectively. The experimental values of these parameters for liquid mercury at 300 K are r_1 = 0.31 nm and r_2 = 0.59 nm [59].

Variations in the wettability that accompany mercury film rolling into a droplet are evident from the temperature dependence of calculated contact angle θ (Figure 10). An initial increase in the $\theta(T)$ function (up to T = 500 K) is due to the predominance of the influence of film heating over the effect relevant to variations in its morphology. It is known that, as the temperature increases, the blunt contact angle of a droplet becomes closer to the flat angle. In spite of a noticeable rise of the droplet over graphene, which begins from

600 K, its separation from the substrate may only be related to a temperature of 800 K. The calculation at 600 K ends when seven Hg atoms are still located at distances r from graphene shorter than distance $r_{\min}$ = 0.3727 nm corresponding to the minimum of the LJ potential describing the $\mathrm{Hg-C}$ interactions. At 700 K two such cases are observed, while, at T = 800 and 900 K, none take place. However, one and two Hg atoms with $r < r_{\min}$ arise at T = 1000 and 1100 K, respectively. Average angle $\bar{\theta}$ = 127.1°, which corresponds to temperatures of 900–1100 K, may be considered to be the contact angle of a 100–atom cluster of Hg on graphene. This angle is noticeably smaller than the contact angle for a macroscopic droplet of mercury on pyrolytic graphite (dashed line in Figure 10) [16]. This agrees with the common ideas of a reduction in angle θ with a decrease in the droplet radius. The inset of Figure 10 shows the time dependence of θ at 600 K. It can be seen that angle θ has begun to noticeably decrease by the end of the calculation at this temperature.

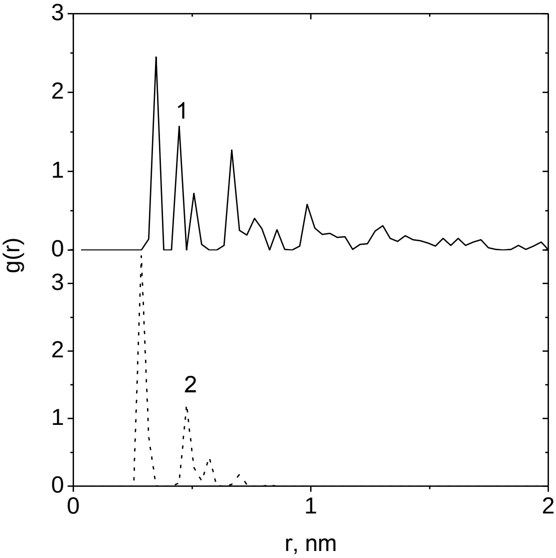


Figure 9. Radial distribution functions calculated for liquid mercury on graphene at different temperatures: (*1*) 300 and (*2*) 1100 K.

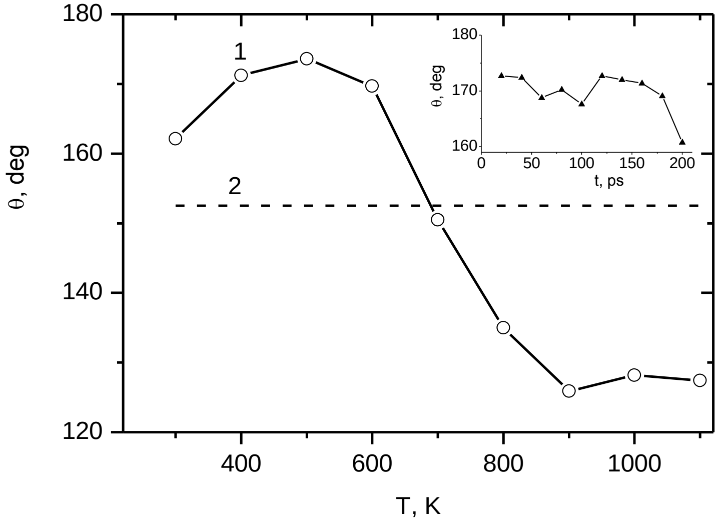


Figure 10. Temperature dependences of contact angles for (*1*) mercury on graphene and (*2*) macroscopic mercury droplet on pyrolytic graphite [16]. The inset shows the temperature dependence of contact angle for a mercury droplet on graphene at *T* = 600 K.

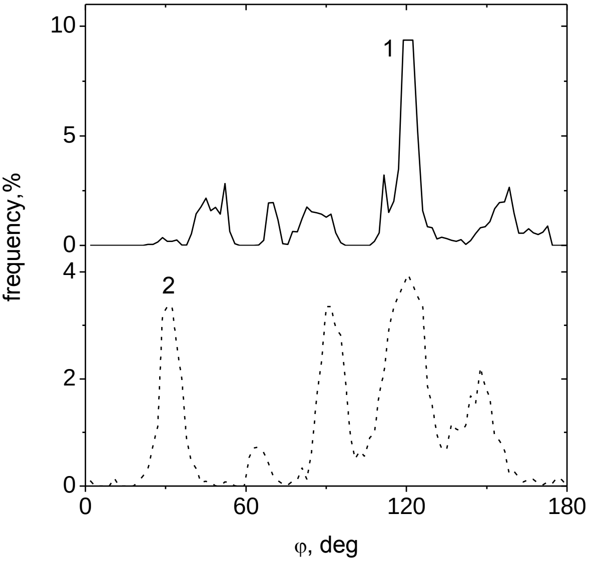


Figure 11. Angular distributions for nearest neighbors in graphene at a high concentration of Stone–Wales defects and different temperatures: (*1*) 300 and (*2*) 1100 K.

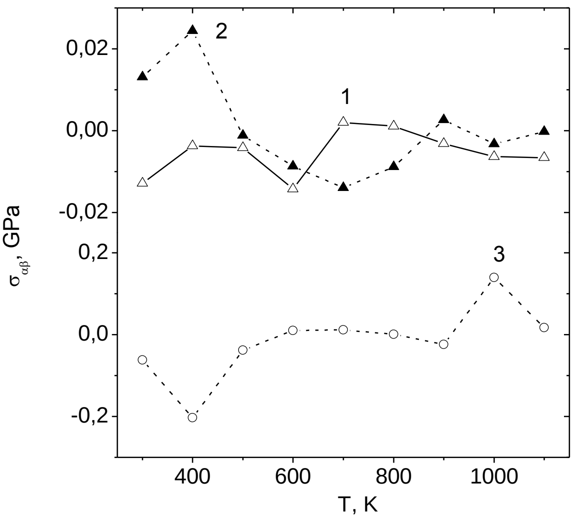


Figure 12. Temperature dependences of the components of the stress tensor in the plane of a mercury–coated defective graphene sheet: (*1*) σ_{zx}, (*2*) σ_{zy}, and (*3*) σ_{zz}.

A peak at 120°, which indicates the presence of the main elements of the two–dimensional structure, i.e., hexagonal honeycombs, dominates in the angular distribution of the nearest neighbors in graphene at T = 300 K (Figure 11). Additional peaks arise in this distribution due to the high density of the Stone–Wales defects (penta– and heptagonal cells). In spite of the fact that 1100 K is not a high temperature for graphene (its melting temperature is T_m = 4900 K), its structure has already suffered from obvious changes. The peak at 120° C has become significantly wider. Moreover, the intensities of peaks at 30°, 90°, and 148° have substantially increased.

These changes indicate the growth of the defects in the graphene structure at T = 1100 K. Stresses σ_{zx} and σ_{zy}, which characterize the action of the internal horizontal forces in the grapheme plane have close values, which weakly vary with an increase in the temperature (Figure 12). A noticeable difference between these stresses, which is observed at T = 300 K, disappears while approaching a temperature of 500 K. The values of stress σ_{zz}, which characterizes the action of the vertically directed forces, have the same order of magnitude as stresses σ_{zx} and σ_{zy} have. The $\sigma_{zz}(T)$ function comprises two regions of the most rapid variations, i.e., a decrease upon heating to 400 K and an increase upon heating after 1000 K. The lowest values of σ_{zz} are observed in a temperature range of 600–800 K, in which the majority of the droplet mass rises over graphene.

Roughness R_a of graphene saturated with the Stone–Wales defects rapidly increases with temperature (Figure 13). As a result of vertical bombardment by Xe_{13} clusters with an energy of 30 eV, graphene containing vacancies and coated with a mercury film acquires a roughness, which is close to R_a at 400 K without the bombardment [34]. The strong bond between carbon atoms in graphene is better preserved at a high temperature (T ≥ 1000 K), when the simulation is performed in terms of the Sch potential than within the framework of the SG potential.

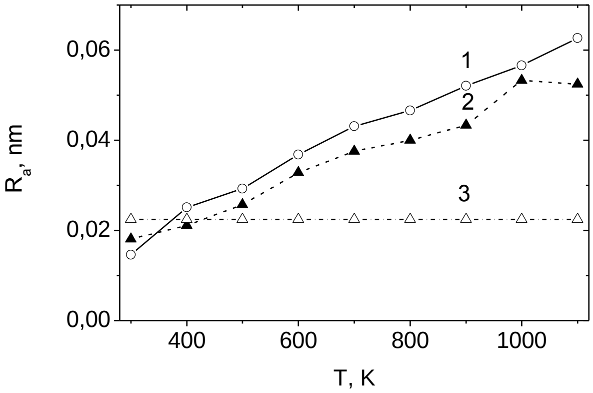


Figure 13. Temperature dependences of roughness coefficient for mercury–coated graphene with regard to Hg–Hg interactions plotted with the use of different models: (*1*) SG potential and (*2, 3*) Sch potential. Temperature is varied by means of (*1, 2*) heating and (*3*) vertical bombardment by Xe_{13} clusters with an energy of 30 eV.

5. THE XENON CLUSTER BOMBARDMENT OF MERCURY ON GRAPHENE

In this section we consider the behavior of the mercury film on graphene when the $\mathrm{Hg-Hg}$ interaction is based on the Schwerdtfeger interaction potential. The cluster bombardment using 125 impacts with an angle of incidence of 0° did not lead to any significant removal of mercury from graphene at all energies of Xe_{13} clusters in the range of 5–30 eV. As a rule, more than half of the Hg atoms after the completion of the bombardment were bound with graphene, as before. The variations in the principal components (σ_{xx}, σ_{yy}, and σ_{zz}) of the stress tensor for the Hg film located on graphene under the action of 5–eV cluster impacts is shown in Figure 14. In the case of structured media, the pressure tensor is not necessarily symmetric, because the extrinsic angular momentum can transform into the intrinsic one, and vice versa. Here, only the total angular momentum must be conserved. In the first approximation, the liquid–metal film on a solid surface can be regarded as a viscous liquid. Successive Xe_{13} cluster impacts force this liquid to move. In this case, the stress tensor is defined as [60]

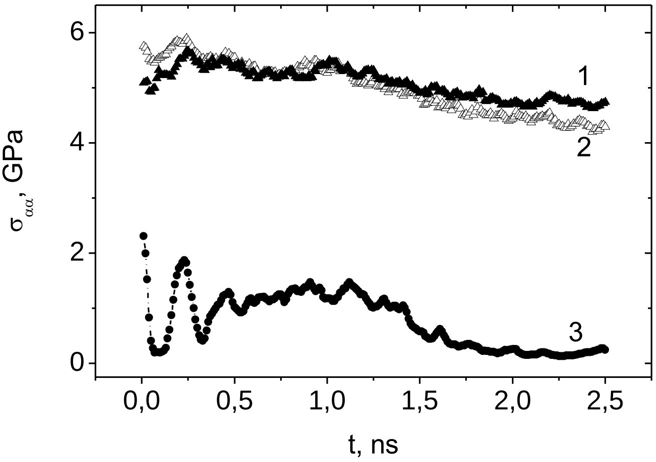


Figure 14. Time dependence of the diagonal stress tensor components (*1*) σ_{xx}, (*2*) σ_{yy}, and (*3*) σ_{zz} for the Hg film on graphene subjected to 5–eV Xe_{13} cluster bombardment.

$$\sigma_{\alpha\beta} = -P\delta_{\alpha\beta} + \sigma'_{\alpha\beta}, \tag{17}$$

where $\sigma'_{\alpha\beta}$ is the viscous stress tensor.

The closeness of the functions $\sigma_{xx}(t)$ and $\sigma_{yy}(t)$ (Figure 14) indicates very small values of σ'_{xx} and σ'_{yy}. At the same time, the function $\sigma_{zz}(t)$ differs significantly from the functions $\sigma_{xx}(t)$ and $\sigma_{yy}(t)$. This is related to the fact that graphene hinders motion in the vertical downward direction. The effective σ'_{zz} turns out to be very large (compared with P). Large oscillations of the function $\sigma_{zz}(t)$ are also related to the presence of graphene reflecting Hg atoms upward after each collision with Xe atoms. It can be seen that the relaxation of stresses σ_{xx} and σ_{yy} occurs faster than the decrease in the stress σ_{zz}.

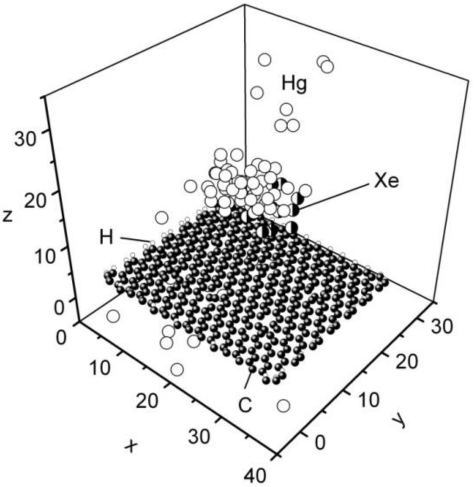


Figure 15. Configuration of a system consisting of a mercury film on a partially hydrogenated imperfect graphene sheet after bombardment by a beam of Xe_{13} clusters at the angle of incidence of 60° and the energy equal to 10 eV. The coordinates of atoms are given in angstroms.

The bombardment at the angle of incidence equal to 45° was considerably more successful. In this case, beginning with the energy of beam equal to 15 eV, graphene was almost completely cleaned of mercury. Only single atoms could remain connected with the graphene sheet; moreover, the majority of these atoms were retained at the edges of the sheet. The remaining atoms of Hg were scattered far beyond the limits of the graphene sheet predominantly in two directions (in the horizontal direction at a sharp angle to the axis ox, and upward). As a rule, the Hg atoms were knocked out from the film one by one and less frequently in the form of dimers and trimers. However, at the energies of the cluster beam $E_{\text{Xe}} \geq$ 15 eV there was always separated also a drop of mercury from graphene. An increase in the angle of incidence of the Xe_{13} clusters to 60° led to the removal of mercury from graphene upon the energy of the beam of 10 eV (Figure 15). A subsequent increase in the energy of the cluster beam at θ = 60° did not give a desired result: graphene was not cleaned of mercury.

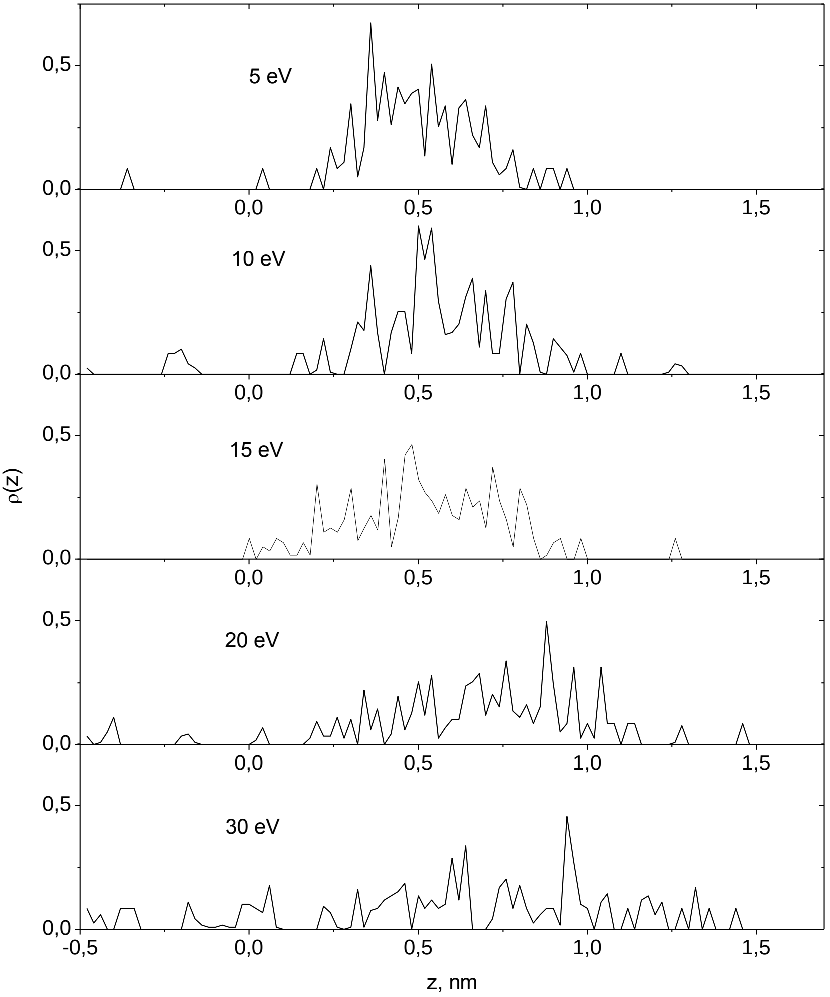


Figure 16. Vertical profiles of the density of liquid mercury on graphene. Numbers (eV) indicate the energies of the falling clusters.

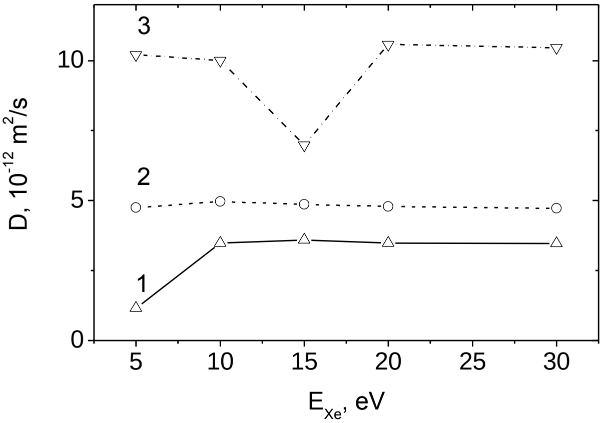


Figure 17. Self-diffusion coefficients of Hg atoms calculated for the cases of bombardment of the target at the angles of incidence (*1*) 0°, (*2*) 45°, and (*3*) 60° depending on the energies of the cluster beam E_{Xe}.

The vertical profiles of the mercury density reflect the displacement of the atoms of metal predominantly upward as a result of the cluster bombardment of the target at the angle of incidence of clusters equal to θ = 0° (Figure 16). The maximum of the density profile is consecutively displaced upward with an increasing energy of the bombarding clusters. A delay in this motion is observed only at E_{Xe} = 15 eV, where the position of the density maximum deflected slightly to the reverse side in comparison with the position for the profile at E_{Xe} = 10 eV. However, already at E_{Xe} = 20 eV the position of the maximum density substantially increased in height and continued increasing at E_{Xe} = 30 eV. The density profiles at $E_{\mathrm{Xe}} \geq$ 20 eV increase their vertical extents in both directions (upward and downward).

With an increase in the angle of incidence of the xenon clusters, there occurs an increase in the self–diffusion coefficient of mercury atoms; especially, this is noticeable on going from the angle θ = 45° to the angle of 60°. The lowest value of the self–diffusion coefficient of Hg atoms is observed upon the vertical bombardment with the energy of Xe_{13} clusters equal to 5 eV (Figure 17). At energies $E_{\mathrm{Xe}} \geq$ 10 eV and at an angle of incidence θ = 0°, there is a very weak dependence of the self–diffusion coefficient on the energy of the falling clusters. A similar weak dependence is manifested in the entire range of cluster energies at the angle of incidence θ = 45°. At the angle θ = 60°, the $D(E_{\mathrm{Xe}})$ function has a deep minimum at 15 eV. The origin of this minimum is most likely connected with the fact that it occurs upon bombardment with precisely such energy of clusters that the most rapid rolling of the mercury film into a drop occurs, from which the Hg atoms can be kicked out only with difficulty. Except for this specific feature, no significant changes in the behavior of the coefficient of self–diffusion is observed upon the variations in the energy with an angle of incidence θ = 60°. The weak change in the $D(E_{\mathrm{Xe}})$ function indicates the effective removal of heat that is separated upon the impacts using the Berendsen thermostat. In other words, the energy is not accumulated in the system in the course of the bombardment.

The $\sigma_{\alpha\beta}(E_{\mathrm{Xe}})$ dependences of the stresses in the plane of graphene caused by horizontal (Figures 18a, 18b) and vertical (Figure 18c) forces exhibit a complex behavior, which is different for the different angles of incidence. As a rule, the stresses σ_{zz} created by vertical forces are noticeably higher than the stresses σ_{zx} and σ_{zy} that appear due to the action of horizontal forces. At cluster energies E_{Xe} that lead to the detachment of the majority of Hg atoms from graphene, the stress σ_{zz} has relatively low values. Recall that this occurs at energies $E_{\mathrm{Xe}} \geq$ 15 eV at an angle of incidence of 45° and at E_{Xe} = 10 eV at the angle θ = 60°.

The roughness R_{a} of graphene increases continuously in the course of cluster bombardment. The inset in Figure 19 gives a representation of the variation of the function $R_{\mathrm{a}}(t)$ in time in the case of bombardment with an energy of Xe_{13} clusters equal to 15 eV at the angle of incidence of 0°. The bombardment has a significant effect on the roughness of graphene. The magnitude of R_{a} increases by 20–40%, even as a result of the bombardment with the energy of clusters equal to only 5 eV; the effect is strongest at an angle of incidence of 60°. The form of functions $R_{\mathrm{a}}(E_{\mathrm{Xe}})$ obtained at different values of the energy of the Xe_{13} clusters is shown in Figure 19. It can be seen that bombardment at an angle of θ = 45° leads to the lowest values of R_a . Thus, after this bombardment at an energy of the beam equal to 30 eV, the value of R_a proves to be below the appropriate characteristics that correspond to the angles of incidence of 0° and 60° by 9.6% and 11.8%, respectively.

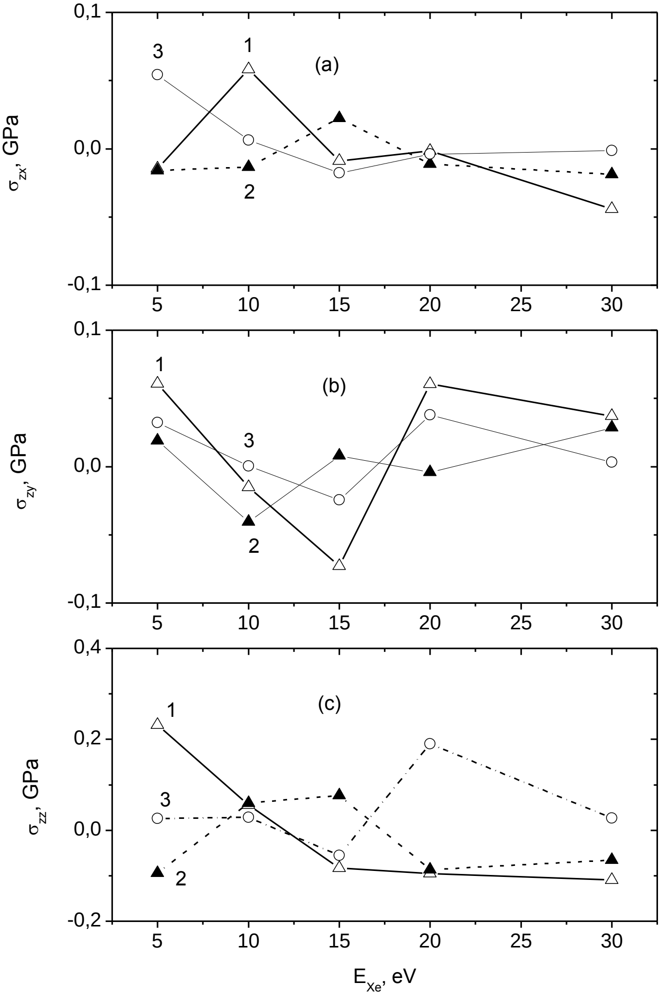


Figure 18. Components of the stress tensor in graphene ((a) σ_{zx}, (b) σ_{zy}, (c) σ_{zz}) obtained for the cases of the bombardment of targets at the angles of incidence (*1*) 0°, (*2*) 45°, and (*3*) 60° depending on the energies of the cluster beam E_{Xe}.

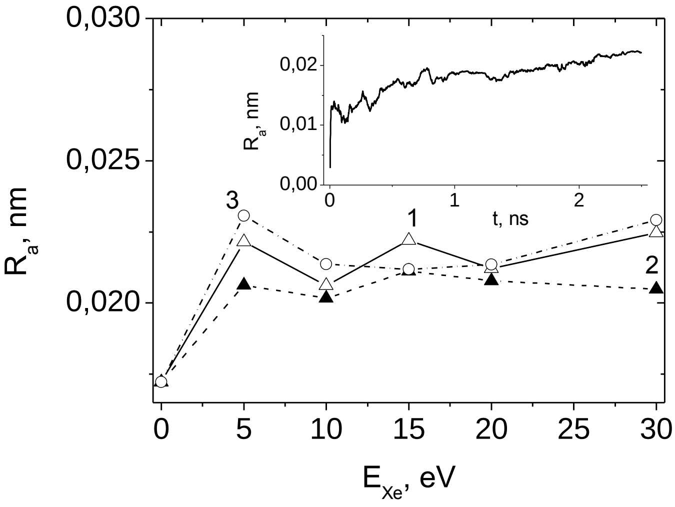


Figure 19. Roughness of graphene obtained as a result of the bombardments of the target at the angles of incidence (*1*) 0°, (*2*) 45°, and (*3*) 60° at the energies of the cluster beam E_{Xe}. Inset shows the change in the roughness of graphene in the course of the bombardment of the target by Xe_{13} clusters at the angle of incidence equal to 0° and at an energy of the cluster beam of 15 eV.

Discussion

Variations in the state of a liquid under a real regime may lead to its superheating, i.e., the existence of the liquid above the boiling temperature upon evaporation. A liquid is superheated as a result of either a rapid heating at a constant pressure or a rapid loss of sealing at a constant temperature. In any case, the liquid enters the region of a nonequilibrium or metastable state, in which its temperature becomes higher than the saturation temperature at normal pressure. The degree of superheating for nonmetal liquids may be as high as several hundred degrees and depends mainly on the rate of heating or reduction in pressure. In the limiting case of complete absence of vapor, very high degrees of superheating may be reached. The superheating is eliminated via an instantaneous change in the phase state, such as explosive boiling up. A high superheating of a liquid is limited by homogeneous nucleation. The ultimate superheating that has been reached for water is (329–333) K [61, 62]. Therewith, a critical nucleus contained nearly 20 molecules [63]. Phase transitions in metal–based systems are distinguished by some specific features.

In this case, the electronic and molecular structures of liquid and vaporous phases occurring at equilibrium are greatly different. For example, liquid mercury and cesium at temperatures close to their ordinary melting points are considered to be normal liquid metals having properties typical of a condensed state. Slight changes in the main properties, such as electrical conductivity or magnetic susceptibility as a result of melting show that the electronic structure of the liquid is similar to that of a crystalline solid. This behavior is commonly explained by the fact that the short–range atomic correlations in a small volume are analogous for a liquid and a crystal. In addition, the ion charges in metals are strongly screened by conduction electrons; therefore, the long–range order of ion potentials is of no importance for either a liquid or a solid. The unusual behavior of a metal–based system is evident from the metal–nonmetal transition, which takes place upon the evaporation of a dense liquid, i.e., when it passes into a rarefied vapor, or in the case of liquid expansion upon heating. The low surface free energy of the majority of nonmetal solids excludes their wettability with inert (nonreactive) liquid metals. However, for mercury located on glass, quartz, or sapphire, a prewetting transition is distinctly observed. The existence of the metal–nonmetal transition noticeably affects the thermodynamic, structural, interfacial, and dynamic properties of metals. The conductivity–density dependence for bivalent mercury may be divided into three regions.

Mercury is a polyvalent metal, which is available for studying in the liquid state at low temperatures. The critical point of its vapor is characterized by the following parameters: T_c = 1751 K, P_c = 167.3 MPa, and ρ_c = 5.8 g/cm^3. Mercury has the lowest critical temperature of those known for all liquid metals. This fact is of importance from the point of view of precise measurement of physical properties at high temperatures and pressures.

The experimental data on droplet evaporation on a hot surface indicate the existence of a discontinuity in the dependence of temperature difference $\Delta T = T^i_\mathrm{vap} - T^i_\mathrm{liq}$ (*i* denotes the interface) on vapor pressure P^i_vap [64, 65]. At a liquid–vapor interface, the temperature is always higher on the side of the vapor. This is explained by the fact that high–energy molecules are primarily evaporated, while molecules with lower energies remain in the droplet. A reduction in the flux of molecules to the vapor phase is mainly observed at high temperatures – for water, at $T / T_c \approx 0.84$ [66]. The value of the temperature discontinuity for water may be higher than 1400 K [66]. Mercury atoms are 11 times heavier than water molecules. Mercury is characterized by another type of interaction. It may be thought that a mercury droplet remains stable at high temperatures because of a reduction in the flux of Hg atoms to the vapor phase; however, the characteristic features of this process differ from the behavior of water.

The high stability of a model mercury droplet may also be explained as follows. The interaction potential between two mercury atoms is, as a rule, considered to be a potential between highly polarizable closed shells, which permit very low migration of electron density from one partner to another; i.e., this potential is, to some extent, similar to a potential function that describes the interaction between atoms of noble gases. We have proven (using the Sch potential) the formation of a mercury droplet on graphene upon rapid heating using a calculation similar to that reported here, but performed in terms of the SG potential.

In [67], it was noted that the model approximations that use pair interaction potentials to describe the liquid–gas transition for mercury are rough [67]. Experimental gas–liquid coexistence curves may be precisely reproduced, provided that the two–atom curves obtained for potential energy from the former principles are supplemented with the many–body potential, which describes the associative interaction of an atom with neighboring atoms that altogether form a virtual cluster. The liquid–gas transition for mercury is distinguished by the fact that the local electronic states change from metal to nonmetal ones because of weakened many–particle interactions and decreased average coordination numbers.

According to the calculation in terms of the Sch potential, Hg film rolls into a droplet upon heating. By the end of the calculation at 600 K, an almost spherical droplet is formed on graphene, with the droplet remaining near the graphene surface even at 1100 K. When the SG potential function is used, the distance between the droplet and graphene rapidly increases up to a temperature of 1000 K. No significant separation of Hg atoms from the droplet takes place in this case. Most likely, the Sch and SG potentials give an overestimated indirect effect of the electron component on the $\mathrm{Hg-Hg}$ interaction, which leads to the high stability of liquid mercury with respect to its vapor.

It is of interest to compare the results of the study of the removal of films of copper and lead by the bombardment with clusters of rare gases with the present investigation of the purification of graphene from mercury. First of all, the different mechanisms of the detachment of these heavy metals from graphene during the irradiation of the target by a cluster beam should be noted.

In the case of the bombardment of the copper film with Ar_{13} clusters, separate Cu atoms are knocked out [24–29]. No regime of bombardment led to the separation of fairly large fragments of the Cu film from graphene. When the lead film is bombarded, separate atoms are also knocked out, but the prevailing mechanism of the removal of the metal form graphene is the separation of islands of a Pb film from the substrate [34]. Since it was only detached away graphene, the island experiences a transformation from a two–dimensional to a three–dimensional structure; otherwise, mercury is separated from graphene. The unique behavior of mercury is due to its liquid state and the poor wetting of graphene; as a result, the Hg film has a tendency to roll into a drop. For this reason, both separate atoms and droplets of significant size are separated from graphene in the course of bombardment. Let us emphasize that it is precisely a drop that is torn off, rather than an island with a two–dimensional morphology. There are several other differences in the processes of the removal of the film of heavy metals from graphene. Thus, the film of copper is not completely removed from graphene, even at an energy of the beam equal to 30 eV at angles of incidence of 0° and 60° [24], and the most efficient method is removal using cluster bombardment at an angle θ = 45°. In the case of lead, the most efficient procedure can be considered to be irradiation by a cluster beam at the angles of incidence of 0° and 60°. In this case, graphene was completely cleaned of metal at energies of the beam equal to 10 and 15 eV. Complete cleaning was also achieved at an angle θ = 45°, but the energy of the cluster beam required in this case was equal to 20 eV. The greatest effect from the bombardment of the mercury–on–graphene target is obtained at an angle of incidence equal to 45°. At this angle of incidence, graphene is cleaned of Hg at all energies $E_{\mathrm{Xe}} \geq$ 15 eV. A less stable cleaning effect was achieved at an angle of incidence of 60°. In the case of an angle of incidence equal to 0°, no significant removal of mercury from graphene occurs in the range of energies of the beam equal to 5–30 eV. Thus, the removal of different heavy metals requires different conditions for bombardment and occurs via different mechanisms.

To check the correctness of the results, we also conducted calculations with another pair potential for mercury and another potential that describes the mercury–graphene interaction. The $\mathrm{Hg}-\mathrm{Hg}$ interactions were determined based on applying the potential proposed by Silver and Goldman with the parameters given in [42]. Here, we obtained results close to those where the Sch potential served as the potential function for mercury. In the calculations that applied the SG potential, upon bombardment, the Hg film was more rapidly transformed into the drop and was separated from graphene. The complete removal of mercury from graphene was only achieved at an angle of incidence equal to 45° at $E_{\mathrm{Xe}} \geq 15$ eV. When using a Morse potential with the parameters given in [68] for the representation of $\mathrm{Hg}-\mathrm{C}$ interactions, mercury upon the bombardment was separated from graphene more difficultly, and the complete cleaning at the angle $\theta = 45°$ was achieved at the energies $E_{\mathrm{Xe}} \geq 20$ eV.

CONCLUSION

The forces of cohesion between mercury and graphene atoms are weak, compared to the ones between mercury atoms. Mercury tends towards its natural boundary angle while wetting decreases and mercury gradually consolidates into individual drops. This phenomenon is largely reproduced using the Sch potential. A tendency toward drop formation is also observed for the LJ and SG potentials, but in these cases there are considerably more individual atoms on the graphene surface, and each drop has a less distinct profile. For real mercury, vaporization proceeds at temperatures above 291 K. Cohesion with modified graphene does not allow Hg atoms to detach from the film at distances much greater than atomic ones at 300 K. However, the tendency toward the vaporization of Hg atoms is still observed in model systems and is clearer when using the SG potential.

Molecular dynamics has been employed to study the stepwise heating of a mercury film on imperfect graphene. A graphene sheet with a high concentration of Stone–Wales defects and hydrogenated edges has been examined. An increase in temperature has been shown to cause gradual rolling of the film into a droplet and a slow movement of the droplet away from graphene. The horizontal component of the mobility coefficient of Hg atoms smoothly decreases in the course of this process, while the vertical component nonmonotonically increases after a reduction reached by a temperature of 600 K. As a whole, the spectra of the horizontal and vertical vibrations of Hg atoms similarly vary with a rise in the temperature; i.e., the small–scale thermal fluctuations in the spectra are smoothed. The vertical profile of mercury density shifts upward and widens to a size that corresponds to the diameter of the formed liquid metal droplet. The formation of the mercury droplet is accompanied by a reduction in the domain of the radial distribution function and a decrease in the number and intensity of pronounced peaks of the $g(r)$ function. An increase in the temperature accelerates the formation of the droplet and decreases the contact angle. In the angular distribution of nearest neighbors, the intensity of the main peak at 120°, which reflects the hexagonal cells, decreases, while intense peaks corresponding to angles of 30°, 90°, and 148° arise. The stresses in the graphene plane that are caused by the horizontal and vertical forces have close magnitudes in the considered temperature range. Graphene roughness rapidly grows with temperature, reaching a maximum value at 1000 K. Hydrogenated graphene edges are not damaged significantly upon heating to high temperatures. Thus, upon rapid heating, a mercury film on graphene is transformed into a droplet with substantial changes in atomic packing and physical properties.

The behavior of a system of mercury–on–partially–hydrogenated–graphene has been investigated under irradiation by a beam of Xe_{13} clusters with energies of 5–30 eV at angles of incidence equal to 0°, 45°, and 60°. Over a wide range of energies ($E_{\mathrm{Xe}} \geq$ 15 eV), the almost complete removal of mercury from graphene was only achieved at an angle of incidence of 45°. The film of mercury, which has a tendency to become rolled up into a drop, is separated from graphene in the form of single atoms, dimers, trimers, and spherical droplets. In the course of the bombardment, mercury exhibits a weak cohesion with graphene. With an increase in the energy of the falling clusters from 5 to 30 eV, the $\rho(z)$ profile evolves in a complex way, demonstrating the formation of a drop of mercury on graphene, as well as the formation of a vapor of Hg monomers. The smallest change in the components of the mobility of Hg atoms upon the variation of the energy of the cluster beam occurs at an angle of incidence equal to 45°. At the energies of the cluster beam under consideration, the stresses in the plane of graphene caused by vertical forces noticeably exceed the stresses created by the horizontally directed forces, regardless of the angle of incidence. The roughness of graphene increases noticeably in the course of cluster bombardment. The lowest roughness is demonstrated by graphene subjected to irradiation by the beam of clusters with an angle of incidence equal to 45°. The hydrogenated edges of graphene do not suffer noticeable damages at all the energies investigated and at all the angles of incidence of the bombarding clusters.

References

[1] Merian, E. *Metals and their compounds in the environment: occurrence, analysis and biological relevance;* VCH Publishers: Weinheim, US, 1991; p. 1438.

[2] Fernandez-Leborans, G. H.; Yolanda, O. *Ecotoxicology Environmental Safety* 2000, 47, 266–276.

[3] Sayari, A.; Hamoudi, S.; Yang, Y. *Chem. Mater.* 2005, 17, 212–216.

[4] Li, R.; Liu, L.; Yang, F. *Chem. Eng. J.* 2013, 229, 460–468.

[5] Bomont, J.-M.; Bretonnet, J.-L. *J. Phys.: Conference Series* 2008, 98, 042018.

[6] Bomont, J.-M.; Bretonnet, J.-L.; Gonzalez, D. J.; Gonzalez, L. E. *Phys. Rev. B* 2009, 79, 144202.

[7] Galashev, A. Y.; Polukhin V. A. *Rus. J. Phys. Chem. A* 2014, 88, 995–999.

[8] Lee, C. W.; Serre, S. D.; Zhao, Y.; Lee, S. J.; Hastings, T. W. *J. Air and Waste Manage. Assoc.* 2008, 58, 484–493.

[9] Huggins, F. E.; Yap, N.; Huffman, G. P.; Senior, C. L. *Fuel Process. Technol.* 2003, 82, 167–196.

[10] Hutson, N. D.; Attwood, B. C.; Scheckel, K. G. *Environ. Sci. Technol.* 2007, 41, 1747–1752.

[11] Wilcox, J.; Sasmaz, E.; Kirchofer, A. *J. Air and Waste Manage. Assoc.* 2011, 61, 418–426.

[12] Awasthi, A.; Bhatt, Y. J.; Garg, S. P. *Meas. Sci. Technol.* 1996, 7, 753–757.

[13] Dujardin, E.; Ebbesen, T. W.; Hiura, H.; Tanigaki, K. *Science* 1994, 265, 1850–1852.

[14] Chen, J. Y.; Kutana, A.; Collier, C. P.; Giapis, K. P. *Science* 2005, 310, 1480–1483.

[15] Galashev, A. Y. *Mol. Simul.* 2010, 36, 273–282.

[16] Kutana, A.; Giapis, K. P. *Phys. Rev. B* 2007, 76, 195444.

[17] Steckel, J. A. *Chem. Phys. Lett.* 2005, 409, 322–330.

[18] Padak, B.; Brunetti, M.; Lewis, A.; Wilcox, J. *Environ. Prog.* 2006, 25, 319–326.

[19] Padak, B.; Wilcox, J. *Carbon* 2007, 47, 2855–2864.

[20] Huggins, F. E.; Huffman, G. P.; Dunham, G. E.; Senior, C. L. *Energy Fuels* 1999, 13, 114–121.

[21] Cao, Y.; Li, X. *Adsorption* 2014, 20, 713–727.

[22] Azamat, J.; Khataee, A.; Joo, S. W. *J. Mol. Graph. Model.* 2014, 53, 112–117.

[23] Kim, H. W.; Yoon, H. W.; Yoon, S.-M.; Yoo, B. M.; Ahn, B. K.; Cho, Y. H.; Shin, H .J.; Yang, H.; Paik, U.; Kwon, S.; Choi, J.-Y.; Park, H. B.; Bum, H. *Science* 2013, 342, 91–95.

[24] Galashev, A. Y.; Polukhin, V. A. *Phys. Met. Metallogr*. 2014, 115, 697–704.

[25] Galashev, A. Y.; Galasheva, A. A. *High Energy Chem*. 2014, 48, 112–116.

[26] Galashev, A. Y. *Fiz. Mezomekh*. 2014, 17, 67–73.

[27] Galashev, A. Y.; Polukhin V. A. *J. Surf. Invest. X-ray, Synchr. Neutr. Techn*. 2014, 8, 1082–1088.

[28] Galashev, A. Y. *Compt. Mater. Sci.* 2015, 98, 123–128.

[29] Galashev, A. Y.; Rakhmanova, O. R. *Chin. Phys. B* 2015, 24, 020701.

[30] Galashev, A. Y.; Rakhmanova, O. R. *High Temp*. 2014, 52, 374–380.

[31] Galashev, A. Y. *High Temp.* 2014, 52, 633–639.

[32] A. E. Galashev, A. Y.; Polukhin, V. A. *Phys. Solid State* 2013, 55, 2368–2373.

[33] Galashev, A. Y.; Rakhmanova, O. R. *Phys.–Usp.* 2014, 57, 970 – 989.

[34] Galashev A. Y.; Galasheva, A. A. *High Energy Chem*. 2015, 49, 117–121.

[35] Zakharchenko, K. V.; Fasolino, A.; Los, J. H.; Katsnelson, M. I. *J. Phys.: Condens. Matter.* 2011, 23, 202202.

[36] Tersoff, J. *Phys. Rev. Lett.* 1988, 61, 2879–2882.

[37] Burgos, E.; Halac, E.; Bonadeo, H. *Chem. Phys. Lett.* 1998, 298, 273–278.

[38] Stuart, S. J.; Tutein, A. V.; Harrison, J. A. *J. Chem. Phys*. 2000, 112, 6472–6486.

[39] Galashev, A. Y.; Polukhin, V. A. *Phys. Solid State* 2013, 55, 1733–1738.

[40] Galashev, A. Y.; Dubovik S. Yu. *Phys. Solid State* 2013, 55, 1976–1983.

[41] Epstein, F.; Powers, M. D. *J. Phys. Chem*. 1953, 57, 336–341.

[42] Munro, L. J.; Johnson, J. K. *J. Chem. Phys.* 2001, 114, 5545–5551.

[43] Kutana, A.; Giapis, K. P. *Nano Lett.* 2006, 6, 656–661.

[44] Schwerdtfeger, P.; Boyd, P. D. N.; Briennes, S.; McFearets, J. S.; Dolg, M.; Liao, M.-S.; Schwarz, W. H. E. *Inorganica Chimica Acta* 1993, 213, 233–246.

[45] Kim, Y. M.; Kim, S.-C. *J. Korean Phys. Soc*. 2002, 40, 293–299.

[46] Li, F.-Y.; Berry, R. S. *J. Phys. Chem*. 1995, 99, 2459–2468.

[47] Ziegler, J. F.; Biersack, J. P.; Littmark, U. *Stopping and ranges of ions in matter*; Pergamon Press: New York, US, 1985; Vol. 1. 321 p.

[48] Delcorte, A.; Garrison, B. J. *J. Phys. Chem. B* 2000, 104, 6785–6800.

[49] Lamari, F. D.; Levesque, D. *Carbone* 2011, 49, 5196–5200.

[50] Wick, C. D.; Martin, M. G.; Siepmann, J. I. *J. Chem. Phys. B*. 2000, 104, 8008–8016.

[51] Verlet, L. *Phys. Rev*. 1967, 159, 98–103.

[52] Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. *J. Chem. Phys*. 1984, 81, 3684–3690.

[53] Hafskjold, B.; Ikeshoji, T. *Phys. Rev. E* 2002, 66, 011203.

[54] Galashev, A. Y. *J. Surf. Invest. X-ray, Synchr. Neutr. Techn.* 2016, 10, 15–22.

[55] Davydov, S. Yu. *Phys. Solid State* 2012, 54, 875–882.

[56] Bosio, L.; Cortes, R.; Segaud, C. *J. Chem. Phys.* 1979, 71, 3595–3600.

[57] Badyal, Y. S.; Bafile, U.; Miyazaki, K.; de Schepper, I. M.; Montfrooij, W. *Phys. Rev. E* 2003, 68, 061208.

[58] Lobo, V. M. M.; Mills, R. *Electrochem. Acta* 1982, 27, 969–971.

[59] Rao, R. V. G.; Murthy, A. K. K. *Phys. Status Solidi B* 1974, 66, 703–707.

[60] Landau, L. D.; Lifshitz, E. M. *Course of Theoretical Physics, Fluid Mechanics*; Nauka: Moscow, US, 1986; Vol. 6, p. 71.

[61] Eberhart, J. G.; Pinks, V. *J. Colloid Interface Sci.* 1985, 107, 574–575.

[62] Salla, J. M.; Demichela, M.; Casal, J. *J. Loss Prevent. Proc. Ind.*, 2006, 19, 690–700.

[63] Hasan, M. N.; Hasan, A.; Ilias, S.; Mitsutake, Y.; Monde, M. *Procedia Engineering* 2014, 90, 618 – 623.

[64] Ward, C. A.; Stanga, D. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* 2001, 64, 051509.

[65] McGaugheya, A. J. H.; Ward, C. A. *J. Appl. Phys.* 2002, 91, 6406–6415.

[66] Lotfi, A.; Vrabec, J.; Fischer J. *Int. J. Heat Mass Transfer* 2014, 73, 303–317.

[67] Kitamura, H. *J. Phys.: Conf. Ser.* 2008, 98, 052010.

[68] Duval, M. C.; Soep, B. *Chem. Phys. Lett.* 1987, 141, 225–231.

In: Computer Vision and Simulation
Editor: Sherri Alexander
ISBN: 978-1-63485-790-1

Chapter 4

INFLUENCE OF YTTRIUM(III) ION ON CALCIUM(II) AND ZINC(II) BIOSPECIATION IN HUMAN BLOOD PLASMA BY COMPUTER SIMULATION

Ivan Ž. Jakovljević[1,*], Djordje Ž. Petrović[2], Milica S. Cvijović[3], Ljubinka G. Joksović[1] and Predrag T. Djurdjević[1]

[1]Faculty of Science, University of Kragujevac, Kragujevac, Serbia
[2]Institute of Nuclear Science “VINČA”,
Laboratory for Radioisotopes Belgrade, Serbia
[3]Faculty of Agriculture, Čačak, Serbia

ABSTRACT

The effect of yttrium(III) ion on calcium(II) and zinc(II) speciation in human blood plasma was studied by computer simulation using the program Hyss2009. Calcium-hydrogen carbonate $[CaHCO_3]^+$ and ternary zinc-cysteinate-citrate $[ZnCysCit]^{-3}$ complexes are predominant species of Ca(II) and Zn(II) ions in normal human blood plasma. Exogenously introduced yttrium(III) ion can compete with Ca(II) and Zn(II) ions for low molecular mass (LMM) ligands in blood plasma, thus influencing

[*] Corresponding Author address. E-mail: ivan_jakovljevic@kg.ac.rs.

their biospeciation. The results showed that at the normal blood yttrium concentration all the Y(III) species are soluble and no precipitate appear. However, at total Y(III) concentration higher than 1×10^{-6} molL^{-1}, the insoluble species become dominant ($Y_2(CO_3)_2$ and YPO_4). At this concentration level of Y(III) the distribution of Ca(II) and Zn(II) species does not change appreciably. If the total concentration of Y(III) is higher than 1×10^{-3} molL^{-1} its influence on biodistribution on Ca(II) and Zn(II) ions is significant. The concentration of free calcium ion increase from 79% to 86% and decreases [$CaHCO_3$] percentage. With further increasing of yttrium concentration (5×10^{-2} molL^{-1}), [$CaHCO_3$] disappear and dominant species is free calcium ion, whit redistribution of zinc species. Main species ZnCysCit (~38%) becomes minor species (<1%), while $ZnCys_2$ (~35%) and ZnCysHis (~20%) become major zinc species.

Keywords: speciation, calcium(II), zinc(II), yttrium(III), blood plasma, computer simulation

INTRODUCTION

Metal ions in human organism may be classified as essential, beneficial, detrimental and toxic. Normally, they are present in trace levels and can exhibit metal-metal interactions. These interactions may be either cooperative or competitive (i.e., synergistic or antagonistic). Metal ions may compete for storage, transport or functional proteins or for binding sites on cellular membranes. In addition, trace metals may compete for blood plasma ligands of low molecular weight especially those forming metal-ligand precipitates.

Externally introduced yttrium (eg, in the form of its radiopharmaceutical complexes) may compete for blood plasma ligands with bivalent ions, Ca and Zn. Since free yttrium ion is present at concentration levels lower than 10^{-7} molL^{-1} no suitable analytical methods exist to measure such low concentrations. Thus, biospeciation could be evaluated only by computer simulation. In this work we used the computer program Hyss2009 [1] to calculate biospeciation of Ca and Zn in the presence of various concentration levels of Y ion in human blood plasma. Earlier, we described the multiphase blood plasma model consisted of about 6000 complexes including insoluble and metal-protein species [2].

Ca(II) and Zn(II) play important roles in the human body [3]. There are many similarities between tri-positive rare earth ions and calcium(II) ions, such as ionic radii, ligand-exchange rates, and coordination numbers.

Thus, tripositive rare earth ions are widely used as calcium(II) probes in biological systems [4]. Moreover, the displacement of calcium(II) ions from biological molecules by rare earth ions closely relates to the biological effects of these ions [4].

Zinc(II) ion is one of the most abundant divalent metal in living organisms and is essential cofactor of many metabolic enzymes and transcription factors. Zinc-deficiency studies of microorganisms followed by those in plants and animals established the importance of zinc to the growth and development in all forms of life [5, 6].

In recent years, application of the rare earths is wide; for example, rare earths as a fertilizer are being applied in agriculture in China [7]. It causes more and more rare earths to enter the environment and human body via the food chain. Therefore, increasing consideration has been given to the effects of rare earths occurring in the biofluids on bioelements essential for life such as Ca and Zn [6]. Because biological effects of metal are controlled by its in vivo speciation; the research on the effect of rare earth ions on bioelement speciation is helpful in understanding the distribution, metabolism, and biological effects of rare earth ions in the life system. However, it is difficult to determine particular metal-complex in the human body using the conventional analytical techniques; thus, computer simulation has been developed as a suitable method to study speciation without disturbing delicate equilibria in body fluids.

Yttrium sources. Yttrium is used to produce electronic devices, including electrodes, electrolytes, electronic filters, lasers and superconductors. At present, yttrium is believed to be hazardous for human health. However, the disposal of yttrium –containing devices enhances the possibility that yttrium may become an environmental pollutant. No biological role has been identified for yttrium although environmentally derived yttrium may concentrate in bones (70 ppb), liver (10 ppb), lung (20 ppb), brain (20 ppb), kidney (6 ppb) and blood (6 ppb) in healthy humans [8]. Human breast milk contains 4 ppm of yttrium. Water soluble yttrium salts, such as the nitrate, are regarded as mildly toxic while its insoluble compounds are non-toxic. However, the element is suspected of being carcinogenic for some animals and humans. Edible plants can have quite a range of yttrium levels, from 20 to 100 ppm (fresh weight) with the highest values being recorded for cabbage [8]. The seeds of woody plants have the highest amounts of all (700 ppm) [8]. Although coal contains yttrium (7-14 ppm) this is not thought to indicate its selective absorption by the organic substances from which it was derived. In

Chinese yttrium mines the dust yttrium concentration ranged from 1.3 to 25.9 mg m^{-3} where 64.1% is yttrium oxide [8].

In experiments on animals, yttrium and its compounds caused lung and liver damage. In rats, inhalation of yttrium citrate caused pulmonary edema and dyspnea, while inhalation of yttrium chloride caused liver edema, pleural effusions and pulmonary hyperemia [9, 10]. Exposure to yttrium compounds in humans may cause lung disease. ^{90}Y obtained from the ^{90}Sr–^{90}Y generator system finds widespread use in the cancer treatment in the form of radiopharmaceutical chelate [11-16]. As other radiopharmaceuticals yttrium is administered by intravenous injection. Radiopharmaceutical yttrium chelates (DTPA, DOTA, etc.) are very stable and usually safe for use. Their injected concentration is about 1×10^{-8} molL^{-1}. The known adverse effect of yttrium radiopharmaceuticals is development of renal toxicity [11]. In blood plasma free yttrium may occur by dissociation of the radiopharmaceutical chelates (e.g., Y-DTPA) or by transmetalation with other ions present in blood plasma, such as zinc and copper ions. These two ions have relatively high stability constants with DTPA ($\log\beta_{\text{Zn-DTPA}}$=18.75, $\log\beta_{\text{Cu-DTPA}}$=21.53) compared to yttrium ion ($\log\beta_{\text{Y-DTPA}}$=22.05). To estimate released free yttrium concentration due to the processes of dissociation and trans-metallation knowledge of the stability constants (under physiological conditions) of metal ions with DTPA are needed. Literature values of Y-DTPA stability constants do not refer to physiological conditions. So, in this work we studied the complexation of Y^{3+} ion with DTPA ligand by potentiometric method under physiological conditions (μ=0.15 molL^{-1} NaCl, t=37°C). To confirm potentiometrically obtained results ESI-MS measurements on Y – DTPA solutions were also performed.

Potentiometric titrations. Potentiometric measurements were made on a Tacussel Isis 20000 pH meter (Courthezon, Vaucluse, France, precision ± 0.1 mV or ± 0.001 pH units) equipped with a Radiometer combined electrode. A Metrohm Dosimat model 665 automatic burette with anti-diffusion tip (Herisau, Switzerland), was used for delivery of the titrant. Potentiometric titrations were carried out in a double-walled glass vessel, thermostatted at 37° C. The ionic strength of all test solutions was adjusted to 0.15 molL^{-1} with sodium chloride. All measurements were performed under a nitrogen atmosphere. The electrode parameters, E_0, Q and E_j from Nernst equation: $E = E_0 + Q\ logh + E_j$ were determined by strong acid-strong base titration to check the system suitability. During the titrations of the test solutions the E_0 and E_j were determined using the data in the acidic region where no hydrolysis or complexation takes place (assuming that h is equal to the analytical concentration of proton), by plotting *E–Q log h* against *h* and extrapolating the straight line so obtained to *h = 0*. The free proton concentration was then calculated through the equation: $logh = (E - E_0 - E_j)/Q$ which was applied to the whole titration curve. All titrations were carried in duplicate. The agreement between duplicate titration was better than 1%. The Y^{3+}-DTPA solutions were titrated with sodium hydroxide and all titration were performed in the pH range from *ca.* 2 to 11 with constant ionic strength (I = 0.15 molL^{-1} NaCl) and under purified nitrogen atmosphere at 37° C. Molar ratios between yttrium ion and DTPA ranged from 1:1 to 1:2, respectively. The concentration stability constants of complexes formed in the solutions were calculated with the aid of the suite of computer programs Hyperquad2006 [1].

The species formed in the studied systems were characterized by the general equilibrium:

$$pY^{3+} + qDTPA^{5-} + rH^{+} \leftrightarrow [M_pDTPA_qH_r]^{(3p-5q+r)}$$

and the corresponding constants are given by:

$$\beta_{p,q,r} = \frac{[Y_p(DTPA_q)H_r]}{[Y]^p[DTPA]^q[H]^r}$$

The obtained results indicated that formation of 1:1 complexes are dominant. Calculated overall stability constants of the yttrium ion with DTPA are given in Table 1. This is in agreement with the majority of published papers [17].

The distribution diagrams of complexes in Y^{3+} -DTPA solutions were calculated with the aid of program HySS2009. Distribution diagram is shown in Figure 1. As can be seen from Figure 1, dominating complex is Y(DTPA), with the maximum concentration at pH=4. Y(DTPA)H complex appear at lower pH, ca. 2, in small amount. Main complex Y(DTPA) is dominant complex in wide pH region.

ESI-MS measurements. ESI MS spectra were collected on an LCQ Fleet 3D Ion Trap Mass Spectrometer (Thermo Fisher Scientific, Waltham, MA, USA). To further confirm the speciation derived from potentiometric measurements, ESI-MS measurements were made on yttrium-DTPA solution. The most intensive signals can be attributed to the 1:1 (metal to ligand) complexe. The ESI-MS data of yttrium-DTPA solution pH 5.5 adjusted with ammonium formate buffer show evidence to formation of the complex

Y(DTPA) as shown in Table 2. Representative ESI-MS spectra is shown in Figure 2.

Table 1. Calculated overall stability constants logβ$_{pqr}$ (SD) for complexation of Y^{3+} ion with DTPA at physiological conditions (T=37° C, I=0.15 molL^{-1} NaCl)

p,q,r	logβ±SD
1,1,0	22.51±0.03
1,1,1	24.41±0.05
statistic	χ^2=11.21; s= 1.23

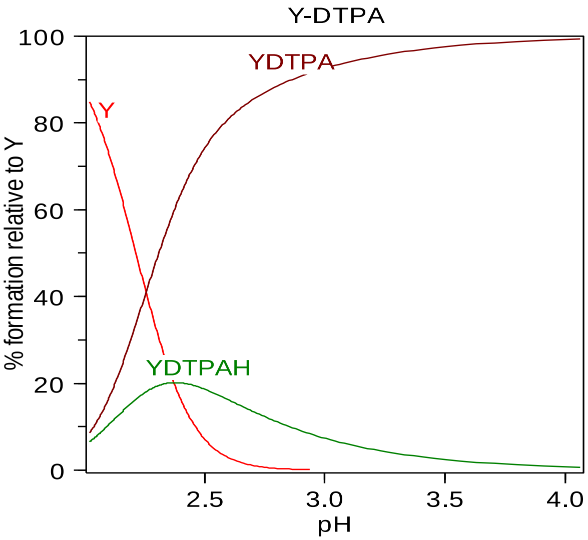


Figure 1. Distribution diagram of Y^{3+} -DTPA species at ligand-to-metal concentration ratio 1:1 and total metal concentration 1.0 mmolL^{-1}.

Table 2. Experimental and theoretical *m/z* values of ESI-MS spectra in Y^{3+}-DTPA solution at pH = 5.5; ((*m/z*)$_e$ and (*m/z*)$_t$ denote experimentally determined and calculated value, respectively)

(*m/z*)$_e$	**(*m/z*) $_t$**	Identificated ions	species ESI-MS ion
200	200	[DTPA +Y^{3+} +2Cl^--3CO_2 –H_2O]$^{2-}$	Y DTPA^{2-}
198	198	[DTPA +Y^{3+} +2Cl^--2CO_2 –2CH_4 –2NH_3]$^{2-}$	Y DTPA^{2-}
161	161	[DTPA +Y^{3+} +2Cl^-- 4CO_2 –2NH_3 –H_2O]$^{2-}$	Y DTPA^{2-}

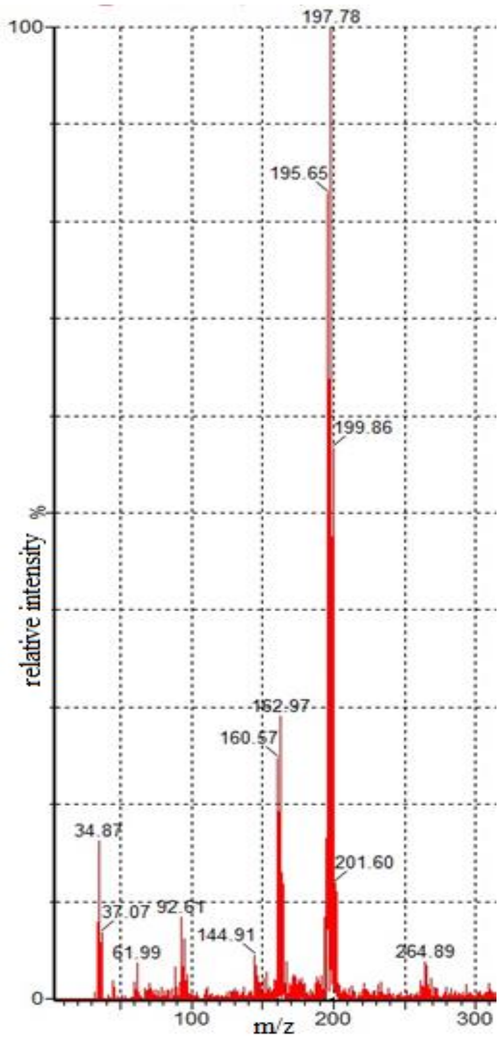


Figure 2. ESI-MS spectrum of Y-DTPA system.

Estimation of yttrium blood concentration. Estimation of yttrium blood concentration could be based on environmental sources and iatrogenic i.e., administrated yttrium chelates (e.g., Y-DTPA) for therapeutic purposes. If Y-DTPA chelate is administrated then free Y^{3+} may appear by dissociation and trans-metallation processes. Assuming that administrated dose is ~10^{-9} molL^{-1}, and taking into account that $log\beta_{\text{Y-DTPA}}$= 22.5, free Y^{3+} is ~ 10^{-12} molL^{-1}.

Trans-metallation could be represented as:

M^{2+} + Y-DTPA↔ MDTPA + Y^{3+}

with equilibrium constant

$$K_{eq} = \frac{[MDTPA][Y^{3+}]}{[M^{2+}][YDTPA]} \tag{1}$$

where M^{2+} could be Zn^{2+}, Cu^{2+} or Fe^{2+}.

The equilibrium constant K_{eq} in equation 1 can be calculated as

$$K_{eq} = \frac{\beta_M^{'}}{\beta_Y^{'}}$$

where β'_M is conditional cumulative stability constant of M-DTPA while β'_Y is conditional cumulative stability constant of Y-DTPA complex.

Cumulative conditional stability constant is defined as

$$\beta_M^{'} = \frac{[M - DTPA]}{[M][H_nDTPA]}$$

and is valid only for a particular pH (pH=7.4). [H_nDTPA] is a mixture of protonated ligand species. Equilibrium and conditional stability constant are related through the relationship:

$$\beta_M^{'} = \alpha_n \times \beta_M$$

where $$\alpha_n = \frac{\beta_n^H [H]^{N-n}}{\sum_{n=0}^{N} \beta_n^H [H]^{N-n}}$$

where N is the maximum number of bound protons and n=1,2,…,N. β^H_n is cumulative protonation constant of the ligand. By convention α^H_0 =1. For DTPA acid $\alpha_5=1.19\times10^{-3}$, for pH=7.4.

Taking into account normal blood plasma concentration of Zn^{2+} (3×10^{-6} $molL^{-1}$), Cu^{2+} (3×10^{-12} $molL^{-1}$) and Fe^{2+} (1×10^{-11} $molL^{-1}$) and that of Y(DTPA) (~10^{-9} $molL^{-1}$) total free Y^{3+} is ~5×10^{-10} $molL^{-1}$. Therefore, this is a reason why the concentration of free yttrium in our modeling was examined in wide range of concentration (10^{-9}- 5×10^{-2} $molL^{-1}$).

*Human blood plasma model and Speciation calculation***.** In developing the computer modeling of blood plasma we improved May et al. [18] model of blood plasma and constructed multi-phase model including 10 metals, 43 ligands and over 6300 complexes. Total concentrations of all components were taken from published papers and Geigy tables [19]. Almost all stability constants of binary and ternary complexes were abstracted from published databases (JESS, IUPAC, NIST [20-22]) and where necessary converted to physiological conditions (t=37° C, I= 0.15 mol L^{-1} NaCl) using the program SIT (Specific Interaction Theory). Part of the stability constants was updated on the basis of recent literature data. A complete list of complexes of components in blood plasma database and constants was described in our previous work [2].

Two general approaches to simulate complex equilibria systems are widely used namely, Gibbs free energy minimization and the equilibrium constant method. The latter is based on the solution of a set of equilibrium conditions satisfying stoichiometric mass balance equations. The system stoichiometric equations

$$T_{Ri} = \sum_{j=1}^{n} \nu_{ji}\beta_j \prod_{i=1}^{m} [R_i]^{\nu_{ji}} + \sum_{k=1}^{s} \nu_{i.k}\, A_k$$

$i = 1 \div m$ number of components

$j = 1 \div n$ number of reactions (products)

β_j - formation constant of particular product, j

ν_{ji} - stoichiometric coeficients, $\nu_{ji} = 1$ for $j = i$ and $j \leq m$ and $\nu_{ji} = 0$ for $j \neq i$ and $i \leq m$ (first m products are identical to components)

A_k - relative amount of the insoluble species, k, formed.

$[R_i]$ – free concentration of components

were solved using the Newton-Raphson iterative method as implemented in computer program Hyss2009. The solution is obtained in the form

$\delta x = -J^{-1} \times F$

where δx is a shift: $[R_{i,new}] = [R_{i,old}] + \delta x$ and $F = T_{Ri,(exp)} - T_{Ri,(calc)}$

The Cholecki factorization of coefficient matrix is used for solution of the set of linear equations.

The results showed that main yttrium complexes at normal blood plasma concentration of yttrium (estimated to be 1×10^{-9} $molL^{-1}$) are YCit and soluble $Y(CO_3)_2$. The percentage distribution of Ca(II) and Zn(II) ions in human blood plasma is shown in Table 1. It shows that most (~79%) of Ca(II) is free metal ion, ~9% is distributed in protonated carbonate, ~4% in citrate, < 3% in phosphate, < 2% in carbonate, ~1% in lactate and the rest in other complexes. Zinc is distributed amongst ternary and binary complexes with cysteinate (Cys), citrate (Cit), histidinate (His) and cysinate (Cis). Main zinc species are ZnCysCit (~38%), $ZnCys_2$ (~20%) and ZnCysHis (~11%). Influence of Y(III) ion on distribution Ca(II) and Zn(II) ions under physiological conditions is shown in Table 3.

Table 3. *Main species and distribution of Ca(II) and Zn(II)** ions in the presence of Y(III) ion (%)**

Species	Total concentration of Y(III) ion ($molL^{-1}$)							
	0	1×10^{-9}	1×10^{-7}	1×10^{-6}	1×10^{-3}	1×10^{-2}	3×10^{-2}	5×10^{-2}
Free Ca(II)	78.62	78.62	78.62	78.62	79.10	86.07	96.41	96.55
$CaHCO_3$	9.26	9.26	9.26	9.26	8.74	4.08	0	0
CaCit	3.83	3.83	3.83	3.83	3.83	3.91	0.02	0.01
$CaPO_4$	2.95	2.95	2.95	2.95	2.96	1.11	0	0
$CaCO_3$	1.54	1.54	1.54	1.54	1.45	0.68	0	0
CaLac	1.12	1.12	1.12	1.12	1.12	1.22	1.10	1.02
Free Zn	0.58	0.58	0.58	0.58	0.58	0.59	1.08	1.10
ZnCysCit	37.77	37.77	37.77	37.77	37.63	36.33	1<	1<
$ZnCys_2$	19.91	19.91	19.91	19.91	19.96	20.48	35.10	35.37
ZnCysHis	10.89	10.89	10.89	10.89	10.92	11.19	19.47	19.67
ZnHis	3.68	3.68	3.68	3.68	3.70	3.80	6.81	6.91
$ZnHCys_2$	2.81	2.81	2.81	2.81	2.82	2.89	4.96	5.00
ZnHCysCis	2.21	2.21	2.21	2.21	2.22	2.27	3.98	4.02
ZnCys	2.14	2.14	2.14	2.14	2.14	2.20	3.88	3.93
$Y_2(CO_3)_{2(s)}$	0	0	5.9	90.44	100	97.52	54.44	32.66
$YPO_{4(s)}$	0	0	0	0	0	2.48	1.27	0.76
Soluble Y species	0	100	94.1	9.56	0	0	44.29	66.58

*Cit-citrate, Lac-lactate, Cys-cysteinate, His-histidinate, Cis-cystinate, charge are omitted for simplicity.

** $[Ca(II)]=1.45\times10^{-6}$ $molL^{-1}$; $[Zn(II)]=3.0\times10^{-6}$ $molL^{-1}$.

The data in Table 3 show that at the concentration of 1×10^{-6} $molL^{-1}$, yttrium ions is almost completely bound to carbonate (90.44%) to form precipitate of $Y_2(CO_3)_{2(s)}$. When the total concentration of Y(III) increases, insoluble species reaches 100% (at 1×10^{-3} molL^{-1} of Y). Further increasing of yttrium concentration leads to increasing of free calcium percentage and increasing of soluble yttrium species. In meantime small amount of another precipitate appears ($YPO_{4(s)}$). In the case of zinc distribution, influence of yttrium is not significant until yttrium reaches the concentration of 1×10^{-2} molL^{-1}. Higher concentration of these lead to redistribution of zinc complexes. Main species ZnCysCit becomes minor species while $ZnCys_2$ and ZnCysHis become major ones.

CONCLUSION

In human blood plasma, the increase of Y(III) level will lead to the increase of free calcium ion and to the redistribution of zinc species. Concentration of calcium complexes $CaHCO_3$ and CaCit decrease (from ~9% and ~4% to ~0%, respectively) while free calcium ion concentration increases from about 79 to 97%. Ternary zinc complex (ZnCysCit) which is main complex in normal blood plasma, with increasing concentration of yttrium completely disappears, while percentage of $ZnCys_2$ and ZnCysHis complexes increase. Bearing in mind that these changes occur at [Y^{3+}] > 0.05 molL^{-1} which is normally not encountered in humans, use of yttrium-based radiopharmaceuticals is safe from the standpoint of view of essential metal ions metabolism.

ACKNOWLEDGMENTS

Financial support from the Ministry of Science and Technological Development of Serbia, under the project 172016, is gratefully acknowledged.

REFERENCES

[1] Gans, P; Sabatini, A; Vacca, A. Investigation of equilibria in solution. Determination of equilibrium constants with the HYPERQUAD suite of programs, *Talanta*, 1996, 13, 1739-1753.

[2] Jakovljevic, I; Petrovic, Dj; Joksovic, Lj; Lazarevic I; Djurdjevic, P. Computer simulation of speciation of trivalent aluminum, gadolinium

and yttrium ions in human blood plasma, *Acta Chim. Slov.*, 2013, 60, 861-869.

[3] Dudev, T; Lim, C. Principles governing Mg, Ca, and Zn binding and selectivity in proteins, *Chem. Rev*., 2003, 103, 773–787.

[4] Bünzli, JC. Luminescent lanthanide probes as diagnostic and therapeutic tools. *Met Ions Biol Syst.*, 2004, 42, 39-75.

[5] Vallee, BL; Falchuk, KH. The biochemical basis of zinc physiology, *Physiol. Rev*., 1983, 73, 79–118.

[6] Zhang, H; Wang, J; Lu, X; Yang, K; Niu, C. Effect of praseodymium(III) on zinc(II) species in human interstitial fluid, *Bio. Trace Element Research*, 2005, 107, 101-110.

[7] Ni, JZ. ed, Bioinorganic chemistry of rare Earths Elements, Science Press, 1995.

[8] Dauber, R; Heim, T. in: Metals and Their Compounds in the Environment, Merian E. ed., VCH: Weinheim, Fed. Rep Ger 1991, 1299-1308.

[9] Gulec, SA; Mesoloras, G; Dezarn, WA; McNeillie, P; Kennedy, AS. Safety and efficacy of Y-90 microsphere treatment in patients with primary and metastatic liver cancer: The tumor selectivity of the treatment as a function of tumor to liver flow ratio, *J. Transl. Med*., 2007, *5*, 15.

[10] Mosconi, C; Cappelli, A; Pettinato, C; Golfieri, R. Radioembolization with Yttrium-90 microspheres in hepatocellular carcinoma: Role and perspectives *World J. Hepatol*., 2015, 7, 738-752.

[11] Yttrium-90 and Rhenium-188 radiopharmaceuticals for radionuclide therapy, IAEA Radioisotopes and Radiopharmaceuticals Series No.5, Vienna, *International Atomic Energy Agency*, 2015, pp. 2.

[12] Liu, S; Edwards, DS. Fundamentals of Receptor-Based Diagnostic Metalloradiopharmaceuticals, *Top. Curr. Chem*., 2002, 222, 259-278.

[13] Liu, S. The role of coordination chemistry in the development of target-specificradiopharmaceuticals, *Chem. Soc. Rev*., 2004, 33, 445-461.

[14] Goldenberg, DM. The role of radiolabeled antibodies in the treatment of non-Hodgkin's lymphoma: the coming of age of radioimmunotherapy, *Crit. Rev. Oncol. Hematol*., 2001, 39, 195-201.

[15] Grillo-Lopez, AJ. Zevalin: the first radioimmunotherapy approved for the treatment of lymphoma, *Expert Rev. Anticancer Ther*., 2002, 2, 485-493.

[16] Davies, A.J; Radioimmunotherapy for B-cell lymphoma: Y^{90}ibritumomab tiuxetan and I^{131} tositumomab, *Oncogen*e, 2007, 26,

3614-3628.

[17] Kumar, K; Chang, CA; Francesconi, LC; Dischino, DD; Malley, MF; Gougoutas, JZ; Tweedle, MF. Synthesis, stability, and structure of gadolinium (III) and yttrium (III) macrocyclic poly (amino carboxylates), *Inorg. Chem.*, 1994, 33 *(16)*, 3567-3575.

[18] May, P.M; Linder, P.W; Williams, D.R; Computer simulation of metal-ion equilibria in biofluids: models for the low-molecular-weight complex distribution of calcium(II), magnesium(II), manganese(II), iron(III), copper(II), zinc(II), and lead(II) ions in human blood plasma, *J. Chem. Soc., Dalton Trans.*, 1977, 588-595.

[19] Lentner C; (Ed). Geigy Scientific Tables, Vol 3. West-Caldwell, NJ:Ciba-Geigy, 1984.

[20] Smith, RM; Martell, AE; Motekaitis, RJ. NIST Standard Reference Database 46, NIST Critically Selected Stability Constants of Metal Complexes Database, Version 8.0. National Institute of Standards and Technology, 2004.

[21] SC-Database. IUPAC stability constants database. Academic Software. UK. 2005.

[22] P. M. May, K. Murray, JESS, A Joint Expert. Speciation System, Part I, *Talanta*, 1991, 38, 1409-1417.

In: Computer Vision and Simulation
Editor: Sherri Alexander

ISBN: 978-1-63485-790-1
© 2016 Nova Science Publishers, Inc.

Chapter 5

SIMULATION OF DIFFRACTION GRATINGS IN THE FRESNEL DIFFRACTION REGIME: USING THE *AB-INITIO* ITERATIVE FRESNEL INTEGRALS METHOD

Kazi Monowar Abedin*[*] and S.M. Mujibur Rahman
Department of Physics, College of Science,
Sultan Qaboos University, Oman

ABSTRACT

Computer-based virtual experiments and simulations in all branches of physical sciences and engineering has attracted wide spread interest among the researchers from all parts of the scientific world due to its multifaceted applications and versatility. Computer simulation of diffraction phenomena, including simulation of diffraction gratings, has widespread applications, since diffraction gratings, especially amplitude diffraction gratings, are used extensively in spectrographs and spectrometers. Usually, these are used in the Fraunhofer (far-field) regime. In this Chapter, we have used the *ab-initio* Iterative Fresnel Integral Method (IFIM) for the complete simulation of the near-field Fresnel diffraction images from any amplitude diffraction grating. The simulations can be performed in any PC in a reasonable amount of time

[*] Corresponding author: Department of Physics, College of Science, Sultan Qaboos University, P.O. Box 36, Al-Khoudh, Muscat, P.C. 123 Oman Email: abedin@squ.edu.om.

and are executed in the MATLAB language. Complete explanations of the computational method, as applied to the diffraction gratings, are described, along with the simulation algorithms. Comparison of the simulated results with certain situations, which can be described by analytical equations, is made. The agreement confirms the correctness of the present simulation methods that will pave the way for future studies. We finally mention some extensions of the N-stilt problem, namely the application to tilted and rotating gratings and multi-wavelength illuminations.

Keywords: computer simulation, Fresnel diffraction, Iterative Fresnel Integrals Method, rectangular apertures, diffraction gratings

1. INTRODUCTION

The use of computers in simulating real-world situations and modeling has received immense interest and attention in recent years. Computer simulations have widely been used in industrial applications, fundamental research and in computer-aided education and visualization [1-4]. With the tremendous improvement in computer hardware and software in the last few decades, it has been possible to simulate some real-world problems even in an ordinary PC or workstations. In particular, in the field of optics, computer simulation and modeling have been used to model wave propagation in various media, diffraction, optical filtering, design of optical instruments of all sorts, teaching of optics, etc. [5-15]. Since optical hardware is not cheap, in some situations, computer simulations can be used as an effective substitute of real-life experiments, and have enabled researchers to perform virtual experiments in a PC.

Diffraction of light and other waves from an aperture or obstacle is a well-known phenomenon and is of tremendous interest in optics [16-19]. In short, diffraction is the ability of light to move around obstacles in a restricted way, and is a manifestation of the wave nature of light itself. Diffraction phenomena affect the real-life behavior of optical systems, such as microscopes, telescopes, lasers, semiconductor lithography, electro-optical modulators, etc. For example, diffraction limits the fundamental resolution achievable in microscopic, telescopic and interferometric systems. For these reasons, it is extremely important to accurately simulate diffraction phenomena in many practical situations.

In general, diffraction phenomena can be classified into two types, near-field (Fresnel) diffraction and far-field (Fraunhofer) diffraction. Fraunhofer diffraction is the simpler to handle mathematically, because *both* the wavefront incident on the aperture and on the observation screen are planar. In this case, the diffraction pattern can be described in terms of the Fourier transform of the aperture function. In the other case, i.e., in the case of near-field (Fresnel) diffraction, where *either* the aperture-source distance or aperture-screen distance is finite, the wavefronts are not planar, and the solution becomes much more complex [16, 17].

In the case of Fresnel diffraction, the analytical form of the diffraction pattern cannot be found even in the simplest cases. Therefore, in these situations, numerical methods are the only viable option. Solutions can be derived in terms of certain types of diffraction integrals, known as Fresnel-Kirchoff, or Rayleigh-Sommerfeld diffraction integrals [17]. The large amount of numerical data in the calculation of the integrals in any real-life situation means that the use of computers and computer simulation techniques can be put to use effectively. With the use of powerful modern computers and advanced software, two-dimensional fast-Fourier transform (FFT) methods can be used to calculate diffraction integrals. For example, the diffraction and propagation of waves from any arbitrary aperture have been simulated in computers by Rudolf et al. [20] using Helmholtz–Kirchoff diffraction integrals. While these Fourier methods are powerful and versatile, they do not provide much insight into the computation process itself. The computation is treated virtually as a black box, and no symmetry in the geometry of the aperture is utilized in simplifying the calculations.

In previous publications [21-25], we introduced a new *ab-initio* technique to simulate the complete diffraction field of apertures having rectangular shapes or rectangular symmetry. This technique is called the Iterative Fresnel Integral Method (IFIM). It uses virtual displacement of the aperture, and repeated calculation of certain non-analytic integrals, knows as the Fresnel integrals, to construct the complete diffraction image (pattern) from a rectangular aperture, or combinations thereof. The technique is quite powerful and general, and despite being limited to rectangular-shaped apertures, can be applied to a variety of interesting problems of practical importance. Using the technique, implemented in any high-level computer language such as MATLAB, the diffraction field from any rectangular-shaped apertures in any arbitrary situation can be calculated in a few minutes or less [21].

An amplitude diffraction grating [16, 19], is an array of small rectangular apertures separated by opaque regions. In a practical grating, both the width of the apertures and the opaque region are of the order of wavelength of light. This type of gratings is widely used in a variety of spectrometers and spectrographs, to separate the light into constituent wavelengths. In all these situations, the wavefront incident on the grating is rendered plane by a collimating mirror or lens, and the wavefront emitted from the grating are also rendered plane by another collimating mirror or lens. In that case, the analytical treatment of the diffraction grating becomes much simpler, because Fraunhofer diffraction regime can be used to calculate the diffraction field. The situation, where this Fraunhofer condition is not valid are not considered or used, partly because of difficulty of treating Fresnel diffraction from a grating.

In this chapter, we describe how the IFIM technique can be used and extended to the non-trivial case of amplitude diffraction gratings, and discuss some potential applications in the case of tilted gratings. Complete and self-contained discussions of the technique as well as the implementation of the algorithms in MATLAB is given and explained.

This Chapter is organized as follows: First, the principle of the IFIM method is introduced as it is used for the rectangular aperture. The algorithm and the simulation technique are discussed, as well as the simulation program. Then this method is applied to the case of a 3-slit aperture, and is then immediately generalized to *N*-slits, which is effectively an amplitude grating. The algorithm used to generalize to *N*-slits, where *N* is an arbitrary odd or even number, is clarified. The details of some typical simulation results for the single slit as well as the amplitude gratings are presented and discussed.. The program is also applied to a real-life grating having grating period of the order of a wavelength of light and having a large number of apertures or slits. Then discussions are made regarding the extension of the technique to tilted gratings and multi-wavelength illuminations, and conclusions are drawn at the end of this Chapter.

2. Introduction to the IFIM Method

2.1. Single Rectangular Aperture: Theory

As a very simple diffraction problem, let us consider the case of a rectangular aperture (Figure1). The solution of this problem by the Iterative

Fresnel Integral Method will help the reader to understand more complicated problems, where there are many slits. This rectangular-aperture diffraction problem has been discussed and analyzed in many textbooks, with the aid of Fresnel integrals and Cornu spirals [16-19]. Light of wavelength λ emitted from a point source *S* is diffracted by a rectangular aperture of dimension *2aX2b* located at a distance p_0 from it. The diffracted light is observed on the screen placed a distance q_0 away. As shown in Figure 1, the coordinate systems on the aperture and on the image planes are chosen to be centered on the optical axis passing through the center of the aperture and normal to it, and are denoted by *(y, z)* and *(Y, Z)* axes, respectively. The Huygens–Fresnel principle is then employed to compute the total electric field at any given point of the image plane *(Y, Z)* by summing up all the contributions (taking into account both amplitude and phase), of all the non-planar elementary wavelets (Huygens wavelets) emitted by different area elements inside the clear rectangular aperture.

The contribution to the complex electric field at the point *P* (located at the origin of the YZ image plane) due to Huygens waves emitted by the small element *dS* located inside the clear aperture is [16, 17],

$$dE = \frac{\varepsilon_0 K(\phi)}{pq\lambda} \exp\left[j\{k(p+q) - \omega_0 t \right] dS \tag{1}$$

where ε_0 is the electric field of the source *S*, $k\ (=2\pi/\lambda_0)$ is the wavenumber of the light waves and K(ϕ) is the obliquity factor, which takes into account the decrease of intensity for secondary Huygens wavelets emitted from *dS* in an oblique direction.

The total electric field at *P* due to the whole aperture can be calculated by summing up (or integrating) the contributions *dE* over the entire clear aperture, i.e.,

$$\mathrm{E_P} = \int_{\mathrm{aprerture}} \frac{\varepsilon_0 \mathrm{K}(\theta)}{\mathrm{p q}\lambda} \exp\left[\mathrm{j\{k(p+q)} - \omega_0 \mathrm{t} \right] \mathrm{dS}\,. \tag{2}$$

Upon integrating the contributions of the secondary Huygens wavelets over the entire aperture, it can be shown [16, 17] that the total complex electric field at *P* is given by the following expression

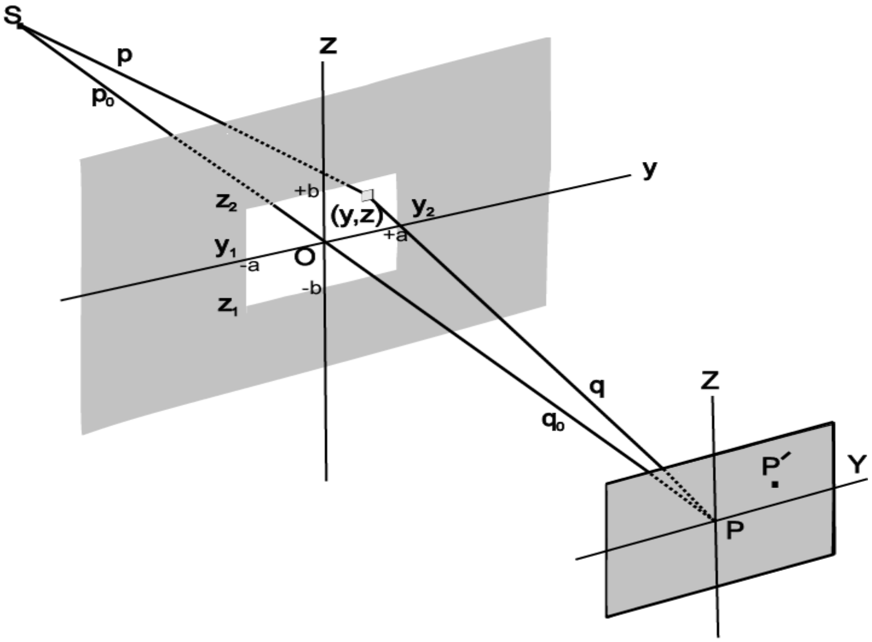


Figure 1. The basic configuration of Fresnel diffraction from a single rectangular aperture (Reprinted from Abedin. et al. Opt. Laser Technol. Vol. 39. pp. 237-246, Copyright, (2007) with permission from Elsevier).

$$E_P = \frac{E_u}{2}\ C(u) + iS(u)\ {}_{u_1}^{u_2}\ C(v) + iS(v)\ {}_{v_1}^{v_2}\ , \tag{3}$$

where E_u is the unobstructed electric field at P (i.e., the electric field that would have existed if the aperture were removed). *C(u)* and *S(u)* are the Fresnel cosine and sine integrals, being defined by,

$$C(w) = \int_0^w \cos(\pi w'^2/2)\, dw' \quad and \quad S(w) = \int_0^w \sin(\pi w'^2/2)\, dw'. \tag{4}$$

Here w represents *either* of the two dimensionless variables u or v,

$$u = y\sqrt{2(p_0+q_0)/\lambda\, p_0 q_0}\,, \qquad v = z\sqrt{2(p_0+q_0)/\lambda\, p_0 q_0}\ . \tag{5}$$

The variables *u* and *v* are proportional to Cartesian coordinates *x* and *y*. The intensity at *P* is given by the square of E_p, in Eq. (3) i.e., by,

$$I_P = \frac{I_0}{4}\{ C(u_2) - C(u_1)^{\ 2} + S(u_2) - S(u_1)^{\ 2}\} \times \{ C(v_2) - C(v_1)^{\ 2} + S(v_2) - S(v_1)^{\ 2}\}\,. \tag{6}$$

In the above expression, I_0 is the unobstructed intensity corresponding to E_u ($I_0 = E_u^2$), i.e., the intensity that would be observed if the aperture were removed. To calculate the intensity at any off-axis point *P'* on the YZ image plane, one can fix the SOP line and instead of moving the point *P*, one can move the aperture itself by small amounts in the *yz* plane, so that the *relative positions* of the aperture in the new position and *P* remains the same. For example, to find the intensity a point *P'* 1mm *above P*, one can keep the screen undisturbed and move the aperture 1mm *downwards*, and find the intensity at *P* in this situation instead. The point *P* will now see a new set of values for z_1 and z_2, and therefore, for v_1 and v_2 in Eq. (6). In principle, the intensity at any point *P'* on the image plane can be found in this way by making appropriate translations of the aperture in the *y* and *z* directions, and in Eq. (6), substituting the correspondingly new values of u_1, u_2, (which indicate the positions of the edges of the aperture in the *y*-direction as seen from *P*) and v_1 and v_2 (which indicate the positions of the edges of the aperture in the *z*-direction as seen from *P*). This implies that new values of Fresnel cosine and sine integrals need to be computed. Using this method, the entire intensity distribution in the image plane can be mapped out provided one is willing to calculate a large number of Fresnel integrals. From Eq. (6), it is clear that the calculation of the intensity at any point *P'* in the image plane requires the evaluation of 8 Fresnel integrals. For the rest of this Chapter, for simplicity, we consider mostly plane-wave illumination of the apertures. This effectively moves the source *S* to infinity, which can be realized by placing *S* in the focal point of a convex lens and allowing the collimated light from the lens to fall on the aperture. If this is not actually done in reality, then Eq. (5) can be used to compute the appropriate values of the dimensionless variables *u* and *v*, and the calculation can proceed from there. For the simple case of plane-wave illumination, p_0 is effectively infinity and, therefore, Eq. (5) can be simplified to,

$$\mathrm{u} = \mathrm{y}\sqrt{2/\lambda \mathrm{q}_0}\,, \qquad \nu = \mathrm{z}\sqrt{2/\lambda \mathrm{q}_0}\ . \tag{7}$$

2.2. The Single Aperture: Simulation Strategy and Associated Algorithm

We next explain the algorithm and the simulation program to construct a two-dimensional matrix of intensity values whose elements represent the intensities at an arbitrary point *P'* in the image plane. A flow chart of the algorithm is presented in Figure 2, and the complete MATLAB program *rectan* is given in the Appendix A. In the beginning, we need to specify certain input values to the program: the aperture dimensions *2aX2b* and the size of the observation area *2WX2W* in the image plane (screen), the aperture-screen distance q_0 (all in millimeters) and the illumination wavelength *l* (in nanometers). These are essential inputs to the simulation program, and are done, by means of a MATLAB GUI (Graphical User Interface), in the first two lines of program. To specify the resolution of the calculations, we need to input a step size *s* (in mm), which indicates the increments *s* in the *y* and *z* directions of the positions of *P'*. This determines the spatial resolution or pixel size, because the number of the pixels in the image is simply given by (*2W/sX2 W/s*). For most of the simulations in this section, we selected a step size so that the number of pixels in the image is of the order of 800X800 pixels. A screenshot of the MATLAB GUI, generated by the *inputdlg* command in the program, is shown in Figure 3.

For calculation of the intensity distribution using Eq.(6), it is convenient to have a function to quickly and efficiently calculate Fresnel cosine and sine integrals for given values of the argument. In MATLAB, Fresnel cosine and sine integrals can be invoked by typing *mfun('FresnelC', w)* and *mfun('FresnelS',w)*, respectively. If *w* is a single-valued variable, then these *mfun* functions return single values of Fresnel cosine or sine integrals. If, on the other hand, we type, for example, *R=mfun('FresnelC', l: s: u)*, then this generates a one dimensional array ***R*** of Fresnel cosine integrals with arguments starting from *l* and ending in *u*, with a step size of *s*. These *mfun('FresnelC', l: s: u)* and *mfun ('FresnelS', l: s: u)* functions will be extensively used in the simulations that follow.

An quick look at Eq. (6) shows that to calculate intensity values for different points *P'* in the image area which will see different values of y_1, y_2, z_1, z_2 (and hence different values of of u_1, u_2, v_1, v_2), we need to calculate arrays of Fresnel integrals with ranges of argument values corresponding to the ranges of values that u_1, u_2, v_1, v_2 will assume for different points *P'* in the image plane. Since the square aperture is symmetrical about the *y* and *z* axes,

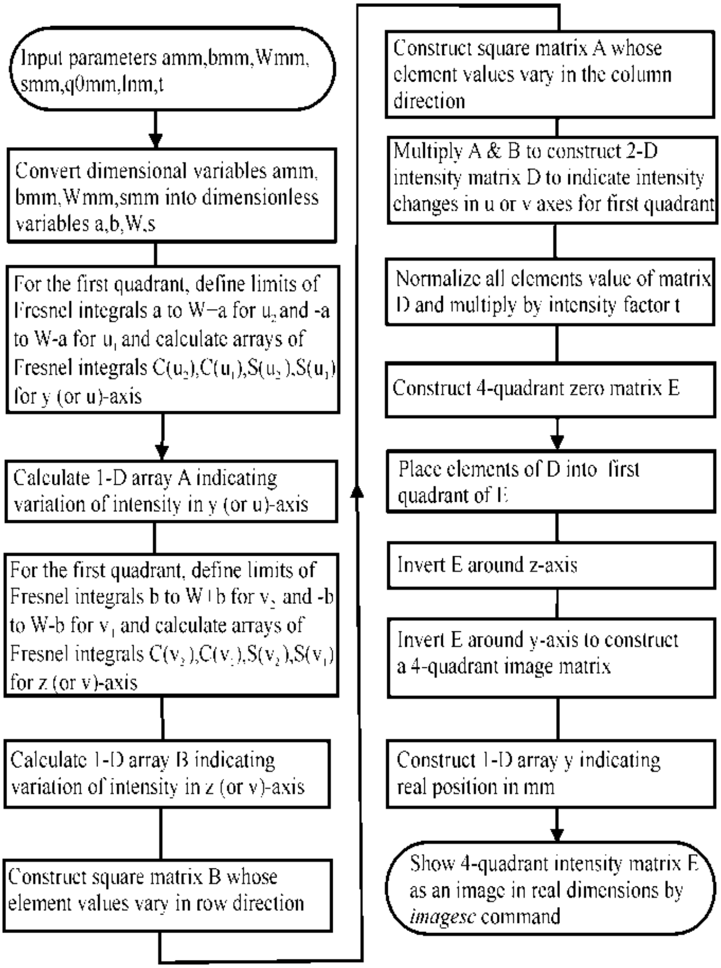
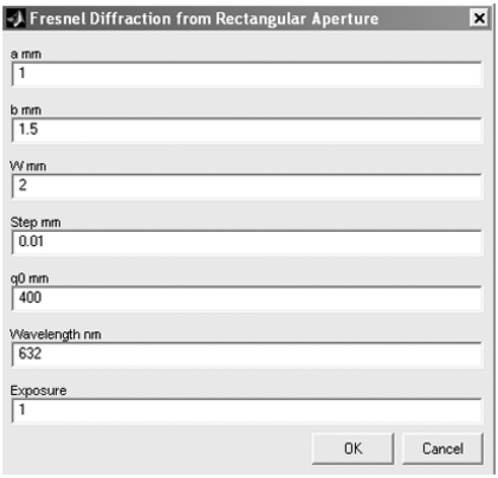

Input parameters amm,bmm,Wmm, smm,q0mm,lnm,t
Convert dimensional variables amm, bmm,Wmm,smm into dimensionless variables a,b,W,s
For the first quadrant, define limits of Fresnel integrals a to W−a for u_2 and -a to W-a for u_1 and calculate arrays of Fresnel integrals $C(u_2)$,$C(u_1)$,$S(u_2)$,$S(u_1)$ for y (or u)-axis
Calculate 1-D array A indicating variation of intensity in y (or u)-axis
For the first quadrant, define limits of Fresnel integrals b to W+b for v_2 and -b to W-b for v_1 and calculate arrays of Fresnel integrals $C(v_2)$,$C(v_1)$,$S(v_2)$,$S(v_1)$ for z (or v)-axis
Calculate 1-D array B indicating variation of intensity in z (or v)-axis
Construct square matrix B whose element values vary in row direction
Construct square matrix A whose element values vary in the column direction
Multiply A & B to construct 2-D intensity matrix D to indicate intensity changes in u or v axes for first quadrant
Normalize all elements value of matrix D and multiply by intensity factor t
Construct 4-quadrant zero matrix E
Place elements of D into first quadrant of E
Invert E around z-axis
Invert E around y-axis to construct a 4-quadrant image matrix
Construct 1-D array y indicating real position in mm
Show 4-quadrant intensity matrix E as an image in real dimensions by *imagesc* command

Figure 2. Flow chart of the simulation algorithm (Reprinted from Abedin. et al. Opt. Laser Technol.. Vol. 39. pp. 237-246, single aperture. Copyright, (2007) with permission from Elsevier).



Fresnel Diffraction from Rectangular Aperture

a mm
1

b mm
1.5

W mm
2

Step mm
0.01

q0 mm
400

Wavelength nm
632

Exposure
1

Figure 3. Screenshot of MATLAB GUI for the single aperture.

the Fresnel diffraction pattern generated by it will also be symmetrical about the *Y* and *Z* axes in the image plane. We can use this inversion symmetry around *Y* and *Z* axes to reduce the amount of calculation in all the simulations. It is necessary to generate diffraction image in only one quadrant in the *YZ* plane (e.g., the first) and we can generate the image in the second quadrant by inverting it around the *Z* axis. Another inversion of these two image parts around the *Y* axis will produce the images in the third and fourth quadrants. By this method, the amount of calculation can be reduced by approximately 4 times. In MATLAB, these inversions can be performed rather easily with a minimum of computation time, so this must be a timesaver.

To generate the image in one quadrant (e.g., the first), the limits that should be used for the calculation of Fresnel cosine and sine arrays *mfun('FresnelC', l:s:u)* and *mfun('FresnelS', l:s:u)* need to be determined. To find these limits, let us imagine that we position ourselves at the right/left or upper/lower edges of the quadrant, and for each edge, determine the corresponding values of u_1, u_2 or v_1, v_2 (i.e., y_1, y_2 or z_1, z_2), for each edge. For example, imagine that we are located on the right edge of the first quadrant, Y = W (see Figure 4). As explained before, this is equivalent to staying at *P* and moving the aperture by *W* to the left. From *P*, the right and left edges of the displaced aperture now appear to be located at $y_2 = (W + a)$ and at $y_1 = (W - a)$, and hence the corresponding values of dimensionless variables u_2 and u_1 must be used for the calculation of $C(u_2)$, $C(u_1)$, $S(u_2)$ and $S(u_1)$ in Eq.(6). On the other hand, when we are located on the left side of the first quadrant, i.e., at $Y = 0$, then the positions of the (un-displaced) aperture edges are simply $y_2 = a$ and at $y_1 = -a$, respectively. The corresponding u_2 and u_1 values should be used for $C(u_2)$, $C(u_1)$, $S(u_2)$ and $S(u_1)$ in Eq. (6) for the calculation of intensity at this observation position. For any observation position between these two extremes, y_2 should range between $(W + a)$ and a, and y_1 should range between W-a and $-a$ (with corresponding ranges for u_2 and u_1). Therefore, the values of the four arrays $C(u_2)$, $C(u_1)$, $S(u_2)$ and $S(u_1)$ should be evaluated for arguments u_1 and u_2 ranging between these extremes (with a chosen step size of s). The same procedure can be adopted for the z (or v) direction, and arrays $C(v_2)$, $C(v_1)$, $S(v_2)$, $S(v_1)$ can be computed for appropriate ranges of v_1 and v_2.

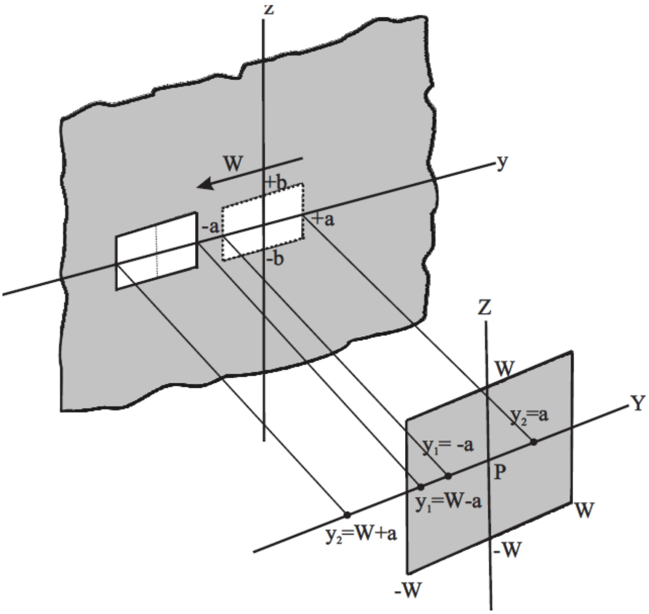


Figure 4. Apparent limits of the displaced aperture as seen from the observation point (Reprinted from Abedin. et al. Opt. Laser Technol..Vol. 39. pp. 237-246, Copyright, (2007) with permission from Elsevier).

To summarize, the limits on the four variables are therefore:

$$\text{for } y_2 = (W + a) \text{ to } a\text{: } y_1 = (W - a) \text{ to } -a \text{ and,}$$

$$\text{for } z_2 = (W + b) \text{ to } b\text{: } z_1 = (W - b) \text{ to } -b,$$

with corresponding ranges on u_2, u_1, v_2 *and* v_1, respectively. The calculations of the Fresnel cosine and sine integrals corresponding to the above ranges are performed in lines 7-10 (for the *u* variable and in lines 12-15 (for the *v* variable) respectively. The first and the second factor in the intensity distributions of the image (see Eq. 6) are computed in lines 11 and 16, respectively, and are contained in matrix *A* and matrix *B*, respectively.

The rest of the program can be understood with the aid of flow chart in Figure 2. The elements of the intensity matrix is normalized, and multiplied by the intensity factor *t* (line 20) to enhance the intensity of images that will be observed at large values of q_0 or for very small apertures, so that they appear with sufficient brightness. Usually the intensity factor is kept at 1 for nearby images, and for distant images, it can be increased to a value greater than unity (2-10). (In a real diffraction experiment, this is equivalent to using long exposures in a camera to capture faint images). The generated image for the first quadrant is contained in matrix *D*. This image is folded twice along the *y* and *z* axes (lines 23-31) to obtain the complete image (matrix *E*) for the four quadrants in the image plane. At the end of the program, the matrix *E* is shown visually as a grayscale image in real dimensions *y* (in mm) by using the MATLAB *imagesc* command.

2.3. Extension to *N* Apertures: Theory

As before in the case of the single aperture, we assume that the *N*-aperture system is centered on the *yz* coordinate system, i.e., the origin of the coordinate system *O* is located at the exact center of the *N*-aperture (Figure 5, shown for $N = 5$). Let *a* be the individual aperture width, *b* be the inter-aperture separation [the center-to-center aperture separation being ($a + b$)], and let *c* be the aperture height in the *z*-direction. The two edges of the central aperture of the *N*-aperture system (called aperture **0** in the figure) are then located at $y_0 = -a/2$ and $y_{0'} = a/2$ respectively, and the edges of the aperture for the next aperture to the right (called aperture **+1** in the figure) are located at $y_1 = a/2+b$ and $y_{1'} = 3a/2+$b respectively. The next aperture to the right (called aperture **+2**) are located at $y_2 = 3a/2+2b$ and $y_{2'}=5a/2+2$b. Finally the edges of the aperture **+*n*** will be at $y_n = (2n-1)a/2+nb$ and $y_{n'}= (2n+1)a/2+nb$. Similarly, the edges of the first left aperture (aperture **-1**) will be at $y_{-1'}= -a/2-b$ and $y_{-1} = -3a/$2-b, and, the edges of the **–*n*** will be at $y_{-n'} = -(2n-1)a/2-nb$ and $y_{-n} = -(2n+1)a/2-nb$.

If illuminated by the light source *S*, the total complex electric field *E* at the center *P* of the YZ image plane consists of the contributions from each of the (*2n +1*) apertures. The electric field contribution from the aperture **1** is given [analogous to equation (3) for a single aperture] by

$$E_{P1} = \frac{E_u}{2}\ C(u) + j\,S(u)\ {}_{u_1}^{u_{1'}}\ C(v) + j\,S(v)\ {}_{v}^{v'}\ . \qquad (8)$$

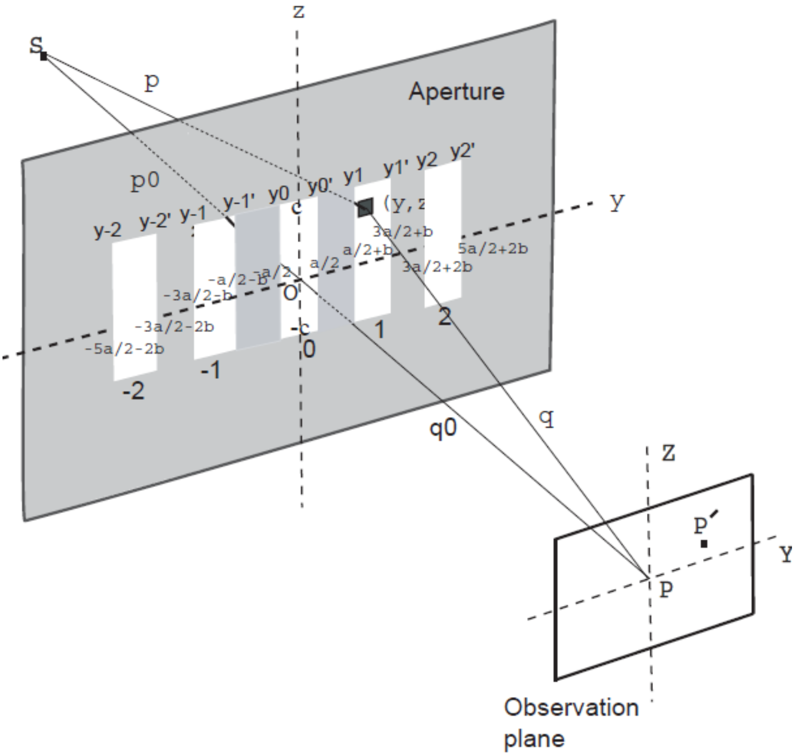


Figure 5. Geometric configuration for Fresnel diffraction for N = 5. a is the individual aperture width and (a + b) is the center-to-center aperture separation. (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

The limits u_1 and $u_{1'}$ are the values of the dimensionless variable *u* corresponding to two edges of aperture **1**, i.e., for $y_1 = a/2+b$ and $y_{1'} = 3a/2+b$, respectively. Similarly, the limits *v* and *v'* are the values of the dimensionless variable *v* corresponding to the lower and the upper edges of this aperture i.e., for *z* = -*c* and *z'* = +*c* respectively.

The electric field contributed by the aperture ***n*** to the right is given by

$$E_{Pn}=\frac{E_u}{2}\left[C(u)+jS(u)\right]_{u_n}^{u_{n'}}\left[C(v)+jS(v)\right]_v^{v'}=\frac{E_u}{2}\left[C(u_{n'})-C(u_n)+j\{S(u_{n'})-S(u_n)\}\right]\left[C(v)+jS(v)\right]_v^{v'} \tag{9}$$

where u_n and $u_{n'}$ are the values of u corresponding to two edges of aperture ***n***, i.e., for $y_n = (2n - 1)a/2 + nb$ and $y_{n'} = (2n + 1)a/2 + nb$, respectively. The values v and v' are the same as in equation (8).

The electric field contributed by the central aperture **0** is given by

$$E_{P0}=\frac{E_u}{2}\;C(u)+jS(u)\;{}_{u_0}^{u_{0'}}\;C(v)+jS(v)\;{}_v^{v'} \tag{10}$$

where u_0 and $u_{0'}$ are the values of u corresponding to two edges of the central aperture, i.e., for $y_0 = -a/2$ and $y_{0'} = a/2$, respectively. The values v and v' are the same as in equation (8) or equation (9). The electric field contribution by the **-1** aperture to the left is similarly obtained from

$$E_{P-1}=\frac{E_u}{2}\;C(u)+jS(u)\;{}_{u_{-1}}^{u_{-1'}}\;C(v)+jS(v)\;{}_v^{v'} \tag{11}$$

where u_{-1} and $u_{-1'}$ are the values of u corresponding to two edges of the **-1** aperture, i.e., for $y_{-1'} = -a/2-b$ and $y_{-1} = -3a/2$-b. The electric field contribution by the last aperture ***-n*** is given likewise by

$$E_{P-n}=\frac{E_u}{2}\;C(u)+jS(u)\;{}_{u_{-n}}^{u_{-n'}}\;C(v)+jS(v)\;{}_v^{v'} \tag{12}$$

where u_n and $u_{n'}$ are the values of u corresponding to two edges of aperture n, i.e., for $y_{-n'} = -(2n-1)a/2 - nb$ and $y_{-n} = -(2n+1)a/2 - nb$, respectively. The values v and v' are as before.

The total complex electric field at P contributed by the ($N = 2n+1$) apertures is given by the simple arithmetic sum of all complex amplitudes E_{Pn}, ... E_{P1}, E_{P0}, E_{P-1}, ...E_{P-n},

$$\begin{aligned}E_N &= E_{Pn}..+E_{P1}+E_{P0}+E_{P1}+....+E_{P-n}\\ &=\frac{E_u}{2}\{\left[C(u)+jS(u)\right]_{u_n}^{u_{n'}}+...+\left[C(u)+jS(u)\right]_{u_1}^{u_{1'}}+\left[C(u)+jS(u)\right]_{u_0}^{u_{0'}}\\ &\qquad+\left[C(u)+jS(u)\right]_{u_{-1}}^{u_{-1'}}+...+\left[C(u)+jS(u)\right]_{u_{-n}}^{u_{-n'}}\}\times\left[C(v)+jS(v)\right]_v^{v'}.\end{aligned} \tag{13}$$

Separating the cosine and sine integrals, and using a summation notation for the Fresnel sines and cosines, we can write the net complex electric field in abbreviated from as,

$$E_N = \tfrac{E_u}{2}\{\sum\nolimits_{i=-n}^{i=n}\left[C(u_{i'}) - C(u_i)\right] + j\sum\nolimits_{i=-n}^{i=n}[S(u_{i'}) - S(u_i)]\}\times \\ \{\left[C(v') - C(v)\right] + j\left[S(v') - S(v)\right]\} \tag{14}$$

The summation on i from $-n$ to $+n$ for both the Fresnel cosine and sine integrals (for the u variable only), in effect, carries out the summation of the complex electric field contributions from aperture ***–n*** to aperture ***n***, a total of $N = (2n + 1)$ apertures in the system. No such summation is required for the v variable, since only two edges (upper and lower) of the apertures are involved in this z-direction for all the $(2n + 1)$ apertures.

The net intensity at P is proportional to the square of the net electric field, i.e.,

$$I_P = \tfrac{I_0}{4}(E_N E_N{}^*), \tag{15}$$

where I_0 denotes the intensity of the unobstructed wave, i.e., $I_0 = E_u^2$.

If instead of $N = 2n+1$, we assume $N = 2n$ (N even), the analysis can be carried out in the same way as shown above. There will be still $2N$ aperture edges, and the after summing up the electric fields from all the $2N$ apertures in the system, equations similar to equation (14) and equation (15) will be obtained in the end. However, these calculations are not shown here.

From equation (14), it is clear that the calculation of electric field or intensity at a point P requires, in general, the evaluation of $2N$ Fresnel cosine and $2N$ Fresnel sine integrals, corresponding to the $2N$ edges of the N aperture system for the $u(y)$ variable. In addition, two pairs of Fresnel cosine and sine integrals are required for the $v(z)$ variable. The cosine integrals form the real parts of the electric field, and the sine integrals form the imaginary parts. After calculation of the complex electric field, the intensity at P is calculated in equation (15) by simply multiplying the field by its complex conjugate. These equations are the basis of calculation of the complete intensity distribution of the Fresnel diffraction pattern from a N-aperture system, as will be explained in section 2.4. Equation (14) was used to calculate first the complex electric field, and then the intensity was calculated by taking the square of it. These equations have two factors, the first factor involves only the u (or y) coordinate and expresses the dependence of electric field or intensity in the *y* direction. The second factor involves only the *v* (or *z*) coordinate and expresses the dependence of electric field or intensity in the *z* direction.

2.4. Extension to *N*-Apertures: Simulation Strategy and Algorithm

Equations (14) and (15) describe the electric field and intensity at *P*, respectively. In order to describe these for an off axis point *P*', a similar technique described in the case of a single aperture in section 2.2 was used. The observation screen and the *SOP* line were fixed. Then instead of moving *P*, the entire aperture in the *yz* plane was moved in the opposite direction, so that the relative position of the aperture is this *new* position and point *P* remains unchanged. We then calculate the electric field at *P* instead of at *P*'. The point *P'* will see a new set of values for *y*'s and *z*'s (and therefore for *u*'s and *v*'s). For example, to find the intensity at point *P'* 1mm to the right of *P*, the screen was kept undisturbed and the aperture system was moved 1mm leftwards, and the intensity at *P* in this configuration is calculated instead of at *P*'. Consequently, the point *P* will now see a new set of values for the *y's* and therefore, for the *u's* in equation (13) and equation (14). As in the case of the single aperture, the electric field and the intensity at any point *P*' on the image plane can be found in this way by making appropriate (virtual) movements of the aperture in the y and *z* directions, and in equation (13) or equation (14), using correspondingly a new set of values for the *u* and *v*'s.

The flow chart of the algorithm is given in Figure 6. For calculation of the electric field distribution using equation (14) at all the points (pixels) on the image plane, a large number of the Fresnel cosine and sine integrals will need to be evaluated quickly, and this is done (as in the single aperture case) by the special functions *mfun (‘FresnelC', i:s:f)* and *mfun (‘FresnelS', i:s:f).* As in the case of a single aperture, inversion symmetry around the *Y* and *Z* axes can be used to reduce the calculation by four times.

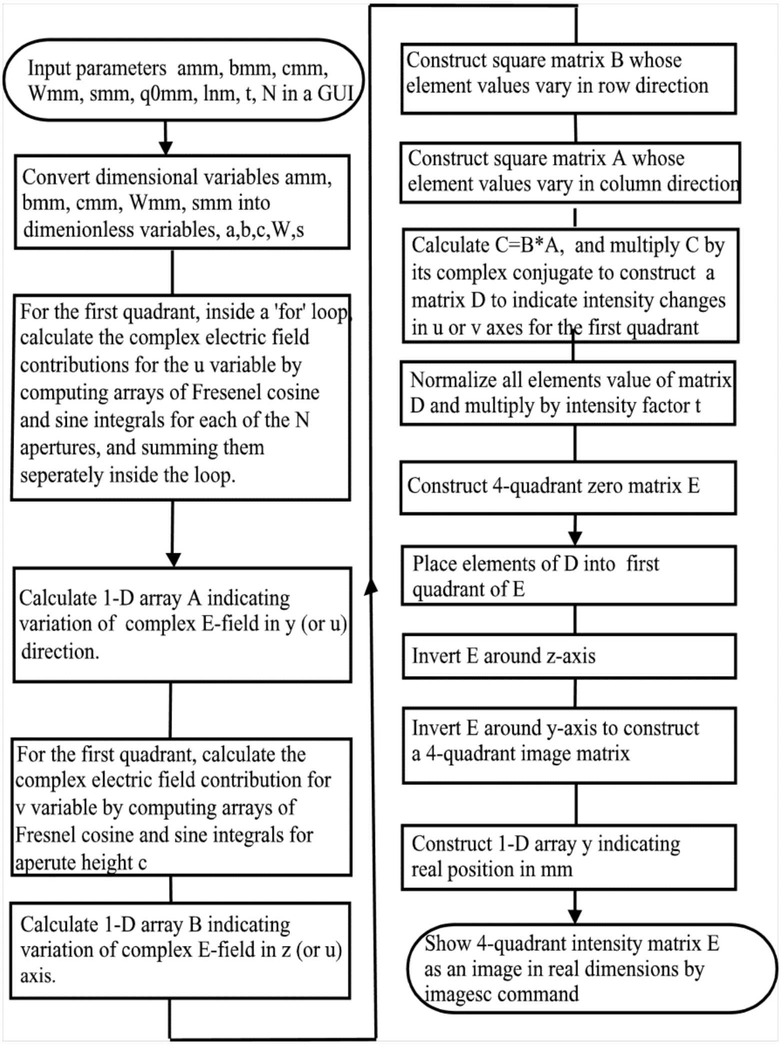

Input parameters amm, bmm, cmm, Wmm, smm, q0mm, lnm, t, N in a GUI
Convert dimensional variables amm, bmm, cmm, Wmm, smm into dimenionless variables, a,b,c,W,s
For the first quadrant, inside a 'for' loop, calculate the complex electric field contributions for the u variable by computing arrays of Fresenel cosine and sine integrals for each of the N apertures, and summing them seperately inside the loop.
Calculate 1-D array A indicating variation of complex E-field in y (or u) direction.
For the first quadrant, calculate the complex electric field contribution for v variable by computing arrays of Fresnel cosine and sine integrals for aperute height c
Calculate 1-D array B indicating variation of complex E-field in z (or u) axis.
Construct square matrix B whose element values vary in row direction
Construct square matrix A whose element values vary in column direction
Calculate C=B*A, and multiply C by its complex conjugate to construct a matrix D to indicate intensity changes in u or v axes for the first quadrant
Normalize all elements value of matrix D and multiply by intensity factor t
Construct 4-quadrant zero matrix E
Place elements of D into first quadrant of E
Invert E around z-axis
Invert E around y-axis to construct a 4-quadrant image matrix
Construct 1-D array y indicating real position in mm
Show 4-quadrant intensity matrix E as an image in real dimensions by imagesc command

Figure 6. Flow chart of the algorithm for the N-aperture problem. (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

As shown in Figure 7 for *N* = 3 (three apertures for simplicity), the whole aperture was displaced (virtually) by an amount *W*, and the extreme limits of the *2N* edges of the displaced aperture was determined and compared to the corresponding limits for the un-displaced aperture to find the required range of *u* and *v* values. In a more general *N*-aperture system, when the aperture is moved by W to the left, the ranges for the $2N$ edges of the $N = (2n + 1)$ apertures will be determined as:

for y_n : (2n -1)a/2+ nb to W+(2n - 1)a/2 + nb for $y_{n'}$: (2n + 1)a/2+ nb
to i+(2n + 1)a/2+ nb

……………………………………………………………………………

for y_1: (a/2 + b) to (W+a/2 + b)   for $y_{1'}$ (3a/2+b) to (W+3a/2+b),
for $y_{0'}$: (a/2) to (W+a/2),   for y_0: (-a/2) to (W-a/2),
for $y_{-1'}$: (-a/2-b) to (W-a/2-b)   for y_{-1} (-3a/2-b) to (W-3a/2-b),

……………………………………………………………………………

for $y_{-n'}$: -(2n-1)a/2-nb to W-(2n-1)a/2-nb for y_{-n} : -(2n+1)a/2-nb
to W-(2n+1)a/2-nb.

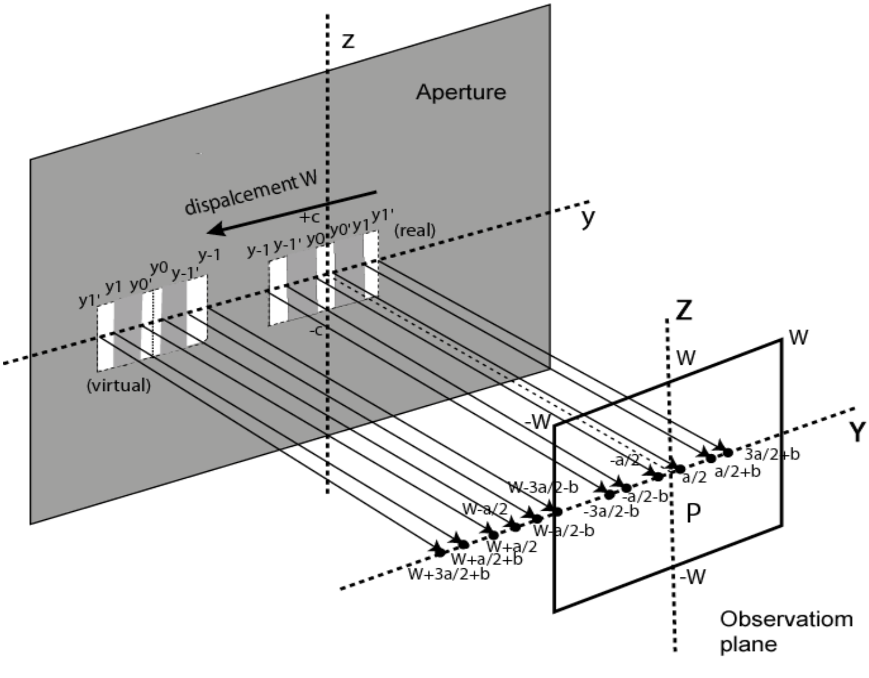


Figure 7. Apparent limits of the virtually displaced aperture seen from the observation plane for N=3. (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

Arrays of both the Fresnel cosine and sine integrals need to be calculated corresponding to these input ranges by the *mfun* statements, with a step size *s*. By using these Fresnel arrays, the electric field or intensity dependence in the *y(u)* direction can be calculated (the first factor in equation 14). Following a similar procedure for the entire aperture, i.e., by moving the aperture by *W* downwards, the z_1 and z_2 ranges are determined: *c* to *(W+c)* for z_2 and –*c* to *(W-c)* for z_1. But only four arrays of Fresnel integrals are to be evaluated for these two ranges, giving numerical values for the calculation of the second factor in equation (14).

The complete MATLAB program for the simulation is given in the Appendix B. The initial parameters to the program are entered through a GUI (Graphical User Interface) generated in lines 1-2 of the program. The GUI accepts the values of aperture width (*a*), aperture separation (*b*), aperture height (*c*), width of the image area (*W*), step size or resolution of the calculations *s*, aperture-image plane distance (q_0) (all in mm), wavelength of light (λ in nm), exposure factor (*t*) and the number of slits (*N*) as user-supplied variables. The dimensionless quantities corresponding to aperture dimensions *a, b* and *c*, step size *s* and image area size *W* are calculated in lines (4-5) of the program. The step size *s* determines the resolution of the simulated images, as in the single aperture. A reasonably small value of the step size should be selected in the simulations. If too small a value is selected, the simulation time will be too long and memory overflow may occur. On the other hand, if too large a value is chosen, a low-resolution, blurred image will be produced. A value of $s = 0.01$ mm is a good starting point in many cases.

The summation of the complex electric field for the *u* variable over the *N* apertures, indicated in equation (14), is carried out inside the repetitive ‘for’ loop (between lines 7-14) wherein *N* iterations are performed. Inside this loop, the required Fresnel cosine integral arrays for the *u* (or *y*) dimension are evaluated in lines (8-9) using *mfun* statements and the Fresnel sine arrays are likewise determined in lines (11-12). The ranges of the input arguments in these integral arrays are controlled by the parameter *j*, which starts from *N*, and goes down to (–*N* + 2) in steps of 2 inside the loop. In lines 10 and 13, the sums are calculated inside the loop, separately for both cosine and sine arrays, thus performing the crucial summation over *N* apertures in equation (13). In line 15, outside the loop, the electric field in *y (u)* is calculated in complex form, corresponding to the first complex factor in equation (14). In the next four lines, the Fresnel cosine and sine integrals are evaluated for the *z (or v)* dimension. Next, in line 20, the second complex electric field in *v* is calculated, corresponding to the second factor in equation (13). In the next lines, the matrix *C*, which contains the complex electric field variation in both *u* and *v* directions, is constructed. Finally, by squaring it, the matrix *D*, which contains the intensity values for the first quadrant of the image plane, in both *y* and *z,* is calculated (line 23). The elements of this matrix is then normalized (line 24), and the complete *E* matrix for the full image plane intensity distribution is constructed by inverting *D* twice (lines 27-35). Finally, the generated image is displayed as a grayscale image by the *imagesc* command with the appropriate scale (line 37). The complete MATLAB program, called *Nslit,* is given in the Appendix A.

3. Simulation Results

3.1. The Single Aperture

Using the program *rectan*, the values of the seven input parameters are inserted to the program: amm, bmm (aperture half-widths in mm), Wmm (image area half-width in mm), smm (step size in mm), lnm (the wavelength in nm), q0mm (the aperture-image plane distance in mm) and the intensity factor t. The computer simulated Fresnel image for λ = 632nm, q_0 = 400 mm and aperture dimensions 2mmX3mm is shown in Figure 8. The choice of wavelength of light (He-Ne laser wavelength) and the aperture-image plane distance is completely arbitrarily. The characteristic checkerboard pattern of Fresnel diffraction can be clearly observed. Though the program is suitable for calculating the diffraction pattern for any apertures, rectangular or square, for brevity, we limit our subsequent attention to square apertures only.

To examine the effect of change of simulation parameters on the diffracted images, we generated a series of image where the size of the square aperture is gradually increased from small to large, keeping the aperture-screen distance and the wavelength constant (at 400 mm and 632nm respectively). These simulation results are shown in Figures. 9(a-d). For the case of a very small aperture 0.3mm*X*0.3mm [see Figure 8(a)], the simulated image resembles a Fraunhofer diffraction pattern, while for the largest apertures 3mmX3mm [Figure 9(d)] we clearly observe a Fresnel-type diffraction pattern. This is expected, since for a small aperture, the waves diffracted from it appears to be approximately planar to a distant observer (the Fraunhofer limit), while for a large aperture, they will no longer be a plane wave at that position (Fresnel limit). As a rough rule of thumb, if we take *a* to be the dimension of square aperture, then Fresnel diffraction will occur if the dimension *a* satisfies the following relation [16],

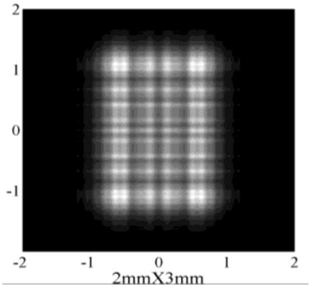


Figure 8. Computer simulated Fresnel diffraction pattern from a rectangular aperture of dimension 2mm×3mm at a aperture-screen distance of 400mm. The wavelength is λ = 632nm. (Reprinted from Abedin. et al. Opt. Laser Technol.Vol. 39. pp. 237-246, Copyright (2007) with permission from Elsevier).

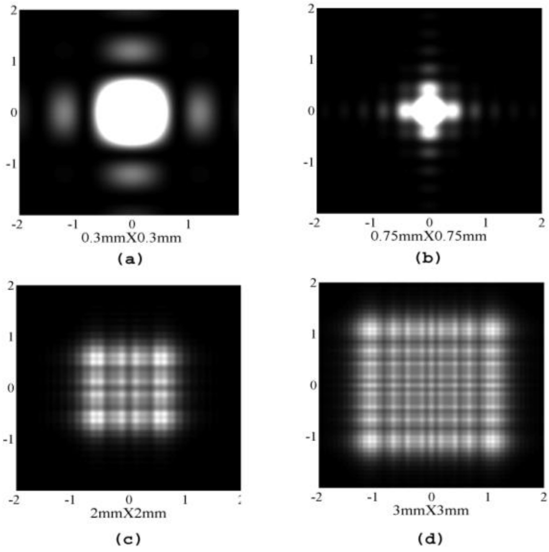


Figure 9. Computer simulated Fresnel diffraction images for increasing aperture size. The wavelength is 632nm and the aperture-screen distance is 400mm. (Reprinted from Abedin. et al. Opt. Laser Technol. Vol. 39. pp. 237-246, Copyright (2007) with permission from Elsevier).

$$q_0 < (a^2 / \lambda). \tag{16}$$

Otherwise Fraunhofer diffraction will occur. Using the given numerical values, it can be immediately verified that this is indeed the case for Figures 9(a) and 9(d). For an aperture of intermediate size of 0.75mm *X* 0.75mm [Figure 9(b)], we obtain a diffraction image that can be described as something between Fresnel and Fraunhofer patterns. This is indeed the *transition regime*, where the diffraction is undergoing a transition from Fraunhofer to Fresnel.

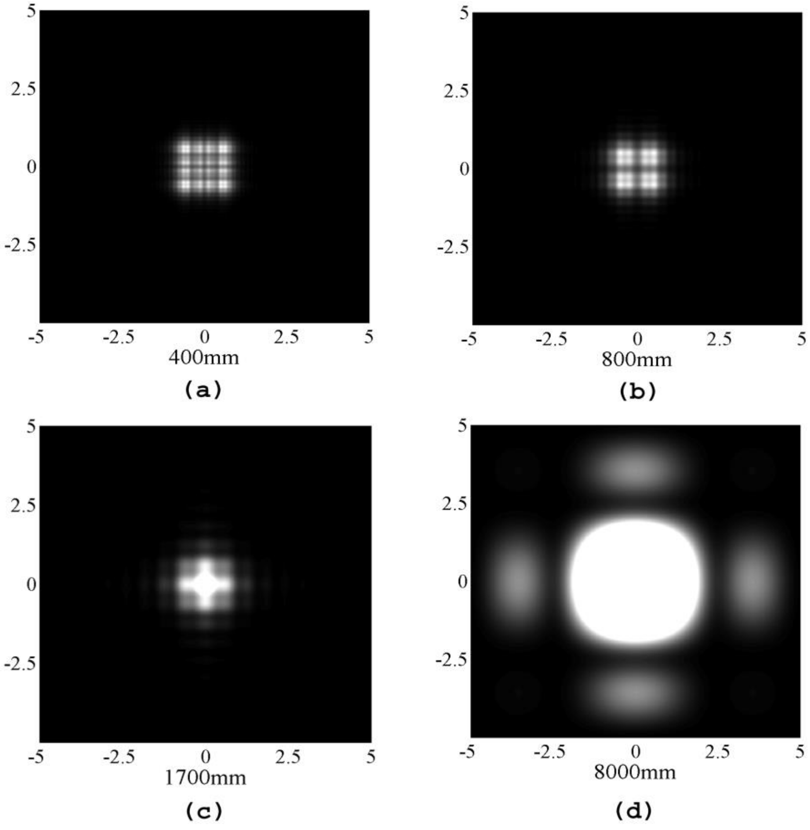


Figure 10. Simulated Fresnel diffraction images for increasing aperture-screen distance q0. The wavelength is 632nm and the aperture size is 2mm×2mm. (Reprinted from Abedin. et al. Opt. Laser Technol. Vol. 39. pp. 237-246, Copyright, (2007) with permission from Elsevier).

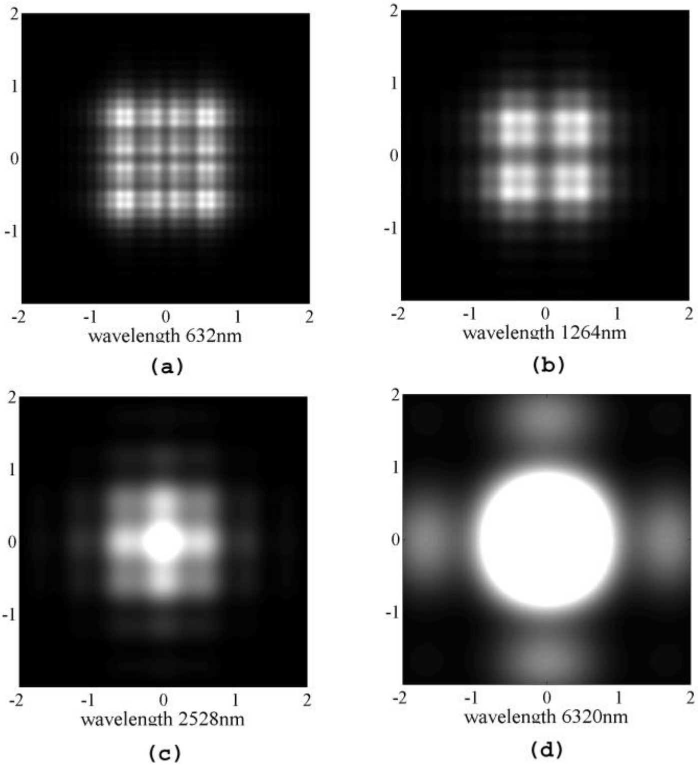


Figure 11. Simulated Fresnel diffraction images for increasing illumination wavelength. The aperture size is fixed at 2mm×2mm and the aperture-screen distance is 400mm. (Reprinted from Abedin. et al. Opt. Laser Technol..Vol. 39. pp. 237-246, Copyright, (2007) with permission from Elsevier).

As a second example of simulations, we fix the aperture size at 2mm*X* 2mm and the wavelength at 632nm, and increase the aperture-screen distance from q_0 = 400mm [same as image 9(c), a case of Fresnel diffraction] to q_0 = 8000mm in suitable steps. According to diffraction theory and the above criterion, we should expect a transition from the Fresnel regime back to Fraunhofer. The simulation images are shown in Figures 10(a)-10(d). Clearly a transition from Fresnel to Fraunhofer diffraction pattern is really observed, with an image [Figure 10(c)] that can be described as being in a state of transition between the two regimes.

It is also possible to easily observe the effect of change of wavelength on the diffraction patterns. For example, starting from the situation in Figure 9(c) [Fresnel case], we can gradually increase the wavelength *lnm* from 632nm (red) to 6320nm (mid-infrared) and can readily observe the expected transition to the Fraunhofer regime. This is clearly shown in Figure 11(a)-(d). (A real experiment to reproduce the effect will be quite difficult to perform in a laboratory.)

It must not be forgotten that in all the above cases, including those where the diffraction pattern is apparently Fraunhofer-like, we are basically using the general Fresnel diffraction formula [Eq. 3] and the Fresnel integrals to compute diffraction fields. No assumptions from the Fraunhofer theory was used. One can ask the question: how do they compare with the results of a purely Fraunhofer calculation? It can be shown that the Fraunhofer diffraction intensity distribution of a rectangular aperture of size *2a X 2b* is exactly given by [16-17],

$$\mathrm{I(Y,Z)} = \mathrm{I}_0 \ (\frac{\sin^2\alpha}{\alpha^2})(\frac{\sin^2\beta}{\beta^2}) \ . \qquad (17)$$

In this equation, I_0 is the intensity at the central image point (*P*), $\alpha = 2\pi aY/R\lambda$, $\beta = 2\pi bZ/R\lambda$ and *R* is the distance between the aperture and the observation plane and is assumed to be sufficiently large. The variables α and β are proportional to image co-ordinates *Y* and *Z*. It is simple to write a MATLAB program which accepts the values of aperture dimensions *2a* and *2b*, distance *R*, wavelength λ, desired image area *2WX2W*, and then computes the normalized intensity distribution [$I(Y,Z)/I_0$] from Eq.(17). The intensity can be shown either as an image, or for quantitative comparison, as a three-dimensional intensity plot (called a mesh plot) in MATLAB.

In Figure 12(a), we show the normalized 3-D mesh plot of the Fraunhofer image for an aperture size 2mm*X*2mm, for λ =632nm and observation distance *R*=8000mm. This is the characteristic Fraunhofer diffraction graphs found in undergraduate textbooks. For quantitative comparisons, in Figure 12(b) we show the 3-D mesh plot of the image [Figure 10(d)] shown previously, which was calculated (simulated) by iterative Fresnel integrals method for the same aperture under identical conditions. Comparing the two 3-D patterns, we observe an excellent agreement, with respect to both the principal maximum and the subsidiary (minor) maxima. This shows that our calculations using the

Fresnel integral method must be correct, since it has a good quantitative agreement with the purely Fraunhofer result (Eq. 17) under similar conditions.

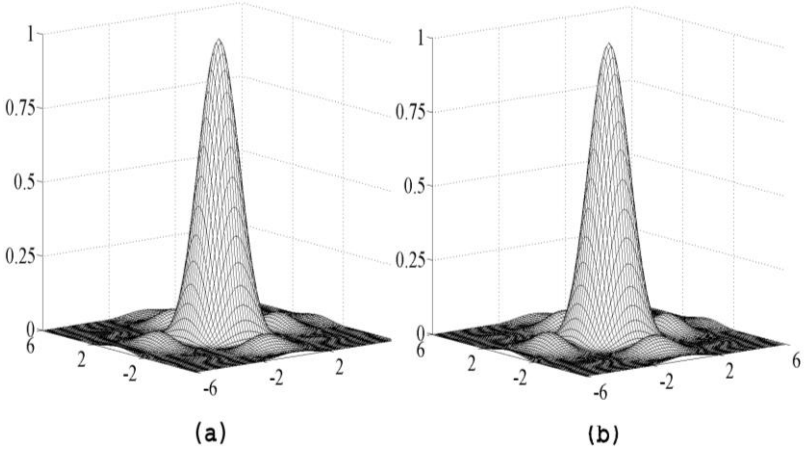
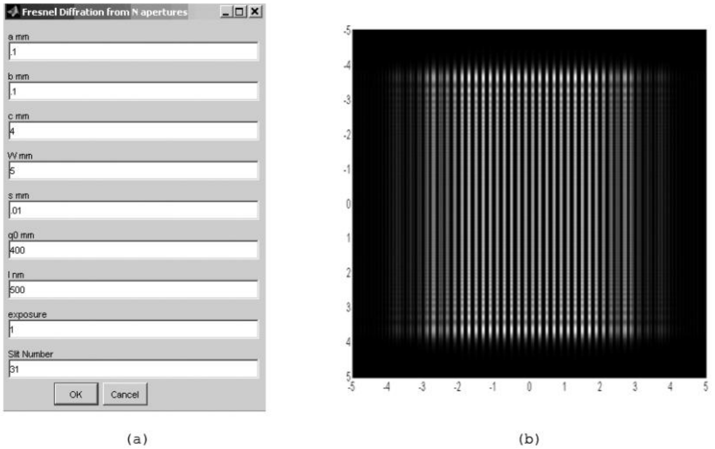


Figure 12. (a) Normalized three-dimensional plot of Fraunhofer diffraction pattern of aperture size 2mm × 2mm at an aperture-screen distance of 8000mm ($\lambda = 632$nm). (b) Simulated Fresnel diffraction pattern of this aperture under identical conditions. (Reprinted from Abedin. et al. Opt. Laser Technol.Vol. 39. pp. 237-246, Copyright (2007) with permission from Elsevier).


Fresnel Diffration from N apertures

a mm: .1

b mm: .1

c mm: 4

W mm: 5

s mm: .01

q0 mm: 400

l nm: 500

exposure: 1

Slit Number: 31

Figure 13. *(*a) Screenshot of the MATLAB GUI, showing the 9 input parameters. (b) The computer simulated Fresnel image from N = 31 apertures. The input parameters correspond to those in the GUI. (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

3.2. *N*-Apertures: Simulation Results

We now present the simulation results for the N-aperture case. Using the program *Nslit*, the values of the inputs, i.e., aperture parameters *a* and *b* in mm (see Figure 2), height *c* in mm, image half-width *W* in mm, step size *s* in mm, the aperture image-image plane distance q_0 in mm, the illuminating wavelength *l* in nm, the exposure factor *t* and the number *N* of the slits are input by a GUI to the program (Figure 13a). The exposure factor *t* is used to control the apparent visual intensity in the generated image, and be adjusted as necessary. The computer generated Fresnel image due to a *N*-aperture for wavelength λ=500 nm, q_0=400mm, image half-width W = 5mm, aperture width a=0.1 mm, b = 0.1 mm (with aperture separation 0.2 mm), height c = 4mm, s = 0.01mm and N = 31 is shown in Figure 13b. Since (a + b) = 0.2mm, this system is equivalent to an amplitude diffraction grating N = 31 with 5 lines/mm. The diffraction image superficially resembles that of a grating in the far-field, with Fresnel-like characteristics exhibited in the top and bottom edges. Some interference effects, resulting in intensity variations can be seen in the left and right edges.

The program equally works for even values of *N*. For example, the computer-simulated images for N = 10 and N = 20 for the input parameters λ = 500 nm, q_0 = 400mm, image area W = 5mm, aperture width a = 0.1 mm, b = 0.1 mm (with aperture separation a + b = 0.2 mm), height c = 4mm, s = 0.01mm are shown in Figures 14(a) and 14(b).

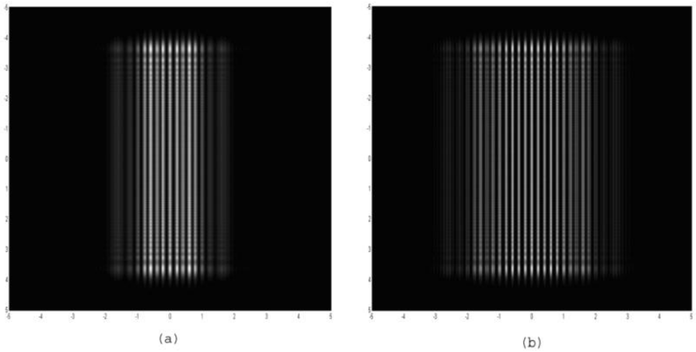
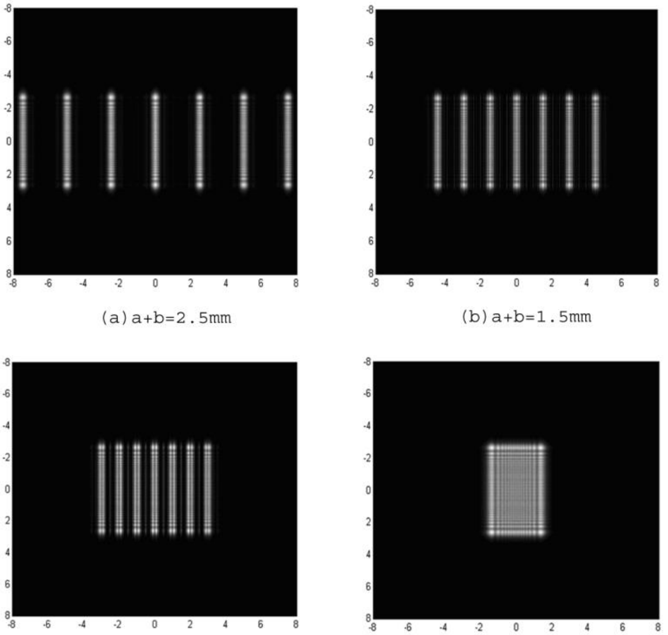


Figure 14. The computer simulated Fresnel image for λ = 500 nm, q_0 = 400mm, W = 5mm, aperture width a=0.1 mm, b=0.1 mm (with aperture separation 0.2 mm), height c = 4mm, s = 0.01mm are shown for (a) N = 10 and (b) N = 20. (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).



(a) a+b=2.5mm

(b) a+b=1.5mm

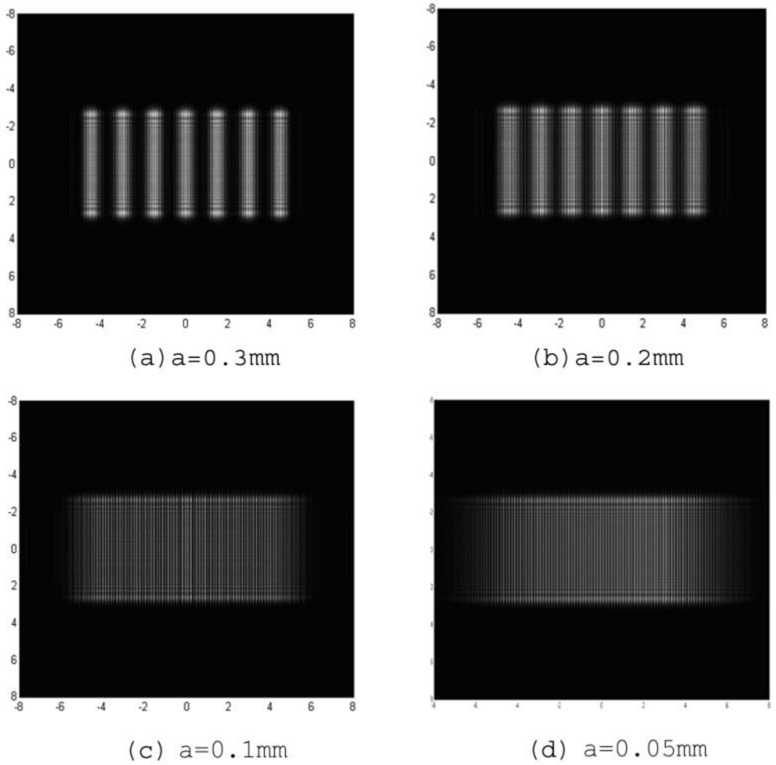
(c) a+b=1.0mm

(d) a+b=0.5mm

Figure 15. Computer simulated Fresnel diffraction images for decreasing (center-to-center) aperture separation (a+b), while keeping the aperture width (a = 0.5mm) constant. for seven slits. Other parameters are: λ = 500 nm, q_0 = 400mm, W = 8 mm, s = 0.01 mm and c = 3mm. Figure 7(d) corresponds to b = 0 (single aperture). (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

To examine the effect of a change of experimental parameters on the diffracted image, we generated several sets of simulations. In the first set of simulations, the aperture separation *(a + b)* was made successively smaller, while keeping the aperture width (*a* = 0.5mm) constant. A series of diffraction images were generated as shown in Figures 15(a)-(d) for a value of *N* = 7. In Figure 15(a), in the region between the apertures, no interference could be detected between the light from each of the aperture diffraction patterns. But in Figure 15(b) and (c), some clear interference between diffracted light from the separate apertures was seen as the separation of the slits is reduced. As the separation was decreased to zero [Figure 15(d) for *b* = 0], a typical single aperture Fresnel diffraction was observed which will be generated by a single aperture of 3.5mmX3mm.

In the next series of simulations, the width *a* of the apertures was made gradually narrower while their separation *(a + b)* was kept constant at 1.5 mm for *N* = 7 (seven slits). This is reproduced in figures 16 (a)-(d). From Figures 16(a) and (b), as the apertures become narrower, the light is diffracted over a wider region and some interference between diffracted light can be observed. In Figures 16(c) and (d), at smaller aperture widths, strong mutual interference between diffracted light generates fringes which look like typical Young's fringes. Light from each aperture is diffracted over a wide portion of the image area, causing mutual interference. Moreover, the individual apertures can no longer be distinguished in the diffracted pattern, as an Figures. 16(a) and 16(b). In Figures 16(c) and (d), fringe density is apparently the same, since this depends only on aperture separation.



(a) a=0.3mm

(b) a=0.2mm

(c) a=0.1mm

(d) a=0.05mm

Figure 16. Computer simulated Fresnel diffraction images for decreasing aperture width a, while keeping the aperture separation (a+b)=1.5mm constant for a seven-slit system. λ=500 nm, q_0 = 400mm, s = 0.01 mm and c = 3mm. (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

3.3. *N*-Apertures: Comparison with *N*-slit Fraunhofer Diffraction

If the aperture-screen distance q_0 is increased, it is well-known that a gradual transition from the Fresnel regime to Fraunhofer regime should be expected. We can use the criterion described in relation (16) to determine where the transition should occur. This is reproduced below:

$$\mathrm{q}_0 < (\mathrm{D}^2 / \lambda).$$

In the present context, *D* represents the maximum dimension of the aperture system in the *Y* or *Z* directions.

In the next series of simulations, for a 7-slit system, we increased aperture-screen distance q_0, and selected input parameters as: the wavelength λ=500 nm, aperture width a=0.1 mm, b=0.3 mm (with aperture separation 0.4 mm), height c=3mm, image area size *W* variable, s=0.01mm or 0.1 mm, as appropriate. We then increased the aperture screen distance q_0 in steps from 400 mm to 64,000 mm while keeping other parameters constant [Figure 17]. A transition from the Fresnel regime to Fraunhofer regime is clearly observed, being consistent with the above relation. For example, for a 7-slit system with $(a + b)$ = 0.4mm, if we take the total lateral dimension of the slit system to be D = 2.8mm, then D^2/λ is about 16,000 mm. We can expect the diffraction to be Fresnel-like if q_0 is less than this value. On the other hand, if, $(D^2/\lambda) > q_0$, and we can expect the diffraction pattern to be Fraunhofer-like. In between these extremes, a transition from Fresnel to Fraunhofer regime is expected to occur. [as in Figures 17(c) and 17(d)].

The usual mathematical analysis of a *N*-slit system, as found in textbooks [16, 19] usually deals with the *N*-slit pattern only in the far-field Fraunhofer regime. The generalized Fraunhofer intensity distribution of an *N*-aperture system in the image plane (*Y*, *Z*) can be exactly calculated by an analytical formula

$$I(Y,Z) = I_0 \left(\frac{\sin^2 \beta}{\beta^2}\right) \left(\frac{\sin^2 N\gamma}{\sin^2 \gamma}\right)\left(\frac{\sin^2 \alpha}{\alpha^2}\right), \tag{18}$$

where β, γ and α are defined as,

$$\beta = (\pi a Y / \lambda R), \qquad \gamma = [\pi (a+b) Y / \lambda R], \qquad \alpha = (\pi c Z / \lambda R). \tag{19}$$

In the above equations, a is aperture width, $(a+b)$ is the aperture separation, c is aperture height, λ is the wavelength and R is the distance between the aperture and the screen, assumed to be sufficiently large. As pointed out above, in our computer simulations, this Fraunhofer conditions can be considered to be sufficiently fulfilled in Figures 17(e) and 17(f), for R (or q_0)=32,000 mm and R=64,000 mm, respectively.

In the conventional analysis of Fraunhofer diffraction for this case [19], the principal maxima of the diffraction pattern occurs when the $[\sin N\gamma/\sin \gamma]^{2.}$ factor have maximum values, i.e., when,

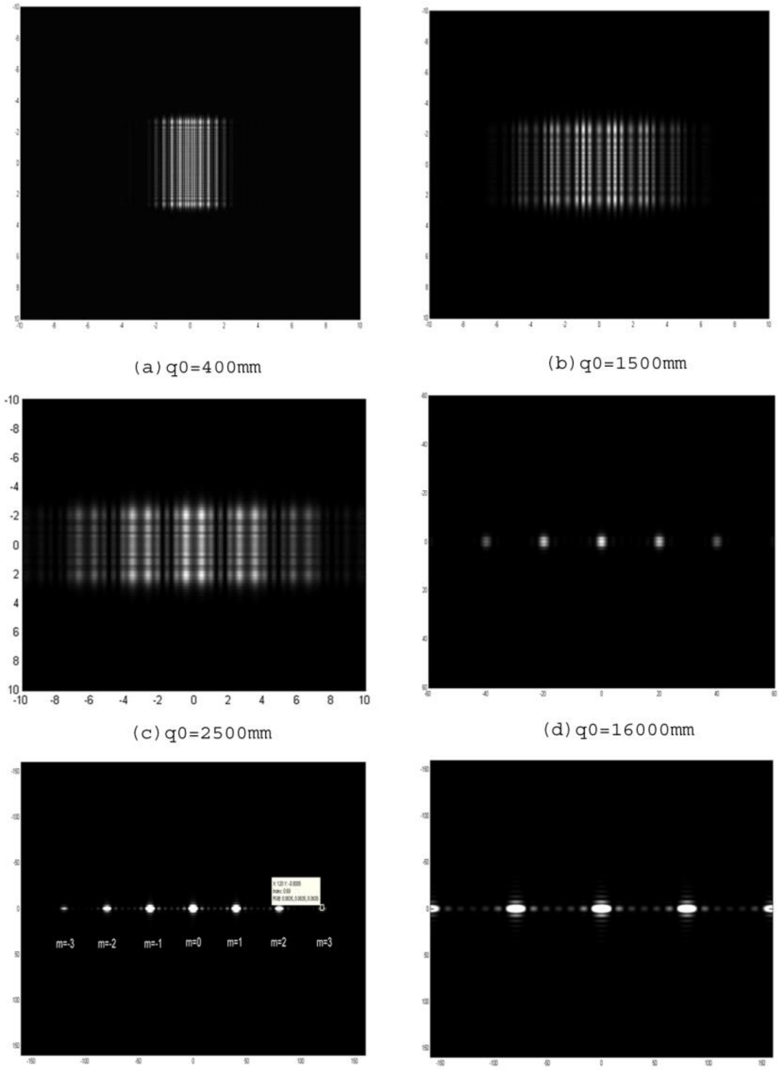
$$(a+b)\sin\theta = m\lambda \tag{20}$$

Here, θ is the diffraction angle and m is known as the *diffraction order*. This equation is known as the *grating equation*. For m = 0, we have the zeroth-order, for m = 1, we have the first order and so on. [Since (a + b)/a = 4 in the above simulation, m = 4 will the *missing order*, and hence it is not expected to occur.] In addition to these principal maxima, several minima of zero intensity and several much weaker secondary maxima are predicted to occur between the principal maxima. If N is the number of grating slits (apertures), then the number of secondary maxima is equal to (N-2), and the number of minima is equal to (N-1) [19].

To compare with simulations quantitatively, Figure 17(e) is enlarged and presented in Figure 18, with the intensity values multiplied by a factor of 10 to bring out the fainter details. We clearly observe 7 principal maxima, with m=0 (at Y = 0), with m=+1 (at Y = 39.8 mm), m = +2 (at Y = 80.0 mm), m = +3 (at Y = 120 mm), with m=-1 (at Y = -39.8 mm), m = -2 (at Y = -80.0 mm), and m = -3 (at Y = -120 mm). *In addition*, 5 (= N-2) secondary maxima are clearly visible between the principal maxima at the zeroth order and the first order, and between the first and the second orders, along with 6 minima of zero intensity, between the secondary maxima. All these observations are consistent with theoretical expectations.

The values of sin θ can be estimated by calculating the ratios (Y/R, with R = 32,000 mm) from the simulated image (Figure 18). These calculated values are: 1.23X10^{-3} (for m = +1 and m= -1), 2.50 X10^{-3} (for m=+2 and m= -2), and 3.75X10^{-3} (for m = +3 and m = -3.)

From equation (20), on the other hand, for λ = 500 nm, (a + b)= 0.4 mm, the following values of sin θ are calculated: 1.24X10^{-3} (for m = +1 and m = -1) 2.50X10^{-3} (for m = +2 and m = -2), and 3.75X10^{-3} (for m = +3 and m = -3).



(a) q0=400mm

(b) q0=1500mm

(c) q0=2500mm

(d) q0=16000mm

(e) q0=32000mm

(f) q0=64000mm

Figure 17. Computer simulated Fresnel diffraction images for a 7-slit system. Aperture-screen distance q_0 is increased in steps, while keeping other parameters constant, with a = 0.1mm, b = 0.3mm, c=3mm, λ = 500 nm. (a) and (b) are in the Fresnel regime, while (e) and (f) are in the Fraunhofer regime, according to equation (16). (Reprinted from Abedin et al. Optik Vol. 126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

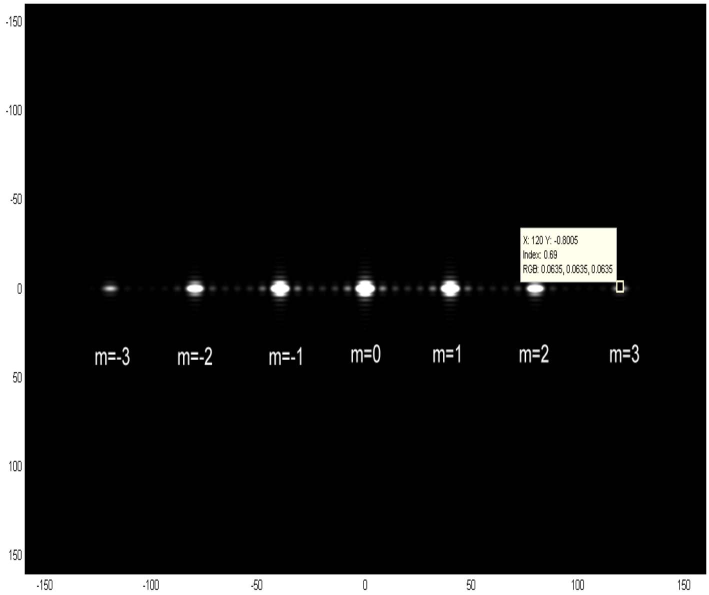


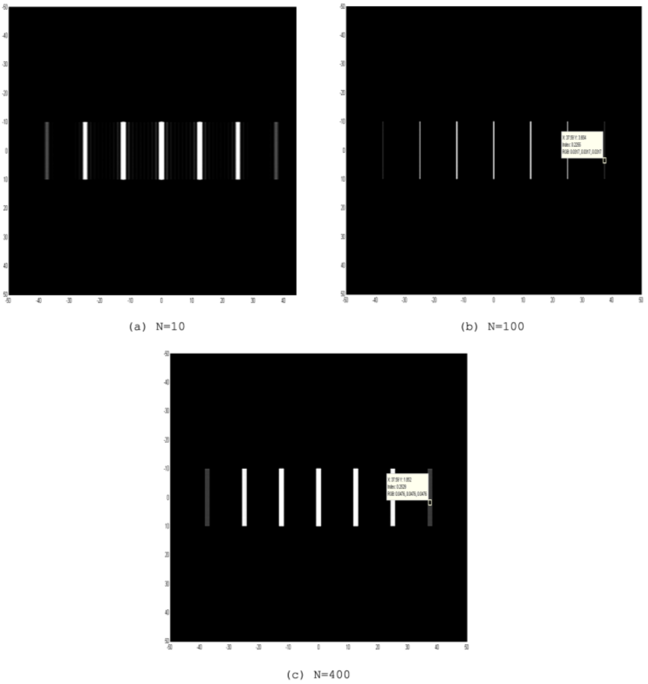
Figure 18. Computer simulated Fresnel diffraction images for a 7-slit system, reproduced for clarity for an aperture-screen distance q_0=32,000 mm (Fraunhofer regime), with a = 0.1mm, b = 0.3mm, c = 3mm and. λ = 500 nm. The intensity values have been multiplied by a factor of 10 to bring out the fainter details. The principal diffraction orders can be seen, as well as the secondary maxima and the minima between them. M = 4 is the missing order, hence it is not expected to occur. (Reprinted from Abedin et al. Optik Vol.126. pp. 3743-3751, Copyright (2015) with permission from Elsevier).

Therefore, we conclude: the computer-simulated results of the diffraction image, in the Fraunhofer limit, almost exactly agree, *both qualitatively and quantitatively*, with the results from the exact Fraunhofer theory, as represented by equations (18)-(20). This agreement strongly supports the validity of the present computation methods and techniques.

3.4. *N*-Apertures: Simulation of Diffraction from Realistic Amplitude Gratings

We simulated diffraction images with a realistic aperture spacing (a + b), so that it corresponds to a real amplitude grating where the grating period is of the order of the wavelength of light. We choose a = 0.001 mm, b = 0.003 mm, and *c* = 5 mm, so that the aperture spacing is 0.004 mm. This is equivalent to a grating with grating constant of 250 lines/mm. Simulations are performed in both Fresnel and Fraunhofer regimes, but are shown in the Fraunhofer regime (with q_0 = 100 mm and λ = 500 nm) to facilitate comparison with the exact grating equation [Eq. 20]. The simulated images are shown in Figure 19 for three different aperture members *N* = 10, *N* = 100 and *N* = 1000. (M=4 will the missing order, and its expected position should be at *Y* = 50 mm according to the grating equation.)

In all the above cases, there is excellent agreement with the observed positions of the principal maxima with those predicted by the exact Fraunhofer theory. The agreement is shown in Table 1.



(a) N=10

(b) N=100

(c) N=400

Figure 19. Computer simulated Fresnel diffraction images for some realistic gratings with different aperture numbers in the far-field regime, with a = 0.001mm, b = 0.003mm, c = 10mm, q_0 = 100 mm and λ = 500 nm. (a) N = 10, (b) N = 100 and (c) N = 400. The intensity values have been multiplied by a factor of 5 to bring out the fainter details.

Table 1. Comparison between Simulations and Theory for the Positions of the Primary Maxima for various grating orders for *N* = 10, *N* = 100 and *N* = 400

Positions of the principal maxima (mm)	From computer simulations in the far-field regime			Calculated from the grating equation (mm)
	N = 10	*N* = 100	*N* = 400	
Zeroth	0	0	0	0
First	12.5	12.5	12.51	12.5
Second	25.0	25.03	25.13	25.0
Third	37.55	37.59	37.49	37.5

From the above table, we observe that there is excellent agreement between the results of simulation and Fraunhofer theory regarding the positions of the principal maximum for a grating with a large number of slits (apertures) which can be put to a practical use. As the number of slits *N* is increased, the positions of the principal maxima remains unchanged, but the number of secondary maxima (*N* - 2) increases and their intensities also decrease significantly, becoming almost invisible for large *N*. The diffraction image is then dominated by the principal maxima only.

4. Discussion

In this Chapter, we have described in detail how the Iterative Fresnel Integrals Method can be applied, first to a single aperture, and then extended to the case of multiple apertures (N apertures). Details of the simulation strategy and methodology are given, and the implementation plan in a MATLAB program. Using these MATLAB codes, the reader can perform these simulations by his own and see the results immediately. Extensive experience is MATLAB programming is not required, as all inputs to the programs are through MATLAB GUIs, and the only outputs are the computed diffraction images, which can be interpreted immediately. Our simulation method (for the single apertures) has already been used for the simulation of diffraction images in a variety of practical situations, such as in fabrication of UV mask for microfluidic systems [26, 27], contact mask lithography [28], and more recently, for micro-nanofabrication [29] and in contact lithography for fabrication of photonics components [30].

4.1. Single Apertures

In the single-aperture case, which is the simplest, the theory is developed *ab-initio* from the classic Huygens-Fresnel principle, and the diffraction integral is evaluated in terms of the Fresnel cosine and sine integrals. By a series of virtual displacements of the aperture, the complete diffraction pattern is mapped out in the image plane in terms of repeated (iterative) calculation of Fresnel integrals. This is the essence of the Iterative Fresnel Integrals method. Typical checkerboard pattern of Fresnel diffraction, which is reproduced in textbooks from real experiments, can be reproduced exactly. The effect of changing any of the experimental parameters, such as aperture size, aperture-screen distance, and wavelength can be immediately observed by changing the input parameters in the simulation program. In a real diffraction experiment, changing the aperture size or the wavelength of light involves a significant amount of investment in experimental equipment, and therefore is not easy. This is particularly true for very small diffraction apertures (where microfabrication techniques have to be used), or very long (infrared) or very short (ultraviolet) illumination wavelengths, which must be monochromatic.

It is well-known that in the limit of large aperture-screen distances, long illumination wavelengths, or small apertures, Fresnel diffraction become Fraunhofer diffraction, and the diffraction pattern can be exactly calculated by analytical equations. We actually observed this happening in our virtual experiments. In particular, in the limit of large aperture-screen distances, we calculated the Fresnel diffraction intensity distribution by our simulation technique, and compared with analytical formula for Fraunhofer diffraction. The excellent quantitative agreement between the two implies that the simulation technique is correct and accurately represents the reality.

4.2. Multiple Apertures and Diffraction Gratings

The theory was extended to the non-trivial case of multiple apertures, and the corresponding algorithm was described. The summation of the complex electric field for *N* apertures was performed inside an iterative loop containing Fresnel cosine and sine integrals, whose iterations depend on the value of *N*. A number of different simulation images for even and odd number of apertures were produced. The intensity value was calculated by squaring the complex electric field. Here the complex number capability of MATLAB was used to full advantage.

The simulated images were generated for a number of different experimental configurations. We found that as the individual apertures of an *N*-aperture system are brought closer, significant interference between diffracted light can be seen in the space between apertures. On the other hand, if the aperture widths of individual apertures are made narrower while keeping their separations constant, the light is observed to be diffracted over a wider region, and Young-like fringes appear on the image. The expected transition to the Fraunhofer diffraction at large aperture-screen distance was obtained by simulation for a diffraction grating having 7 slits for different experimental conditions. The diffraction pattern was compared with those predicted by the Fraunhofer theory. In the far-field Fraunhofer limit, the positions of the maxima correspond exactly to those predicted by the exact grating equation. The number and positions of the secondary maxima and minima also agree with the theory.

The amplitude diffraction grating is extremely important in spectroscopy, as pointed out in the Introduction. Because of the difficulty of treating the near-field Fresnel diffraction theoretically, almost always the diffraction from an amplitude grating is treated in the far-field Fraunhofer regime. To render the incident and diffracted rays parallel, as required in the Fraunhofer limit, collimating lenses or mirrors are almost always used.

Our simulation method enables one to treat the near-field Fresnel regime and also to examine the effects of the changing the various experimental parameters, such as grating spacing, number of lines, wavelength, and other parameters with a minimum of effort. In fact, complete virtual experiments can be performed with our program to simulate any experimental condition, without ever performing any real experiments. We are reasonably certain that all the outcomes will agree with the experiments, as long as our assumptions of the iterative Fresnel integrals method are valid to a reasonable degree.

Finally, simulations were performed for realistic amplitude gratings having a large number of apertures *N* with spacings comparable to the wavelength of light. The MATLAB program was successfully able to handle large number of apertures without any memory overflow or errors. The simulations produced the expected results in the far-field Fraunhofer regime. In addition, it is able to generate the diffraction images in the Fresnel regime too.

4.3. Simulation Times

All the above simulations can be performed in any modern PC where MATLAB is installed. The requirements on computational resources for all of the above simulations are quite modest. For most of the computer simulations, we used an ordinary office PC or a general purpose lab PC running under Windows 8. The CPU clock speed was about 2.4 GHz, and the installed RAM in the PC was 8GB. If the image sizes are of the order of 1000*X*1000 pixels, the computation time is of the order of less than a minute in most cases for single apertures. On the other hand, for much larger images (for example, 3000*X*3000 pixels or larger), the computation time will be increased considerably. Using high-performance PCs or workstations using multicore processors and large amounts of RAM, the computation times for even large images can be expected to be significantly smaller. For *N*-apertures or diffraction gratings, computation time depends on the image size as well as the value of *N*, and for large images with large values of N (of the order of hundreds or thousands), computation time is can be of the order of tens of minutes or even an hour. But there is no limitations on RAM for large *N*, and the requirement of memory space do no increase with the value of N, since the summation of the electric field *E* is done inside a *for* loop, without any increase of memory requirements. The CPU speed then becomes the limiting factor.

Though we have presented our program in MATLAB, it should be possible to translate the programs into other high-level scientific languages, such as Mathematica or MathCad. Depending on the efficiency of calculation of the Fresnel integrals and on the efficiency how the matrix calculations are performed, however, the computation times for the programs written in the above languages maybe significantly different.

4.4. Further Extensions of the Problem

Two variations of the *N*-slit problem can be considered. The first is the illumination by multiples optical wavelengths, which would be expected in a real-life situation, for example, a spectrograph analyzing light from a mercury discharge lamp emitting multiple wavelengths. The second will be the simulation of the near-field diffraction pattern for the case where the light is incident at an oblique angle on the grating which may be tilted and rotated by some mechanical means. This situation also occurs in a practical spectrograph

[31-33], for example, in the popular Czerny-Turner spectrograph configuration. In this case, the problem would be to combine the geometry of the tiled grating [23] with the geometry of the *N*-slits [25] and to find the total diffracted field in the case of a tilted *N*-aperture system.

CONCLUSION

In conclusion, we can say that we have explained the complete solution of the problem of near-field diffraction from an amplitude diffraction grating, with complete algorithms and MATLAB codes. Anybody can do the virtual experiments in his or her PC, using the codes we have provided in the Appendix. The codes can be used also as tool of learning diffraction phenomena from gratings in an educational environment, for example in an undergraduate course in optics and diffraction, or as a demonstration of a computer simulation experiment.

APPENDICES

A. **rectan**: *Complete MATLAB program for the single aperture*

```
1. u=inputdlg({'a mm','b mm','W mm','Step mm','q0 mm','Wavelength nm','Exposure'}, …
2. 'Fresnel Diffraction from Rectangular Aperture', [1,1,1,1,1,1,1]);
3. for i=1:7; v(i)=str2num(u{i}); end
4. l=v(6)*1e-6; t=v(7); q0=v(5);
5. a=v(1)*sqrt(2/(l*q0)); b=v(2)*sqrt(2/(l*q0));
6. W=v(3)*sqrt(2/(l*q0)); s=v(4)*sqrt(2/(l*q0));
7. Cu2=mfun('FresnelC', a:s:W+a);
8. Cu1=mfun('FresnelC',-a:s:W-a);
9. Su2=mfun('FresnelS', a:s:W+a);
10. Su1=mfun('FresnelS', -a:s:W-a);
11. A=(Cu1-Cu2).^2+(Su1-Su2).^2;
12. Cv2=mfun('FresnelC', b:s:W+b);
13. Cv1=mfun('FresnelC', -b:s:W-b);
14. Sv2=mfun('FresnelS', b:s:W+b);
15. Sv1=mfun('FresnelS', -b:s:W-b);
16. B=(Cv1-Cv2).^2+(Sv1-Sv2).^2;
```

```
17. B=B';n=size(B);B=repmat(B(:,1),1,n);
18. A=A';A=repmat(A(:,1),1,n); A=(A)';
19. D=B*A;
20. D=t*D/max(max(D));
21. m=2*fix(((2*W)/s)/2);
22. E=zeros(m+1,m+1);
23. for q=1:1:m/2+1;
24. for p=1:1:m/2+1; E(m/2+2-p,q+m/2)=D(p,q); end
25. end
26. for q=m+1:-1:m/2+2;
27. for p=1:1:m/2+1; E(p,-q+2+m)=E(p,q); end
28. end
29. for q=1:1:m+1;
30. for p=1:1:m/2; E(-p+2+m,q)=E(p,q); end
31. end
32. y=-W:s:W;ymm=y*sqrt((l*q0/2));
33. imagesc(ymm,ymm,E,[0 1]);colormap(gray);
```

B. **Nslit:** *Complete MATLAB program for the N-aperture*

```
1. u=inputdlg({'a mm','b mm','c mm','W mm','s mm','q0 mm','l nm',...
2. 'exposure', 'Slit Number'},'Fresnel Diffration from N apertures',
 [1,1,1,1,1,1,1,1,1]);
3. for i=1:9; v(i)=str2num(u{i}); end
4. q0=v(6); t=v(8); l=v(7)*1e-6;f=sqrt(2/(l*q0));
5. a=v(1)*f; b=v(2)*f; c=v(3)*f; W=v(4)*f; s=v(5)*f;N=v(9);r=(N-1)/2;
6. CuS=0; SuS=0;
7. for j=N:-2:-N+2
8. Cu2=mfun('FresnelC',j/2*a+r*b:s:W+j/2*a+r*b);
9. Cu1=mfun('FresnelC',(j-2)/2*a+r*b:s:W+(j-2)/2*a+r*b);
10. CuS=CuS+Cu2-Cu1;
11. Su2=mfun('FresnelS',j/2*a+r*b:s:W+j/2*a+r*b);
12. Su1=mfun('FresnelS',(j-2)/2*a+r*b:s:W+(j-2)/2*a+r*b);
13. SuS=SuS+Su2-Su1;r=r-1;
14. end
15. A=complex(CuS,SuS);
16. Cv2=mfun('FresnelC',c:s:W+c);
17. Cv1=mfun('FresnelC',-c:s:W-c);
18. Sv2=mfun('FresnelS',c:s:W+c);
```

```
Sv1=mfun('FresnelS',-c:s:W-c);
B=complex(Cv1-Cv2,Sv1-Sv2);
B=B';n=size(B);B=repmat(B(:,1),1,n);
A=A';A=repmat(A(:,1),1,n);A=(A)';
C=B*A;D=C.*conj(C);
D=t*D/max(max(D));
m=2*fix(((2*W)/s)/2);
E=zeros(m+1,m+1);
for q=1:1:m/2+1;
for p=1:1:m/2+1; E(m/2+2-p,q+m/2)=D(p,q); end
end
for q=m+1:-1:m/2+2;
for p=1:1:m/2+1;E(p,-q+2+m)=E(p,q);end
end
for q=1:1:m+1;
for p=1:1:m/2;E(-p+2+m,q)=E(p,q);end
end
y=-W:s:W;ymm=y/f;
imagesc(ymm,ymm,E,[0,1]);colormap(gray);
```

REFERENCES

[1] Birdsall, CK; Langdon, AB. *Plasma Physics via Computer Simulation*, Taylor and Francis, 2004.

[2] Winsberg, E. *Science in the Age of Computer Simulation*, University of Chicago Press, 2010.

[3] Ingham, J; Dunn, IJ; Heinzle, E; Prenosil, JE. *Chemical Engineering Dynamics: Modeling with PC Simulation* (2nd ed.), John Wiley, 2000.

[4] Cohen, L; Manion, L; Morrison, K. *Research methods in Education*, Routledge, 2007.

[5] Schmidt, JD. *Numerical Simulation of Wave Propagation*, SPIE Press, Bellingham, USA, 2010.

[6] Coles, WmA; Filice, JP; Frehlich, RG; Yadlowsky, M. Simulation of wave propagation in three-dimensional random media. *Appl. Opt.*, 1995, 34, 2089-2101.

[7] Varslot, T. Computer simulation of forward wave propagation in soft tissue. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control.*, 2005, 52, 1473-1482.

[8] Arsenault, HH; Sheng, Y. *An Introduction to Optics in Computers*, SPIE Press, Bellingham, 1992.

[9] Alexandrova, IV; Valieva, RZ. Computer simulation of X-ray diffraction patterns of nanocrystalline materials. *Philosophical Magazine Part B*., 1996, 73, 861-872.

[10] Voelz, D. *Computational Fourier Optics*; Tutorial Texts in Optical Engineering Volume TT89, SPIE Press, Bellingham, 2011.

[11] Mu, G; Wang, X; Wang, Z. Amplitude-compensated matched filtering. *Appl. Opt.*, 1988, 27 3461-3463.

[12] Horner, JL; Gianino, PD. Phase-only matched filtering. *Appl. Opt.*, 1984, 23, 812-816.

[13] Walk, M; Niklaus, J. Some remarks on computer-aided design of optical lens systems. *Journal of Optimization Theory and Applications*., 1988, 59, 173-181.

[14] Lamb, DJ; Chipman, RA; Hillman, LW; Takahashi, Y; Dimmock, JO. Computer modeling of optical systems containing Fresnel lenses. *AIP Conf. Proc.*, 1997, 433, 434-438.

[15] Eylon, B; Ronen, M; Ganiel, U. Computer simulations as tools for teaching and learning: Using a simulation environment in optics. *J. of Sci. Educ. Technol.*, 1966, 5, 93-110.

[16] Hecht, E. *Optics*, 4th ed. Singapore: *Pearson Education*, 2002 [Chap. 10].

[17] Born, M; Wolf, E. *Principles of Optics*, **7**th ed. Cambridge: Cambridge University Press, 1999 [Chap. 8].

[18] Guenther, RD. *Modern Optics*, John Wiley and Sons, 1990 [Chap. 11].

[19] Jenkins, FA; White, HE. *Fundamentals of Optics*, 4th ed. McGraw-Hill, 1981 [Chap. 18].

[20] Rudolf, PG; Tollett, JJ; McGowan, MM. Computer modeling of wave propagation with a variation of the Helmholtz-Kirchhoff relation. *Appl. Opt.* 1990, 29, 998.

[21] Abedin, KM; Islam MR; Haider, AFMY. Computer simulation of Fresnel diffraction from rectangular apertures and obstacles using the Fresnel integrals approach. *Opt. Laser Technol.*, 2007, 39, 237-246.

[22] Abedin, KM; Rahman, SMM. Computer simulation of Fresnel diffraction from double rectangular apertures in one and two dimensions using the iterative Fresnel integrals method *Opt. Laser Technol.*, 2012, 44, 394-402.

[23] Abedin, KM; Rahman, SMM. The Iterative Fresnel Integrals Method for Fresnel Diffraction from Tilted Rectangular Apertures: Theory and Simulations. *Opt. Laser Technol.,* 2012, 44, 939-947.

[24] Al-Saiari, FH; Rahman, SMM; Abedin, KM. Computer simulation of Fresnel diffraction from triple apertures by iterative Fresnel integrals method, *Photonics Optoelectron.,* 2012, 1(2), 33–42.

[25] Abedin, KM; Rahman, SMM. Fresnel Diffraction from N-apertures: Computer Simulation by Iterative Fresnel Integrals Method. *Optik,* 2015, 126(23), 3743-3751.

[26] Macken, S; Filippini, D. Monolithic SU-8 macrocavities for efficient fluorescence collection. *J. Micromech. Microeng.,* 2009, 19, 085011.

[27] Macken, S; Filippini, D. Monolithic SU-8 macrocavities for efficient fluorescence collection. *Procedia Chem*., 2009, 1, 1115.

[28] Kim, H; Kim, J; Kim, EG; Heinz, AJ; Kwon, S; Chun, H. Optofluidic *in situ* maskless lithography of charge selective nanoporous hydrogel for DNA preconcentration. *Biomicrofluidics,* 2010, 4, 43014.

[29] Zhou, L; Dong, X; Zhou, Y; Su, W; Chen, X; Zhu, Y; Shen, S. Multiscale Micro−Nano Nested Structures: Engineered Surface Morphology for Efficient Light Escaping in Organic Light-Emitting Diodes. *Applied Materials and Interfaces,* 2015, 7, 26989.

[30] Markey, L; Zacharatos, F; Weeber, JC; Prinzen, A; Waldow, M; Nielsen, MG; Tekin, T; Dereux, A. Recess Photomask Contact Lithography and the fabrication of coupled silicon photonic and plasmonic waveguide switches. *Microelectron. Eng.,* 2015, 141, 129.

[31] Loewen, EG; Popov, E. *Diffraction Gratings and Applications,* Mercel Dekker, New York, 1997.

[32] Schrenk, WG. *Analytical Atomic Spectroscopy,* Plenum, New York and London, 1975.

[33] Tkachenko, NV. *Optical Spectroscopy: Methods and Instrumentations,* Elsevier, Amsterdam, 2006.

BIOGRAPHICAL SKETCHES

Dr. Kazi Monowar Abedin
Department of Physics, Sultan Qaboos University, Muscat, Oman

Education: Doctor of Philosophy (Ph.D.)

Address: Department of Physics, College of Science, Sultan Qaboos
University, P.O. Box 36, Al-Khoudh, Muscat, 123 Oman

Research and Professional Experience:

Teaching in the undergraduate and graduate levels in the university (since 1994)

Research experience in laser physics, laser applications, optical and speckle metrology, Laser spectroscopy and applications, diffraction theory (since 1991)

Professional Appointments:

Assistant Professor, (1994-2000), Associate professor (2000-2003) and Professor (since 2003): Department of Physics, University of Dhaka

Associate Professor, Department of Physics, Sultan Qaboos University (since 2014)

STA Research Fellow, Mechanical Engineering Laboratory, Japan (1996-97)

Commonwealth Fellow, Clarendon Laboratory, University of Oxford, UK (2001-2)

Visiting Scientist, Max-Planck Institute for the Science of Light, Erlangen, Germany (2008-9)

Honors:

Razzak-Shamsun Research prize, (1999-2000) awarded in 2006 for outstanding contribution to research work

Ibrahim Memorial Gold Medal (2002), awarded in 2006 for research in ESPI

Publications Last 3 Years:

1. Fresnel Diffraction from N-apertures: Computer Simulation by Iterative Fresnel Integrals Method. K.M. Abedin and S.M.M. Rahman, Optik- International Journal for Light and Electron Optics, 126 (23), 3743-3751 (2015).
3. Survey of the Water Bodies for Ecotoxic Metals by Laser-Induced Breakdown Spectroscopy. AFMY Haider, B Rahman ZH Khan and

KM Abedin, Environmental Engineering Science (USA) 32 (4): 284-291 (2015).

4. Radiative lifetime measurement of excited neutral nitrogen atom by Time Resolved Laser-induced breakdown spectroscopy. A.F.M.Y. Haider, M. K. Ira, Z. H. Khan and K.M. Abedin, Journal of Analytical Atomic Spectrometry (RSC, UK), 29, 1385-1392 (2014).
5. Detection of trace amount of arsenic in groundwater by laser-induced breakdown spectroscopy and adsorption. AFMY Haider, M.H. Ullah, ZH Khan, F Kabir and KM Abedin, Optics and Laser Technology (UK) 56, 299- 303 (2014).
6. Determination of the Ash Content of Coal without Ashing: A Simple Technique Using Laser-Induced Breakdown Spectroscopy. AFMY Haider, MA. Rony and KM Abedin, Energy and Fuels (ACS, USA), 27(7), 3725-3729 (2013).
7. Elemental Profiling and Identification of Eco-Toxic Elements in Agricultural Soil by Laser-Induced Breakdown Spectroscopy. AFMY Haider, F. Kabir, M. Ullah, ZH Khan and KM. Abedin, Applied Ecology and Environmental Sciences, 1(4), 41-44 (2013).

Dr. S. M. Mujibur Rahman
Department of Physics, Sultan Qaboos University, Muscat, Oman

Education: Doctor of Philosophy (Double

Address: Department of Physics, College of Science, Sultan Qaboos
University, P.O. Box 36, Al-Khoudh, Muscat, 123 Oman

Research and Professional Experience:

University teaching at the undergraduate and graduate levels since 1974. Research experience in theoretical investigation of the structural, thermodynamic, transport and thermo-mechanical properties of metals and binary alloys. Recent interests also include dynamics of carbon fluids and simulation of optical properties. Also supervising research projects at the undergraduate, postgraduate and PhD levels.

Professional Appointments:

Date	Level of Appointment	Organization
1973-1975	Research Scholar	University of Dhaka, Dhaka, Bangladesh
1975-1976	Lecturer in Physics	University of Dhaka, Dhaka, Bangladesh
1976-1979	Commonwealth Scholar	University of Bristol, Bristol, England
1979-1982	Assistant Professor	University of Dhaka, Dhaka, Bangladesh
1982-88	Associate Professor	University of Dhaka, Dhaka, Bangladesh
1986-1986	Nuffield Fellow	Royal Society, London, England
1987-1987	IAEA-UNESCO Visiting Scientist	International Centre for Theoretical Physics, Trieste, Italy
1988-1989	Professor	University of Dhaka, Dhaka, Bangladesh
1988-1989	Humboldt Fellow	Alexander von Humboldt Stiftung, Germany
1989-1992	Assistant Professor	Sultan Qaboos University, Muscat, Oman
1992-2001	Associate Professor	Sultan Qaboos University, Muscat, Oman
2001…	Professor	Sultan Qaboos University, Muscat, Oman
2010…	Head of Physics	Sultan Qaboos University, Muscat, Oman

Honors:

Bangladesh Academy of Science Gold Medal for outstanding contribution to research in Theoretical Condensed Matter Physics [1982].

A.R. Chowdhury Gold Medal for outstanding contribution in Physics [1982].

Recent Selected (representative) Publications

1. *Electron Transport in Liquid Na, K, and Rb: t-Matrix Formalism Revisited*, A B Abdellah, K Bouziane, B Grosdidier, S M Mujibur Rahman* and J G Gasser*Physica B*, 405, 4978 (2013) [*ELSEVIER*]
2. *Velocity Profiles for Flow of Omani Crude Oils and Other Liquids,* S Arafin and S M Mujibur Rahman, *SQU Journal for Science*, 19, 87-94 (2014)
3. *Exact Solution for Velocity Profile of Flow of Multilayer Immiscible Liquids*S Arafin and S M Mujibur Rahman, *Phys. Chem. Liq*. DOI:10.1080/00319104.2014.947371 (2014) [*ELSEVIER*].
4. *Fresnel Diffraction from N-Apertures: Computer Simulation by Iterative Fresnel Integrals Method* Kazi M Abedin and S M Mujibur Rahman, *Optik* 126, 3743-3751 (2015) [*ELSEVIER*].
5. *Spin-Sate Dependence of Electrical Resistivity and Thermoelectric Power of Molten Al-Mn Alloys: Experiment and Theory*, A B Abdellah, B Grosdidier, S M Osman, S M Mujibur Rahman, M Mayoufi, J Ataati and J G Gasser, Journal of Alloys and Compounds 658, 1010-1019 (2016) [*ELSEVIER*]. Online version is already in the web: doi.org/10.2016/j.jalicom.201510.27

In: Computer Vision and Simulation
Editor: Sherri Alexander

ISBN: 978-1-63485-790-1

Chapter 6

VISUAL FEEDBACK CONTROL OF A MOBILE ROBOT FOR MECHATRONICS EDUCATION

***Fusaomi Nagata*[1], *Toshiyuki Tatai*[1], *Mamadou Ngom*[1],**
***Akimasa Otsuka*[1] *Maki K. Habib*[2] *and Keigo Watanabe*[3]**
[1]Tokyo University of Science, Yamaguchi, Japan
[2]American University in Cairo, Egypt
[3]Okayama University, Japan

Abstract

Recently, visual feedback control system is becoming more attractive for mechatronics education due to the development of RGB-D cameras such as Kinect and Xtion. In this paper, an example of a simple visual feedback control system of a mobile robot with an axis-symmetric shape is introduced for mechatronics education which has to be demonstrated within a time limit of a lecture. Positions of a robot in image plane and projected plane can be calculated by referring to RGB stream and depth stream obtained from Xtion camera, respectively. As the first simple exercise, a virtual barrier fence is designed, so that a mobile robot can move within the virtual fence even without a real one. In addition, if a mobile robot has an axis-symmetric shape, e.g., circle, from the top view, it is difficult for a vision system to identify the orientation of the robot in the coordinate system. Another exercise is introduced to deal with an orientation following control using a forward direction vector. The forward direction vector can be calculated from point cloud data obtained by making the robot move forward for a short distance, e.g., 30 mm, every dynamic sampling period. The effectiveness and usability of the presented work is demonstrated experimentally.

1. Introduction

In the previous research, the authors developed a network-based subsamption architecture for multiple mobile robots whose memory specifications and software development environments were not sufficient for a large-scaled application [1]. Each robot has six PSD sensors as shown in Figure 1 to detect objects around the body. Recently, visual feedback control system is becoming more attractive for robotics and mechatronics systems because of the appearance of RGB-D cameras such as Kinect and Xtion. First of all, several potential research results using visual feedback are briefly introduced.

For example, Yu et al. developed a system which had standard network protocol and an interactive human-machine interface. An operator could control a mobile robot to navigate in their laboratory while receiving visual feedback information. The designed user interface enabled both researchers and students to control and program mobile robots and to do some interesting experiments from a remote computer [2]. Uchikado et al. coped with a problem concerning navigation of a mobile robot with a camera in indoor environment. A visual feedback control system using a vanishing point of parallel lines at both sides of the corridor was proposed for guidance and obstacle avoidance [3]. Nierobisch et al. developed a novel approach to large view visual servo of a mobile robot with a pan-tilt camera. The rotational, lateral and longitudinal motions are controlled separately by selecting appropriate image features which can decouple the rotational and translational velocity components. This function was effective for traversing a confined indoor space [4]. Then, Slawinski et al. proposed a control scheme for teleoperation of mobile robots with visual feedback in presence of undesirable time-varying delay. The controller was designed using a model of the human operator to combine the velocity command generated by the human operator in a delayed time instant, the received information in such moment, and the current state of the remote site to set the velocity reference for the mobile robot [5]. Lutvica et al. conducted a design and implementation of a remote position control system for a mobile robot. The system was composed of the mobile robot, PC-based positioning controller, camera and wireless communication device using ZigBee. The camera captured images of the mobile robot. Developed image processing algorithms was able to estimate the position and orientation of the robot [6]. Also, Hong et al. reported a visual and force feedback method to enhance a human operator's situational awareness in multi-robot teleoperation environment by fabricating a global view of the multi-robot system and transmitting its velocity information, while using only local information of the robots [7]. Futher, Machida et al. proposed a tracking control system of human motion with Kinect for control of a mobile robot, in which the 3D position information of human obtained from Kinect enabled to control the velocity and attitude of the mobile robot [8]. Furthermore, Wang et al. presented a new controller for the trajectory tracking of nonholonomic mobile robots using visual feedback without direct position measurement. This controller was developed based on the basis of a novel adaptive algorithm for estimating the global position of the mobile robot, in which natural visual features measured by a vision system, its orientation and velocity measured by odometry, and Attitude and Heading Reference System (IMU & Compass) sensors were online used [9].

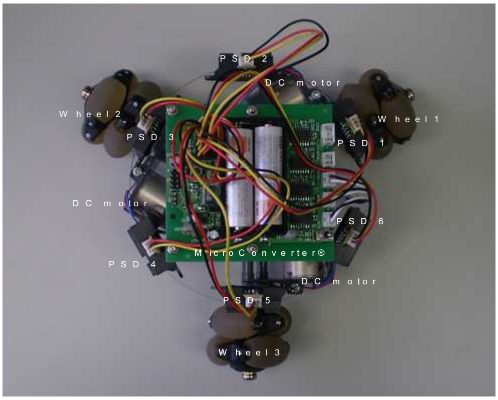


Figure 1. Three wheeled mobile robot with six PSD sensors used for past experiments.

However, it seems that mechatronics education systems using a visual feedback control have not been adequately developed and provided for undergraduate students in mechanical engineering course. In this paper, a simple visual feedback control system of a mobile robot with an axis-symmetric shape is introduced for mechatronics education that should be concluded with a time limit of a lecture. Positions of a robot in image plane and projected plane can be calculated by referring RGB stream and depth stream obtained from a Xtion camera, respectively. As the first simple exercise, a virtual fence system is designed, so that a mobile robot can move within the fence even without a real physical one. In addition, if a mobile robot has an axis-symmetric shape as a circle from the top view, it is not easy for a vision system to identify the orientation of the robot in the coordinate system. Of course, although an axis-asymmetric marker on the mobile robot easily overcome this problem, such a marker is not used due to the problem setting in this paper. Odometry is commonly used as a typical dead reckoning for mobile robots. However, in this test bed, the friction between the table and wheels is very small, so that undesirable slips and measurement errors due to the integration tend to occur. To cope with this problem, an orientation following control is considered using a forward direction vector. The forward direction vector can be calculated from point cloud data obtained by making the robot move forward for a short distance, e.g., 30 mm, every dynamic sampling period. The promise and usability as a mechatronics education system are demonstrated experimentally.

2. Experiment System

Figure 2 shows the experimental setup, in which a three-wheeled mobile robot with no sensors used is controlled by a server PC. A Xtion PRO LIVE camera is taking view of the table top. A table coordinate system $o - xyz$ is fixed on the center of the table. Figure 3 shows the top view of the robot. The main body is provided by TOSADENSHI LTD., on which a micro control unit called MicroConverter is mounted. A simple DC motor without an encoder is built in each wheel, so that the robot has a high cost performance. The robot can communicate with a server PC through Bluetooth and has only to conduct the reflex action command transmitted from the server PC. The server PC, which was developed in the past, transmitted a reflex action command generated by the already proposed network-based subsamption architecture [1]. The nine kinds of reflex action commands are tabulated in Table 1. In this system, the same reflex action commands are used.

As can be guessed, it is not easy for the camera system to identify the robot's orientation ϕ in Figure 2 because of the axis-symmetric shape from the top view. Note that the symmetric shape of the mobile robot is the important problem setting for students.

3. Design of Virtual Fence

The whole work area on the table is given by $x_{\mathrm{lft1}} < x < x_{\mathrm{rht1}}$, $y_{\mathrm{btm1}} < y < y_{\mathrm{top1}}$. $[x\ y\ z]^T$ is the estimated position of the robot mea-

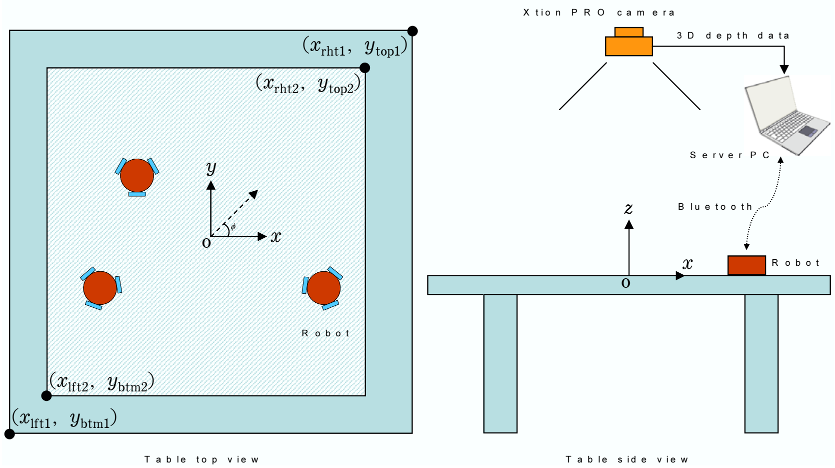

Xtion PRO camera
3D depth data
Server PC
Bluetooth
z
x
o
Robot
$(x_{\mathrm{rht1}},\ y_{\mathrm{top1}})$
$(x_{\mathrm{rht2}},\ y_{\mathrm{top2}})$
y
ϕ
o
x
Robot
$(x_{\mathrm{lft2}},\ y_{\mathrm{btm2}})$
$(x_{\mathrm{lft1}},\ y_{\mathrm{btm1}})$
Table top view
Table side view

Figure 2. Experimental setup, in which the inner rectangle is called the virtual fence.

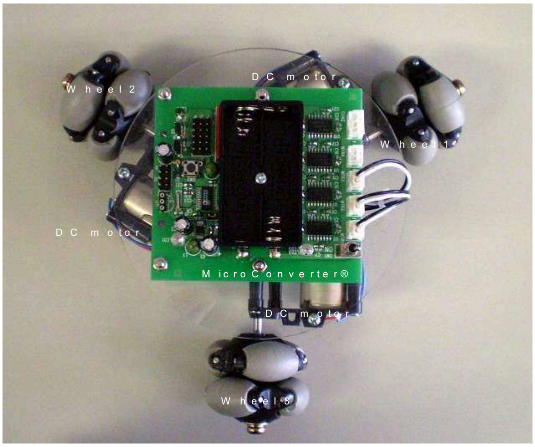


Figure 3. Three wheeled mobile robot with no sensors used in experiments this time.

Table 1. The most simply subdivided reaction behaviors, i.e., reflex actions, for a mobile robot

Cmd. code $c(k)$	**Corresponding reflex actions**
0x30	Halt at the position
0x31	Move to the direction 1
0x32	Move to the direction 2 (Front)
0x33	Move to the direction 3
0x34	Move to the direction 4
0x35	Move to the direction 5 (Rear)
0x36	Move to the direction 6
0x37	Rotate to clockwise direction
0x38	Rotate to counterclockwise direction

sured by the Xtion. A virtual barrier fence is assumed that $x_{\mathrm{left2}} < x < x_{\mathrm{rht2}}$, $y_{\mathrm{btm2}} < y < y_{\mathrm{top2}}$. If the robot moves outside the virtual fence, the server PC generates an opposite directional command and transmits a packet including the command to make the robot return into the area, so that the virtual fence is simply realized. Since the mobile robot takes only the actions shown in the Table 1, a reflex behavior like bounding from a fence can be easily done by generating the opposite directional command according to the last conducted one. The control law of virtual fence is given by

If $(x_{\mathrm{lft2}} < x < x_{\mathrm{rht2}}) \wedge (y_{\mathrm{btm2}} < y < y_{\mathrm{top2}})$ is false,

$$c(k) = \begin{cases} c(k-1) + 3 & \text{if } 0\text{x}30 < c(k-1) < 0\text{x}34 \\ c(k-1) - 3 & \text{if } 0\text{x}33 < c(k-1) < 0\text{x}37 \end{cases} \tag{1}$$

where $c(k)$ is the reflex action command code at discrete time k shown in Table 1. Figures 4 and 5 show the software flowcharts of the reflex action program built-in the mobile robot and the timer interrupt in the virtual fence mode on the server PC, respectively.

In Figure 5, BGR image and point cloud map are first retrieved, then the threshold image of the target mobile robot with red color is extracted from the BGR image. The resolution is 640×480, so that the zero-order moment, first-order ones in x- and y-directions are given as [10]

$$m_{0,0}(k) = \sum_{i=0}^{639} \sum_{j=0}^{479} f_b(i,j) \tag{2}$$

$$m_{1,0}(k) = \sum_{i=0}^{639} \sum_{j=0}^{479} i f_b(i,j) \tag{3}$$

$$m_{0,1}(k) = \sum_{i=0}^{639} \sum_{j=0}^{479} j f_b(i,j) \tag{4}$$

where $f_b(x,y)$ is a binarization function according to a threshold and the inten-

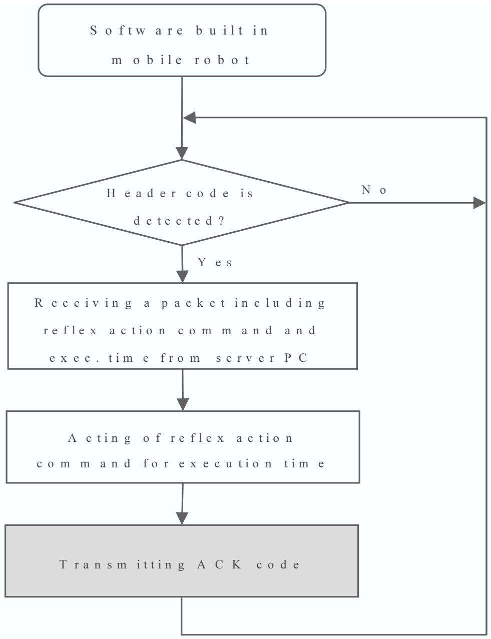

Software built in mobile robot
Header code is detected?
No
Yes
Receiving a packet including reflex action command and exec. time from server PC
Acting of reflex action command for execution time
Transmitting ACK code

Figure 4. Software flowchart of the reflex action program built-in the mobile robot.

sity value $f(x, y)$ as given by

$$f_b(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq threshold \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

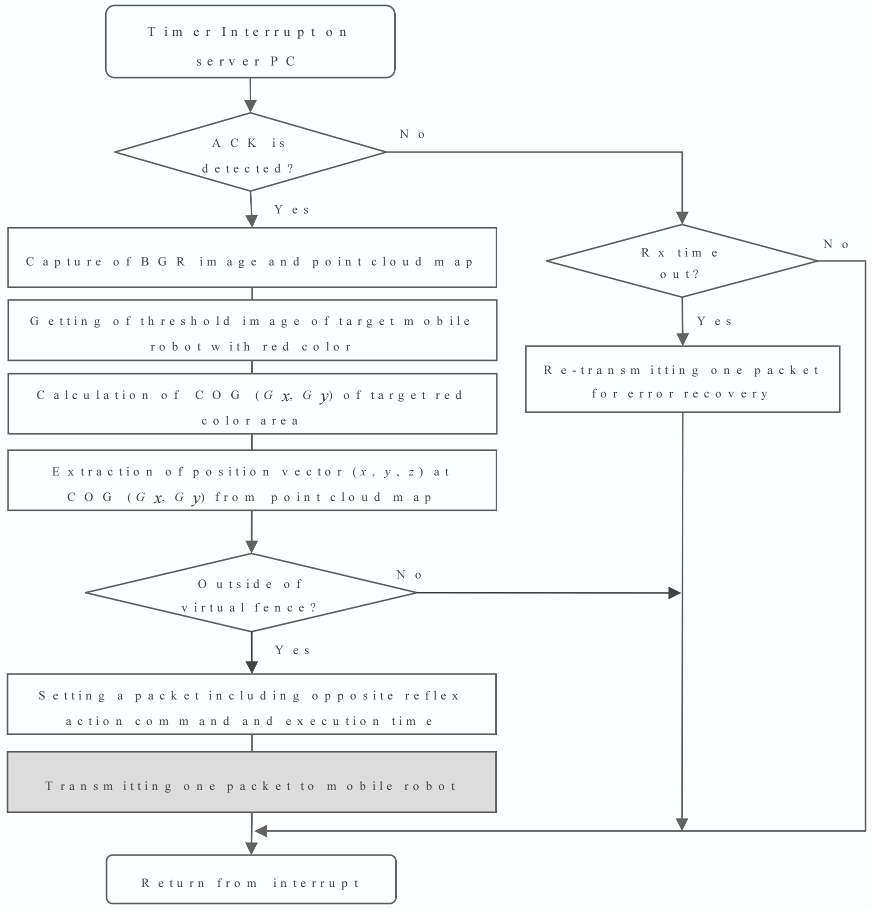

Timer Interrupt on server PC
ACK is detected?
No
Yes
Capture of BGR image and point cloud map
Getting of threshold image of target mobile robot with red color
Calculation of COG (Gx, Gy) of target red color area
Extraction of position vector (x, y, z) at COG (Gx, Gy) from point cloud map
Rx time out?
No
Yes
Re-transmitting one packet for error recovery
Outside of virtual fence?
No
Yes
Setting a packet including opposite reflex action command and execution time
Transmitting one packet to mobile robot
Return from interrupt

Figure 5. Software flowchart of the timer interrupt in the virtual fence mode on the server PC.

The COG $[G_x(k)\ G_y(k)]^T$ in the image of the target robot is calculated by [10]

$$[G_x(k)\ G_y(k)]^T = \left[\frac{m_{1,0}(k)}{m_{0,0}(k)}, \frac{m_{0,1}(k)}{m_{0,0}(k)}\right]^T \tag{6}$$

The position $\boldsymbol{x}(k) = [x(k)\ y(k)\ z(k)]^T$ in table coordinate system corresponding to the COG can be extracted from the point cloud data. In this paper, $\boldsymbol{x}(k)$ is used as the robot's current position estimated by the Xtion. If the robot is outside of the virtual fence, a command packet consisting of a header code of 'S', i.e., 0x53, an opposite reflex action command based on Eq. (1) and an execution time, e.g., 100 ms, are transmitted to the mobile robot.

On the other hand, as shown in Figure 4, the mobile robot is waiting for the command packet with a header code. After receiving the command packet, the robot has only to conduct the reflex action for a constant execution time. A recovery process for communication error, i.e., retransmission of the packet, is equipped in the timer interrupt routine on the server PC. Because undesirable lack of code is easy to occur when the DC motor is running or just after running.

4. Orientation Following Control

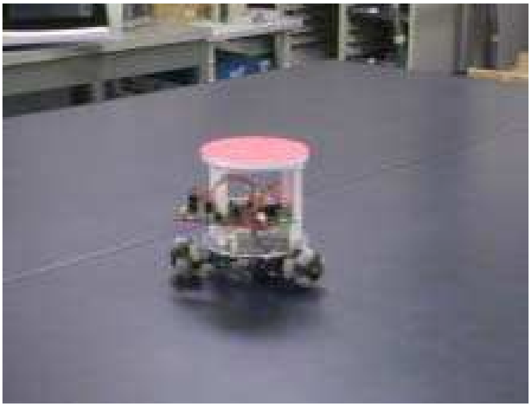


Figure 6. Mobile robot with an axis-symmetric red circular shape from the top view.

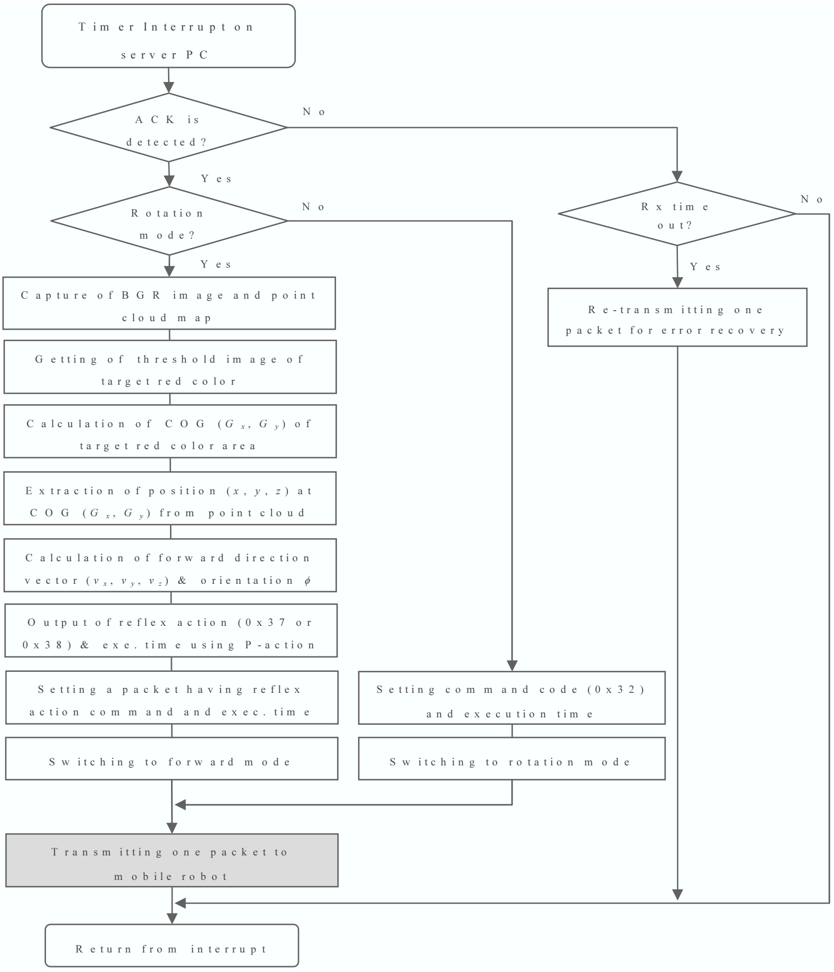

Timer Interrupt on server PC
ACK is detected?
No
Yes
Rotation mode?
No
Yes
Capture of BGR image and point cloud map
Getting of threshold image of target red color
Calculation of COG (G_x, G_y) of target red color area
Extraction of position (x, y, z) at COG (G_x, G_y) from point cloud
Calculation of forward direction vector (v_x, v_y, v_z) & orientation ϕ
Output of reflex action (0x37 or 0x38) & exe. time using P-action
Setting a packet having reflex action command and exec. time
Switching to forward mode
Setting command code (0x32) and execution time
Switching to rotation mode
Transmitting one packet to mobile robot
Return from interrupt
Rx time out?
No
Yes
Re-transmitting one packet for error recovery

Figure 7. Software flowchart of the timer interrupt in the orientation following control mode on the server PC.

It is not easy to estimate the orientation from images captured by a camera in case of the axis-symmetric shape as shown in Figure 6. In this paper, an orientation following control method is designed for the mobile robot with the axis-symmetric shape. The orientation can be estimated by using the direction vector which is measured by making the robot move a little to a direction, e.g., forward direction. By using the position $\boldsymbol{x}(k)$ in table coordinate system measured by the Xtion, the moving direction vector $\boldsymbol{v}(k) = [v_x(k)\ v_y(k)]^T$ is calculated as

$$\boldsymbol{v}(k) = \boldsymbol{x}(k) - \boldsymbol{x}(k-1) \tag{7}$$

Hence, the orientation $\phi(k)$ [rad] of the robot shown in Figure 2 is easily estimated by

$$\phi(k) = \tan^{-1}\frac{v_y(k)}{v_x(k)} \tag{8}$$

Here, let's consider the execution time $|t_\phi(k)|$ calculated by a simple P-action given by

$$t_\phi(k) = K_\phi\{\phi_d(k) - \phi(k)\} \tag{9}$$

where $\phi_d(k)$ is the desired orientation of the robot.

Figures 7 shows the software flowchart of the timer interrupt routine in the orientation following control mode on the server PC. The server PC transmits a command packet consisting of one byte header code and two bytes command vector $\boldsymbol{C}_{md}(k) = [c(k)\ \ |t_\phi(k)|]^T$ to the mobile robot. Note that, in the mobile robot side, the same built-in program shown in Figure 4 is used, so that the mobile robot has only to conduct $c(k)$ for $|t_\phi(k)|$. After executing a reflex action, the mobile robot returns ACK code 'S', i.e., 0x52 to the server PC as shown in Figure 4. As can be seen from Figure 7, the forward mode and the rotation mode are alternately switched. The switching period depends on the time required for the handshake process between the program of the mobile robot shown in Figure 4 and the timer interrupt of the server PC shown in Figure 7. In this paper, the switching period is called the dynamic sampling period.

The forward mode is important to steadily generate the direction vector given by Eq. (7). In the forward mode, $c(k)$ has a value of 0x32. On the other hand $c(k)$ has a value of 0x37 or 0x38 when the orientation is controlled by the server PC in the rotation mode. The control law in the rotation mode on the server PC is represented by

$$\boldsymbol{C}_{md}(k) = \begin{cases} [\text{0x38}\ \ t_\phi(k)]^T & \text{if } t_\phi(k) \geq 0 \\ [\text{0x37}\ \ -t_\phi(k)]^T & \text{otherwise} \end{cases} \tag{10}$$

Finally, experiments were conducted to evaluate the effectiveness and usability of the designed system. Figure 8 shows the experimental scenes, in which the red-colored line is the robot's actual trajectory and the white-colored arrow is the final controlled orientation. As can be seen, the mobile robot could successfully follow the two desired orientations, i.e., ϕ_d =90° and 45°, by alternately acting the forward mode and the rotation mode as shown in Figure 7.

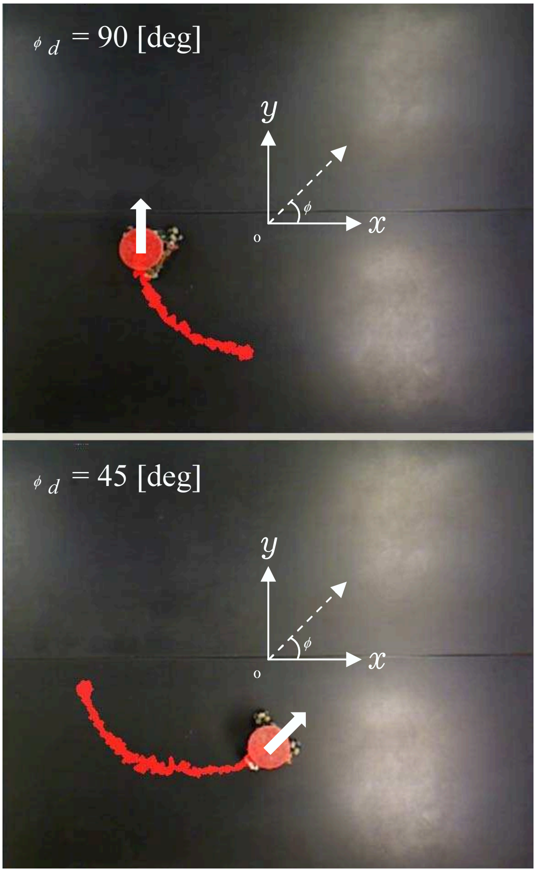


Figure 8. Experimental results of orientation following control.

Conclusion

In this paper, a simple visual feedback control system of a mobile robot with an axis-symmetric shape has been introduced for mechatronics education which has to be concluded successfully within a time limit of a lecture. Positions of a robot in image plane and projected plane could be calculated by referring RGB stream and depth stream obtained from a Xtion camera, respectively. As an exercise, a virtual barrier fence system was designed, so that a mobile robot could move around within the fence without using any sensors. Furthermore, an orientation following control using a forward direction vector is presented for a mobile robot with an axis-symmetric shape from top view. The forward direction vector could be generated from the point cloud data obtained by making the robot move forward for a short distance, e.g., 30 mm, every dynamic sampling period. The effectiveness and usability were demonstrated experimentally.

The authors are planning to apply the proposed mechatronics education system to an experimental lecture for the third year students, department of mechanical engineering, Faculty of Engineering, Tokyo University of Science, Yamaguchi. It is expected that students will be able to learn the basic application technique of RGB-D camera such as Kinect and Xtion through experiential learning.

References

[1] F. Nagata, A. Otsuka, K. Watanabe, M.K. Habib, "Network-based subsumption architecture for broadcast control of multiple mobile robots based on a poor hardware/software platform," Procs. of the 14th International Symposium on Advanced Intelligent Systems, pp. 231–247, 2013.

[2] L. Yu, P.W. Tsui, Q. Zhou, H. Hu, "A Web-based telerobotic system for research and education at Essex," Procs. of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 37–42, 2001.

[3] S. Uchikado, S. Lili, M. Nagayoshi, "New visual feedback control design about guidance of a mobile robot using vanishing point," Procs. of IEEE IECON 2004, pp. 627–632, 2004.

[4] T. Nierobisch, W. Fischer, F. Hoffmann, "Large view visual servoing of a mobile robot with a pan-tilt camera," Procs. of 2006 IEEE/RSJ Inter-

national Conference on Intelligent Robots and Systems, pp. 3307–3312, 2006.

[5] E. Slawinski, V. A. Mut, J.F. Postigo, "Teleoperation of mobile robots with time-varying delay," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1071–1082, 2007.

[6] K. Lutvica, N. Kadic, G. Dzampo, H. Muminovic, J. Velagic, N. Osmic, "Remote position control of mobile robot based on visual feedback and ZigBee communication," Proceedings of ELMAR2011, pp. 169–172, 2011.

[7] A. Hong, H.H. Bulthoff, H.I. Son, "A visual and force feedback for multi-robot teleoperation in outdoor environments: A preliminary result," Procs. of 2013 IEEE International Conference on Robotics and Automation (ICRA2013), pp. 1471–1478, 2013.

[8] E. Machida, C. Meifen, T. Murao, H. Hashimoto, "Human motion tracking of mobile robot with Kinect 3D sensor," Procs. of SICE Annual Conference (SICE2012), pp. 2207–2211, 2012.

[9] K. Wang, Y. Liu, L. Li, "Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 1026–1035, 2014.

[10] S. Dey, "A simple architecture for computing moments and orientation of an image," *Fundamenta Informaticae*, vol. 52, no. 4, pp, 285–295, 2002.

RELATED NOVA PUBLICATIONS

COMPUTER VISION

Sota R. Yoshida

ISBN: 978-1-61209-399-4 Publication Date: 2011

Computer vision is the science and technology of machines that see, where seeing in this case means that the machine is able to extract information from an image that is necessary to solve some task. This new book presents topical research in the study of computer vision, including computer vision systems in micromechanics; genetic algorithm-based face recognition; algebraic topology for computer vision and computer vision by laser metrology and algorithms of artificial intelligence.

COMPUTER SIMULATIONS: TECHNOLOGY, INDUSTRIAL APPLICATIONS AND EFFECTS ON LEARNING

Boris Nemanjic and Navenka Svetozar

ISBN: 978-1-62257-580-0 Publication Date: 2012

The use of computers in simulating real-world situations has received widespread interest and attention. Computer simulations have widely been used in industrial applications, fundamental research and in computer-aided education and visualization. In this book, the authors present current research in the study of the technology, applications and effects on learning of computer simulations. Topics include computer modeling of rearrangement in liquid phase sintering; generating near-field fresnel diffraction patterns by Iterative Fresnel Integrals Method; computer simulation of the ion beam modification and analysis of single crystal surfaces under grazing incidence conditions; computer simulation of carbon and carbon-metal nanostructures; computation of residue curves using Mathematica and MATLAB; educational computer simulations of lightning and associated thunders; and designing energy efficient buildings using computer simulation.

COMPUTATIONAL TECHNIQUES IN MODELING AND SIMULATION

Victor Krasnoproshin, Anna M. Gil Lafuente and Constantin Zopounidis

ISBN: 978-1-62808-017-9 Publication Date: 2013

The present volume is concerned with Computational Techniques in Modeling and Simulation. Initially, there are four research papers and apart from those, the rest of the research papers are coming from the International Conference on Modeling and Simulation (MS’2012), held in Minsk, 2-4 May 2012.

MS'12 was co-organized by the AMSE Association and the Belarusian State University in cooperation with other scientific establishments: the Belarusian Academy of Sciences (United Institute of Problems of Informatics), the Belarusian Society of International Association of Pattern Recognition and the International Association for Fuzzy Set Management and Economy (SIGEF, Spain). The Conference was sponsored by the Belarusian State University and Byelex Multimedia Products BV (The Netherlands). It offered a unique opportunity for researchers, professionals and students to share ideas concerning modeling, simulation and implementation of the results in the real world.

We would like to thank all contributors, referees, honorary committees for their co-operation within MS'12, in particular: Jaime Gil Aluja (AMSE President), Sergey Ablameyko (the Rector of the Belarusian State University) as honorary chairmen.

INDEX