

Vol. 25, No. 1/4, 2016

Machine
GRAPHICS & VISION

International Journal

Published by
The Faculty of Applied Informatics and Mathematics – WZIM
Warsaw University of Life Sciences – SGGW
Nowoursynowska 159, 02-776 Warsaw, Poland

in cooperation with
The Association for Image Processing, Poland – TPO

ACOUSTIC CARPETS

Maciej Janowicz, Joanna Kaleta, Piotr Wrzeciono, Andrzej Zembrzusi

Faculty of Applications of Informatics and Mathematics – WZIM

Warsaw University of Life Sciences – SGGW

Nowoursynowska 159, 02-775 Warsaw, Poland

<http://www.wzim.sggw.pl>

Abstract. Initial-boundary value problem for linear acoustics has been solved in two spatial dimensions. It has been assumed that the initial acoustic field consists of two Gaussian distributions. Dirichlet boundary conditions with zero acoustic pressure at the boundaries have been imposed. The solution has been obtained with the help of a split-operator technique which resulted in a cellular automaton with uncountably many internal states. To visualize the results, the Python library `matplotlib` has been employed. It has been shown that attractive graphical output results in both the transient and stationary regimes. The visualization effects are similar to, but different from, the well-known quantum-mechanical carpets.

Key words: acoustics, Euler equations, split-operator method, visualization of physical fields.

1. Introduction

As is well known, interference effects omnipresent in the propagation of both linear and non-linear physical fields, very often lead to formation of beautiful patterns. Let us only mention here caustic patterns which can be in coffee cup due to the multiple reflection of light rays with the cup serving as a kind of quasi-cylindrical mirror. Many interesting and important physical effects are known in quantum mechanics due to the interference of probability waves. This includes formation of sometimes spectacular patterns emerging in the propagation of probability waves in an infinite potential well [2, 3, 4, 5]. Very interesting and beautiful images have been obtained by simulation of the dynamics of quantum mechanical wavepackets provided that generic initial conditions have been assumed. By *generic* initial conditions we mean those with many modes being initially excited. Interesting revivals of patterns have been reported.

Needless to say, one can and should expect appearance of similarly interesting structures in other physical fields, for instance, electromagnetic or acoustic. The role of quantum-mechanical infinite-well potential is naturally played by perfectly reflecting walls. It is to be mentioned that such walls actually do not exist. Nevertheless, a model with perfectly reflecting walls is a convenient starting point and can provide a reasonable approximation if we do not consider the fields too close to such mirrors.

In this work we concentrate on the case of evolution of acoustic fields in two spatial dimensions. The main body of the paper is organized as follows. In Section 2 we provide

Euler's partial differential equations which constitute the mathematical model of our system. In Section 3 a simple but efficient algorithm to integrate those equations is worked out. Section 4 contains a series of figures in which our results are illustrated. Some concluding remarks can be found in Section 5.

2. The model

We are about to solve the following system of the partial differential equations [1]:

$$\frac{\partial p}{\partial \tau} + \frac{\partial v_x}{\partial \xi} + \frac{\partial v_y}{\partial \eta} = 0, \quad (1)$$

$$\frac{\partial v_x}{\partial \tau} + \frac{\partial p}{\partial \xi} = 0, \quad (2)$$

$$\frac{\partial v_y}{\partial \tau} + \frac{\partial p}{\partial \eta} = 0. \quad (3)$$

All quantities in the above system, including the independent variables, have been made dimensionless. The quantity p has the physical meaning of the acoustic pressure while v_x and v_y are two components of the field of velocity of particles. The variable τ is the dimensionless time while ξ and η are dimensionless coordinates in the plane.

The above system describes propagation of the acoustic field in two spatial dimensions. In this work we will employ the simple Dirichlet boundary conditions: $p = 0$ on any boundary.

Equations (1)-(3) can be represented in a matrix form:

$$\frac{\partial}{\partial \tau} \begin{pmatrix} p \\ v_x \\ v_y \end{pmatrix} = \widehat{\mathbf{M}} \begin{pmatrix} p \\ v_x \\ v_y \end{pmatrix}, \quad (4)$$

where the matrix operator $\widehat{\mathbf{M}}$ is equal to

$$- \begin{pmatrix} 0 & \frac{\partial}{\partial \xi} & \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \xi} & 0 & 0 \\ \frac{\partial}{\partial \eta} & 0 & 0 \end{pmatrix}. \quad (5)$$

In the following section this matrix notation will be used to develop a split-operator algorithm for numerical computations of (p, v_x, v_y) .

3. Split-operator algorithm

Equations (1)-(3) can be formally integrated to give:

$$\begin{pmatrix} p(\tau + \Delta\tau, \xi, \eta) \\ v_x(\tau + \Delta\tau, \xi, \eta) \\ v_y(\tau + \Delta\tau, \xi, \eta) \end{pmatrix} = \exp(\widehat{\mathbf{M}}\Delta\tau) \begin{pmatrix} p(\tau, \xi, \eta) \\ v_x(\tau, \xi, \eta) \\ v_y(\tau, \xi, \eta) \end{pmatrix}. \tag{6}$$

Let us write the matrix $\widehat{\mathbf{M}}$ as a sum of four terms:

$$\widehat{\mathbf{M}} = \widehat{\mathbf{M}}_1 + \widehat{\mathbf{M}}_2 + \widehat{\mathbf{M}}_3 + \widehat{\mathbf{M}}_4, \tag{7}$$

where

$$\widehat{\mathbf{M}}_1 = \widehat{\mathbf{M}}_3 = -\frac{1}{2} \begin{pmatrix} 0 & \frac{\partial}{\partial \xi} & 0 \\ \frac{\partial}{\partial \xi} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{8}$$

$$\widehat{\mathbf{M}}_2 = \widehat{\mathbf{M}}_4 = -\frac{1}{2} \begin{pmatrix} 0 & 0 & \frac{\partial}{\partial \eta} \\ 0 & 0 & 0 \\ \frac{\partial}{\partial \eta} & 0 & 0 \end{pmatrix}. \tag{9}$$

Let us notice here that the matrices with even and odd indices do not commute.

For small $\Delta\tau$ we can approximately “split” the matrix exponential $\exp(\widehat{\mathbf{M}}\Delta\tau)$ as follows [6, 7, 8]

$$\exp(\widehat{\mathbf{M}}\Delta\tau) \approx \prod_{n=1}^4 \exp(\widehat{\mathbf{M}}_n\Delta\tau). \tag{10}$$

The following remark is necessary here. In our numerical simulations $\Delta\tau$ is actually not small: it is exactly equal to 2. However, it is sufficient to require that $\Delta\tau$ be small when compared to characteristic lengths introduced by the initial conditions (or boundary conditions, or, in a more general settings, by frequencies of the external currents). In our case $\Delta\tau$ will be significantly smaller than the half-widths of Gaussians which form the initial conditions.

Taking into account that

$$\begin{aligned} \exp(\widehat{\mathbf{M}}_1 \Delta\tau) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cosh\left(\frac{\Delta\tau}{2} \frac{\partial}{\partial\xi}\right) + \\ &- \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \sinh\left(\frac{\Delta\tau}{2} \frac{\partial}{\partial\xi}\right) + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned} \quad (11)$$

$$\begin{aligned} \exp(\widehat{\mathbf{M}}_2 \Delta\tau) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cosh\left(\frac{\Delta\tau}{2} \frac{\partial}{\partial\eta}\right) + \\ &- \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \sinh\left(\frac{\Delta\tau}{2} \frac{\partial}{\partial\eta}\right) + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \end{aligned} \quad (12)$$

as well as considering the simple action of hyperbolic sine and cosine functions of first-order operators on any function $f(\xi, \eta)$, e.g.:

$$\cosh\left(\frac{\Delta\tau}{2} \frac{\partial}{\partial\eta}\right) f(\xi, \eta) = \frac{1}{2} \left[f\left(\xi, \eta + \frac{\Delta\tau}{2}\right) + f\left(\xi, \eta - \frac{\Delta\tau}{2}\right) \right], \quad (13)$$

$$-\sinh\left(\frac{\Delta\tau}{2} \frac{\partial}{\partial\eta}\right) f(\xi, \eta) = \frac{1}{2} \left[f\left(\xi, \eta - \frac{\Delta\tau}{2}\right) - f\left(\xi, \eta + \frac{\Delta\tau}{2}\right) \right], \quad (14)$$

we end up with a simple four-step algorithm to integrate (1)-(3). For instance, the first step is obtained upon using the exponential of $\widehat{\mathbf{M}}_4$:

$$\begin{aligned} p^{(1)}(\xi, \eta) &= \frac{1}{2} \left[p\left(\tau, \xi, \eta + \frac{\Delta\tau}{2}\right) + p\left(\tau, \xi, \eta - \frac{\Delta\tau}{2}\right) \right] + \\ &- \frac{1}{2} \left[v_y\left(\tau, \xi, \eta + \frac{\Delta\tau}{2}\right) - v_y\left(\tau, \xi, \eta - \frac{\Delta\tau}{2}\right) \right], \end{aligned} \quad (15)$$

$$v_x^{(1)}(\xi, \eta) = v_x\left(\tau, \xi, \eta\right), \quad (16)$$

$$\begin{aligned} v_y^{(1)}(\xi, \eta) &= -\frac{1}{2} \left[p\left(\tau, \xi, \eta + \frac{\Delta\tau}{2}\right) - p\left(\tau, \xi, \eta - \frac{\Delta\tau}{2}\right) \right] + \\ &\frac{1}{2} \left[v_y\left(\tau, \xi, \eta + \frac{\Delta\tau}{2}\right) + v_y\left(\tau, \xi, \eta - \frac{\Delta\tau}{2}\right) \right]. \end{aligned} \quad (17)$$

In the second step, using exponential of $\widehat{\mathbf{M}}_3$ we obtain:

$$p^{(2)}(\xi, \eta) = \frac{1}{2} \left[p^{(1)}\left(\xi + \frac{\Delta\tau}{2}, \eta\right) + p^{(1)}\left(\xi - \frac{\Delta\tau}{2}, \eta\right) \right] + \frac{1}{2} \left[v_x^{(1)}\left(\xi + \frac{\Delta\tau}{2}, \eta\right) - v_x^{(1)}\left(\xi - \frac{\Delta\tau}{2}, \eta\right) \right], \tag{18}$$

$$v_x^{(2)}(\xi, \eta) = -\frac{1}{2} \left[p^{(1)}\left(\xi + \frac{\Delta\tau}{2}, \eta\right) - p^{(1)}\left(\xi - \frac{\Delta\tau}{2}, \eta\right) \right] + \frac{1}{2} \left[v_x^{(1)}\left(\xi + \frac{\Delta\tau}{2}, \eta\right) + v_x^{(1)}\left(\xi - \frac{\Delta\tau}{2}, \eta\right) \right], \tag{19}$$

$$v_y^{(2)}(\xi, \eta) = v_y^{(1)}(\xi, \eta), \tag{20}$$

and so on.

We have discretized our simulation cell, being a square, as $\xi = m\Delta\xi$, $\eta = n\Delta\eta$, $m, n = 0, 1, \dots, N - 1$, and set $\Delta\xi = \Delta\eta = 1$, $\Delta\tau = 2$. Thus, the fields p, v_x, v_y have become functions of discrete variables m and n . The boundary conditions $p = 0$ result in the following prescription for $v_x(0, n)$, $v_x(N - 1, n)$, $v_y(m, 0)$, $v_y(m, N - 1)$:

$$\begin{aligned} v_x(0, n) &= -p(1, n) + v_x(1, n), \\ v_x(N - 1, n) &= p(N - 2, n) + v_x(N - 2, n), \\ v_y(m, 0) &= -p(m, 1) + v_y(m, 1), \\ v_y(m, N - 1) &= p(m, N - 2) + v_y(m, N - 2). \end{aligned}$$

Let us notice that the algorithm per se is unitary and it preserves the energy exactly. However, the numerical boundary conditions slightly spoil the energy conservation so that it exhibits oscillations with amplitudes up to 0.5%.

The above algorithm while being simply a particular version of finite-difference method, also forms a kind of cellular automaton with uncountably many states. Similar automata have also been constructed e.g. in [9, 10, 11].

4. Numerical results

In all our numerical simulations the size N of the computational cell has been equal to 1000. The initial conditions have been of the form:

$$p(m, n) = 10 \sum_{i=0}^1 \exp(-((m - m_i)^2 + (n - n_i)^2)/(2\sigma^2))/(2\pi\sigma^2)$$

with $m_0 = n_0 = N/4$, $m_1 = n_1 = 3N/4$. The initial velocity has been assumed to vanish everywhere. We have performed simulations for several values of σ .

The the dependence of $|p|$ on (ξ, η) for various τ and for $\sigma = 20$ has been shown in Figure 1, and further on, in Figure 2 for $\sigma = 60$.

In addition, we have also performed simulations with vanishing initial conditions but with point-like external current in the form of an additional boundary condition

$$p(m_0, n_0) = p_0 \sin(\nu\tau),$$

with $p_0 = 1$ and $\nu = 0.01$ (thus, $\Delta\tau \ll 1/\nu$). The results are shown in Figure 3.

Let us ask an important question whether we can draw any conclusions from the images in Figures 1-3. The answer is, naturally, affirmative. Just by looking at these images we may infer that any “geometroptical”, i.e. ray-tracing approach to the system must immediately result in a failure. The interference effects kill the well-defined wave packets and the “carpet” structure arises. On the other hand, the initial presence of small-width wavepackets leads to a fine, granular structure of the carpet. Larger widths provide larger and better visible structures in the square.

5. Conclusion

In this paper we have provided a simple yet efficient cellular-automaton-like, unitary algorithm to integrate Euler’s equations of linear acoustics. Numerical simulations of the dynamics of acoustic fields starting with two Gaussian wavepackets have been carried out inside a square with perfectly reflecting walls. What is more, similar simulations have been performed for the fields with zeroth initial condition but under the presence of an external source with sinusoidal time variation. Our results have been illustrated with the help of the Python library `matplotlib`. We believe they form a valuable supplement to what is known about interference in linear field structures evolving inside very high-quality resonators.

Let us notice that visualization has, of course, already become a standard tool of acoustic analysis. Let us quote only recent papers [12, 13, 14] in this connection.

We plan to investigate further similar systems with special emphasis on the presence of vortices and other topological structures in the field.

References

- [1] Morse P. *Vibrations and Sound*. McGraw-Hill, New York, 1948.
- [2] Marzoli I., Saif F., Białynicki-Birula I., Friesch, O.M., Kaplan, A.E., and Schleich W.P. Quantum carpets made simple. *Acta Phys.Slov.* **48**:323-333, 1998
- [3] Hall M.J.W., Reineker M.S. and Schleich W.P. Unravelling quantum carpets: a travelling wave approach. *J. Phys. A* **32**:8275-8291, 1999.
- [4] Kaplan A.E., Marzoli I., Lamb Jr., W.E., and Schleich W.P. Multimode interference: Highly regular pattern formation in quantum wave-packet evolution. *Phys. Rev. A* **61**:032101, 2000.

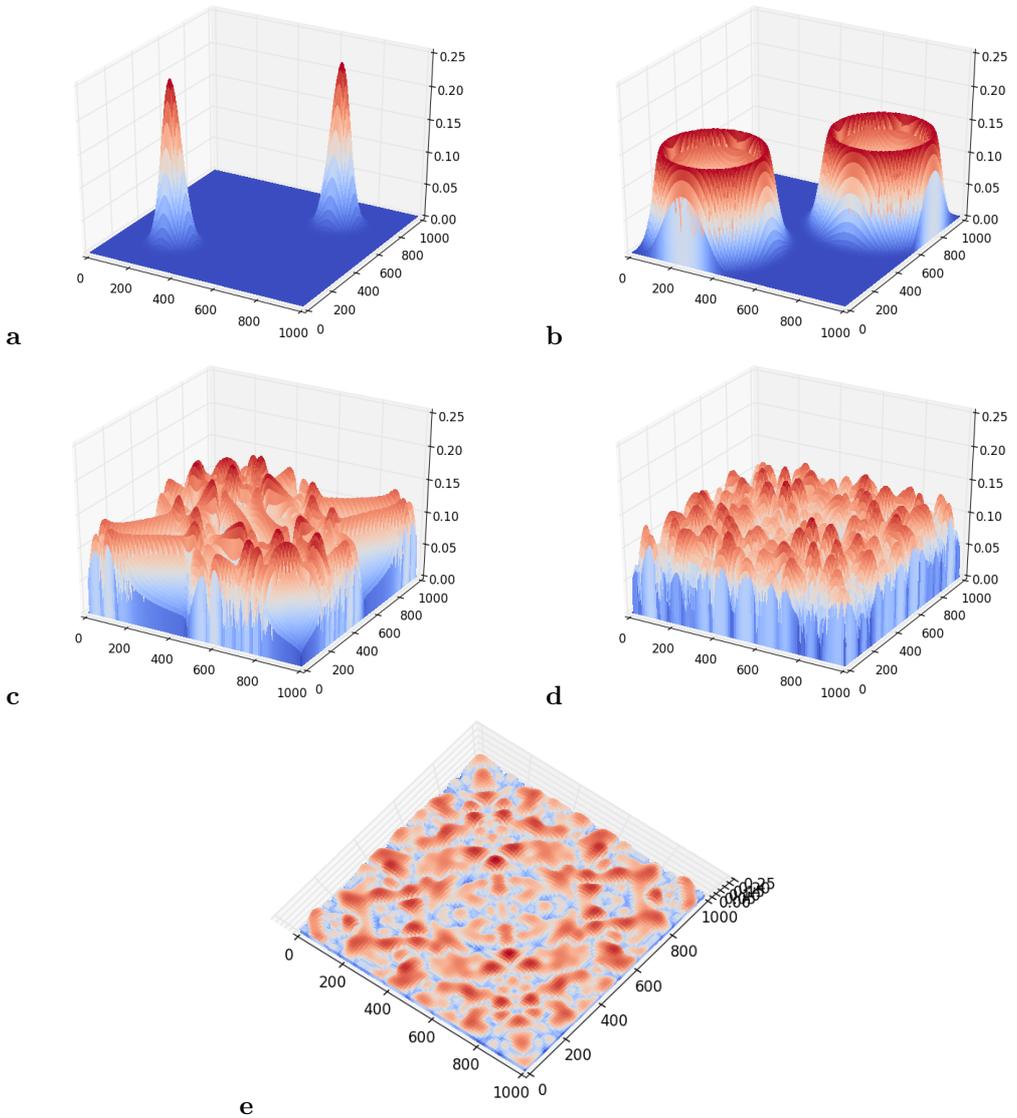


Fig. 1. Snapshots of the time evolution of acoustic field from two Gaussian wavepackets for $\sigma = 20.0$ (see text). The quantity $|p|^{1/4}$ is plotted as a function of ξ and η for several τ . (a) $\tau = 1$; (b) $\tau = 100$; (c) $\tau = 500$; (d) $\tau = 5000$; (e) same as d but at a different projection.

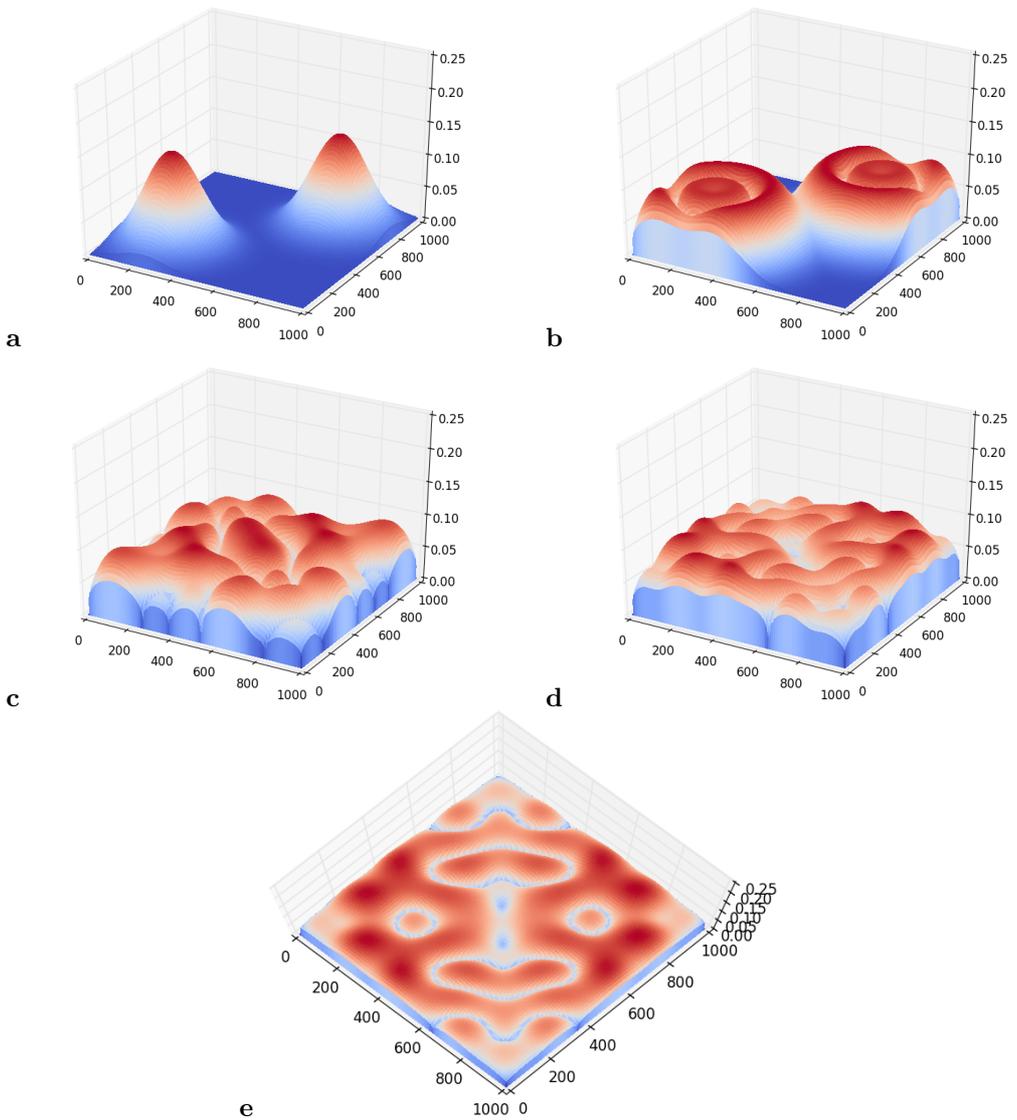


Fig. 2. Snapshots of the time evolution of acoustic field from two Gaussian wavepackets for $\sigma = 60.0$ (see text). The quantity $|p|^{1/4}$ is plotted as a function of ξ and η for several τ : (a) $\tau = 1$, (b) $\tau = 100$, (c) $\tau = 500$, (d) $\tau = 5000$. (e) Same as d but at a different projection.

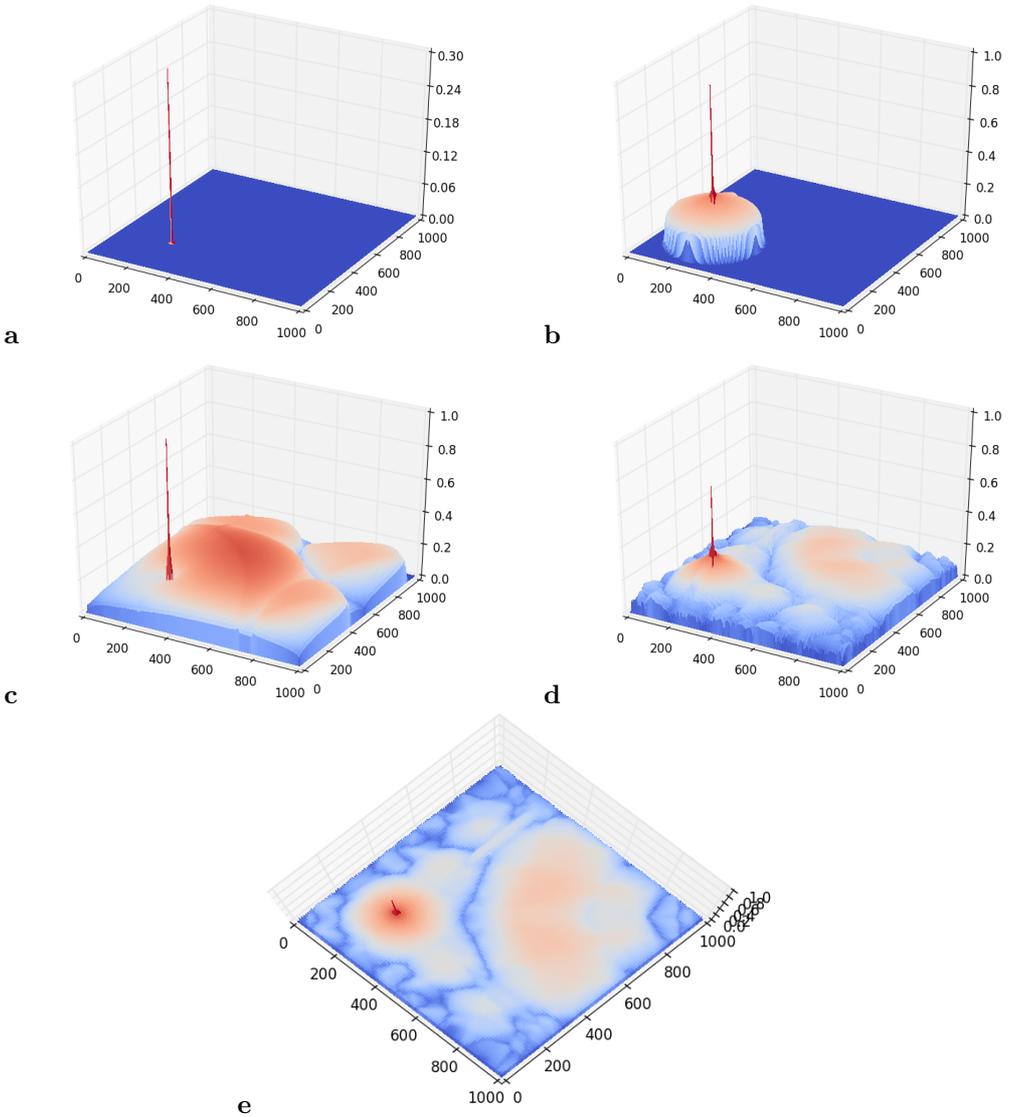


Fig. 3. Snapshots of the time evolution of acoustic field with zeroth initial conditions but in the presence of external current (see text). The quantity $|p|^{1/4}$ is plotted as a function of ξ and η for several τ : (a) $\tau = 1$, (b) $\tau = 100$, (c) $\tau = 500$, (d) $\tau = 5000$. (e) Same as d but at a different projection.

- [5] Belloni M., Doncheski M.A., and Robinett R.W. Wigner quasi-probability distribution for the infinite square well: Energy eigenstates and time-dependent wave packets. *Am. J. Phys.* **72**:1183-1192, 2004.
- [6] Trotter H.F. On the product of semi-groups of operators. *Proc. Am. Math. Soc.* **10**:545-551, 1959.
- [7] Feit M.D., Fleck Jr. J.A., and Steiger A. Solution of the Schrödinger equation by a spectral method. *J. Comp. Phys.* **47**:412-433, 1982.
- [8] Suzuki M. Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics. *J. Math. Phys.* **26**:601-612, 1985.
- [9] Białynicki-Birula I. Weyl, Dirac, and Maxwell equations on a lattice as unitary cellular automata. *Phys. Rev. D* **49**:6920, 1994.
- [10] Succi S. *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*. Clarendon Press, Oxford, 2001.
- [11] Janowicz, M.W., Ashbourn J.M.A., Orłowski A. and Mostowski J. Cellular automaton approach to electromagnetic wave propagation in dispersive media. *Proc. Roy. Soc. London Ser. A* **462**:2927-2948, 2006.
- [12] Miyasaka C., Telschow K.L., Tittmann B.R., Sadler J.T., and Park I.K. Direct visualisation of acoustic waves propagation within a single anisotropic crystalline plate with hybrid acoustic imaging system. *Research in Nondestructive Evaluation* **23**:197-206, 2012.
- [13] Szczodrak M., Kurowski A., Kotus J., Czyżewski A., and Kostek B. A system for acoustic field measurement employing Cartesian robot. *Metrol. Meas. Syst.* **23**:333-343, 2016.
- [14] Rambach R.W., Taiber J., Scheck C.M.L., Meyer C., Reboud J., Cooper J.M., and Franke T. Visualization of Surface Acoustic Waves in Thin Liquid Films. *Sci. Rep.* **6**:21980, 2016. DOI:10.1038/srep21980.

VISUALIZATION OF NONLOCALITY IN COUPLED MAP LATTICES

Maciej Janowicz, Joanna Kaleta, Piotr Wrzeciono,
Andrzej Zembrzuski, Arkadiusz Orłowski
Faculty of Applications of Informatics and Mathematics – WZIM
Warsaw University of Life Sciences – SGGW
Nowoursynowska 159, 02-775 Warsaw, Poland
<http://www.wzim.sggw.pl>

Abstract. Numerical simulations of coupled map lattices with various degree of nonlocality have been performed. Quantitative characteristics of recently introduced for local coupling have been applied in the nonlocal case. It has been attempted to draw qualitative conclusions about nonlocality from the emerging pictures.

Key words: coupled map lattices, nonlocality, density matrix, visualization.

1. Introduction

Coupled map lattices (CML) [1, 2] have long become a valuable theoretical tool to investigate pattern formation effects, chaos in many-mode systems, cooperative effects and synchronization [3, 4, 5, 6]. Some of them have proved to be a valuable tool in modeling physical processes [7, 8, 9, 10, 11].

Some of the most characteristics introduced to study various CMLs are: co-moving Lyapunov spectra, mutual information flow, spatiotemporal power spectra, Kolmogorov-Sinai entropy density, pattern entropy [10]. Also, the detrended fluctuation analysis, structure function analysis, local dimensions, embedding dimension and recurrence analysis have also been applied to CMLs [11].

More recently, research in the field of systems of coupled nonlinear oscillators (including CML) have been, among other things, connected with the so-called chimera states [12, 13, 14, 15], associated with nonlocality occurring in such systems.

In the work of two of us [16], some additional characteristics of CMLs have been introduced. In particular, the reduced density matrix, the *wave function of CML* (being the eigenvector corresponding to the dominant eigenvalue of the reduced density matrix), which can serve as an order parameter and the *number of particles* which have been shown to be useful quantitative properties of the system.

In this work we concentrate on the case of evolution of the coupled map lattices with nonlocality in two spatial dimensions. Our main concern is to find qualitative information about the system by studying its visualization. Therefore, we do not employ

sophisticated measures. We rather concentrate upon the pictures and their properties because they also reflect the properties of the reduced density matrices.

The main body of the paper is organized as follows. In Section 2 we describe our model. In Section 3 the important parameters and characteristics are defined. The Section 4 contains a series of figures which contain our basic results. Some concluding remarks can be found in Section 5.

2. The model

Let f be a quadratic function which defines the logistic map

$$f(x) = cx(1 - x) \quad (1)$$

and let ξ and η be discrete spatial variables, $\xi \in [0, N - 1]$, $\eta \in [0, N - 1]$. In this work we shall consider a lattice of coupled maps with the following evolution in the discrete time τ :

$$\psi(\tau + 1, \xi, \eta) = (1 - 4Pd)f(\psi(\tau, \xi, \eta)) + d \sum_{k=-P}^P \sum_{l=-P}^P f(\psi(\tau, \xi + k, \eta + l)), \quad (2)$$

where the sums over k and l are taken mod N , so that periodic boundary conditions are imposed. From the above definition of the dynamics of the field ψ in the discrete time τ it follows immediately that P is a degree of nonlocality. The larger P , the more maps contribute to the calculations of the next (in τ) value of a given $\psi(\xi, \eta)$. On the other hand, the parameter d measures the *strength* of the influence of neighboring maps. It can be considered as a kind of a *diffusion constant*.

Our choice of the function f is, in a sense, the standard one. It is connected with the very well known striking properties of the simple quadratic map which exhibits qualitatively very different behavior for varying c . Here, we have chosen $c = 3.8$ (this choice corresponds to the chaotic behavior of the single map in time) and three different values for P and d . We have taken $P = 4, 16$ and 64 . The values of d have been such that the product $4Pd = 0.4$ has been kept constant.

3. Quantitative characteristics of coupled map lattices

In this Section we introduce some quantitative characteristics of our CML: the reduced density matrix, and entropy. The latter, however, though interesting per se, has a marginal meaning in this work.

Let us start with the definition of the reduced density matrix. We begin with the quantity $\bar{\rho}(\xi, \xi', \tau, T)$:

$$\bar{\rho}(\xi, \xi', \tau, T) = \langle \psi(\tau, \xi, \eta) \psi(\tau, \xi', \eta) \rangle_T, \quad (3)$$

where T is the time over which averaging is performed, and $T \leq \tau$. Then the reduced density matrix is normalized to have trace equal to 1:

$$\rho(\xi, \xi', \tau, T) = \bar{\rho}(\xi, \xi', \tau, T) / \sum_{\xi} \bar{\rho}(\xi, \xi, \tau, T). \tag{4}$$

In the above equations the sharp bracket denotes averaging in the following sense:

$$\langle (\cdot) \rangle_T = \frac{1}{T} \sum_{t=\tau-T}^{\tau} (\cdot) \tag{5}$$

We define the (von Neumann) entropy of the CLS in the standard way:

$$S = - \sum_{\xi} \rho(\xi, \xi, \tau, T) \log(\rho(\xi, \xi, \tau, T)).$$

The entropy, defined in this way, is apparently a function of τ and T . We shall comment on this point below.

4. Numerical results

In all our numerical simulations the size N of the computational cell has been equal to 500. The initial conditions have been of the form:

$$\psi(\xi, \eta) = 0.5 \cdot \delta_{\xi, N/2}, \delta_{\eta, N/2}.$$

We have conducted simulations for $\tau = 10, 100, 1000$ and 10000 . These values have to be considered rather small, and our results show, in general, the transient, not stationary, regime.

In Figures 1-3 we have shown the snapshots of the dependence of ψ on (ξ, η) for several τ and $P = 4, 16$ and 64 . Those figures have been produced from `.ppm` files. The contents of the red (R), green (G) and blue (B) colors at each pixel (i, j) have been computed as $R = G = B = 256 - \lfloor 256 \psi(i, j) \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function. Then the resulting `.ppm` file has been converted to `.png` with the GIMP program.

Although the grayscale figures have their obvious advantages, giving immediate indications, where a quantity is large and where it is small, more information about the spatial structure of a solutions is provided by the colored figures. We have created the following color figures as follows. Again, `.ppm` files have been first created with the help of the following formulae for the contents of the three basic colors at the pixel (i, j) :

$$\begin{aligned} R &= \lfloor 256 \times 0.5(1 + \sin(20 \psi(i, j))) \rfloor, \\ G &= \lfloor 256 \times 0.5(1 + \cos(30 \psi(i, j))) \rfloor, \\ B &= \lfloor 256 \times 0.5(1 - \sin(25 \psi(i, j))) \rfloor. \end{aligned}$$

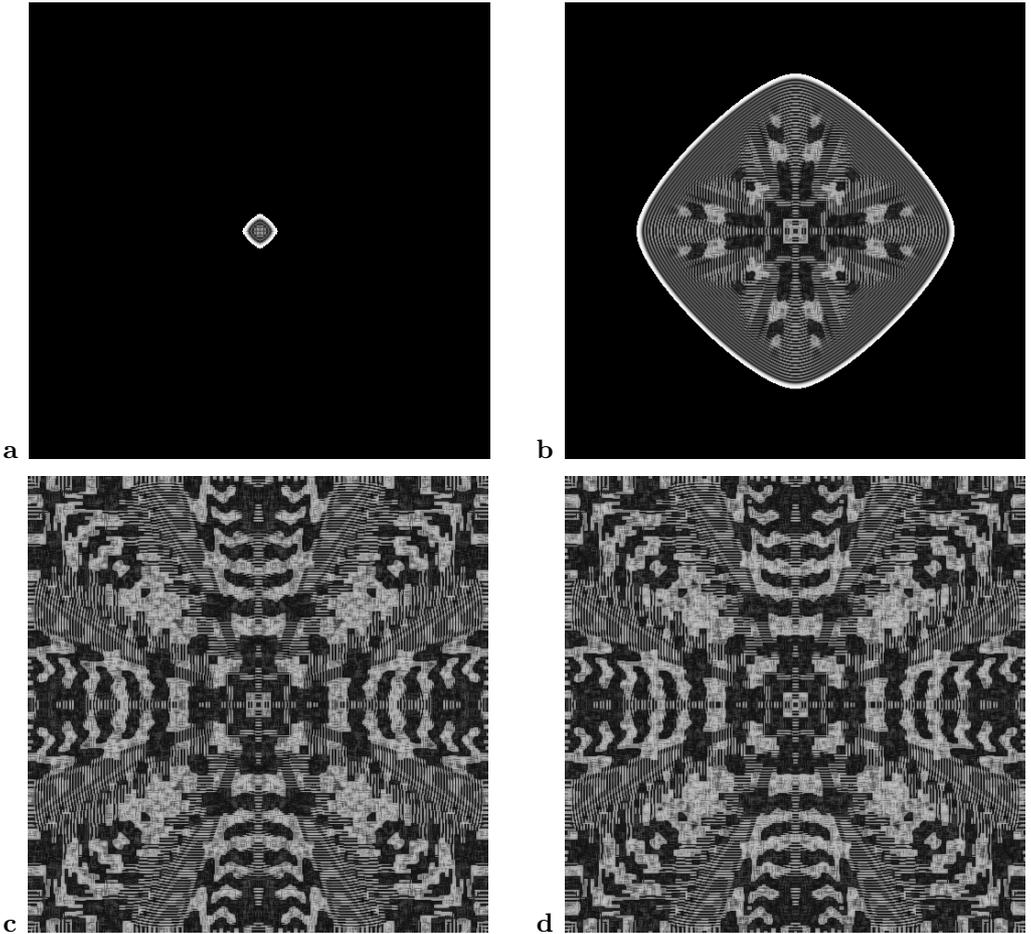


Fig. 1. Grayscale shaded graphics representing the field ψ for $P = 4$ (please see the the main tekst). The quantity ψ is displayed as a function of ξ and η for several τ . Brighter regions correspond to smaller values of ψ (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

The results of such coloring are shown in Figures 4-6.

In Figures 7-9 the reduced density matrices as functions of (ξ, ξ') for various τ have been displayed. The averaging has been performed over time in $[\tau/2 + 1, \tau]$ (that is, $T = \tau/2$). The contents of the red (R), green (G) and blue (B) colors at each pixel (i, j)

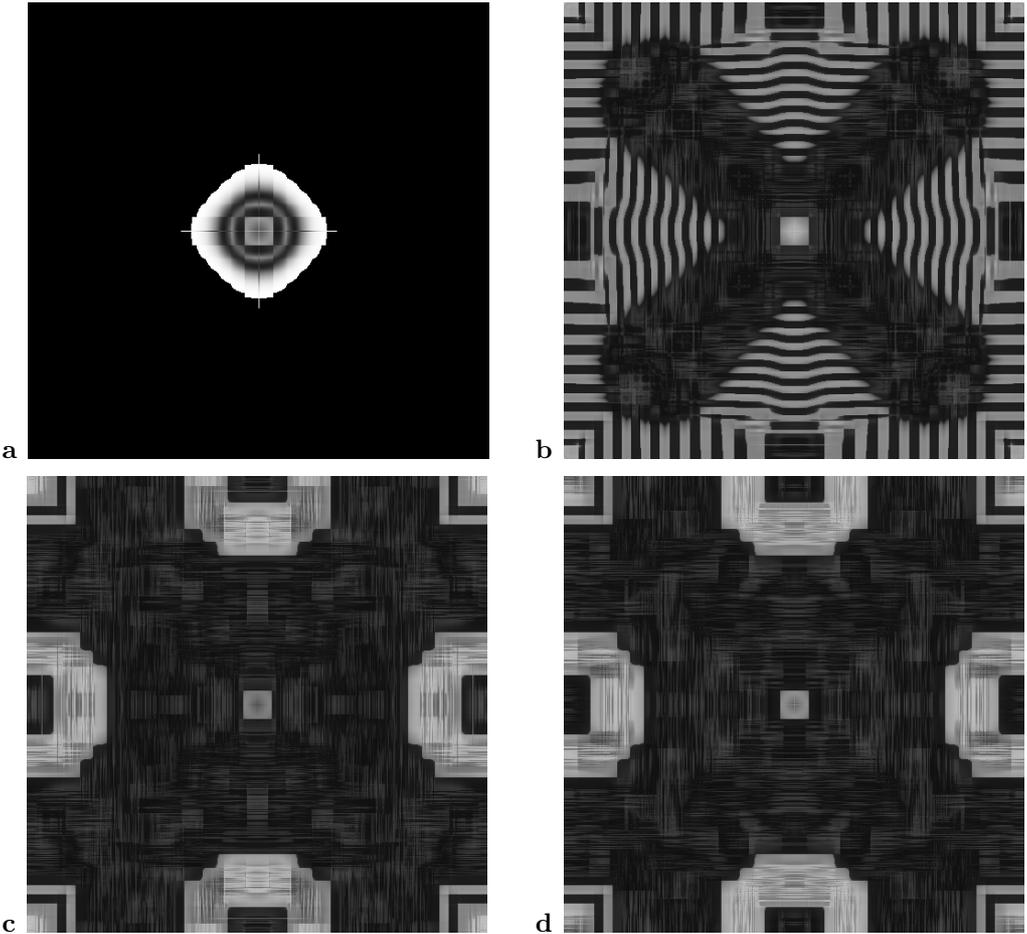


Fig. 2. The same as in 1 but for $P = 16$. (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

have been computed as $R = G = B = \lfloor 256 \rho(i, j) \max(\rho) \rfloor$. The GIMP has again been used to transform `.ppm` to `.png` files.

Let us start a qualitative analysis of the pictorial representation of the results with two trivial remarks. Firstly, all figures clearly reflect the symmetry of the computational cell as well as the anisotropy of the system, with the horizontal and vertical directions clearly distinguished. Secondly, the larger nonlinearity, the faster non-zero values of ψ

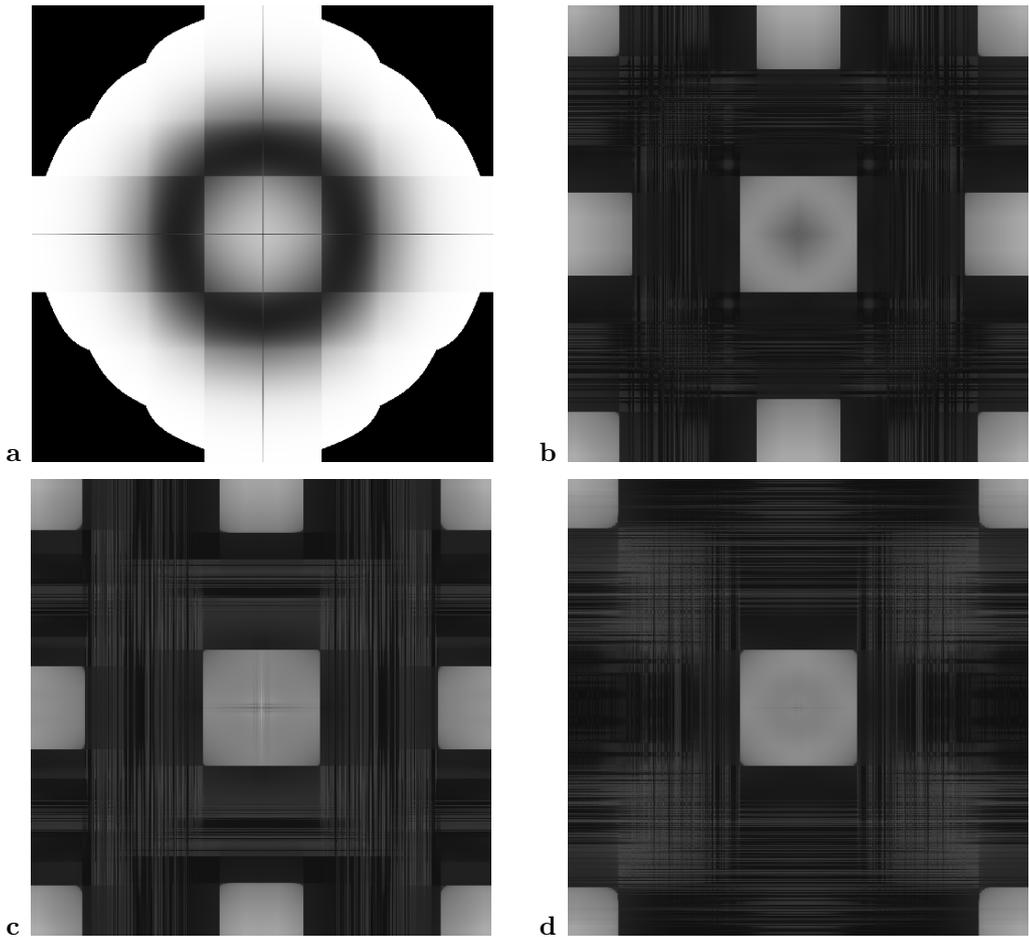


Fig. 3. The same as in 1 but for $P = 64$. (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

cover the computational cell. What we can also immediately guess from the pictures is the number of nontrivial structures in system, which clearly evolve in a different way. Again, that number is clearly associated with P . That is, just having a look at the figure, one can guess, upon some experience, approximate magnitude of nonlinearity. It is particularly interesting to observe a kind of a discontinuity in the pictures: the regions with varying dynamical patterns break the symmetry.

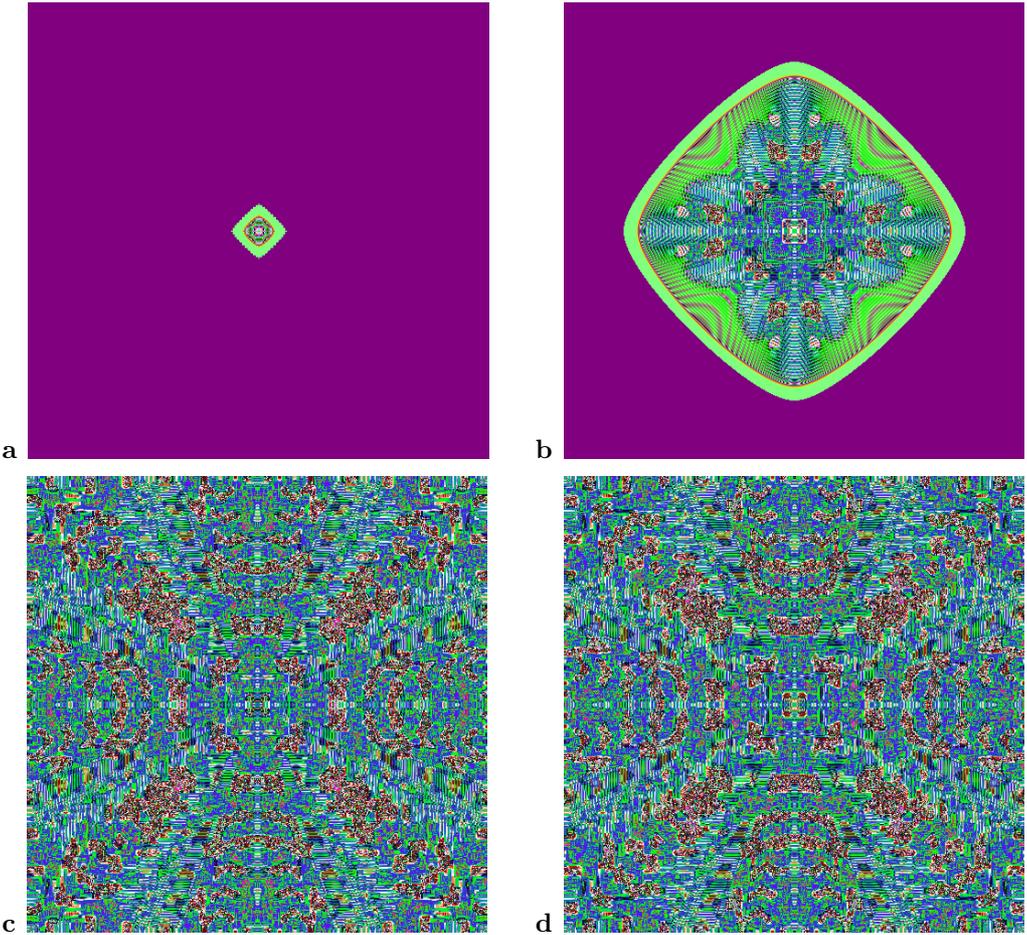


Fig. 4. Snapshots of the time evolution of the field ψ for $P = 4$ (please see the main tekst). The quantity ψ is displayed as a function of ξ and η for several τ .
 (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

As for the pictures illustrating the reduced density matrix let us observe the prominent visibility of both main diagonals. This feature seems to disappear for larger values of P . Square- and rectangle-like structure are clearly able to survive averaging even for large values of τ and T .

While the concept of entropy has played a minor role in this work, we would like

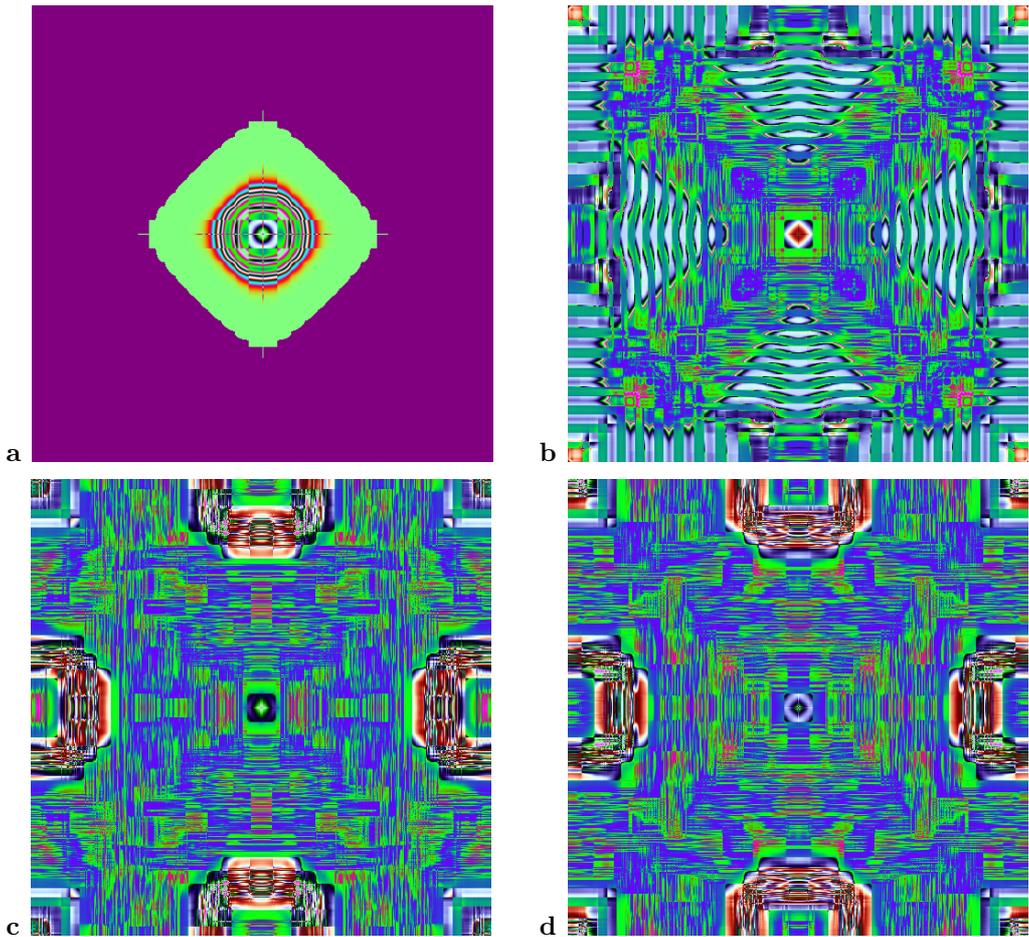


Fig. 5. Snapshots of the time evolution of the field ψ for $P = 16$ (please see the main tekst). The quantity ψ is displayed as a function of ξ and η for several τ .

(a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

to report that it appears to be independent of averaging time T for sufficiently large τ (i.e., larger than a few hundreds). Also, it appears to become asymptotically constant in τ , and independent of P . It is an interesting question to what extent it is independent of the parameters of the system, the initial and boundary conditions, and a specific function f used to define individual maps. Certainly it does depend on the size of the system.

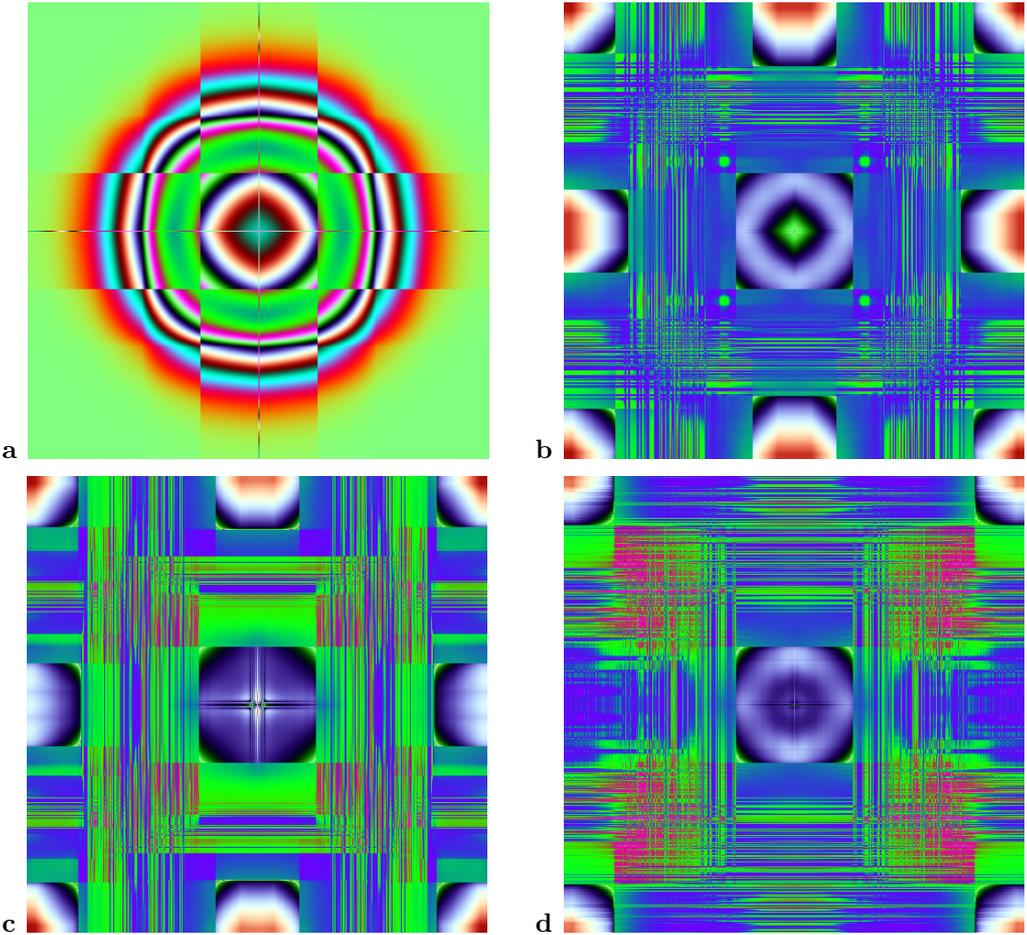


Fig. 6. Snapshots of the time evolution of the field ψ for $P = 64$ (please see the main tekst). The quantity ψ is displayed as a function of ξ and η for several τ .

(a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

5. Conclusion

To conclude, we have performed numerical simulations of nonlocal coupled map lattices. We have demonstrated usefulness of visualization to draw qualitative conclusions about the nonlocal nature of interactions in the system. In particular, it is to some extent possible to guess the degree of nonlocality just by looking at the figures. The larger

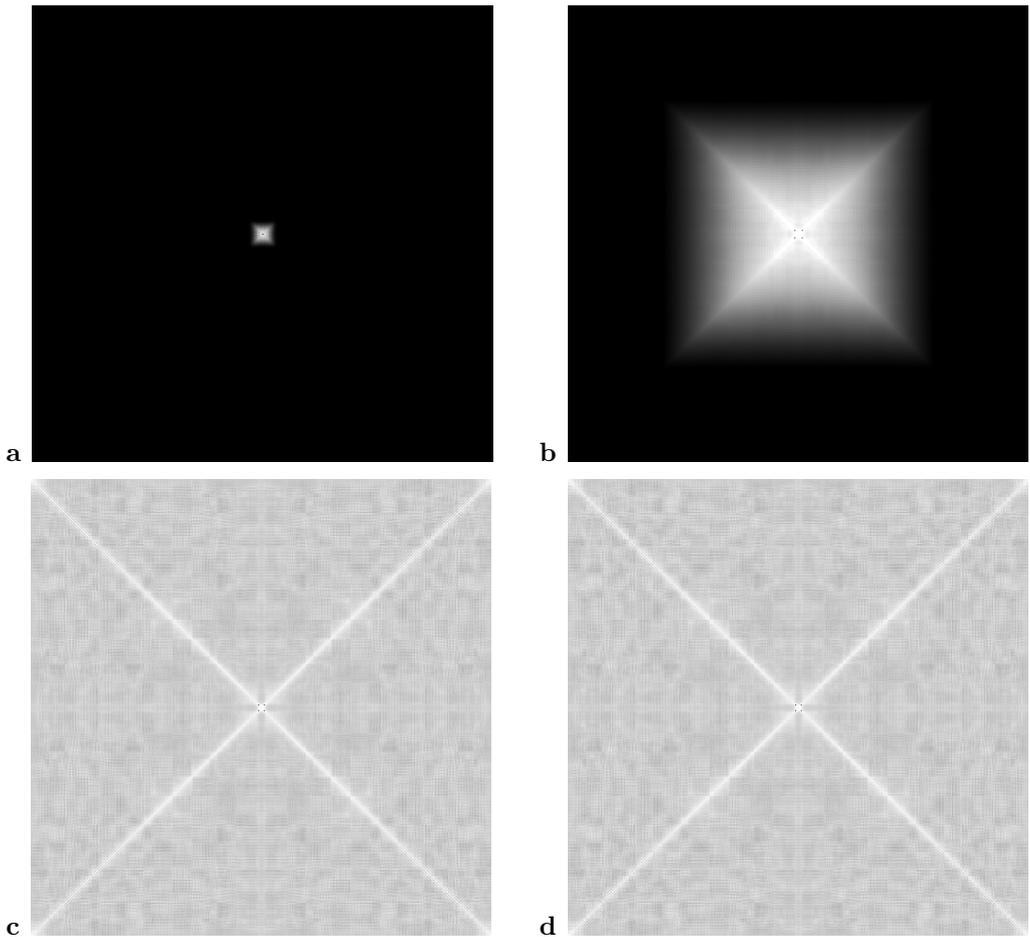


Fig. 7. Grayscale shaded graphics representing the reduced density matrix ρ for $P = 4$ (please see the main tekst). The quantity ρ is displayed as a function of ξ and ξ' for several τ . Brighter regions correspond to larger values of ρ
 (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

degree of nonlocality, the larger amount of interesting, spatially separated structures. Let also mentioned aesthetic appeal of some of the color pictures obtained from the nonlocal maps. Finally, remarkable asymptotic stability of entropy has been observed.

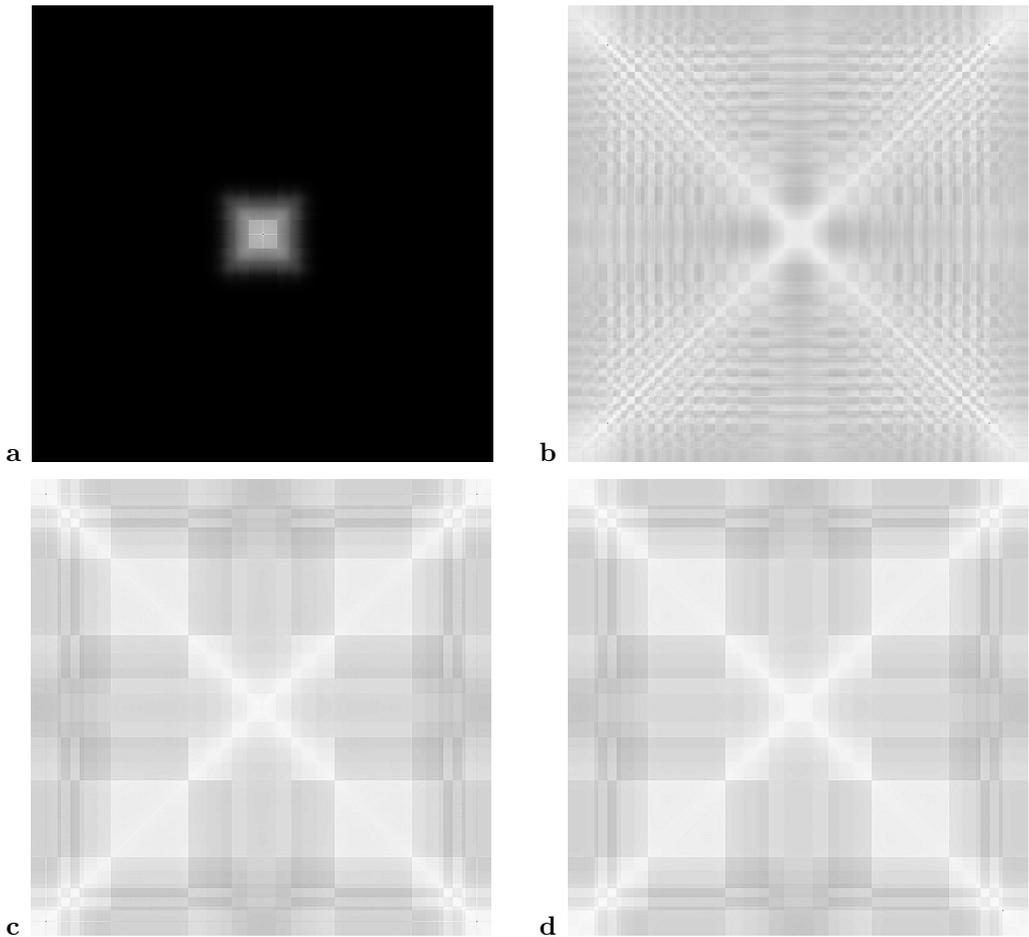


Fig. 8. The same as in Figure 7 but for $P = 16$
 (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

References

- [1] *Dynamics of Coupled Map Lattices and Related Spatially Extended Systems*. J.R. Chazottes and B. Fernandez (Eds.), Springer, New York, 2005.
- [2] Ilachinski A. *Cellular Automata. A Discrete Universe*. World Scientific, Singapore 2001.
- [3] Kaneko K. Period-doubling of kink-antikink patterns, quasiperiodicity in antiferro-like structures

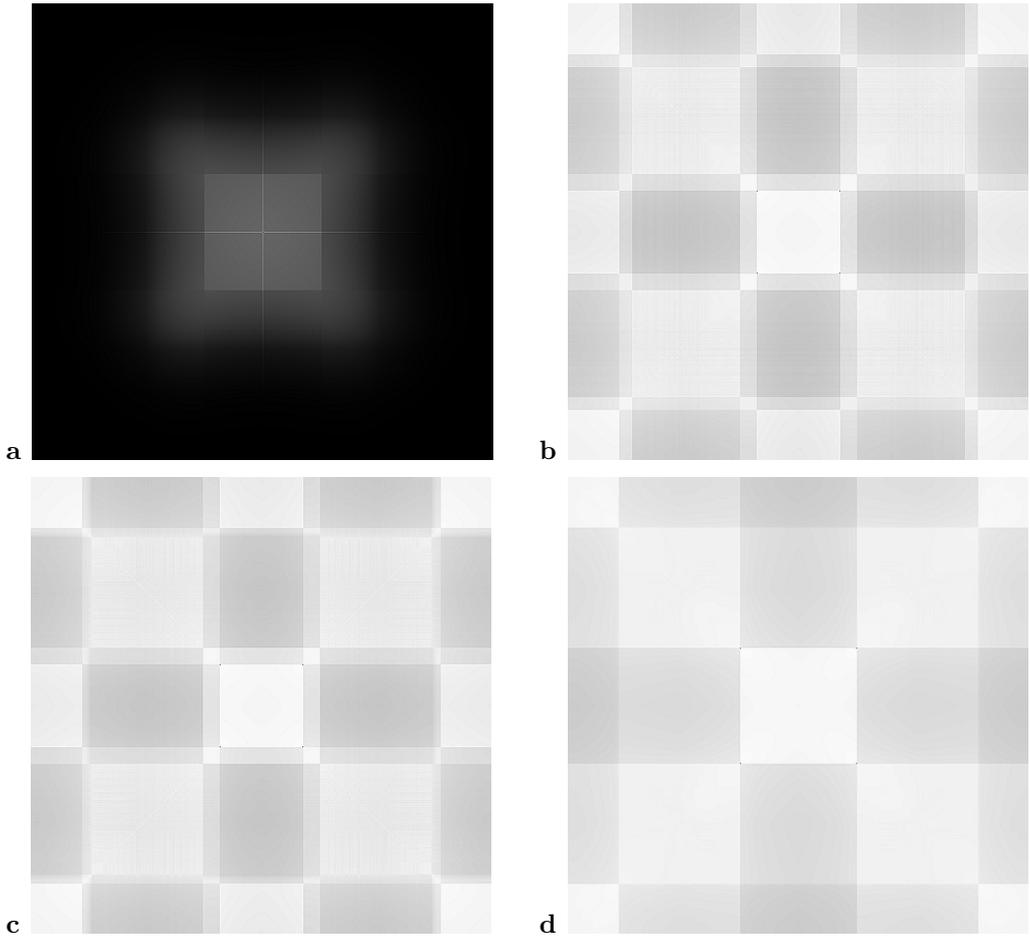


Fig. 9. The same as in Figure 7 but for $P = 64$.
 (a) $\tau = 10$; (b) $\tau = 100$; (c) $\tau = 1000$; (d) $\tau = 10000$.

and spatial intermittency in coupled logistic lattice: towards a prelude of a “Field theory of chaos”.
Prog. Theor. Phys. **72**:480-486, 1984.

[4] Waller I. and Kapral R. Spatial and temporal structure in systems of coupled nonlinear oscillators.
Phys. Rev. A **31**:2047-2055, 1984.

[5] Kapral R. Pattern formation in two-dimensional arrays of coupled, discrete-time oscillators. *Phys. Rev. A* **31**:3868-3879, 1985.

[6] Kaneko K. Pattern dynamics in spatiotemporal chaos: Pattern selection, diffusion of defect and

- pattern competition intermittency. *Physica D* **34**:1-41, 1989.
- [7] Yanagita T. and Kaneko K. Rayleigh-Bénard convection patterns, chaos, spatiotemporal chaos and turbulence. *Physica D* **82**:288-313, 1995.
- [8] Yanagita T. Coupled map lattice model of boiling *Phys. Lett. A* **165**:405-408, 1992.
- [9] Yanagita T. and Kaneko K. Modeling and characterization of cloud dynamics *Phys. Rev. Lett.* **78**:4297-4300, 1997.
- [10] Kaneko K. In: Pattern Dynamics, Information Flow, and Thermodynamics of Spatiotemporal Chaos. K. Kawasaki, A. Onuki, and M. Suzuki (Eds.), World Scientific, Singapore 1990.
- [11] Muruganandam P., Francisco F., de Menezes M., and Ferreira F.F. *Chaos, Solitons and Fractals* **41**:997, 2009.
- [12] Abrams D.M. and Strogatz S.H. Chimera states for coupled oscillators. *Phys. Rev. Lett.* **93**:174102, 2004.
- [13] Omelchenko I., Maistrenko Y., Hövel P. and Schöll E. Loss of coherence in dynamical networks: spatial chaos and chimera states. arxiv:1102.4709v2 [nlin.AO]
- [14] Panaggio M.J. and Abrams D.M. Chimera states: coexistence of coherence and incoherence in networks of coupled oscillators. *Nonlinearity* **28**:R67-R87, 2015.
- [15] Maistrenko Y., Sudakov. O., Osiv. O, and Maistrenko V. Chimera states in three dimensions. *New J. Phys.* **17**:073037, 2015.
- [16] Janowicz M. and Orłowski A. Coherence and large-scale pattern formation in coupled logistic-map lattices via computer algebra systems. In *Computer Algebra in Scientific Computing. CASC 2014*, V. Gerdt, W. Koepf and W.M. Seiler (Eds.), pp. 230-241. Lecture Notes in Computer Science, vol. 8660, 2014.

IMPROVED GENDER CLASSIFICATION USING DISCRETE WAVELET TRANSFORM AND HYBRID SUPPORT VECTOR MACHINE

Jakub Walczak, Adam Wojciechowski

Institute of Information Technology, Łódź University of Technology

ul. Wólczańska 215, 90-924 Łódź, Poland

<http://it.p.lodz.pl>

Abstract. Gender recognition, across different races and regardless of age, is becoming an increasingly important technology in the domains of marketing, human-computer interaction and security. Most state-of-the-art systems consider either highly constrained conditions or relatively large databases. In either case, often not enough attention is paid to cross-racial age-invariant applications. This paper proposes a method of hybrid classification, which performs well even with a small training set. The design of the classifier enables the construction of reliable decision boundaries insensitive to an aging model as well as to race variation. For a training set consisting of one hundred images, the proposed method reached an accuracy level of 90%, whereas the best method known from the literature, tested under the restrictions imposed on the database, achieved only 78% accuracy.

Key words: computer vision, gender recognition, DWT, SVM.

1. Introduction

The ability to correctly determine gender is important for both inter-human communication and advanced human-computer interfaces, which tend to simulate real life interaction. Knowing the gender balance of an audience is a valuable information in the field of marketing, allowing message content to be adjusted so that advertisements can reach their targets more effectively. Algorithms for gender recognition need to be reliable under difficult conditions, since they are frequently employed in places where good facial exposition is rare. In this paper, we present an improved method of gender classification – robust against facial expressions, race and age. An additional advantage is that in the classifications based on a small training data set the method outperforms the current state-of-the-art solutions.

In our research, we used a randomly selected set of 200 subjects (100 samples for training and 100 for testing), from the FERET database [1, 2] including a range of races, ages and facial expressions (see fig. 1). During the investigation, a 100×100 pixel box was used to limit the visible area of each face.

While most of the solutions in the literature ([3, 4, 5]) provide high classification accuracy due to the large training datasets used, our method focuses on a combined



Fig. 1. Exemplary faces used in the project

classification model applicable even to small training sets (like in our case, 100 training samples).

2. Related work

An approach to gender recognition developed by Nazir *et al.* in [3] used the Discrete Cosine Transform to extract image traits. The collected features were then classified using the k-Nearest Neighbors (kNN) algorithm. This method attained accuracy of 99.3% with the Stanford University medical student frontal facial images database (SUMS) of 400 images, 200×200 pixels in size. The drawback of these studies is that the training set comprised good quality and well-lit subjects of approximately the same age. The application of the kNN classifier also does not seem to be a good solution for larger datasets, since all objects must be stored, making the method demanding in terms of memory.

Another method has been suggested by Alrashed *et al.* [5] who used eye region images processed either with the Discrete Cosine Transform or the Discrete Wavelet Transform. The features obtained were classified using a Support Vector Machine with a Radial Basis Function kernel. The accuracy of the systems reached 99.62% for DCT and 99.49% for DWT. Tests were performed on two databases: *Faces94* [6] and *AR database* [7, 8]. The data set comprised 691 images of males and 692 of females. Subjects with sunglasses were excluded, making classification easier. In addition, the first dataset contained images captured under similar lighting conditions.

Singh *et al.* in [4] proposed a method for feature extraction based on Local Binary Patterns and the Histogram of Oriented Gradients [9]. A Support Vector Machine was employed for classification. The accuracy level reached 89.43% for LBP and 95.56% for HOG for *Indian face database* [10]. Although this study took into account variations in lighting, it considered solely the subjects of South Asian ethnicity – focusing on a narrow range of human appearances.

It is worth noting that each of the aforementioned methods relied on the immensity of information provided by large datasets rather than providing a well-generalized gender model based on classification of carefully chosen features. In this paper, we present



Fig. 2. Example of ambiguous face according to our method

a novel method of gender recognition which combines multiple classifiers to boost precision in even small training sets [11].

3. Method

The method, developed by inference from empirical results, is composed of three main stages: preprocessing, features retrieval and classification [12]. In the first stage, an input image face is prepared for further processing by means of noise reduction including Gaussian filtering. The image is then subjected to feature extraction, performed using Discrete Wavelet Transform [5], since DWT and DCT achieved the best precision, and DWT was found to be better at preserving high and low frequency information [13]. As Aguado *et al.* observed, filtering either high or low frequencies deteriorates the precision of gender recognition [14]. The feature vector calculated with DWT is finally passed to our hybrid classifier.

The superiority of the Support Vector Machine over other standard classifiers was confirmed by experimentation. This dominance was also achieved by the application of kernel transformation. It was noticed that the gender model, in the context of DWT coefficients, was highly non-linear and transformations of higher orders worked better. Two particular kernels were determined as paramount transformations: the Quadratic Kernel Function (QKF) and Radial Basis Function (RBF). Classification results for both SVMs indicated that, in most cases, they provided the same answer (*woman* or *man*). In the majority of the other cases, QKF SVM was superior over RBS, since for true decisions it revealed higher classification probability (expressed in (3) and (4) further in the text), enforcing the right decision of the classification ensemble.

The final classifier was composed of two separate Support Vector Machines, one with

a Quadratic Kernel Function $K_q(\beta, \delta) = (\beta^T \delta)^2$ and the other with a Radial Basis similarity Function $K_r(\beta, \delta) = \exp(\gamma||\beta - \delta||)$. For both SVMs, the success rate was calculated according to the following formulas, respectively.

$$s_q = \sum_{i=1}^m a_i y_i K_q(\beta_i, \beta) + b, \quad (1)$$

$$s_r = \sum_{i=1}^m a_i y_i K_r(\beta_i, \beta) + b, \quad (2)$$

where β_i is the feature vector of the i -th training example, y_i is a class label taking values -1 for class *woman* and 1 for class *man*, m is the number of training observations, a_i is a coefficient associated with an observation (zero if observation is not a support vector), b is a scalar, β is a feature vector of the test object and $K(\beta_i, \beta)$ is a kernel function.

Posterior probabilities $P(s_q)$ and $P(s_r)$ are then measured for two Support Vector Machines. The probability is a function of the success rate s . For the Quadratic kernel, the probability function is presented in (3), while that for the Radial Basis Function kernel it is shown in (4).

$$P(s_q) = \frac{1}{1 + e^{A_1 s_q + B_1}}, \quad (3)$$

$$P(s_r) = \frac{1}{1 + e^{A_2 s_r + B_2}}, \quad (4)$$

where s_q and s_r are success rates for the Quadratic kernel and RBF kernel respectively. A and B are function parameters.

The functions (3) and (4) provide a qualitative measure of a subject's membership to the class with label $\mathbf{1} - \textit{man}$.

The value of $P(s_q)$, denoted in short as x , is the probability of being classified as a *man* by the SVM with a Quadratic kernel. The probability of being classified as a *woman* by the same classifier is $1 - P(s_q)$ (denoted as \hat{x}) according to the probability theorem. In the second equation, the value of $P(s_r)$ (y) indicates the probability of being a *man* according to the SVM with RBF kernel, and the probability of being a women for this classifier is $1 - P(s_r)$ denoted as \hat{y} .

Based on the complementarity of the two SVMs, one can assume that the arithmetical average of posterior probability for each class gives a more reliable probability than each classifier separately. Hence, having calculated the posterior probability for the two SVMs, the mean is calculated according to standard formulas

$$z = (x + y)/2, \quad (5)$$

$$\hat{z} = (\hat{x} + \hat{y})/2. \quad (6)$$

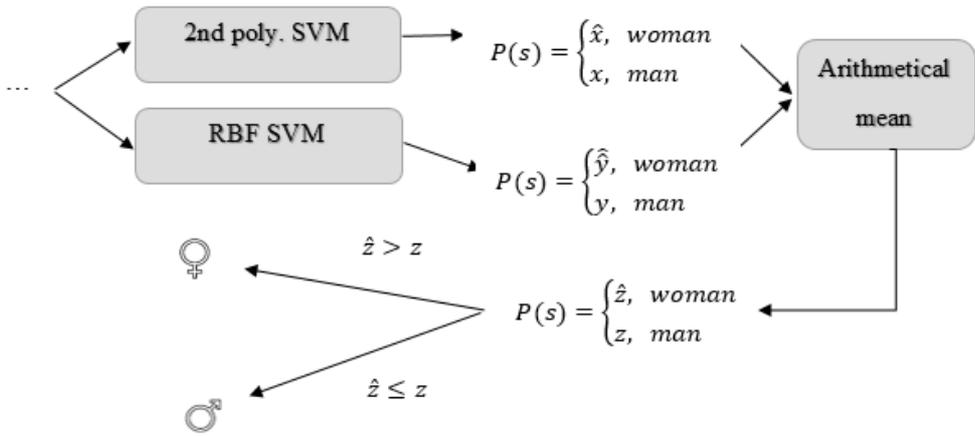


Fig. 3. Proposed classification method scheme

The final decision is then made depending on the values determined for z and \hat{z} . If the average posterior probability z is greater or equal to \hat{z} then the examined object is classified as a *man*, otherwise it is classified as a *woman*. The classification method is visually presented in Figure 3.

4. Test and Results

The tests consisted of two parts: validation of systems described in the literature [3, 4, 5] and validation of the proposed hybrid method. Validation was carried out with tests performed on half of the previously described set (100 subjects out of 200) from the FERET database, randomly selected. This set contains people of various races (Caucasian, Black and Asian), of various ages (from round 20 up to around 60), under varying lighting conditions (uniform light, spot light from different directions) and with different facial expressions (neutral, smiling, grinning, sad, disgusted). The images were taken from gray-scaled thumbnail photographs of the *FERET database*. This database was chosen due to the versatility that it provides with regard to age, race, facial expressions and lighting. In spite of the fact that it is not a new database, it is still valid since the general perception of gender does not change over time. To obtain a qualitative comparison, an accuracy factor was used

$$accuracy = \frac{\text{number of correct classifications}}{\text{capacity of test set}}. \tag{7}$$

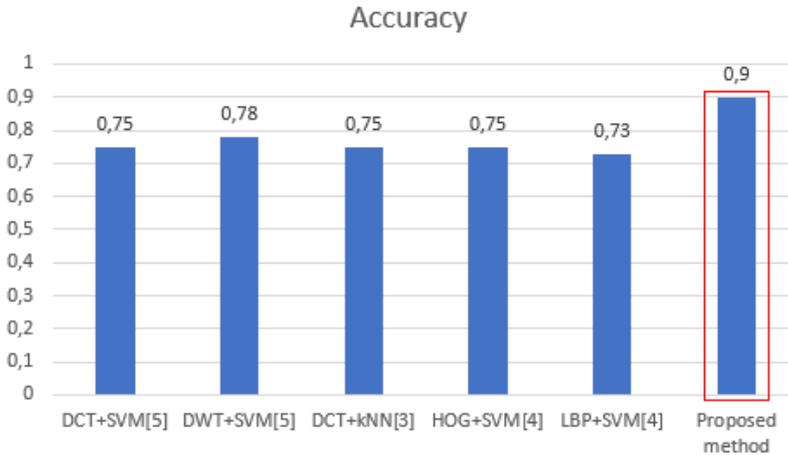


Fig. 4. Comparison of the pure state-of-art algorithms with the proposed system

Below, Figure 4 shows the accuracy of algorithms in the literature for the proposed image set, in comparison to the proposed system. The first two methods presented in the chart were proposed in [5]. They used either Discrete Cosine Transform or Discrete Wavelet Transform together with the SVM classification method. The variant with DWT achieved better results (78%) than that with DCT (74%). The results were worse than those reported by the authors, which may have been due to the significantly smaller database used in our study. The third bar depicts the accuracy achieved by the algorithm presented in [3]. It uses Discrete Cosine Transform applying the kNN classifier. It gained precision of 75%. Again, the lower accuracy with respect to that reported by the authors may have been caused by differences in databases. Nazir *et al.* drew on a larger database of students, hence the ages of the subjects were also quite similar. The next two methods suggested in [4] used a Histogram of Oriented Gradients and Local Binary Patterns as feature descriptors. Both methods were tested with the Support Vector Machine Classifier. The variant with HOG obtained a slightly better accuracy (75%) than that with LBP (73%). The authors trained and tested their system on an Indian database, rejecting cross-racial comparisons. The database used in our study contains subjects of different races, which has significant implications for the accuracy of the method.

As shown in Figure 4, the system from the literature with the highest accuracy (78%) was the Discrete Wavelet Transform with Support Vector Machine. The algorithms suggested by Nazir *et al.* [3], Alrashed *et al.* [5] and Singh *et al.* [4] achieved at most 78% accuracy with pictures from the *FERET database*. However, the accuracy of the

proposed method reached 90%. This means that with the proposed system 10 more people in every hundred were classified correctly.

In this way, the superiority of SVM in terms of gender classification [15] was clearly confirmed.

The execution time was measured to assess the feasibility of real-time applications. For tests performed with images of 100×100 px, the accuracy of the method was preserved and the time required to perform the classification was 0.2s. This enables complete classification of five faces per second.

5. Conclusion

The hybrid classification method presented here, consisting of two Support Vector Machines, one with Quadratic and one with Radial Basis kernels, deals very well with gender recognition, even when the subjects are of different races and ages, have different facial expressions, or are taken under different lighting conditions. The results compare favorably to those for algorithms presented in the literature, in [3, 4] and [5], which performed considerably worse with the small training data set.

Further research should focus on making the proposed method insensitive to rotation and scale and on making the whole system applicable for real use without any manual pre-processing. Finally, low-dimensional image representation pointed out by Amine *et al.* in [16] might be used to improve the accuracy and efficiency.

References

- [1] P. J. Phillips, J. Huang H. Wechsler and, and P. Rauss. The FERET database and evaluation procedure for face recognition. *Image and Vision Computing*, 13:259–306, 1998. doi:10.1016/S0262-8856(97)00070-X.
- [2] P. Phillips, H. Mon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000. doi:10.1109/34.879790.
- [3] M. Nazir, M. Ishtiaq, A. Batoool, M. A. Jaffar, and A. M. Mirza. Feature selection for efficient gender classification. In *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing. Proc. 11th WSEAS Int. Conf. on Neural Networks and 11th WSEAS Int. Conf. on Evolutionary Computing and 11th WSEAS Int. Conf. on Fuzzy Systems*, NN'10/EC'10/FS'10, pages 70–75, Iasi, Romania, 2010. dl.acm.org/citation.cfm?id=1863431.1863444.
- [4] V. Singh, V. Shokeen, and B. Singh. Comparison of feature extraction algorithms for gender classification. *International Journal of Engineering Research and Technology*, 2(5):1313–1318, May 2013. www.ijert.org.
- [5] H. A. Alrashed and M. A. Berbar. Facial gender recognition using eyes images. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6):2441–2445, 2013. www.ijarcce.com.

- [6] L. Spacek. Collection of facial images: Faces94. The Ohio State University. <http://cswww.essex.ac.uk/mv/allfaces/faces94.html> [Online; accessed 09 Sep 2016].
- [7] A. M. Martinez. AR Face Database. The Ohio State University. <http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html> [Online; accessed 09 Sep 2016].
- [8] A. M. Martinez and R. Benavente. The AR face database. CVC Technical Report #24, June 1998.
- [9] A. Wojciechowski. Potential field based camera collisions detection in a static 3D environment. *Machine Graphics & Vision*, 15(3/4):665, 2006.
- [10] R. Sharma and M. S. Patterh. Indian face age database: A database for face recognition with age variation. *International Journal of Computer Applications (0975 -8887)*, 126(5):21–27, 2015.
- [11] V. N. Pawar and S. N. Talbar. Hybrid machine learning approach for object recognition: Fusion of features and decisions. *Machine Graphics and Vision*, 19(4):411–428, 2010.
- [12] R. Staniucha and A. Wojciechowski. Mouth features extraction for emotion classification. In *Proc. 2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1685–1692. IEEE, Sept 2016. ieeexplore.ieee.org/document/7733480/.
- [13] A. Katharotiya, S. Patel, and M. Goyani. Comparative analysis between DCT & DWT techniques of image compression. *Journal of Information Engineering and Applications*, 1(2):9–17, 2011. <http://iiste.org/Journals/index.php/JIEA>.
- [14] L. Aguado, I. Serrano-Pedraza, S. Rodríguez, and F. J. Román. Effects of spatial frequency content on classification of face gender and expression. *The Spanish Journal of Psychology*, 13(2):525–537, 2010. doi:10.1017/S1138741600002225.
- [15] S.A. Khan, M. Katameneni, and P. M. Latha. A comparative analysis of gender classification techniques. *Middle-East Journal of Scientific Research*, 20(1):1–13, 2014. www.idosi.org/mejsr/mejsr.htm.
- [16] A. Amine, S. Ghouzali, M. Rziza, and D. Aboutajdine. An improved method for face recognition based on SVM in frequency domain. *Machine Graphics and Vision*, 18(2):187–199, 2009.

PARTICLE EFFECT SYSTEM FOR THE NEEDS OF A MODERN VIDEO GAME USING THE GPU

Wojciech Zeler, Paweł Rohleder

Techland Sp. z o.o.

ul. Żółkiewskiego 3, 63-400 Ostrów Wlkp, Poland

<http://techland.pl>

Abstract.

A new system of creation and management of particle effects created for the needs of the future productions of Techland Co. Ltd. is presented. By a proper organisation of memory buffers it provides for maximum data density in the memory. This makes it possible to simplify the calculations and to use a smaller number of threads and less memory readings.

Key words: GPU, particles, FX, compute, shader, real time, graphics

1. Introduction

A particle effect is any physical phenomenon to replicate and visualize which it is necessary to simulate the behavior and interaction of a number of unrelated, so called, *particles*, i.e. individual entities subject to certain behaviors specified by the artist. Depending on the intended result, the particles can be e.g. sparks, dust particles, clouds of smoke, individual insects, flames, liquid streams, etc. (see Fig. 1 for an example).

In the discussed implementation it has been tried to fully use the benefits offered by today's graphics cards and game consoles. In order to operate, the equipment compatible with the Shader Model 5.0 [3] is necessary.

2. Review of competing solutions

Prior to the implementation, a review of competitive solutions was conducted. It helped us to create list of features that our new system must have. Among them there were:

- particle effects editor in the Unity engine,
- particle effects editor in the Unreal 4 engine,
- particle effects editor in the CryEngine.

The comparative analysis was made in the following aspects:

- editor shape and appearance, artist's work process with the editor – window appearance, integration with the rest of the editor;
- ways to control the particles, such as e.g. curves, gradients;



Fig. 1. Example of the particle effect with fire and smoke.

- additional options, such as e.g. sub-emissions (i.e. particle emission by other particles), and depth buffer collisions.

The comparison revealed that none of the existing systems had all the features we required. These features will be considered in the context of the methodology developed within the present paper.

3. Tasks of the system

The main tasks to be executed by every particle effect system are as follows.

New particle emission Each individual particle is created by a, so called, *emitter*. It defines in what way the particles created in it will act throughout their lifespan. The operation of emitting particles is quite complicated and should take into account, e.g. the necessity to emit more than ten thousand particles from one emitter during one animation frame.

Individual particle movement simulation Each particle has certain physical properties, such as instantaneous velocity, size, acceleration, and orientation. Each of those properties can be further modified by a set of modifiers freely set by the artist (in case of e.g. acceleration, it is the force of gravity).

Particle sorting Because of the translucency of most of the particles the order in which

they are painted on the scene becomes important. A lack of sorting would result in a lot of artifacts in the generated image.

Taking into account different types of particles Translucent particles are the most popular. However, the possibility of using particle effects at any stage of rendering of frame animation is necessary because it gives access to many potentially interesting effects, such as e.g. refraction.

Drawing particles This is the last stage of the system's operation. It is at this time that the particles are placed on the scene in a way visible to the user (up till now they were only the compressed structures in the graphic card's memory).

4. Algorithm

The algorithm was designed in such a way as to strive for storing the maximum density of data in the smallest area of memory. This makes it possible to keep the calculations simple and to use a smaller number of threads and make less memory readings.

The discussed implementation works in two stages.

In the first stage basic calculations for the continuous operation of the entire system are performed, i.e., calculations of state changes of all particles, emission of new particles and deleting the old ones.

The purpose of the second stage is to determine a subset of particles visible from every camera known to the game, and then to perform calculations associated with their presence and to prepare data needed for instantiating.

First stage The order of operations included in the first stage is shown in the chart in Fig. 2.

First, the information on the amount of particles in the main buffer is read. This information allows to determine the amount of particles at the start of the animation frame, as well as how many thread groups must be used to perform the simulation.

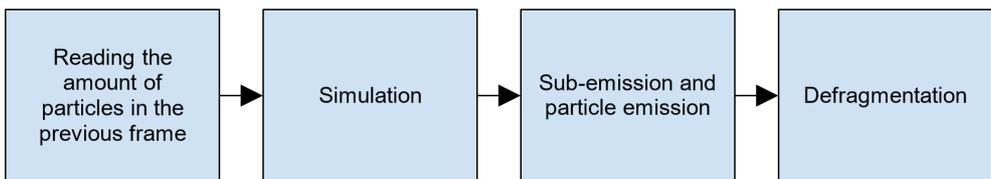


Fig. 2. Chart of operations of the stage one of the algorithm.

Let us introduce the following denotations:

P – set of all particles,

P_{prev} – number of particles in the previous frame (equal to the index of the last live particle in the set),

P_{curr} – number of particles in the current frame (at the beginning of the frame it is $P_{\text{curr}} := P_{\text{prev}}$),

Th_{opt} – optimal number of threads in one group. For contemporary GPUs it is most commonly $Th_{\text{opt}} = 64$,

P_{prev} – number of particles in the previous frame (equal to the index of the last live particle in the set),

G_{sim} – number of thread groups needed to perform the simulation.

Therefore

$$G_{\text{sim}} = \text{ceil} \left(\frac{P_{\text{prev}}}{Th_{\text{opt}}} \right). \quad (1)$$

During simulation, excluding velocity, position, acceleration, orientation and any other physical property change, particles may be assigned to the following sets:

set D containing particles whose lifetime came to an end in the current animation frame;

set S containing particles capable of emitting other particles.

The sets S and D are reset with the beginning of each consecutive animation frame.

The next step is the, so called, *sub-emission*. It is limited from above to a maximum volume, due to the easy to cause exponential increase in particle quantity visible on the screen (which may result in a sudden drop in the animation speed, and even the loss of system stability). The principle of *first come, first served* is in force.

Immediately afterwards, a classic particle emission occurs, just like in the older particle effect systems based on calculations performed by the CPU.

Both emissions work according to the following algorithm:

- If the set D is not empty, take the index of that dead particle.
 - Otherwise increase P_{curr} and take its previous value – this is the new index of the particle.
- Fill a particle, with this index, with data from the emitter.

The last stage of the first step is the, so called, *defragmentation*, i.e. transferring a given number of particles, whose indices are close to P_{curr} , to places marked with indices of particles in the set D . This operation is rather light because only up to a few thousand particles are being transferred per animation frame and it takes about 100 microseconds on an Xbox One console. Its goal is to decrease the P_{curr} value, so that in the next animation frame it would be possible to create a smaller G_{sim} quantity. Without this procedure, the buffer, even for a small quantity of particles (< 1000), can very quickly fill in the whole prepared pool ($\sim 0.5 \times 10^6$), thus forcing a maximally high G_{sim} value or looking for free spaces in the memory for new particles each time.

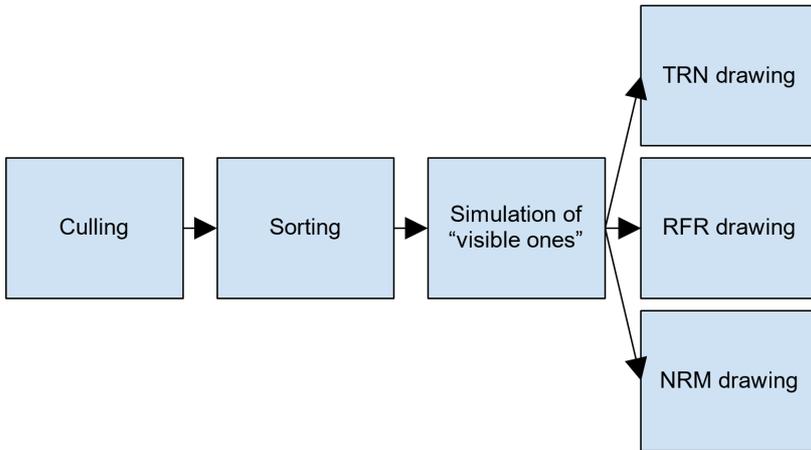


Fig. 3. Chart of operations of the stage two of the algorithm.

Second stage After performing a simulation of all the particles and emitting new ones or culling old ones, the second stage takes place, in which calculations for individual cameras are performed. They are directly connected with instantiating, being the culmination of the entire process. The stage is comprised of the steps shown in Fig. 3.

Another reason for which it is worthwhile to pursue the reduction of the P_{prev} value is the frustum culling step. It must be performed for all P_{curr} particles (including the possibility that dead particles are inside that block). After performing this operation we get the set denoted A_0 on the output, in which the indices of all particles visible to a given camera, denoted as C_0 , are located.

With the information about which particles are necessary to properly draw a scene (their indices are known), as well as knowing their distance from a given camera, it becomes possible to sort them according to distance to ensure a proper order of drawing. To achieve this we used the bitonic sort algorithm [1] and we have based its implementation on AMD's publication [4] presenting a very minimalistic particle effects system.

The particles in the memory are sorted not only by distance to the camera, but also according to the technique they will be drawn with. Thanks to that, after sorting, the buffer contains a few continuous blocks with particles ready to be drawn with the use of various techniques.

Example For example: the buffer containing the set A_0 (i.e. containing particles which passed the visibility tests) before sorting looks like shown in Tab. 1, where I_x – particle

Tab. 1. The buffer with set A_0 before sorting.

$I_0 = 123$	$I_1 = 345$	$I_2 = 234$	$I_3 = 4$	$I_4 = 3$	$I_5 = 45$	$I_6 = 1222$
$D_0 = 12.3$	$D_1 = 4$	$D_2 = 3$	$D_3 = 59$	$D_4 = 22.4$	$D_5 = 1.2$	$D_6 = 23$
$T_0 = 0$	$T_1 = 0$	$T_2 = 1$	$T_3 = 1$	$T_4 = 2$	$T_5 = 0$	$T_6 = 0$

Tab. 2. The buffer with set A_0 after sorting.

$I_0 = 123$	$I_1 = 345$	$I_2 = 234$	$I_3 = 4$	$I_4 = 3$	$I_5 = 45$	$I_6 = 1222$
$D_0 = 12.3$	$D_1 = 4$	$D_2 = 3$	$D_3 = 59$	$D_4 = 22.4$	$D_5 = 1.2$	$D_6 = 23$
$T_0 = 0$	$T_1 = 0$	$T_2 = 1$	$T_3 = 1$	$T_4 = 2$	$T_5 = 0$	$T_6 = 0$

index in the set P , D_x – particle distance from the camera in which it is drawn, T_x – technique with which the particle will be drawn (to be explained further).

After sorting, the buffer will look as shown in Tab. 2.

Final steps The next step is *simulating visible particles*, that is, calculating all variables needed to properly draw each particle. Up until now we have used a representation containing only data related to movement, position, lifespan, etc. Information on textures, color, gradients, and other features related solely to drawing the particles was not needed and additionally would increase the amount of transferred data. After this step, the data representation optimized for drawing in various techniques alone becomes correct. It lacks e.g. information on the particle's velocity or the forces acting on it. Such an approach makes it possible to decrease the amount of data needed in a given moment from over 40 MB to around 20. This is critical in the case of using Xbox One and its 32 MB of ESRAM.

The last stage, which crowns the whole process, is drawing. Various techniques are used in the rendering engine for this task. Each of them is dedicated to drawing different type of objects. The technique used to draw translucent objects is called TRN, the one used to draw refractive objects is called RFR, opaque objects are drawn with OPQ technique, etc.

As it was mentioned before – it takes place in a few stages of scene rendering, because particles can have different purposes. The most popular ones are translucent and are an ideal fit for drawing, e.g. fire, smoke, dust, snow, etc. The TRN technique fills this need. The screenshot shown in Fig. 4 presents fire and smoke generated with this technique. Background image was used by our artists to show refraction of light.

Another part of particles in our system consists of ones responsible for the refraction of light. As it was said before – they are drawn with the RFR technique – see fig. 5

The combination of the two effects above allows for a quite realistically looking fire.



Fig. 4. Example of the particle effect with fire and smoke.

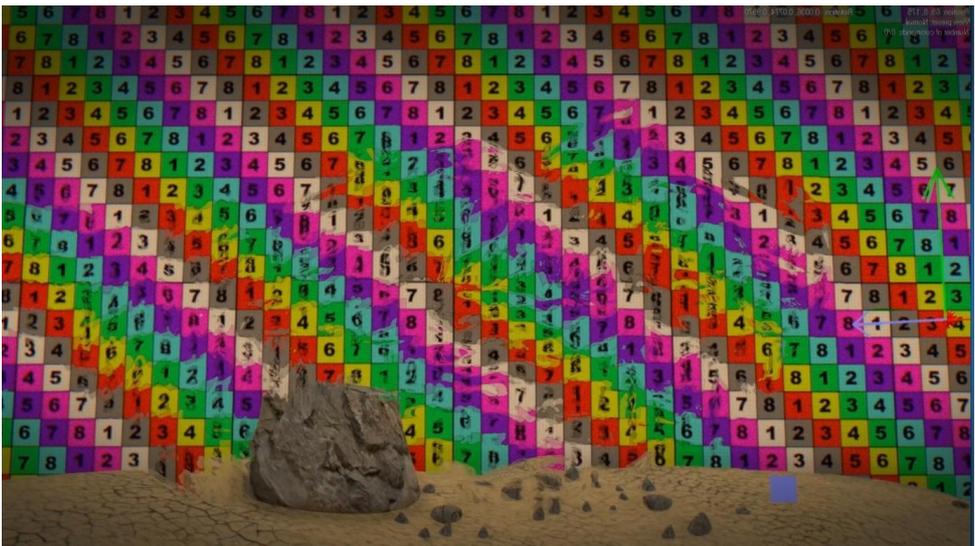


Fig. 5. Example of the light refraction.

Thanks to the data being sorted in the above described way, it is very easy to implement particle drawing in each of the techniques. Each time the input data are exactly the same – the buffer with data created after the *simulation of visible particles* and the buffer with a number of particles for each technique. Knowing the order of techniques in the buffer, it is easy to determine parameters for further drawing operations (using the `DrawInstancedIndirect` method [2]).

Referring to the previously given example with sorting let us observe the following:

- `DrawInstancedIndirect` for the TRN technique (i.e., $T = 0$) will draw the first 4 particles;
- `DrawInstancedIndirect` for the RFR (i.e., $T = 1$) will draw 2 further ones (without first 4 TRN particles);
- `DrawInstancedIndirect` for the NRM technique (those responsible for modifying normal vectors in `GBuffer`) (i.e., $T = 2$, without first 6 TRN and RFR particles).

Of course, the above operations will not take place one after the other, but only when each of them is required. All techniques are rendered in specific order.

5. Data storage

From the point of view of the HLSL code, the particle, during the first stage of simulation, is represented by the quite highly compressed structure shown below, constituting the description of its physical properties:

```
struct GpuFx_ComputeParticle
{
    float3    position;
    uint      sizeXY;
    uint2     flags_emitterId_ttl_seed;
    uint2     velocity_orientation;
    uint2     acceleration_ttlPhase;
    uint2     orbitOffsetXYZ_parentId;
};
```

Floating-point numbers are used only for storing the position in the world in view of precision. All the other variables are stored as a half type. These types of measures aim to decrease the size of an individual particle stored in the memory. It should be remembered that one of the main objectives of the designed system told of supporting up to 0.5 million particles in real time with the use of current generation consoles, i.e. Playstation 4 and Xbox One.

The stage of detecting particles in the frustum and sorting allows for using an even smaller structure:

```
struct GpuFx_ParticleInstance
{
    uint    technique_index; // 8 bytes - technique (NRM/RFR/OPQ...)
                                // 24 bytes - index
    float   distance;
};
```

Only the data prepared in this way make it possible to prepare a buffer which will be the input for the drawing operation. It takes place in the so called *simulation of visible particles* stage and works well for filling the frustum composed only of some tens of bytes of data that will be read accordingly and interpreted by the vertex shader for one of the techniques. Each technique uses a different set of data, but all the particles share the same buffer.

6. Conclusion

The presented algorithm makes it possible to minimize the number of places in which data are loaded – further data are added only when they are actually needed.

The main course of the algorithm does not assume special cases. Using one buffer to store particles belonging to various techniques allows to maximally utilize the allocated memory without the need of establishing any smaller budgets. Everything takes place dynamically and is adjusted to production needs.

Drawing particles, thanks to the locality of data, favors the operations related to the data cache.

7. Further development

It is key to solve the problem of the low, so called, *fill-rate*, emerging while drawing large amounts of big particles close to the cameras, for example, by using smaller particles or hiding the invisible ones in a better way. Achieving this will require better co-operation with the team responsible for creating particle effects in order to work out an optimal way to design the effects.

Additionally, further work is planned, adding minor functionality such as support of vector fields and improvements in support of animated textures.

Also, the optimization is of priority here – in terms of both memory as well as performance.

The new system also includes a particle effect editor deliberately omitted in this paper. In further work it is expected that the emphasis will be put on ergonomics of its use and its stability.

Acknowledgement

The paper was created as the result of the project “Industrial Research on the Technology of Scaled City for the Needs of Computer Games” (original title in Polish: „Badania przemysłowe nad technologią skalowanego miasta na potrzeby gier komputerowych”).

The project is co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme.

References

- [1] Vardit Cohen, Brian Hillman, Kyle Suarez, and Sesh Venugopal. Bitonic sort. http://www.cs.rutgers.edu/~venugopa/parallel_summer2012/bitonic_overview.html [Online; accessed Dec 2016].
- [2] Microsoft Corporation. ID3D11DeviceContext::DrawInstancedIndirect method. [https://msdn.microsoft.com/en-us/library/windows/desktop/ff476413\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ff476413(v=vs.85).aspx) [Online; accessed Dec 2016].
- [3] Microsoft Corporation. Shader Model 5. [https://msdn.microsoft.com/en-us/library/windows/desktop/ff471356\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ff471356(v=vs.85).aspx) [Online; accessed Dec 2016].
- [4] Jason Stewart (jstewart-amd). AMD GPU Particles Sample. <https://github.com/GPUOpen-LibrariesAndSDKs/GPUParticles11> [Online; accessed Dec 2016].

OFFLINE HANDWRITTEN PRE-SEGMENTED CHARACTER RECOGNITION OF GURMUKHI SCRIPT

Munish Kumar¹, Manish K. Jindal², Rajendra K. Sharma³ and Simpel R. Jindal⁴

¹Maharaja Ranjit Singh Punjab Technical University,
Department of Computer Applications, GZS Campus College of Engineering & Technology,
Bathinda, Punjab, India munishcse@gmail.com

²Panjab University Regional Centre, Department of Computer Science & Applications,
Muktsar, Punjab, India manishphd@rediffmail.com

³Thapar University, Department of Computer Science & Engineering,
Patiala, Punjab, India rksharma@thapar.edu

⁴Yadavindra College of Engineering, Talwandi Sabo, Computer Science & Engineering Section,
Bathinda, Punjab, India simpler.jindal@rediffmail.com

Abstract. In this paper, we have proposed a feature extraction technique for recognition of segmented handwritten characters of Gurmukhi script. The experiments have been performed with 7000 specimens of segmented offline handwritten Gurmukhi characters collected from 200 different writers. We have considered the set of 35 basic characters of the Gurmukhi script and have proposed the feature extraction technique based on boundary extents of the character image. PCA based feature selection technique has also been implemented in this work to reduce the dimension of data. We have used k-NN, SVM and MLP classifiers. SVM has been used with four different kernels. In this work, we have achieved maximum recognition accuracy of 93.8% for the 35-class problem when SVM with RBF kernel and 5-fold cross validation technique were employed.

Key words: feature extraction, classification, PCA, k-NN, SVM, MLP

1. Introduction

Optical Character Recognition (OCR) is the process which helps to convert the handwritten or printed text into a format that is processable by machine. Handwritten character recognition is more complicated due to the variation in styles of writing. When a text is handwritten and scanned, a large amount of noise will occur while recognizing such handwritten characters. We have divided handwritten character recognition into two categories, *viz.* online and offline. This paper focuses on offline handwriting recognition. Therefore, in this paper we have exhibited a recognition technique for offline handwritten character recognition of Gurmukhi script in light of the fact that a little work has been done on Gurmukhi script to date, to the best of our knowledge, in spite of the fact that a considerable measure of work has been done on various scripts, for example, Bangla, Devanagari and so forth.

2. Related work

Bhattacharya *et al.* [1] have presented Bangla character recognition system and they have obtained maximum recognition accuracy of 94.7%.

Bunke and Varga [2] have reviewed the state of the art in offline Roman cursive handwriting recognition. They identified the challenges in Roman cursive handwriting recognition.

Kunte and Samuel [8] have presented an efficient framework for printed Kannada text recognition. They considered invariant moments and Zernike moments as features and Neural Network (NN) as classifier. They achieved a recognition accuracy of 96.8% using 2500 characters.

Grosicki and Abed [3] proposed a French handwriting recognition system in a competition held in ICDAR 2011. In this competition, they have presented comparisons between different classification and recognition systems for French handwriting recognition. Kacem *et al.* [4] have used structural components for recognition of Arabic names.

Kumar *et al.* [7] have presented efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. They have also presented a hierarchical technique for offline handwritten Gurmukhi character recognition [5]. Using this technique, they accomplished a recognition accuracy of 91.8%. Kumar *et al.* [6] have presented a study on various transformation techniques for offline handwritten Gurmukhi character recognition.

Lorigo and Govindaraju [9] have introduced a critical review on offline Arabic handwriting recognition frameworks. They have presented various techniques employed at different stages of the offline handwritten Arabic character recognition system.

Pal *et al.* [10] dealt with recognition of offline handwritten Bangla compound characters using modified quadratic discriminant function. They have acquired 85.9% recognition precision by using five-fold cross validation technique. Pal *et al.* [11] have assimilated a comparative study of handwritten Devanagari character recognition. Sharma *et al.* [12] considered a quadratic classifier based scheme for the recognition of off-line Devanagari handwritten characters. In this system, they have used chain code directional features and achieved a recognition accuracy of 80.4%.

Tran *et al.* [13] have considered the problem of French handwriting recognition using 24800 samples. They have worked on both online and offline handwritten character recognition. Wang *et al.* [14] have presented a technique for recognition of Roman alphabets and numeric characters. They had a recognition rate of about 86.0%.

Zhu *et al.* [16] have described a robust model for online handwritten Japanese text recognition. They obtained a recognition accuracy of 92.8% using 35686 samples.

The work carried out in this paper focuses on the recognition of segmented offline handwritten character recognition for Gurmukhi script of a 35-class problem.

3. Gurmukhi script

The name Gurmukhi signifies “from the mouth of the Guru” and originates from the old Punjabi word *Guramukhi*. Gurmukhi script is utilized for composing the Punjabi dialect. A portion of the properties of the Gurmukhi script are as follows.

- In Gurmukhi script, there are 35 character constants out of which the first three are vowel bearers.
- There are six consonants which are created by placing a dot (*bindi*) at the foot (*pair*) of the consonant which is called Gurmukhi constants with subscript dots.
- Auxiliary Gurmukhi symbols denote double consonants, or conjunct adjacent constants.
- Double Consonants: These symbols are also called *adhak*.
- In Gurmukhi script, various punctuation symbols are used. These symbols signify the partition of heading from the text, or line break. Various punctuation symbols are *visarg*, *dandi*, and *dodandi*.
 - *Visarg*: it is symbolized as two circles, one circle is above the other circle, just as a colon used in English language. This symbol points towards the division of heading from the text.
 - *Dandi*: it is a single vertical line which indicates the completion of a sentence.
 - *Dodandi*: signifies two parallel lines which indicate a break in a line.

4. The proposed recognition system

Proposed recognition framework comprises of different stages: data collection, digitization, preprocessing, feature extraction, and classification. In our proposed framework, we have also used Principal Component Analysis (PCA) to reduce the length of the feature vector. The general plan of the proposed framework is depicted in Fig. 1.

4.1. Data collection

In this work, we have collected 7000 specimens of segmented Gurmukhi characters from 200 different writers. They were requested to write each of the fundamental 35 characters of the Gurmukhi script. We have collected this type of dataset from different government offices, schools, colleges etc.

4.2. Digitization and pre-processing

Digitization is the way toward changing over the handwritten document into electronic shape. This procedure is accomplished by using HP-1400 model of scanner. Digitization stage creates the digital image which is sustained to the pre-processing stage. Pre-processing stage is the underlying phase of character recognition framework. In this,

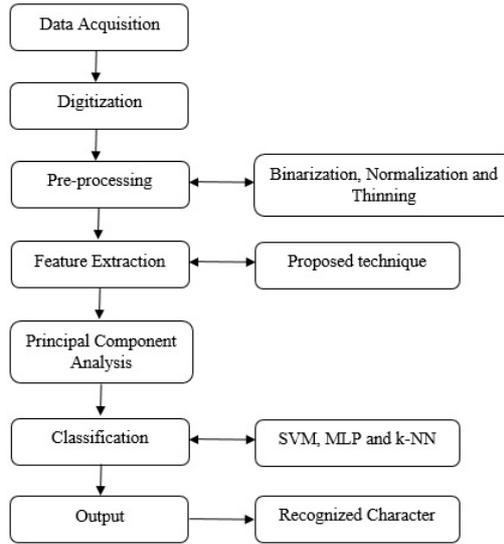


Fig. 1. Block diagram of the offline handwritten Gurmukhi character recognition system.



Fig. 2. Image of a Gurmukhi character: (a) digitized; (b) thinned.

stage we change over the image into a gray scale format and bitmap image. From that point onward, bitmap image is changed over into the thinned image as shown in Fig. 2 by using parallel thinning algorithm proposed by Zhang and Suen [15].

4.3. Proposed feature extraction technique

In our proposed technique, a character image is horizontally divided into n horizontal regions and then the width of the character in this region, that is, the distance between the leftmost and rightmost element of the boundary of the character is taken as a feature in each region, thus forming n features. Similarly, the height of the character, that is, the vertical distance between the uppermost and lowermost element of the boundary of the character in n vertical regions is found, thus forming further n features. For the

regions that do not have a foreground pixel, the feature value is taken as zero. By using this process, $2n$ features are extracted for an image of one character. In our case it was $n = 100$ so there were $2n = 200$ features in total.

In the algorithm, the background pixel (OFF pixel) is considered as 0 and foreground pixel (ON pixel), that is the pixel belonging to the character, is considered as 1. The steps made to extract the features for horizontal regions are as follows.

```

I : given image ( $x \times y$ )
V ← Allocate( $n$ ): vector of  $n$  features for horizontal regions
for  $i := 1$  to  $n$  step 1 do
    for  $j := 1$  to  $x$  step 1 do
        for  $k := 1$  to  $y$  step 1 do
            if  $I_{jk} = 1$  then
                 $a := j$ 
                break loops for  $j, k$ 
            end if
        end for
    end for
    for  $j := x$  to 1 step -1 do
        for  $k := 1$  to  $y$  step 1 do
            if  $I_{jk} = 1$  then
                 $b := k$ 
                break loops for  $j, k$ 
            end if
        end for
    end for
     $V_i := b - a$ 
end for
Return V

```

4.4. Principal Component Analysis

Principal Component Analysis (PCA) is a methodology which changes various correlated features into a number of uncorrelated features called principal components. PCA has been broadly used in the field of pattern recognition. PCA is a technique which is used to reduce the dimension of data and to extract the meaningful feature subset. We covered 95% variance in this work for PCA.

4.5. Classification

Classification phase is the last phase of character recognition framework; this phase uses the feature extracted in the previous phase. The preliminary aim of the classification

Tab. 1. Recognition accuracy of the proposed feature extraction technique with different classifiers. Recognition accuracy was calculated as an average for five folds.

Feature Extraction Technique	Classifier	False Positive Rate (FPR) [%]	False Rejection Rate (FRR) [%]	Recognition Accuracy [%]
without PCA	k -NN ($k = 5$)	0.80	9.10	90.10
	MLP	1.20	11.60	87.20
	Linear SVM	1.70	8.10	90.20
	Polynomial SVM	1.60	19.60	78.80
	RBF SVM	0.60	8.80	90.60
	Sigmoid SVM	1.20	16.40	82.40
	with PCA	k -NN ($k = 5$)	1.10	6.60
MLP		0.89	9.71	89.40
Linear SVM		0.84	7.46	91.70
Polynomial SVM		1.40	16.20	82.40
RBF SVM		0.60	5.60	93.80
Sigmoid SVM		1.20	11.70	87.10

phase of an OCR framework is to build up the constraint for decreasing the misclassification relevant to feature extraction. In order to reduce the complexity of classifier, we have used PCA to reduce the dimension of the feature vector, as written above. In present work, we have used three different classifiers, namely, k -NN ($k = 5$), SVM and MLP. SVM classifier has also been considered with four different flavours, namely, Linear-SVM, Polynomial-SVM, RBF-SVM and Sigmoid-SVM.

5. Experimental results and discussion

In this section, we have presented the results of experiments by using proposed feature extraction technique with different classifiers. We have used 5-fold cross validation technique for obtaining the recognition accuracy. For the purpose of experiments, we have considered 7000 samples of offline handwritten Gurmukhi characters of the 35-class problem. Various types of recognition accuracies obtained forthwith have been depicted in Tab. 1. False Positive Rate (FPR), Rejection Rate (RR) and recognition accuracy achieved with various classifiers, without PCA and with PCA, are graphically depicted in Figs. 3a, b and c, respectively.

We have accomplished the best recognition accuracy of 93.8% using the proposed feature extraction and RBF-SVM classifier. Confusion matrix for this case has been depicted in Tab. 2.

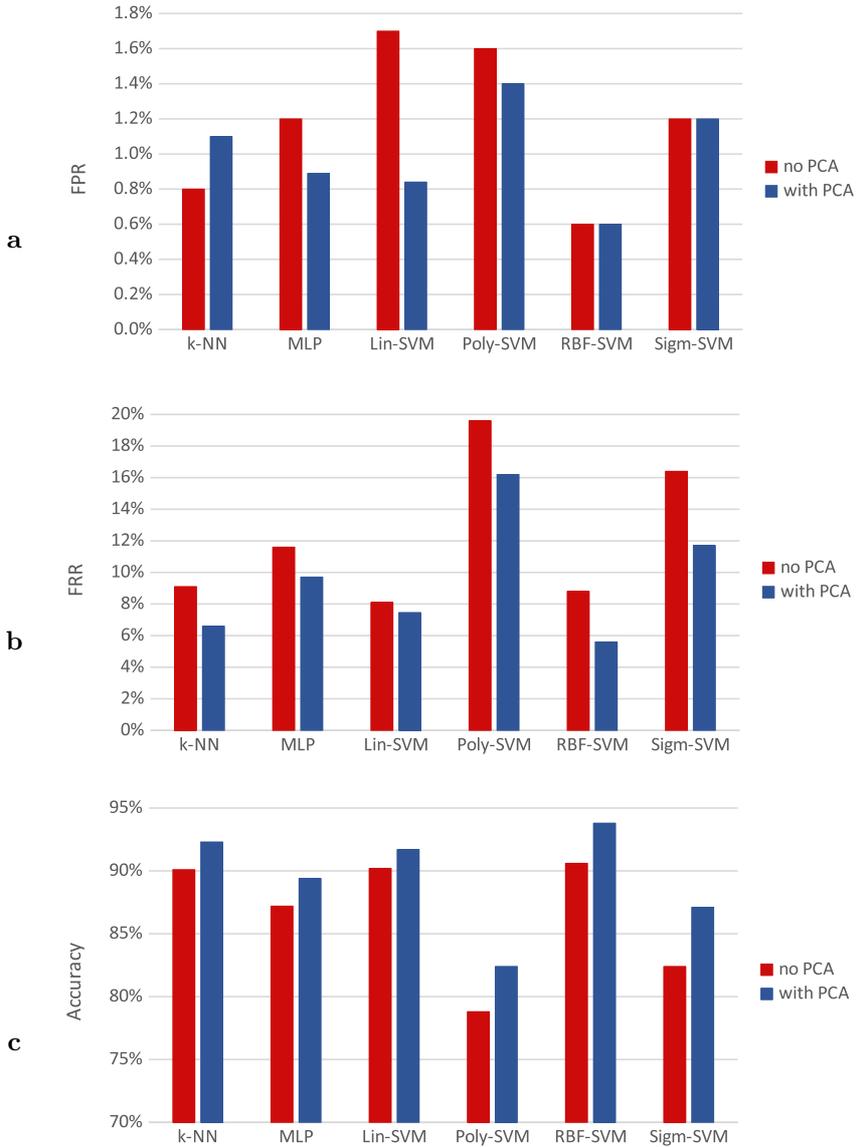


Fig. 3. Quality measures of classification with classifiers as listed in Tab. 1 (here, names abbreviated w.r.t. table for clarity). (a) False Positive Rate (FPR); (b) False Rejection Rate (FRR); (c) Recognition Accuracy (average of five folds).

Tab. 2. Confusion matrix for proposed feature extraction with PCA and RBF-SVM classifier.

Character	Confused with characters					
ੳ	ੳ 95%	ੜ 2%	ਤ 1%	ਰ 2%		
ਅ	ਅ 98%	ਘ 2%				
ੲ	ੲ 91%	ੳ 3%	ਟ 3%	ਰ 2%	ਵ 1%	
ਸ	ਸ 89%	ੜ 1%	ਗ 6%	ਣ 1%	ਮ 3%	
ਹ	ਹ 94%	ਕ 1%	ੜ 2%	ਤ 1%	ਰ 2%	
ਕ	ਕ 93%	ਧ 1%	ਰ 4%	ੜ 2%		
ਖ	ਖ 96%	ਘ 1%	ਬ 2%	ਪ 1%		
ਗ	ਗ 91%	ਸ 7%	ਧ 1%	ਰ 1%		
ਘ	ਘ 95%	ੜ 2%	ਅ 3%			
ਙ	ਙ 91%	ਬ 4%	ੜ 2%	ਪ 2%	ੜ 1%	
ਚ	ਚ 92%	ਜ 2%	ਧ 5%	ਬ %		
ਛ	ਛ 91%	ੳ 2%	ੜ 2%	ੳ 3%	ਯ 2%	
ਜ	ਜ 93%	ੲ 2%	ੳ 4%	ਠ 1%		
ੲ	ੲ 91%	ਕ 4%	ਙ 1%	ੜ 3%	ੳ 1%	
ੳ	ੳ 91%	ਚ 2%	ਧ 2%	ਵ 5%		
ਟ	ਟ 92%	ਫ 6%	ਵ 2%			
ਠ	ਠ 92%	ਚ 2%	ਛ 1%	ਟ 2%	ਤ 3%	
ੜ	ੜ 92%	ਤ 3%	ਰ 3%	ੜ 2%		
ੳ	ੳ 91%	ਸ 3%	ਚ 2%	ਛ 4%		
ਣ	ਣ 93%	ੲ 4%	ਛ 3%			
ਤ	ਤ 92%	ੜ 2%	ਬ 3%	ੜ %		
ਬ	ਬ 87%	ਧ 8%	ਖ 4%	ਪ 1%		
ਦ	ਦ 92%	ਚ 4%	ੳ 2%	ਵ 1%	ੜ 1%	
ਧ	ਧ 83%	ਬ 5%	ਬ 4%	ਪ 5%	ਵ 2%	ੜ 1%
ਠ	ਠ 92%	ਗ 1%	ਠ 2%	ਫ 2%	ਲ 2%	ਵ 1%
ਪ	ਪ 92%	ਖ 4%	ਬ 3%	ਵ 1%		
ਫ	ਫ 95%	ਗ 1%	ਛ 3%	ਟ 1%		
ਬ	ਬ 91%	ਸ 1%	ਧ 2%	ਠ 4%	ੜ 2%	
ਭ	ਭ 93%	ਖ 2%	ਙ 3%	ੲ 1%	ਠ 1%	
ਮ	ਮ 95%	ਸ 1%	ਖ 2%	ਙ 1%	ੲ 1%	
ਯ	ਯ 93%	ਠ 2%	ਫ 3%	ੳ 2%		
ਰ	ਰ 89%	ਠ 5%	ੲ 2%	ੜ 2%	ਠ 1%	ੜ 1%
ਲ	ਲ 92%	ੲ 2%	ਠ 4%	ਠ 2%		
ਵ	ਵ 94%	ੲ 1%	ਛ 1%	ੲ 3%	ਣ 1%	
ੜ	ੜ 92%	ੲ 4%	ਜ 1%	ੜ 3%		

6. Conclusion and scope of future research

A feature extraction technique has been presented for offline handwritten Gurmukhi character recognition of 35-class problem. For classification, we have used k -NN, SVM and MLP classifiers. The best recognition accuracy of 93.8% has been accomplished using 5-fold cross validation technique with the proposed feature extraction technique and RBF-SVM classifier. We have used 7000 specimens of isolated offline handwritten Gurmukhi characters in experimentation work. This precision can likely be increased by considering a larger data set while training the classifier. This work can likewise be extended to other Indian scripts: Bengali, Devanagari, Tamil, and so forth.

References

- [1] U. Bhattacharya, M. Shridhar, and S. K. Parui. On recognition of handwritten bangla characters. In P. K. Kalra and S. Peleg, editors, *Proc. 5th Indian Conf. Computer Vision, Graphics and Image Processing ICVGIP 2006*, pages 817–828. Springer Berlin Heidelberg, Madurai, India, December 13–16, 2006. doi:10.1007/11949619-73.
- [2] H. Bunke and T. Varga. Off-line Roman cursive handwriting recognition. In B. B. Chaudhuri, editor, *Digital Document Processing: Major Directions and Recent Advances*, pages 165–183. Springer, London, 2007. doi:10.1007/978-1-84628-726-8_8.
- [3] E. Grosicki and H. E. Abed. ICDAR 2009 Handwriting Recognition Competition. In *Proc. 10th Int. Conf. Document Analysis and Recognition ICDAR 2009*, pages 1398–1402, July 2009. doi:10.1109/ICDAR.2009.184.
- [4] A. Kacem, N. Aouiti, and A. Belaïd. Structural features extraction for handwritten arabic personal names recognition. In *Proc. Int. Conf. Frontiers in Handwriting Recognition ICFHR 2012*, pages 268–273, September 2012. doi:10.1109/ICFHR.2012.276.
- [5] M. Kumar, M. K. Jindal, and R. K. Sharma. A novel hierarchical technique for offline handwritten Gurmukhi character recognition. *National Academy Science Letters*, 37(6):567–572, December 2014. doi:10.1007/s40009-014-0280-1.
- [6] M. Kumar, M. K. Jindal, and R. K. Sharma. Offline handwritten Gurmukhi character recognition: Analytical study of different transformations. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 87(1):137–143, March 2017. doi:10.1007/s40010-016-0284-y.
- [7] M. Kumar, R. K. Sharma, and M. K. Jindal. Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. *National Academy Science Letters*, 37(4):381–391, August 2014. doi:10.1007/s40009-014-0253-4.
- [8] R. S. Kunte and R. D. S. Samuel. A simple and efficient optical character recognition system for basic symbols in printed Kannada text. *Sadhana*, 32(5):521–533, October 2007. <http://www.ias.ac.in/article/fulltext/sadh/032/05/0521-0533>.
- [9] L. M. Lorigo and V. Govindaraju. Offline Arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):712–724, May 2006. doi:10.1109/TPAMI.2006.102.
- [10] U. Pal, T. Wakabayashi, and F. Kimura. Handwritten Bangla compound character recognition using gradient feature. In *Proc. 10th Int. Conf. Information Technology ICT 2007*, pages 208–213, December 2007. doi:10.1109/ICT.2007.62.

- [11] U. Pal, T. Wakabayashi, and F. Kimura. Comparative study of Devnagari handwritten character recognition using different feature and classifiers. In *Proc. 10th Int. Conf. Document Analysis and Recognition ICDAR 2009*, pages 1111–1115, July 2009. doi:10.1109/ICDAR.2009.244.
- [12] N. Sharma, U. Pal, F. Kimura, and S. Pal. Recognition of off-line handwritten Devnagari characters using quadratic classifier. In P. K. Kalra and S. Peleg, editors, *Proc. 5th Indian Conf. Computer Vision, Graphics and Image Processing ICVGIP 2006*, pages 805–816. Springer Berlin Heidelberg, Madurai, India, December 13-16, 2006. doi:10.1007/11949619_72.
- [13] D. C. Tran, P. Franco, and J. M. Ogier. Accented handwritten character recognition using SVM – application to French. In *Proc. 12th Int. Conf. Frontiers in Handwriting Recognition ICFHR 2010*, pages 65–71, November 2010. doi:10.1109/ICFHR.2010.16.
- [14] X. Wang, V. Govindaraju, and S. Srihari. Holistic recognition of handwritten character pairs. *Pattern Recognition*, 33(12):1967–1973, 2000. doi:10.1016/S0031-3203(99)00204-6.
- [15] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239, March 1984. doi:10.1145/357994.358023.
- [16] B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa. A robust model for on-line handwritten Japanese text recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(2):121–131, June 2010. doi:10.1007/s10032-009-0111-y.



Munish Kumar received his Master's degree in Computer Science & Engineering from Thapar University, Patiala, India, in 2008. He received his Ph.D. degree in Computer Applications from Thapar University, Patiala, India in 2015. He started his career as an Assistant Professor in computer application at Jaito centre of Punjabi University, Patiala. Presently, he is working as Assistant Professor in Department of Computer Applications, GZS Campus College of Engineering & Technology, Bathinda, Punjab, India. His research interests include Character Recognition and Pattern Recognition.



Prof. Manish Kumar Jindal received his Bachelor's degree in science in 1996 and Post Graduate degree in Computer Applications from Punjabi University, Patiala, India, in 1999. He holds a Gold Medal in his Post graduation. He received his Ph.D. degree in Computer Science & Engineering from Thapar University, Patiala, India in 2008. He is working as Professor in Panjab University Regional Centre, Muksar, Punjab, India. His research interests include Character Recognition.



Prof. Rajendra Kumar Sharma received his Ph.D. degree in Mathematics from the University of Roorkee (Now, IIT Roorkee), India, in 1993. He is currently working as Professor at Thapar University, Patiala, India, where he teaches, among other things, statistical models and their usage in computer science. He has been involved in the organization of a number of conferences and other courses at Thapar University, Patiala. His main research interests are statistical models in computer science, Neural Networks, and Pattern Recognition.



Simpel Rani Jindal received her Ph.D. degree in Computer Science from Punjabi University, Patiala, India, in 2016. She is working as Associate Professor in Yadavindra College of Engineering, Talwandi Sabo, Bathinda, Punjab, India. Her research interests include Character Recognition.

