

ISSN 1213-6972

Volume 24, Number 2, 2016

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

EDITOR – IN – CHIEF

Václav Skala

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
<http://www.VaclavSkala.eu>

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

Hardcopy: **ISSN 1213 – 6972**
CD ROM: **ISSN 1213 – 6980**
On-line: **ISSN 1213 – 6964**

Journal of WSCG

Editor-in-Chief

Vaclav Skala

c/o University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitni 8
CZ 306 14 Plzen
Czech Republic

<http://www.VaclavSkala.eu>

Journal of WSCG URLs: <http://www.wscg.eu> or <http://wscg.zcu.cz/jwscg>

Editorial Advisory Board MEMBERS

Baranoski,G. (Canada)

Benes,B. (United States)

Biri,V. (France)

Bouatouch,K. (France)

Coquillart,S. (France)

Csebfalvi,B. (Hungary)

Cunningham,S. (United States)

Davis,L. (United States)

Debelov,V. (Russia)

Deussen,O. (Germany)

Ferguson,S. (United Kingdom)

Goebel,M. (Germany)

Groeller,E. (Austria)

Chen,M. (United Kingdom)

Chrysanthou,Y. (Cyprus)

Jansen,F. (The Netherlands)

Jorge,J. (Portugal)

Klosowski,J. (United States)

Lee,T. (Taiwan)

Magnor,M. (Germany)

Myszkowski,K. (Germany)

Oliveira,Manuel M. (Brazil)

Pasko,A. (United Kingdom)

Peroche,B. (France)

Puppo,E. (Italy)

Purgathofer,W. (Austria)

Rokita,P. (Poland)

Rosenhahn,B. (Germany)

Rossignac,J. (United States)

Rudomin,I. (Mexico)

Sbert,M. (Spain)

Shamir,A. (Israel)

Schumann,H. (Germany)

Teschner,M. (Germany)

Theoharis,T. (Greece)

Triantafyllidis,G. (Greece)

Veltkamp,R. (Netherlands)

Weiskopf,D. (Germany)

Weiss,G. (Germany)

Wu,S. (Brazil)

Zara,J. (Czech Republic)

Zemcik,P. (Czech Republic)

Board of Reviewers

Abad,Francisco (Spain)
Adzhiev,Valery (United Kingdom)
Agathos,Alexander (Romania)
Ahmad,Khurshid (Ireland)
Akleman,Ergun (United States)
Amditis,Angelos (Greece)
Ammi,Mehdi (France)
Ammi,Mehdi (France)
Ariu,Davide (Italy)
Assarsson,Ulf (Sweden)
Aveneau,Lilian (France)
Ayala,Dolors (Spain)
Backfrieder,Werner (Austria)
Barthe,Loic (France)
Battiat,Sebastiano (Italy)
Baum,David (Germany)
Benes,Bedrich (United States)
Benger,Werner (United States)
Benoit,Crespin (France)
Biasotti,Silvia (Italy)
Bilbao,Javier,J. (Spain)
Biri,Venceslas (France)
Birra,Fernando (Portugal)
Bittner,Jiri (Czech Republic)
Bosch,Carles (Spain)
Bouatouch,Kadi (France)
Boukaz,Saida (France)
Bourdin,Jean-Jacques (France)
Bourke,Paul (Australia)
Bouville,Christian (France)
Bruckner,Stefan (Austria)
Bruder,Gerd (Germany)
Brun,Anders (Sweden)
Bruni,Vittoria (Italy)
Brunnett,Guido (Germany)
Buehler,Katja (Austria)
Bulo,Samuel Rota (Italy)
Buriol,Tiago Martinuzzi (Brazil)
Cakmak,Hueseyin (Germany)
Camahort,Emilio (Spain)
Casciola,Julio (Italy)

Chaine,Raphaelle (France)
Chaudhuri,Debasis (India)
Chen,Falai (China)
Chmielewski,Leszek (Poland)
Choi,Sunghee (Korea)
Chover,Miguel (Spain)
Chrysanthou,Yiorgos (Cyprus)
Chuang,Yung-Yu (Taiwan)
Cline,David (United States)
Coquillart,Sabine (France)
Corcoran,Andrew (Ireland)
Cosker,Darren (United Kingdom)
Daniel,Marc (France)
Daniels,Karen (United States)
de Amicis,raffaele (Italy)
de Geus,Klaus (Brazil)
de Oliveira Neto,Manuel Menezes (Brazil)
De Paolis,Lucio Tommaso (Italy)
Debelov,Victor (Russia)
Dingliana,John (Ireland)
Doellner,Juergen (Germany)
Dokken,Tor (Norway)
Drechsler,Klaus (Germany)
Durikovic,Roman (Slovakia)
Eisemann,Martin (Germany)
Erleben,Kenny (Denmark)
Falcidieno,Bianca (Italy)
Faudot,Dominique (France)
Feito,Francisco (Spain)
Ferguson,Stuart (United Kingdom)
Fiorentino,Michele (Italy)
Flaquer,Juan (Spain)
Fuenfzig,Christoph (Germany)
Gain,James (South Africa)
Galo,Mauricio (Brazil)
Garcia Hernandez,Ruben Jesus (Germany)
Garcia-Alonso,Alejandro (Spain)
Gavrilova,M. (Canada)
Gianelli,Carlota (Germany)
Giannini,Franca (Italy)
Gobron,Stephane (Switzerland)

Goebel,Martin (Germany)
Gonzalez,Pascual (Spain)
Grau,Sergi (Spain)
Gu,Xianfeng (United States)
GuĂŠrin,Eric (France)
Gudukbay,Ugur (Turkey)
Guthe,Michael (Germany)
Habel,Ralf (Switzerland)
Hall,Peter (United Kingdom)
Hansford,Dianne (United States)
Haro,Antonio (United States)
Hast,Anders (Sweden)
Hauser,Helwig (Norway)
Havemann,Sven (Austria)
Havran,Vlastimil (Czech Republic)
Hege,Hans-Christian (Germany)
Hernandez,Benjamin (United States)
Herout,Adam (Czech Republic)
Hicks,Yulia (United Kingdom)
Hildenbrand,Dietmar (Germany)
Hinkenjann,Andre (Germany)
Horain,Patrick (France)
Horain,Patrick (France)
House,Donald (United States)
Ihrke,Ivo (Germany)
Iwasaki,Kei (Japan)
Jeschke,Stefan (Austria)
Jiang,Jianmin (China)
Jones,Mark (United Kingdom)
Juan,M.-Carmen (Spain)
Juettler,Bert (Austria)
Kanai,Takashi (Japan)
Kim,H. (Korea)
Klosowski,James (United States)
Kohout,Josef (Czech Republic)
Kolcun,Alexej (Czech Republic)
Krueger,Jens (Germany)
Kumar,Subodh (India)
Kurillo,Gregorij (United States)
Kurt,Murat (Turkey)
Kyratzi,Sofia (Greece)
Lanquentin,Sandrine (France)
Larboulette,Caroline (France)
Lee,Jong Kwan Jake (United States)
Lengyel,Eric (United States)
Lien,Jyh-Ming (United States)
Lindow,Norbert (Germany)
Liu,SG (China)
Liu,Damon Shing-Min (Taiwan)
Lopes,Adriano (Portugal)
Loscos,Celine (France)
Lucas,Laurent (France)
Lutteroth,Christof (New Zealand)
Maciel,Anderson (Brazil)
Maddock,Steve (United Kingdom)
Magnor,Marcus (Germany)
Manak,Martin (Czech Republic)
Mandl,Thomas (Germany)
Manzke,Michael (Ireland)
Mas,Albert (Spain)
Masia,Belen (Spain)
Masood,Syed Zain (United States)
Matey,Luis (Spain)
Matkovic,Kresimir (Austria)
Max,Nelson (United States)
McDonnell,Rachel (Ireland)
McKisic,Kyle (United States)
Meng,Weiliang (China)
Mestre,Daniel,R. (France)
Metodiev,Nikolay Metodiev (United States)
Meyer,Alexandre (France)
Mokhtari,Marielle (Canada)
Molina Masso,Jose Pascual (Spain)
Molla,Ramon (Spain)
Montruccio,Bartolomeo (Italy)
Morigi,Serena (Italy)
Muller,Heinrich (Germany)
Murtagh,Fionn (United Kingdom)
Myszkowski,Karol (Germany)
Niemann,Henrich (Germany)
Nishita,Tomoyuki (Japan)
Okabe,Makoto (Japan)
Oliveira, Jr.,Pedro Paulo (Brazil)
Oyarzun Laura,Cristina (Germany)
Pala,Pietro (Italy)
Pan,Rongjiang (China)
Papaioannou,Georgios (Greece)
Paquette,Eric (Canada)
Pasko,Alexander (United Kingdom)
Pasko,Galina (United Kingdom)
Pastor,Luis (Spain)

Patane,Giuseppe (Italy)
Patow,Gustavo (Spain)
Pedrini,Helio (Brazil)
Pereira,Joao Madeiras (Portugal)
Perret,Jerome (France)
Peters,Jorg (United States)
Pettre,Julien (France)
Peytavie,Adrien (France)
Pina,Jose Luis (Spain)
Platis,Nikos (Greece)
Plemenos,Dimitri (France)
Post,Frits,H. (Netherlands)
Poulin,Pierre (Canada)
Praktikakis,Ioannis (Greece)
Puig,Anna (Spain)
Puppo,Enrico (Italy)
Rafferty,Karen (United Kingdom)
Reisner-Kollmann,Irene (Austria)
Renaud,Christophe (France)
Renaud,Christophe (France)
Reyes-Lecuona,Arcadio (Spain)
Richardson,John (United States)
Ritschel,Tobias (Germany)
Ritter,Marcel (Austria)
Rojas-Sola,Jose Ignacio (Spain)
Rokita,Przemyslaw (Poland)
Runde,Christoph (Germany)
Ruther,Heinz (South Africa)
Sacco,Marco (Italy)
Sadlo,Filip (Germany)
Sakas,Georgios (Germany)
Salvetti,Ovidio (Italy)
Sanna,Andrea (Italy)
Santos,Luis Paulo (Portugal)
Sapidis,Nickolas,S. (Greece)
Savchenko,Vladimir (Japan)
Schultz,Thomas (Germany)
Schumann,Heidrun (Germany)
Segura,Rafael (Spain)
Seipel,Stefan (Sweden)
Sellent,Anita (Switzerland)
Shesh,Amit (United States)
Sik-Lanyi,Cecilia (Hungary)
Slavik,Pavel (Czech Republic)
Sochor,Jiri (Czech Republic)
Solis,Ana Luisa (Mexico)
Sommer,Björn (Germany)
Sourin,Alexei (Singapore)
Sousa,Augusto (Portugal)
Sramek,Milos (Austria)
Sreng,Jean (France)
Staadt,Oliver (Germany)
Stricker,Didier (Germany)
Stroud,Ian (Switzerland)
Subsol,Gerard (France)
Suescun,Angel ()
Sunar,Mohd-Shahrizal (Malaysia)
Sundstedt,Veronica (Sweden)
Svoboda,Tomas (Czech Republic)
Szecsi,Laszlo (Hungary)
Tang,Min (China)
Tang,Qian (China)
Taubin,Gabriel (United States)
Tavares,Joao Manuel R.S. (Portugal)
Teschner,Matthias (Germany)
Theussl,Thomas (Saudi Arabia)
Tian,Feng (United Kingdom)
Tobler,Robert (Austria)
Todt,Eduardo (Brazil)
Tokuta,Alade (United States)
Torrens,Francisco (Spain)
Trapp,Matthias (Germany)
Triantafyllidis,Georgios (Greece)
Tytkowski,Krzysztof (Poland)
Umlauf,Georg (Germany)
Vanderhaeghe,David (France)
Vasa,Libor (Czech Republic)
Vazquez,Pere-Pau (Spain)
Vazquez,Pere Pau (United States)
Viola,Ivan (Austria)
Vitulano,Domenico (Italy)
Vosinakis,Spyros (Greece)
Walczak,Krzysztof (Poland)
WAN,Liang (China)
Wang,Charlie,C.L. (Hong Kong SAR)
Weber,Andreas (Germany)
Wenger,Raphael (United States)
Westermann,Ruediger (Germany)
Wu,Enhua (China)
Wu,Shin-Ting (Brazil)
Wuensche,Burkhard,C. (New Zealand)
Wuetrich,Charles (Germany)

Xin,Shi-Qing (Singapore)
Yoshizawa,Shin (Japan)
YU,Qizhi (United Kingdom)
Yue,Yonghao (Japan)
Zachmann,Gabriel (Germany)
Zalik,Borut (Slovenia)
Zara,Jiri (Czech Republic)
Zemcik,Pavel (Czech Republic)

Zhang,Xiaopeng (China)
Zhang,Xinyu (United States)
Zhu,Ying (United States)
Zillich,Michael (Austria)
Zitova,Barbara (Czech Republic)
Zwettler,Gerald (Austria)

Journal of WSCG

Vol.24, No.2, 2016

Contents

Maisch, S., Skanberg, R., Ropinski, T.: A Comparative Study of Mipmapping Techniques for Interactive Volume Visualization	35
Lee,D.Y., Kim,K.R.,Kim,T.Y.,Cho,H.G.*: Comparison-specialized Visualization Model for Whole Genome Sequences	43
Azam, S., Gavrilova, M.: Gender Prediction using Individual Perceptual Image Aesthetics	53
Svoboda,P., Hradis,M., Barina,D., Zemcik,P.: Compression Artifacts Removal Using Convolutional Neural Networks	63

A Comparative Study of Mipmapping Techniques for Interactive Volume Visualization

Sebastian Maisch

Ulm University

sebastian.maisch@uni-ulm.de

Robin Skånberg

Ulm University

robin.skanberg@uni-ulm.de

Timo Ropinski

Ulm University

timo.ropinski@uni-ulm.de

Abstract

Many large scale volume visualization techniques are based on partitioning the data into bricks, which are stored and rendered using mipmaps. To generate such mipmaps, in most cases an averaging is applied such that an area in a lower mipmap level is presented by the areas' average in the next higher mipmap level. Unfortunately, this averaging results in the fact that mipmaps are not feature-preserving, as details are often lost. In this paper, we discuss and compare mipmap modification schemes which have been developed to support feature-preserving reconstruction during rendering. In particular, we focus on reconstruction schemes which are capable to support anisotropic and non-linear reconstruction, as these are promising to preserve features that are often sacrificed by averaging. The presented techniques are discussed in detail and are thoroughly compared in a quantitative and qualitative analysis. We will discuss their impact on performance, memory footprint and visual quality with respect to feature preservation. Based on the findings we present guidelines for generating and using mipmaps in various visualization scenarios.

Keywords

Volume Rendering, Mipmapping

1 INTRODUCTION

Large data in volume visualization is a very common case in modern visualization [1]. With current methods of data acquisition the resolution of volume data gets too big to be handled in a straightforward way, even on modern graphics hardware. To be able to handle these amounts of data, multi-resolution techniques are often used. Besides plain bricking, most of these techniques rely on downsampling at least parts of the volume to achieve lower resolutions in more distant parts of the volume [2]. This results in a smaller memory footprint of the data to be rendered. Although we will compare different three dimensional downsampling strategies towards mipmapping, our results can also be applied to other approaches in large volume rendering, especially multi-resolution techniques. Mipmaps were originally introduced in computer graphics for two dimensional textures [3] and have later also been extended to volumetric data sets (e.g., [4]). Volume visualization benefits from mipmapping, as it reduces aliasing problems and at the same time lowers the amount of memory necessary for rendering.

While most modern volume visualization algorithms rely on mipmaps (or similar techniques for data down-sampling) to support large data [2], mipmaps do not

preserve the underlying data very well and thus details are often lost. We identify two main reasons for this loss: the representation of the entire data range of a cell to a single average value, and the disregard of directional information during the averaging process. Due to the importance of mipmapping in large-scale volume visualization, we investigate and analyze mipmapping modifications, which tackle these shortcomings. We focus on those approaches which do maintain a low memory footprint and enable a direct interactive transfer function change, without requiring extra computations. We see those two criteria as essential capabilities for a technique which shall be scalable to large-scale data. Furthermore, as mipmapping can be considered sufficient for the first downscaling level, where linear interpolation would be used on the GPU, we primarily focus on higher levels, where averaging forbids feature-preservation in the final visualization.

In this paper, we compare two groups of mipmapping techniques in order to investigate their impact on the averaging and directional issues as mentioned above. The first method is named '*Non-Linear Reconstruction*' and uses an enhancement of mipmaps that not only stores the average value of each lower resolution voxel but also minimum and maximum values of the covered area. During rendering, we use these values to reconstruct a power function inside each voxel that tries to resemble the intensity distribution.

With '*Anisotropic Mipmaps*' we try to encode directional information in lower resolution volumes to be able to better reconstruct anisotropy inside each voxel. We use piecewise linear functions to achieve this which are also evaluated during rendering. For each of these techniques we compare different variants with respect to performance, memory footprint and image quality.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

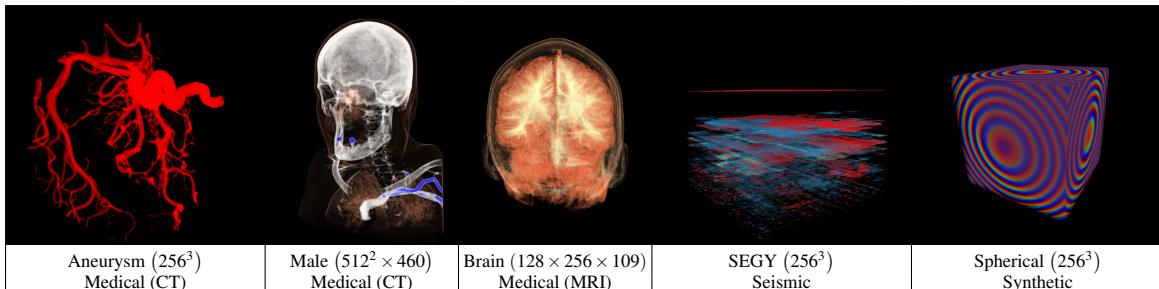


Table 1: Overview of the datasets used throughout the paper; the name, dimensions, and acquisition modality are listed. The images show renderings of the highest resolution with the respective transfer functions applied.

2 RELATED WORK

Today, several different approaches exist to support rendering of multi-resolution volumes. On a data structure level, octrees and kd-trees are widely used techniques to handle large-scale volumetric data sets [2, 5]. Other techniques use hierarchical grids to represent the data [6, 7]. In both cases mipmaps [3] are commonly used for storing and downsampling the volume.

In the recent past several approaches have been put forward to improve the visual quality of mipmapping by trying to reconstruct the intensity distribution of the region covered by the downsampled voxel. Such approaches are based on Gaussians [8] or sparse PDFs [9, 10]. The latter group of techniques uses a hierarchical representation of the data making use of Gaussians to resemble the data range. This comes at the need of performing calculations with every change in transfer function. Independent of the method used to store the intensity distribution, the memory footprint of these methods is larger than with plain mipmaps. Therefore, we have omitted these approaches from our comparison, as we aim for a low memory footprint while still being able to edit the transfer function during rendering without a significant computational overhead. Instead, we investigate an improved aggregation technique which is similar to the Min-/Maxmaps used for 2D shadow mapping by Guennebaud et al. [11] and for storing minimum and maximum positions of geometry in Mipmaps by Carr et al. [12]. Thus, to reconstruct the data distribution of each voxel, we use a volume texture storing for each voxel the minimum, maximum, and average values of the region they cover. We found this relevant, because the usage of minimum and maximum values in volume rendering is a frequently used approach. Lacroute and Levoy [13] for instance introduced Min-Max Octrees in which each octree node stores a minimum and maximum value of the contained volume to allow empty space skipping. A similar concept is used by Dong et al. [14]. As many of the features suppressed in standard mipmapping have a directional nature, we also look into encoding the change of intensity within a downsampled voxel by using a technique similar to deep shadow mapping [15]. Thus, anisotropic features can be captured with anisotropic voxels [16, 17].

Kraus and Ertl [18] introduced an approach to downsampling that tried to preserve the topology of isosur-

faces. Their memory requirements are the same as conventional mipmapping, but there are cases where it is not possible to preserve the topology, for example, if there are more than one local minima or maxima in the original resolution data that corresponds to a single downsampled voxel. Later Kraus and Ertl have addressed a similar issue for downsampling RGBA volume data [19]. We have not included these approaches in our evaluation, as the first one is intended for extracting isosurfaces from a downsampled volume. While some ideas as the need to preserve extreme intensity values also apply to raycasting volumes this approach is not entirely suitable. The latter technique is only applicable to volumes that evaluate the transfer function as a preprocessing step which we want to avoid, because of the inability to change the transfer function without recomputing the volume.

While we will not focus on compression techniques, we would like to mention that they can be combined with the tested techniques to further reduce the required memory footprint. Often, for instance, wavelet representations are used to compress large volumes and thus reduce the loss of information [20, 21]. These or other techniques, may also be combined with the approaches discussed in this paper.

3 METHODOLOGY

The evaluation presented in this paper is motivated by two common problems that arise when using mipmaps. One problem that is also addressed in other papers is the loss of information about the intensity distribution due to averaging the voxels intensity values. We use a simple non-linear reconstruction approach to address this issue and show, that even under very harsh conditions and high frequency transfer functions, we can provide convincing results. We have selected the method described in Section 4.1 among the known techniques addressing similar issues as it is simple to realize and can be directly integrated into existing volume renderers. Furthermore, it fulfills our requirements with respect to low memory footprint and the flexibility of post-classification. Kraus and Ertl [18] point out the importance of preserving the topology of isosurfaces within the volume. This includes the ability to reconstruct the extreme values for each downsampled voxel. We do not target the generation of isosurfaces, but the importance of keeping the extreme values for a better reconstruction of intensity values is still applicable.

The second problem when using mipmaps is that directional information is lost, also due to the averaging which occurs in the process of generating the mipmaps. To deal with this issue, we aim at evaluating how the encoding of directional intensity changes can improve feature preservation. The idea is to derive a more accurate intensity value from this directional information, to which we then apply the transfer function during rendering. The underlying method for realizing this encoding is described in Section 4.2.

In this paper, the i -th mipmap level is specified using the convention ℓ_i where ℓ_0 represents the full resolution data. For each subsequent level ℓ_{i+1} the resolution is halved compared to the previous level ℓ_i .

By taking the two exemplary techniques into account, we want to evaluate the impact of non-linear reconstruction vs. anisotropic reconstruction, when using mipmaps in volume rendering. Based on our own observations as well as the results reported in recent papers, e.g., [10], we consider the mipmap levels ℓ_0 and ℓ_1 (the highest and second highest resolution) sufficient in terms of quality of the resulting image, and therefore focus on the higher levels. The representative approaches are implemented inside a basic raycaster for volume data which incorporates early ray termination.

Our testing environment contains five different data sets (see Table 1). We use several volumes from the medical domain. We also include a seismic data set, and one synthesized by using a simple spherical function (as was also used by Younesy et al. [8]). These different volumes represent a wide range of different applications. Transfer function design is discussed in Appendix A.

The measurements we used to determine the quality of each technique in the different scenes are as follows. We measure memory footprint and performance compared to rendering a full resolution volume and the original mipmapping. We also take into account the error rates (PSNR) of the images in comparison to an image produced using the highest resolution data as a ground truth. In terms of quality we will look at different features of the datasets that are visible if we render the high resolution data and discuss how well these are preserved in downscaled renderings.

4 EVALUATED RECONSTRUCTION TECHNIQUES

In this section, we describe the two techniques we use to improve reconstruction of downsampled values in different situations. One technique aims to preserve the intensity distribution of each downsampled voxel using a non-linear function. We will call this technique ‘non-linear reconstruction’. The other approach is to preserve a directional distribution of values in downsampled voxels. This technique will be called ‘anisotropic reconstruction’.

4.1 Non-Linear Reconstruction

To reconstruct the intensity distribution of a downsampled voxel in a non-linear manner, we want to find a

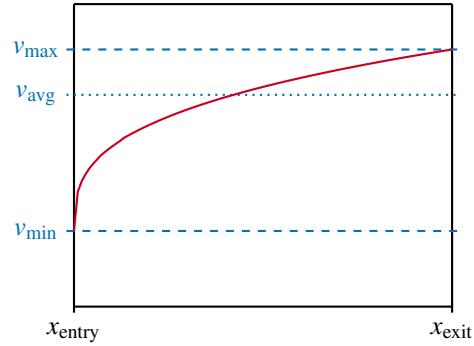


Figure 1: Reconstruction function (red) inside a voxel defined by the ray entry (x_{entry}) and exit (x_{exit}) points and the minimum (v_{\min}), maximum (v_{\max}) and average (v_{avg}) values of the covered high resolution voxels in ℓ_0 .

simple representation of that distribution encoded using low memory footprint. Using only three values – the minimum, maximum, and average in a specified region, we fit a power function to these values covering the interval between the minimum and the maximum value while still preserving the average. To be able to create such a representation during rendering we need to preprocess the volume data and store for each downsampled voxel the minimum (v_{\min}), maximum (v_{\max}), and average (v_{avg}) value of that voxel’s equivalent region in the original volume.

When ray-casting the volume represented by ℓ_2 or lower we use these values to reconstruct a non-linear function representing the intensity distribution inside each voxel. Younesy et al. [8] use a Gaussian to reconstruct this value from the mean and standard deviation values they store. They then use a preintegrated transfer function to calculate the color value at each voxel. Our method works without any precalculations directly using the transfer function. To generate the final color value we sample the reconstructed function, apply the transfer function to each value, and composite the colors using the scheme determined by the raycasting algorithm. Therefore the reconstructed function is interpreted as a one dimensional function along the ray used to calculate the current pixels color. This introduces an error as we have no actual information about the locations of each value contained in the intensity distribution, but imply these locations by using our reconstruction. However, as Kraus and Ertl have discovered [18], this omission of the actual sample orders can be in most cases neglected. The function we use for reconstruction, is a scaled power function:

$$f(x) = (v_{\max} - v_{\min}) \left(\frac{x - x_{\text{entry}}}{x_{\text{exit}} - x_{\text{entry}}} \right)^a + v_{\min} \quad , \quad (1)$$

where x_{entry} is the rays entry point into the voxel and x_{exit} is its exit point. The parameter a is determined by the voxels average value (v_{avg}) to ensure the average of $f(x)$ matches that of the voxel. Figure 1 illustrates this function and the parameters used for its reconstruc-

tion. This parameter is calculated on the fly during ray-casting by solving the definition of the average value:

$$v_{\text{avg}} = \frac{1}{x_{\text{exit}} - x_{\text{entry}}} \int_{x_{\text{entry}}}^{x_{\text{exit}}} f(x) dx , \quad (2)$$

which gives us a as follows:

$$a = \frac{v_{\max} - v_{\min}}{v_{\text{avg}} - v_{\min}} - 1 \quad (3)$$

The impact of minimum, maximum and average value on the shape of the function is illustrated in Figure 1.

The resulting function is then evaluated at different positions within the voxel and the transfer function is applied to these results. The final values are then composited in a straightforward way by using front to back composition. To further improve the coverage of the intensity distribution inside the voxel we adapt the step size we use to fit the function. The same step size is used for the composition. In general we use values that are evenly distributed across the intensity range, whereby we use the minimum and maximum values to determine the possible range of values. When sampling the function at N different positions inside the voxel we use the values v_i ($i = 0 \dots N - 1$) defined as $v_i = \frac{i}{N-1} (v_{\max} - v_{\min}) + v_{\min}$ and calculate the step sizes needed to correctly weight those values during composition. The corresponding sampling points for the values v_i are x_i defined as:

$$x_i = \sqrt[a]{\frac{i}{N-1}} (x_{\text{exit}} - x_{\text{entry}}) + x_{\text{entry}} \quad (4)$$

With these points we use the following equation to calculate the step sizes (Δ_i) for ray-casting by using the midpoints between two sampling points to separate steps:

$$\Delta_i = \min\left(\frac{x_{i+1} + x_i}{2}, x_{\text{entry}}\right) - \max\left(\frac{x_i + x_{i-1}}{2}, x_{\text{exit}}\right) \quad (5)$$

Figure 2 illustrates the different v_i , x_i and Δ_i determined by a given function $f(x)$.

To directly see the impact of the non-linear density distribution reconstruction, we will also evaluate the same approach, whereby we replace the non-linear with a linear function. This approach only uses the minimum and maximum values, and we will refer to it as ‘Linear Reconstruction’ throughout this paper.

4.2 Anisotropic Reconstruction

The second identified downside of plain mipmapping is the fact, that directional features are not preserved. To deal with this issue, intensity changes for selected

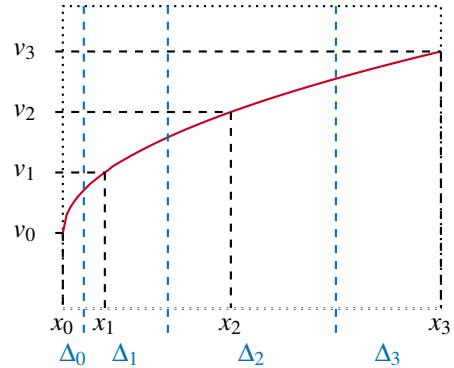


Figure 2: Step sizes (blue) for raycasting a reconstructed function (red) with $N = 4$ sampling points. The sampled values v_i are shown with their corresponding sampling positions x_i .

directions in each region would need to be encoded. This idea forms the basis for the evaluated anisotropic approach, where we – for each major axis within a downsampled voxel – encode how the intensity changes along the axis, and store this information as a piecewise linear function. This approach requires an explicit pre-processing step where intermediate axis functions are computed and later encoded into piecewise linear functions. These functions can later in the rendering part be sampled and composited into intensity values. In the following three subsections, we discuss how to compute and encode the axis functions, as well as how to combine values of the three piecewise linear functions into one representative intensity value.

4.2.1 Computing Axis Functions

To compute the axis functions, we begin by studying each of the major axes of a voxel which we intend to downsample (Fig. 3a). The three axes are split into stacks of 2D slices (Fig. 3b), and the mean of each slice is calculated and stored to represent the overall intensity at this location (Fig. 3c). This is always done at the highest possible resolution inside of the region of voxels to capture as many details as possible of the underlying data.

4.2.2 Encoding Axis Functions

With the axis functions represented as lists of N intensity values (Fig. 3c), we aim to compress these using a piecewise linear function *PWLF*. In our case, a piecewise linear function is a function that sparsely stores a set of 2D sample-points (x, y) and the function value can be evaluated at any position x using linear interpolation between the neighbouring points of x . In order to encode the list of intensity values into a *PWLF*, we employ the following greedy algorithm:

1. We initialize the *PWLF* (Fig. 3d) by adding the full set of axis intensity values with their corresponding positional values along the axis.

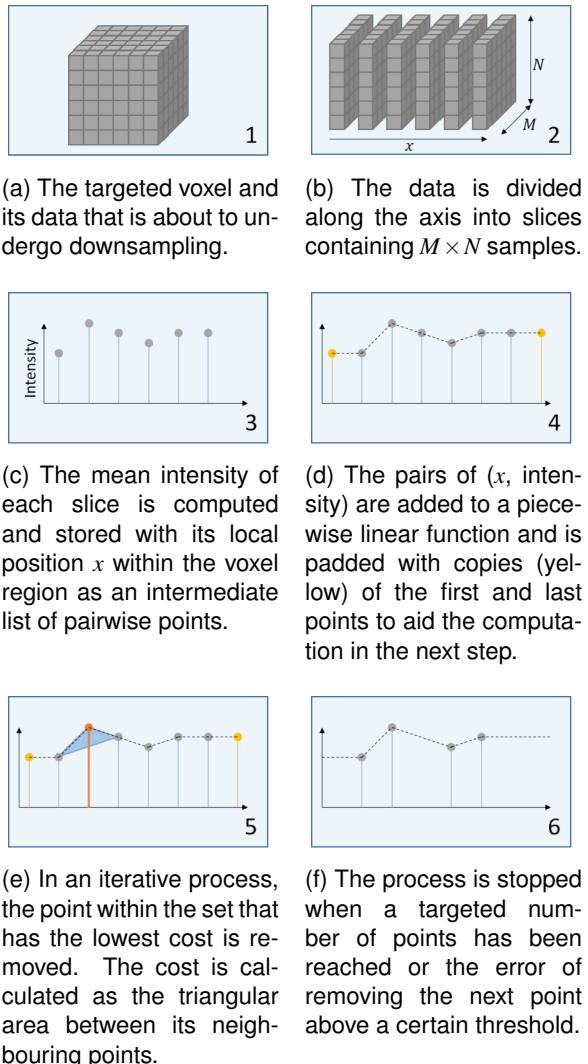


Figure 3: The preprocessing step of computing a piecewise linear function that approximates the intensity values along one major axis.

2. We then remove the point in the set which is deemed as the least destructive to overall shape of the function using a cost metric (Fig. 3e). The cost of removing a point is equal to the triangular area which spans the point and its two closest neighbours before and after removal of this point.
3. Step 2 is repeated until we are left with a desired amount of points (Fig. 3f).

4.2.3 Sub-sampling Anisotropic Voxels

When traversing the voxels a sub-sampling approach is used to reconstruct the intensity values within the voxel using the piecewise linear functions stored within it (see Fig. 4). The input arguments for the PWLFs are the local coordinates ($x, y, z \in [0, 1]$) of the sub-sample within the frame of the voxel. The intensity values (Eq. 7) returned by the three PWLFs are then combined into one single intensity value using the dot product of the squared viewing vector \mathbf{d} and intensity values $v(\mathbf{x})$ as shown in Eq. 6

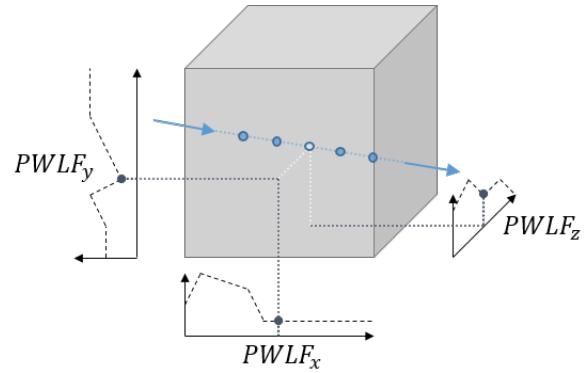


Figure 4: A sub-sample (Light-blue dot) along the traversed ray (Blue) is being composited into a representative value of the intensity within a voxel by using the piecewise linear functions that provide the directional intensity along each major axis.

$$\text{intensity}(\mathbf{x}) = \text{dot}(\mathbf{d} \odot \mathbf{d}, v(\mathbf{x})) \quad , \quad (6)$$

where

$$v(\mathbf{x}) = \{PWLF_x(x_x), PWLF_y(x_y), PWLF_z(x_z)\} \quad . \quad (7)$$

The major downside of this technique is that it does not allow trilinear interpolation. If trilinear interpolation is used, the resulting intensity values from all of the piecewise linear functions will be averaged over the neighbouring region of the voxel thus resulting in an average intensity, almost indistinguishable from plain mipmapping. This forces us to only use values from within the traversed voxel disregarding the neighbourhood of it. The visual impact of this is that the boundaries of the voxels can be seen in most of the cases.

5 EVALUATION

In this section we analyze the impact of the mipmap shortcomings resulting from the area averaging and the non-directional reconstruction. We do so by comparing standard mipmapping against non-linear and anisotropic reconstruction, as they have been described in the previous section. The achieved results are evaluated quantitatively and qualitatively. A more comprehensive list of results can be found in Appendix B.

5.1 Quantitative Evaluation

The presented techniques are meant to produce better images than standard mipmapping while still using less GPU memory and being faster as compared to rendering the highest resolution data. For performance measurements we ray-casted the volume data with a screen resolution of 1280×720 with the volume fitting roughly to the screen, whereby we have used the same images in the subsequent error analysis. The results of the performance evaluation, as summarized in Table 2, have been measured on a 3.60 GHz Intel i7 Haswell system with 16GB RAM and a NVIDIA GTX 980 GPU.

As it can be seen in Table 2, the results of non-linear reconstruction highly depends on the size of the volume.

Dataset	Non-Linear	Anisotropic	Mipmaps
Aneurysm (ℓ_2)	76%	60%	122%
Male (ℓ_3)	232%	51%	385%
Brain (ℓ_2)	69%	16%	109%
Spherical (ℓ_3)	138%	90%	207%
SEGY (ℓ_2)	102%	16%	150%

Table 2: Performance measurements of different techniques relative to rendering at the original resolution (ℓ_0). A value of 200% signifies that the technique is twice as fast as the reference, while a value of 50% signifies that only half the speed of the reference was achieved.

For volumes that use a lot of memory (see Table 1) in the first place texture fetches are more expensive (due to cache misses) and a reduction of the volumes’ resolution improves the frame rates drastically. For lower resolution volumes this is not the case, and the computation time overhead for the evaluation of the reconstructed function seems to have more impact. The rendering performance of the ‘Spherical’ data set is in general rather high, as the chosen transfer function makes the volume appear very dense which triggers the early ray termination.

The performance results of the anisotropic voxels seem worse than rendering the high resolution data, this is due to a bigger number of texture fetches which are needed to retrieve the data of the piecewise linear functions. The memory requirements for the anisotropic voxels is noticeably larger than with the other techniques, but has been implemented in a very naïve way where one voxel stores three unique piecewise linear functions. In practice, many of the piecewise linear functions are identical or very similar, which makes them a good candidate for lossless or lossy compression through clustering.

With no compression applied the memory requirements for the tested techniques directly depend on the resolution of the original data. The piecewise linear functions of the anisotropic mipmaps are always represented using eight 8-bit integers per pixel – four for position and four for intensity. The non-linear reconstruction on the other hand uses the same bit-depth as the original volume. Thus, the memory requirements of the reconstruction methods (as well as regular mipmapping) are a fixed percentage of the original memory (118% for non-linear reconstruction, 116% for linear reconstruction and 114% for regular mipmapping), whereas the anisotropic mipmaps use 156% of the original data sets memory for 8-bit volumes. The percentage gets lower for higher bit-depths of the original volume due to the used 8-bit quantization.

The more interesting figures are the error measurements of the rendered final images. The images can be seen in Appendix B. We calculated the Peak Signal to Noise Ratio (PSNR) of the resulting images, by taking into account the images produced with the high resolution data sets as a ground truth. We measured these values

for both, a high resolution (1280×720 Pixels) as well as a low resolution (128×72) image. The high resolution was chosen to be able to clearly see differences and errors introduced by both types of methods. The lower resolution represents the actual size of the images when the size of one voxel in the data set should correspond to the size of one pixel, whereby the mipmap levels have been chosen accordingly. To avoid aliasing we use downsampled versions of the high resolution images. The results are shown in Table 3.

In general, the lower resolution images show better quality with respect to the PSNR because quality shortcomings depending on the resolution are avoided. These differences can be seen best for the anisotropic mipmaps as this technique does not support interpolation and thus produces ‘blocky’ results when rendering them in a resolution higher than they are meant for. When comparing the different reconstruction techniques the non-linear reconstruction provides the best results in nearly all cases. The exception is the Spherical data set where the intensity values are linear with the distance from the center. The linear reconstruction ensures that this linearity is preserved well while the non-linear reconstruction might introduce some errors.

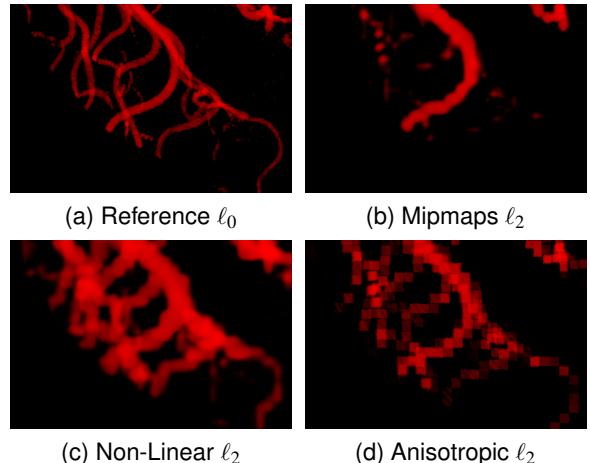


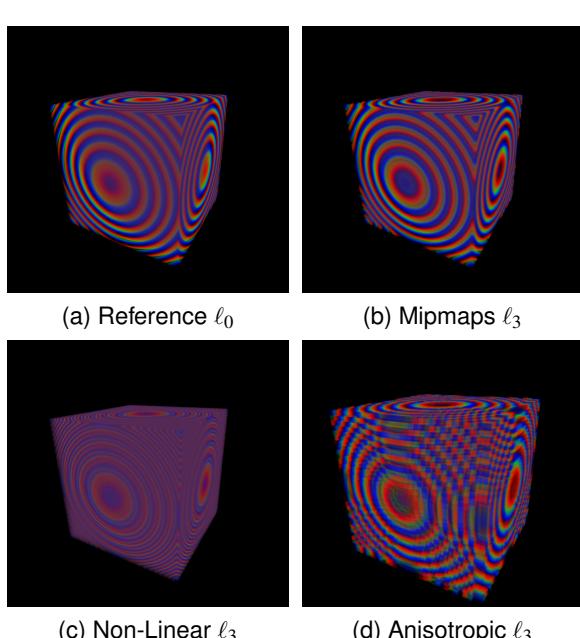
Figure 5: The Aneurysm dataset, rendered at its original resolution ℓ_0 (a) and at resolution ℓ_2 for the down-sampled methods. It shows a region with a lot of high frequency details.

5.2 Qualitative Evaluation

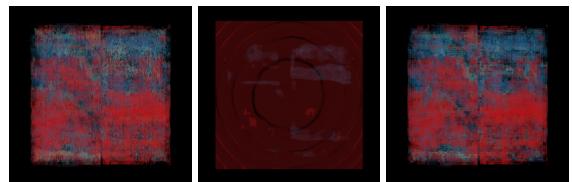
By visually comparing the results of the different techniques we can make assumptions about how to address mipmapping problems in general. In Figures 5 and 7 the plain mipmapping clearly shows a loss of information. In Figure 5b the high frequency details are lost due to averaging. The most obvious case is the seismic data set. There nearly all information is lost as can be seen in Figure 7b. When looking at the Spherical data set in Figure 6b most features seem to be preserved. However, when taking a closer look the colors observed seem to be exaggerated and are shifted towards the corners of the volume.

Dataset	Non-Linear		Linear		Anisotropic		Mipmaps	
	high Resolution	low Resolution						
Aneurysm (ℓ_2)	20.62 dB	22.18 dB	16.23 dB	17.00 dB	24.49 dB	30.35 dB	23.82 dB	26.95 dB
Male (ℓ_3)	21.02 dB	22.25 dB	18.13 dB	18.84 dB	22.86 dB	24.78 dB	18.96 dB	19.78 dB
Brain (ℓ_2)	26.20 dB	28.33 dB	23.60 dB	24.77 dB	25.05 dB	26.60 dB	23.27 dB	24.79 dB
Spherical (ℓ_3)	23.04 dB	27.45 dB	24.09 dB	28.88 dB	19.93 dB	23.52 dB	22.02 dB	26.04 dB
SEGY (Top) (ℓ_2)	22.23 dB	22.96 dB	21.36 dB	21.98 dB	26.97 dB	32.04 dB	17.69 dB	17.97 dB

Table 3: Error ratios (PSNR) of different techniques, datasets and resolutions.

Figure 6: The Sphere dataset, rendered at its original resolution ℓ_0 (a) and at resolution ℓ_3 for the downsampled methods. The fading to grey in Non-Linear reconstruction (c) is an expected artifact to avoid aliasing.

From a qualitative point of view, the anisotropic voxels seem to preserve details better over the plain mipmapping technique, but have the downside of looking ‘blocky’. This ‘blockiness’ comes from the fact that doing trilinear interpolation is too expensive in terms of performance to be practical, and if applied, the intensity is again averaged over the neighbouring region and the result ends up being indistinguishable from mipmapping. A similar problem arises when the different directions need to be interpolated due to a view that is not parallel to the volumes coordinate axes. This can be seen clearly in the geological data set. When choosing a view that is parallel to the volumes up coordinate as can be seen in Figure 7c the results are very similar to the reference image. When dealing with high frequencies as in the Aneurysm data set, anisotropic approaches also generate a good reconstruction of most details as can be observed in Figure 5d. The worst scenario is presented in Figure 6d. Here, preservation of the anisotropy is not important in the final rendering as the volume is very dense. Due

Figure 7: The SEGY dataset, rendered from a top view at its original resolution ℓ_0 (a) and at resolution ℓ_2 for the downsampled methods. Plain mipmapping does not preserve the structure of this dataset. From this perspective the Anisotropic Mipmapping (c) provides results that are very similar to the reference image.

to the interpolation of axis functions the result is very similar to the plain mipmapping.

When studying the non-linear reconstruction approaches, the different regions classified by the transfer function are well preserved. Due to the lower resolution of the volume these areas seem bigger in all cases. This can be seen especially well in Figure 5c where the high frequency details are rendered as bigger structures. This also leads to more saturated colors in nearly all of the results achieved using this approach. Several images show that preserving not only the extreme values but also the average value is important to create convincing results. In nearly all cases the non-linear reconstruction which preserves the average value provides results that are closer to the ground truth. The only exception is the synthetic Spherical data set which only contains intensity values linear in respect to the distance to the volumes center. This explains the close fit that a linear reconstruction provides.

5.3 Guidelines

We have seen different approaches which in general provide better quality compared to plain mipmapping depending on the type of data and the transfer function used. Based on these observations we provide guidelines for using the mentioned approaches in different scenarios.

As mentioned above the non-linear reconstruction exaggerates the color saturation. Generally the regions colored by a transfer function are the ones with features that the observer wishes to emphasize. This may prove useful in cases like seismic visualization to use these or similar approaches to be able to observe all details

in the volume. The same applies to medical datasets where it may be crucial for the observer not to miss any details of significance. In these cases using a non-linear reconstruction will benefit the visualization.

In the cases where you want to preserve high frequencies within data sets, an anisotropic approach may be a viable option. And if the data set contains intensities layered along one major direction, an anisotropic approach may also provide better results.

If rendering speed is the top priority plain mipmaps are the most obvious choice to use due to the native hardware integration existent in all modern GPUs. The Non-Linear Reconstruction provides reasonable speed especially with bigger datasets while providing an improved image quality.

When dealing with large volumes the overall difference in memory between plain mipmaps and techniques better preserving the intensity distribution of downsampled voxels is very small. Using such a technique may be a viable option in those cases.

6 CONCLUSION & FUTURE WORK

We have discussed two shortcomings of plain mipmaps and evaluated techniques that address these using different data sets from different domains together with transfer functions that have been designed to visualize interesting features in these volumes. As a conclusion of the conducted evaluation, we can say that all tested approaches can be implemented as an improvement to plain mipmapping. Also, conserving the intensity distribution of a downsampled voxel looks like a very promising approach that is applicable in the general case. An anisotropic approach also provides promising results but lacks hardware support and a proper solution for feature preserving interpolation in the technique presented in this paper. The different strengths of the provided techniques calls for a combination of both approaches that should provide a reconstruction of the intensity distribution of downsampled voxels as well as a directional component.

In our opinion, the ideal technique for mipmaps in volumetric visualization should fulfill three important requirements. First, it should preserve the intensity distribution of all the data contained within the region of the voxel. Second, it should utilize the GPU's hardware support for interpolation when fetching data. And third, the intensity distribution contained within the voxel should be able to be integrated with the transfer function in an efficient manner enabling interactive editing of the transfer function together with interactive frame rates.

As it seems to be more important to preserve the intensity distribution in a correct way rather than to store the directional intensity changes within a voxel an interesting approach would be to approximate this distribution by using piecewise linear functions. On the other hand the directional interpolation problems of Anisotropic Mipmaps could be tackled encoding directional intensity using spherical harmonics.

REFERENCES

- [1] E. W. Bethel, H. Childs, and C. Hansen, *High Performance Visualization: Enabling Extreme-Scale Scientific Insight*. CRC Press, 2012.
- [2] J. Beyer, M. Hadwiger, and H. Pfister, "State-of-the-Art in GPU-Based Large-Scale Volume Visualization," *CGF*, May 2015.
- [3] L. Williams, "Pyramidal parametrics," *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 3, pp. 1–11, Jul. 1983.
- [4] E. C. La Mar, B. Hamann, and K. I. Joy, "Multiresolution Techniques for Interactive Texture-based Volume Visualization," in *VIS '99*. IEEE Computer Society, Oct. 1999.
- [5] C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann, "GigaVoxels: Ray-Guided Streaming for Efficient and Detailed Voxel Rendering," in *I3D '09*. ACM Press, Feb. 2009, p. 15.
- [6] M. Hadwiger, J. Beyer, and H. Pfister, "Interactive Volume Exploration of Petascale Microscopy Data Streams Using a Visualization-Driven Virtual Memory Approach," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2285–2294, Dec. 2012.
- [7] T. Fogal, A. Schiewe, and J. Krüger, "An Analysis of Scalable GPU-Based Ray-Guided Volume Rendering," *IEEE Symposium on Large-Scale Data Analysis and Visualization*, vol. 2013, pp. 43–51, Oct. 2013.
- [8] H. Younesy, T. Möller, and H. Carr, "Improving the quality of multi-resolution volume rendering," in *IEEE VGTC '06*. Eurographics Association, May 2006, pp. 251–258.
- [9] M. Hadwiger, R. Sicat, J. Beyer, J. Krüger, and T. Möller, "Sparse PDF maps for non-linear multi-resolution image operations," *ACM TOG*, vol. 31, no. 6, p. 1, Nov. 2012.
- [10] R. Sicat, J. Krüger, T. Möller, and M. Hadwiger, "Sparse PDF Volumes for Consistent Multi-Resolution Volume Rendering." *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2417–26, Dec. 2014.
- [11] G. Guennebaud, L. Barthe, and M. Paulin, "Real-time soft shadow mapping by backprojection," in *EGSR '06*. Eurographics Association, Jun. 2006, pp. 227–234.
- [12] N. A. Carr, J. Hoberock, K. Crane, and J. C. Hart, "Fast GPU ray tracing of dynamic meshes using geometry images," in *GI '06*. Canadian Information Processing Society, Jun. 2006, pp. 203–209.
- [13] P. Lacroute and M. Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation," in *SIGGRAPH '94*. ACM Press, Jul. 1994, pp. 451–458.
- [14] F. Dong, M. Krokos, and G. Clapworthy, "Fast Volume Rendering and Data Classification Using Multiresolution in Min-Max Octrees," *CGF*, vol. 19, no. 3, pp. 359–368, Sep. 2000.
- [15] T. Lokovic and E. Veach, "Deep shadow maps," in *SIGGRAPH '00*. ACM Press, 2000, pp. 385–392.
- [16] J. Mitchell, G. McTaggart, and C. Green, "Shading in valve's source engine," in *ACM SIGGRAPH 2006 Courses*. ACM, 2006, pp. 129–142.
- [17] C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann, "Interactive indirect illumination using voxel cone tracing," *Computer Graphics Forum (Proceedings of Pacific Graphics 2011)*, vol. 30, no. 7, sep 2011.
- [18] M. Kraus and T. Ertl, "Topology-Guided Downampling," in *Volume Graphics 2001*, K. Mueller and A. E. Kaufman, Eds. Springer Vienna, 2001, pp. 223–234.
- [19] M. Kraus and K. Bürger, "Interpolating and Downsampling RGBA Volume Data." in *VMV*, 2008, pp. 323–332.
- [20] S. Muraki, "Volume data and wavelet transforms," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 50–56, Jul. 1993.
- [21] R. Westermann, "A multiresolution framework for volume rendering," in *VVS '94*. ACM Press, Oct. 1994, pp. 51–58.

Comparison-specialized Visualization Model for Whole Genome Sequences

Da-Young Lee Kyung-Rim Kim Taeyong Kim Hwan-Gue Cho*

Dept. of Electrical and Computer Engineering
Pusan National University, South Korea

{schematicque, alflskfl, ktyong22, hgcho}@pusan.ac.kr

ABSTRACT

Analyzing and visualizing the whole genome sequence is very important to finding genetic evolution. Many researchers have used 2D or 3D DNA random walk plots to study short DNA sequences. However, visualizing a whole genome sequence is difficult because of overlapping, self-intersection, and biases. In this paper, we propose a 3D graphical representation of a whole-genome sequence based on a random walk plot. Our 3D graphical representation can reduce the overlaps or biases that can occur during the visualization of large sequences by using the 2D or 3D DNA walk plot algorithm. We visualized and compared data on the whole genomes of 10 species, including humans and anthropoid apes. In our experiment, the 3D graphical representation showed similarities between humans and apes and differences between other species.

Keywords

DNA visualization, random walk, DNA similarity

1 MOTIVATION

By the late 1990s, genetic maps were developed by using genome projects based on Sanger's method of gene analysis. A great deal of research was carried out. Since then, next-generation sequence (NGS) techniques such as Roche 454 and Illumina have been developed, and mass gene data have become available. Now, gene data consisting of billions of base points can be obtained in only a few hours. Obtaining gene data has become more easier. Gene analysis plays an important role in understanding biological features such as genetic expression and diversity and in the medical diagnosis, prevention, and treatment of genetic illnesses.

The Bio information in genetic material is decided by order of base composition, geneticists analyze what bio information is retained on particular sequence of particular position by comparing the order of encoded gene information. The most popular analysis method for gene information stored as data is the alignment algorithm. After the reference sequences are read, they are compared with the query sequence that the user wants to understand. Thus, not only the similarity score but also similar points between the two sequences and indel

mutations can be confirmed. The alignment algorithm provides the advantage of detailed information about the characteristics or mutation of a gene. However, it requires many computational operations running at $O(n^2)$ times and a large amount of memory space. This makes it difficult to analyze considerably long sequences like a whole genome.

In contrast, analysis methods based on static components are used for obtaining rough gene information. These methods use statistical features such as the proportion and distribution of A, G, T, and C. Statistical results can be obtained from a single pass of reading the gene information. However, as the amount of gene data increases, the overall statistical proportion of the gene information converges to one point. Consequently, it is difficult to obtain feature points of long sequences and confirm details of the sequence.

We propose a visualization method of the geometric space that uses random walk with base sequences consisting of character strings. The computed DNA sequence information by preprocessing can be rapidly visualized, and the visualization results make it easy for the user to check the structure of the DNA base sequence.

2 RELATED WORK

2.1 Visualization for Sequence Data

Sequence data are a set of ordered data that change with time, such as economic indicators or weather, and order-dependent data such as documents and DNA

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

base sequences. The most common type of sequence data is the time series, which is utilized to study the current state or predict the future state. Research in this field is based on correlation analysis of collected information with an associative relationship but unknown rules, such as between the stock price and interest rate or between the temperature, humidity, and weather. Most sequence data processing involves collecting and analyzing a great deal of data. The given data are mostly stored and managed in a text or compressed format. Examining such huge amounts of data directly has a high computational cost. Developing a more efficient method for data visualization is an important topic of research.

To determine the correlation between data more effectively, Alencar et al. [1] and Krstajic et al. [2] proposed methods for comparing different types of data at a time by compressing the memory space. In addition, Graells and Jaimes [3] presented a method for visually grasping the data progress by compressing data that are not chosen by the user as a significant part with a helical form. Thakur and Hanson [4] suggested a visualization tool based on location information to compare the data trends among regions.

2.2 Genomic Analysis

As previously stated, the DNA base sequence is bio-information encrypted in the form of bases. It is typical sequence data because biological features are determined by the order of bases. By analyzing the function of the encrypted base sequence, the characteristics can be obtained through a comparison with other well-known base sequences.

A typical analytical comparison of the base sequences involves using a tool based on the alignment algorithm, such as BLAST [5], Bowtie [6], or BWA[7] to grasp the similarity between the two sequences. Such tools eventually determine the similarity by using the alignment algorithm, which was described by Smith and Waterman [8]. However, the algorithm has a space complexity of $O(n^2)$. Thus, the longer the base sequence length, the greater the required computing time and memory space.

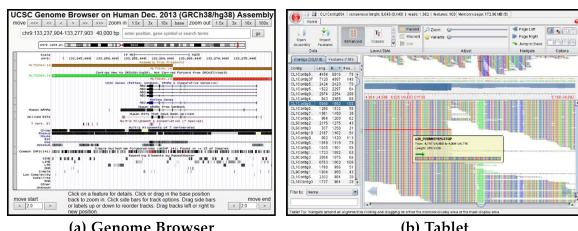


Figure 1: The web based visualization tool (a) Genome Browser and also visualization tool but provides mapping result (b) Tablet. Using these visualization tool, we can check the detail information about genome, but grasp of the whole structure of DNA base sequence is difficult.

Research has also been carried out on analyzing base sequence information by visualizing the results of base sequence analysis is also carried out. Genome Browser [9, 10] visualizes various genome analysis results that can immediately be obtained from a website. The Tablet tool [11] provides the mapping result based on Bowtie or BWA in visual form for the user. These visualization tools give very detailed analysis results but cannot provide information about the overall features of the genome.

2.3 Random Walk Visualization Model

The random walk plot is a visualization technique that allocates states to features of the data and represents the change in states with the coordinates according to the priority so that users can easily visualize the data contents. Overall, a two-dimensional walk plot is visualized with four states. The characters A, G, T, C are allocated to the DNA base sequences of each state to confirm the overall form of the genome universally.

The process of DNA visualization by using a two-dimensional walk plot is as follows.

1. Read the base sequences from the DNA data.
2. Convert the base to unit vectors.
3. Visualize the sequence by using the unit vectors.

The unit vector can be represented with various form which depends on the setting of direction for ‘A’, ‘G’, ‘T’, ‘C’, in case of WS-curve, the i th base of sequence is S_i , the unit vector $Unit^{WS}(i)$ is defined as follows:

$$Unit^{WS}(i) = \begin{cases} (-1, 0) & \text{if } S_i = A \\ (0, +1) & \text{if } S_i = G \\ (+1, 0) & \text{if } S_i = T \\ (0, -1) & \text{if } S_i = C \end{cases} \quad (1)$$

And then finally visualize the result of the sum of unit vector in order of base reading.

A two-dimensional random walk generally visualizes a sequence by mapping A, G, T, and C in each direction, as discussed earlier. This produces three main varieties: the WS-curve, RY-curve, and MK-curve. These depend on which bases are allocated in directly opposite directions of each other. The RY-curve allocates puRine(R = A, G) and pYrimidine(Y = C, T) as complementary relations. The MK-curve assigns aMino(M = A, C) and Keto (K = G, T) as complementary relations. The WS-curve allocates Weak (W = A, T) and Strong (S = G, C) as complementary relations. The two bases are combined in a double helix. The visualization results change depending on how we set the complementary axis, even if the same sequence is used.

Authors	Dimensions	Main characteristics
Kim et al. [12]	2	visualization in polygon form using K-Convex and comparison.
Liao and Ding [13]	3	Allocate the z axis with time and visualize the 1*1*1 space with a walk plot form and compute the entropy of those results to compare the similarity between genomes.
Bai et al.[14]	1	Read the base sequences, combine the complementary form into the genome as per the classifications of WS, RY, MK, visualize according to the order of sequences, and convert the results into entropy to compare the similarity between genomes.
Lo et al. [15]	3	Define the unit vector of A, G, C, T as a tetrahedron in a three-dimensional space and visualize the DNA.
Xie and Mo [16]	3	Add the axis representing the order of the sequence to the previous two-dimensional walk plot and visualize the DNA. With the additional axis, the beginning and end points of RY, MK, and WS provide the number of each base A, G, T, and C.

Table 1: Characteristics of DNA visualization and analysis research using a random walk plot

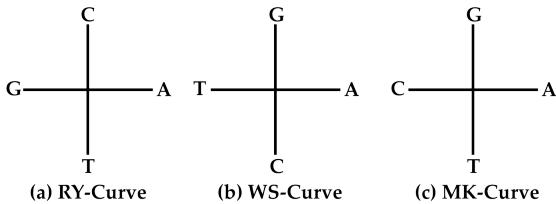


Figure 2: 2 dimensional walk plot.

Thus, we generally utilize all three curves to compare sequences. However, the genome bases A and T are generally much more common than G and C. If the sequence length is too long, the visualization result of the sequence tends to become lopsided.

The WS-curve, RY-curve, and MK-curve are each divided into four kinds depending on how the complementary axis is selected. For example, besides the above unit vector, the WS-curve is divided into four kinds depending on the directions of A, T, and G, C:

$$\text{Unit}^{SW}(i) = \begin{cases} (-1, 0) & \text{if } S_i = A \\ (0, +1) & \text{if } S_i = G \\ (+1, 0) & \text{if } S_i = T \\ (0, -1) & \text{if } S_i = C \end{cases} \quad (2)$$

These graphs are used to prevent different resulting values depending on the three curves of the axis when computing similarities or dissimilarities. Thus, the visualization results for the walk plot vary with the conversion method from each base to the unit vector. Many studies have examined methods for effective visualization. Table 1 introduces different studies on how to utilize the random walk plot.

There have been many studies on visualization with a three-dimensional random walk based on the concepts for a two-dimensional random walk. The biggest problem with the two-dimensional random walk is the loss of data by the two bases in the opposite directions. To prevent such a problem, many studies have allocated the z axis as a time stamp. Liao and Ding

[13] represented the z axis as $1 - i/1$ so that the whole sequence appears in the space between $(0,0,0)$ and $(1,1,1)$ instead of showing just a simple cumulative sum of vectors about A, G, T, C. By representing the whole sequence in a given space, comparison results such as the similarity can be numerically expressed by $0 \leq \text{Similarity/Dissimilarity}(x) \leq 1$. However, the longer the sequence, the more difficult it is for the user to understand the visualization results. Xie and Mo [16] visualized the data by adding an axis to the basic two-dimensional visualization method and increasing the additional axis value for each base pair. Thus, just the location of the end point of the random walk provides the rate of each base (A, G, T, and C), and entropy can help with determining the similarity between different sequences. However, a longer sequence, causes the axis representing the order of base pairs to be out of proportion to the other axis, and the visualization results are too compressed for massive sequences.

To avert the loss of base information, Lo et al. [15] mapped each base to each vertex of a tetrahedron in three dimensions instead of to the axes x and y. This method has the advantage of effectively representing the sequence characteristics, but the features of a base cannot be clearly shown in each direction. In addition, information is lost by the other three bases.

3 NEW VISUALIZATION MODEL

3.1 Usefulness of Three-dimensional Walk Plot Visualization

Random walk visualization is useful for visualizing DNA data in string form to make the results easier to understand. However, previous research focused on short DNA data lengths; their methods present drawbacks for visualizing long sequences. For example, table 2 presents the rate of the 2-mer base of human chromosome 1.

The ratio of the base pairs AT and GC was about 19.23%. The visualization process based on the WS-Curve incurred heavy data losses. To prevent this prob-

2mer	bp	ratio	2mer	bp	ratio
AA	21,411,841	9.50%	CC	12,264,137	5.44%
GG	12,274,467	5.45%	TT	21,464,952	9.53%
AC(CA)	29,523,084	13.11%	AT(TA)	31,095,833	13.80%
AG(GA)	27,699,140	12.30%	CG(CG)	12,235,412	5.43%
CT(TC)	27,746,387	12.32%	GT(TG)	29,565,347	13.12%

Table 2: The rate of two pair of base in Human chromosome



Figure 3: The transition process of sequence. Read the sequence with overlapped form per 2-mer unit. Through this process, we can find the complementary 'AT(TA)', 'GC(CG)' in sequence.

lem, other curves such as the MK-curve and RY-curve are used to check the sequence. However, A and T are generally more common than G and C, so a longer sequence makes the graph form more lopsided. Thus, this approach is inappropriate for visualizing long sequences. Three-dimensional visualization tries to reduce this problem by using a time axis, but a long sequence makes the time axis too big. It is difficult to grasp the visualization results at a glance. In this paper, we propose a new three-dimensional visualization method to improve the visualization results for a long DNA sequence. To reduce the loss of data, which is the repeated sections of AT and GC, we can represent that repetition through the z axis if the sequences AT and GC are repeated. Thus, we can get the same results of existing two-dimensional visualizations by using orthogonal projection for the XY plane with our new method, and we can check for loss of information caused by repetition of the AT and GC bases by viewing the results from another angle.

3.2 Proposed Walk Plot Procedure

To determine the AT and GC parts in DNA sequence data more easily, we reset the relation between the unit vector and each base, as given in Table 3, based on existing definitions given in section 2.

2mer	Symbol	Vector	2mer	Symbol	Vector
AA	A	(2, 0, 0)	AG(GA)	U	(1, 1, 0)
AC(CA)	V	(1, -1, 0)	AT(TA)	W	(0, 0, -2)
CC	C	(0, -2, 0)	CG(CG)	X	(0, 0, +2)
CT(TC)	A	(-1, -1, 0)	GG	G	(0, 2, 0)
GT(TG)	Z	(-1, 1, 0)	TT	T	(-2, 0, 0)

Table 3: Reset the vector for visualization of 3 dimensional random walk.

As indicated in Table 3, to detect AT and GC in the sequence, we combine the base pairs and convert the sequence into coordinates. In order to prevent the wrong combination of 2-mer unit bases, the base pairs are read in overlapped form, as shown in Figure 3.

The base pairs AT and GC are represented on the z axis. The other base pairs are represented as the sum

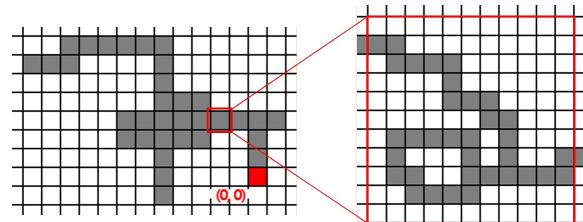


Figure 4: The phenomenon that simplification of widespread data of visualization result on limited screen. To visualize the whole data in limited space, the random walk such as (b) is simplified into (a).

of two unit vectors for each base, as given by the WS-curve method. After the vector transition for DNA genome data information, those vectors are visualized in three-dimensional space. The method of visualization is the same as that of two-dimensional visualization, where the sum of vector values is computed according to the order of sequences and the results are connected with a line to provide the final visualization result. For the random walk plot R , the beginning point is $R(0) = (X_0, Y_0, Z_0)$ ($X_0 = Y_0 = Z_0 = 0$). $Unit^{3d}(i)$ is the converted value of the i th 2mer of the unit vector. The i th point $R(i) = (X_i, Y_i, Z_i)$ of the random walk plot is computed as follows:

$$R(i) = R(i-1) + Unit^{3d}(i) = \sum_0^i Unit^{3d}(i) \quad (3)$$

3.3 Simplification and Normalization of the 3D Walk Plot

When trying to visualize a base sequence bigger than 100 Mbp like a chromosome with random walk, the screen size is more limited than the range of coordinates represented on the walk plot. Thus, it is impossible to represent the whole sequence. Usually, when representing these values with such a wide range in coordinates on a screen, the points in a certain range are generally simplified into one pixel, as shown in Figure 4.

However, reading these many points with little influence on the results and reflecting them in the visualization greatly wastes time and memory space during operation. Therefore, we fixed the screen size and developed a simplification preprocess for the vector transition time to reduce the unnecessary operation and processing time.

The i th point of walk plot R is $R(i)$, and the screen range of visualization is $[-v, v]$. Then, the simplification results can be defined as follows:

- Find the maximum value $\max(R)$ from the absolute values of the x, y, z components for each point of R .
- Determine the range of simplification $ran = \max(R)/v$ by using the maximum value.

3. Convert all points of R into the simplified point $R'(i) = R(i)/ran$ by using the range of simplification ran .
4. Remove the continuous overlapping points among the converted points $R'(i)$.

The preprocess to simplify all of the points in the range ran into one point, as shown in Figure 4, reduces the number of points that need to be visualized. As explained before, however, in the case of DNA base sequences, A and T are generally more common than G and C. Thus, all curves excluding the WS-Curve have lopsided visualization results. This problem is also encountered for the z axis of the three-dimensional walk plot, which is determined by combining the high-frequency base AT (TA) and low-frequency base GC (CG). The rate of AT is 13.80%, which is much higher than that of GC (5.43%), as given in Table 2. This imbalance in the range of data slightly differs for each base sequence of the chromosome. However, similar trends can also be observed in the chromosomes of other species besides humans. In the case of short sequences, this does not greatly affect the result. However, with longer sequences, the range of the z axis is much bigger than that of the x and y axes. This makes the results difficult for the user to check. Table 4 presents the resulting maximum values of each axis for human chromosomes 1–5 based on the computed results for the walk plot R .

The z axis has a much larger value than the x and y axes. Because of this difference in the range of data, when the whole sequence is visualized, the changes in the x and y axes are too slight compared to that of the z axis. Thus, it is difficult to sense the changes in x and y. To solve this problem, when we simplify the data, we compute the ranges of simplification for x, y, and z independently. We defined this process as the normalization of the three-dimensional walk plot. The maximum values among the absolute values of the x and y components in the walk plot R are $\max_{XY}(R)$. The maximum value of the z component is $\max_Z(R)$, and the view size for visualization is v . Then, the results of normalizing each point can be defined as follows:

$$R_{regular}(i) = \left(X_i \frac{v}{\max_{XY}(R)}, Y_i \frac{v}{\max_{XY}(R)}, Z_i \frac{v}{\max_Z(R)} \right) \quad (4)$$

In the preprocess, when we simplify a long base sequence by using the normalization method in advance, it is cumbersome to change the screen size or check the detailed information because the preprocess needs to be run again. However, if we visualize a chromosome of 100 Mbp in three-dimensional space in the range of $[-400, 400]$, the number of points is reduced to below $\frac{1}{100}$ (this slightly differs depending on the range of data and direction of progress), so the visualization results

can be quickly obtained. Furthermore, if the screen size is fixed, the data can be visualized without computing each point by always using the preprocess results.

4 COMPARISON AND 3D VISUAL SHAPE

The walk plot results generated by simplification can be visualized on three-dimensional coordinates as a set of points. A simple method to grasp the similarity within a set of points is comparing the similarity of the polygon with the form containing all of the points. However, finding areas with a polygon form that contains all of the points and comparing these areas are not easy to do.

In this section, we compare orthographic projection planes such as XY, YZ, and XZ instead of comparing the visualization results in three-dimensional space to reduce the complicated computation and compare the results more easily. To effectively determine the area on the two-dimensional space and compare the area generated by each sequence, we assumed a "beta" shape. The similarity comparison method uses this beta shape.

Finding the area that contains the results of the walk plot is very important to comparing similarities based on the visualization result. In the case of a relatively dense area, if there are slight mutations in each sequence, they will be recognized as different. In contrast, if the area is too large, the results will not demonstrate the characteristics of the sequence. The simplest method for finding the area that contains all of the points on the two-dimensional plane is to use a bounding box or convex hull. The bounding box can define an area more easily if the maximum and minimum values of each coordinate are found. However, this tetragonal area is too rough to comprehend the characteristics of sequences.

The convex hull is the connection of the outermost points to make the smallest hull that includes all of the points. The convex hull provides a more concrete area than the bounding box but uses the outermost points; if there are many large empty spaces, it cannot consider the problem, as shown in Figure 5(a).

The alpha hull (alpha shape) [17, 18] was proposed in 1983 and is shown in Figure 5(b). This involves first selecting two points and drawing a circle with the radius of alpha. If there are no points in the circle, the two points are connected. This continues until the area that contains all points is found.

The alpha hull is robust against empty spaces that cannot be considered by the convex hull. The expected area can be determined by regulating the range with the α value.

However, the alpha hull requires a computation time of $O(n \log n)$ for n points. When there are many points, like in a long sequence, the computation time becomes

	Chr1	Chr2	Chr3	Chr4	Chr5
Max_X	233,197	121,923	215,564	126,126	218,859
Max_y	112,716	106,743	87,480	74,728	90,302
Max_Z	18,589,777	14,769,847	19,379,909	20,991,662	17,923,569

Table 4: The result of the maximum value of each axis for human chromosome 1-5 which is computed result for walk plot R. The maximum values of the axis of x and y are similar to each other, but the z axis has dozens of time as big value as x, y.

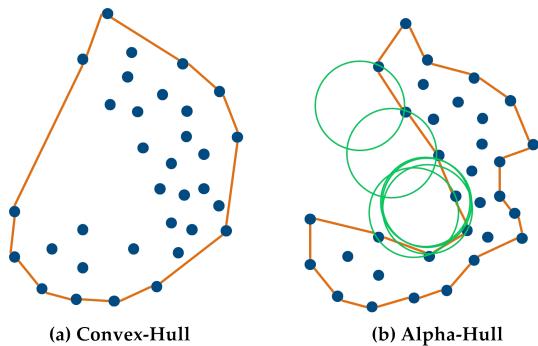


Figure 5: The example of Convex Hull (a) and Alpha-Hull(b). In the case of Convex-Hull (a), Because it uses outer-most points, if there are much big empty space, it can't consider this. Alpha-Hull, on the other hand, it can extract much detailed area using circle with a radius of alpha.

excessive. The alpha hull algorithm also prints edges that connect the two outermost points of the area. This result is not sorted, so additional computation is needed.

4.1 Beta Shape Similarity Algorithm

In this section, we propose a new algorithm for finding the area by comparing the visualization result with the above alpha hull. The alpha hull method generates the result in edge form by comparing the results located in α with each other. However, the beta shape searches the coordinate space and prints the dot-marked coordinate space in matrix form. The beta shape method takes a computation time of $O(n^2)$ to search $n \times n$ space because the search space is the visualization of the vertex coordinates. As previously explained about simplifying the method for the walk plot, if there are many vertices and the range of coordinates is limited, the visualization time for the beta shape method is less than that for the alpha hull method. The process to obtain the beta shape is defined as follows:

1. Mark the visualization results with two-dimensional matrix coordinates as $mat[x][y] = 1$.
2. Search the whole coordinate space for the marked vertices.
3. Check if there are other marked vertices within the range of $(i - \beta, j)$ and $(i + \beta, j)$. The location of the marked vertex is found by searching (i, j) .
4. If there are other marked vertices, fill the empty space by checking the range of (i, j) .

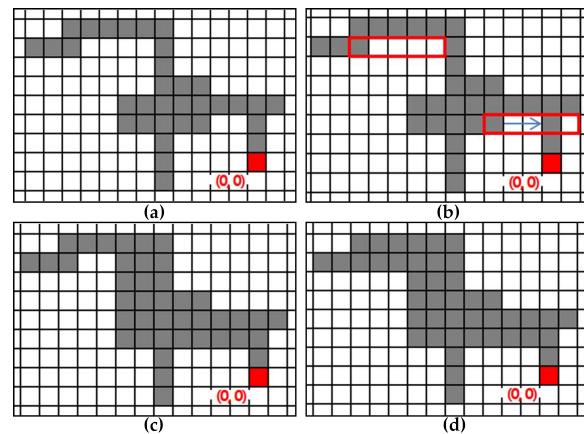


Figure 6: The example of generation process when the beta value is 5. Find the marked point (i, j) on the orthogonal projected walk plot as (a), and find the other marked point in the axis of x, y direction within the limit of β range. If there is another point in the β range, it fills the empty space with marked points as (c). Those process (b), (c) are repeated until there no space to be filled more. Figure (d) is final result for (a) using Beta-shape when $\beta = 5$.

Figure 6 presents the above process. The range of β is searched in the x and y axis directions, as shown in Figure 6(b). The empty space is filled, and coordinates of the area containing the vertices are stored to compute the whole area. Through this process, we can determine the area that includes the visualization result within a computation time of $O(n^2)$.

4.2 Comparing the Similarity of Walk Plots by Using the Beta Shape

Comparing the area on three-dimensional coordinates is a difficult problem, so we computed the similarity by using the results of orthogonal projection for each three walk plots on the two-dimensional planes XY, YZ, and XZ. If the two beta shapes beta_a , beta_b are generated from the two base sequences a, b that are orthogonally projected onto a single two-dimensional plane k , we can get a broad outline of similarity by computing their overlapping area. The similarity of the areas beta_a and beta_b is defined in equation (5):

$$\text{Sim}_{ab}^k = \frac{\sum (\text{beta}_a \cap \text{beta}_b)}{\sum \text{beta}_b} \quad (5)$$

On the left-hand side, $\sum (\text{beta}_a \cap \text{beta}_b)$ is the size of the overlapping areas of beta_a and beta_b . This equation can be used to determine how much of a overlaps with

b. If most of beta_a overlaps with beta_b , this may be only a small part of beta_b , so the two areas may not be similar to each other. Therefore, to compare the similarity of two areas, we should consider $\text{Sim}_{ab}^k, \text{Sim}_{ba}^k$. After considering this point, we defined the similarity between beta_a and beta_b as follows:

$$\text{Sim}^k(a, b) = \sqrt{\frac{(\text{Sim}_{ab}^k)^2 + (\text{Sim}_{ba}^k)^2}{2}} \quad (6)$$

By using equation 6, we can compute the visualization results on a three-dimensional walk plot and get the similarity rate $\text{Sim}(a, b)$ between different two sequences. The similarity rate for each plane can be represented as a vector with three components ($\text{Sim}^{XY}(a, b), \text{Sim}^{YZ}(a, b), \text{Sim}^{XZ}(a, b)$) that are orthogonally projected on the XY, YZ, and XZ planes. By using the sizes of the vector values, we computed the similarity of two sequences with the rate $\text{Sim}(a, b)$. The computed results are given below.

$$\text{Sim}(A, B) = \sqrt{(\text{Sim}^{XY}(a, b))^2 + (\text{Sim}^{YZ}(a, b))^2 + (\text{Sim}^{XZ}(a, b))^2} \quad (7)$$

5 EXPERIMENT

5.1 Environment and Testing Data

The two main techniques were a visualization model for base sequences of 10 Mbp on a three-dimensional walk plot by using the simplification method and a comparison model based on the beta shape by using the visualization results. Using the simplification method, we could visualize the base sequences of 10 Mbp within a very short time on the three-dimensional walk plot and compare the visualization results of two sequences more intuitively. To represent the results as numerical values, we compared the result of two bases with the beta shape. To visualize the three-dimensional walk plot in real time, we developed a web service system based on Web-GI Language.

All of the data used for the experiment were obtained from UCSC [10] and NCBI [19]. We obtained the human chromosome data from the Human Genome Project; "homo sapiens gr 37" is offered free by UCSC [10]. The chromosome data of three apes (gorilla, chimpanzee, orangutan) and other species (milk cow, dog, green monkey, chicken, rat, pig) were obtained from the taxonomy in NCBI [19]. The average chromosome length was approximately 30–250 Mbp, and 261 chromosomes were used. The sex chromosomes were too short, so they were not selected for analysis. Table 5 presents the data for this experiment.

5.2 Visualization Results

5.2.1 Verification of the DNA Visualization Result with the Proposed Method

We checked if the three-dimensional walk plot visualization results could be used to distinguish the charac-

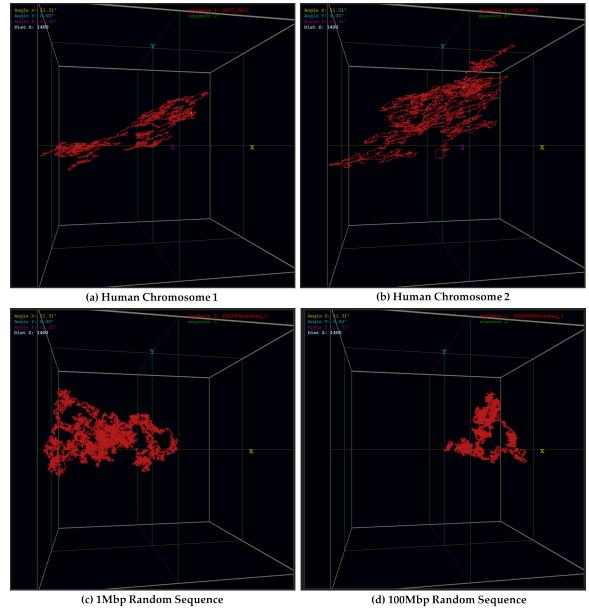


Figure 7: The visualization results of human chromosome(GR37) 1, 2 are (a), (b). And those of randomly generated sequences which has 1Mbp, 100Mbp are (c), (d). The walk plots of random sequences are lumpy generally, but those of human chromosome are scattered. We hereby can figure out that the chromosome data has certain pattern which is distinguished from random sequences.

teristics of real genomes. To judge the utility of the visualization result, we set three standard judgments based on other research about visualization-based walk plots [15]:

- Are the characteristics of the base sequences represented in the three-dimensional walk plot results?
- Is it possible to distinguish the different base sequences from each other by using the visualization results?
- Is it possible to check similar sequences against each other by using the visualization results?

To confirm the base sequences form the visualization results, we generated the human chromosome (GR37) 1, 2 and random sequences by using the visualization tool, as shown in Figure 7. The random sequences were provide by a random number generator in quantities of 1 and 100 Mbp. We then compared the results of both sequences. The total view size was limited to [-400, 400] in the preprocess for all experiments.

When we compared the human chromosomes 1 and 2, chromosome 1 was spread out densely, but chromosome 2 was distributed widely. Thus, they had different characteristics. For the random sequences created by the random number generator, the results were generally lumpy (Figures 7(c) and (d)), but the results of the human chromosome were not. Thus, we confirmed that the human chromosome has certain rules for genome

sequence ID	scientific name	name	Number of chromosome	notes
s01	Bos taurus	milk cow	38	-
s02	Canis lupus familiaris	dog	38	-
s03	Chlorocebus sabaeus	green monkey	29	-
s04	Gallus gallus	chicken	except 28(32)	29,30,31,32
s05	Gorilla gorilla gorilla	gorilla	23	chromosome 2A, 2B
s06	Homo sapiens	human	22	-
s07	Mus musculus	rat	19	-
s08	Pan troglodytes	chimpanzee	23	chromosome 2A, 2B
s09	Pongo abelii	orangutan	23	chromosome 2A, 2B
s10	Sus scrofa	pig	18	-

Table 5: The table shows the information of DNA Chromosome sequences with each sort which are used for experiment. The information for the mitochondria and X,Y Chromosome which is related with gender is excepted for this experiment.

construction, and those characteristics were revealed by the walk plot.

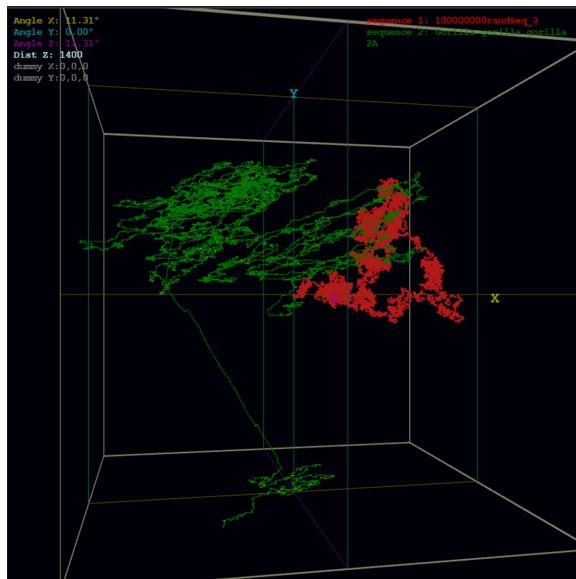


Figure 8: The comparison result of visualization for random sequence which has 100Mbp and gorilla. In case of random sequence, the visualization result is represented with overlapped form as (d). Also the visualization result of random sequence has more lumpy than the gorilla chromosome 2A which has similar size with random sequence. That way, we can confirm that the DNA sequence is not irregular data.

To get more precise results, when we compare the visualization results of two sequences on screen, we considered the scale of the gorilla chromosome 2A with a similar size to that of a random sequence, as shown in Figure 8. For the gorilla chromosome 2A, the whole sequence length was approximately 111 Mbp, which was different from the random sequence by 10%. However, the random sequence was expressed at a point on the coordinate plane in lumpy form, and there were no changes in the walk plot. On the other hand, the gorilla chromosome showed large changes within the data range.

Figure 9 compares the results of human chromosome 1 and chromosome 1 of the dog and chimpanzee. The two chromosomes clearly had different visual results, as shown in Figure 9(a). This result was true not only for the dog but also for all species excluding the apes. On the other hand, the chimpanzee chromosome 1, which is known to have similar genome information as human chromosome 1, had similar visualization results. The chromosomes of other apes such as the gorilla and orangutan also had generally similar results.

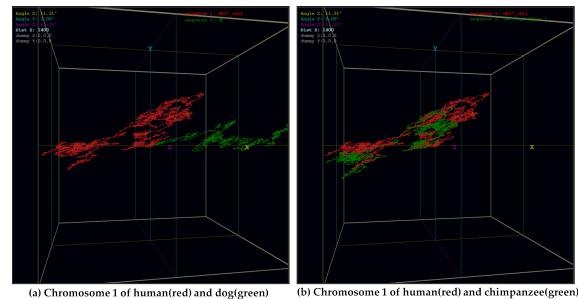


Figure 9: The comparison result of the chromosome 1 of human and dog, chimpanzee. The chromosome of human and dog is quite different as (a), but the chromosome of human and chimpanzee, which is known that more than 90% of chromosome is similar, is also highly similar.

5.2.2 Verification of the Similarity of Genome Information by Using the Beta Shape

By using the above three-dimensional walk plot method, as shown in Figure 9, we compared the information of base sequences with different genome information according to the visualization results for validation. The chromosomes of the human and apes, which are known to be quite similar, were also checked and showed only small differences. We computed the similarity of the visualization results of two different sequences by using the beta shape and verified the comparison method for similarity based

on the results. The similarity comparison method was verified according to the following criterion:

- Is it possible to confirm the degree of similarity with numerical values by using the similarity comparison method?

To confirm this criterion, we performed a comparative experiment to check whether the results of the similarity comparison using the beta shape met this requirement. Table 6 presents the computed similarity for each chromosome 1 introduced in Table 5.

In the case of the human (S06), other apes were very similar in the order of S08, S05, and S09, as given in Table 6. In addition, the comparison results were better with the human than with the other species. Figure 10 shows the visualization results of chromosome 1 for the human and gorilla. Figure 10(a) shows a three-dimensional walk plot. Compared with the human, the front part of the gorilla sequence was weighted towards the y axis, but the general forms were very similar. The similarity between the two base sequences could be checked more concretely when their results were orthogonally projected on the (b) XY, (c) XZ, and (d) YZ planes. There was no difference between the two sequence in the orthogonal projection on the XZ plane. For the orthogonal projections in the XY and YZ planes, the random walk was slightly weighted towards the y axis, but the general forms were quite similar. However, the orangutan (S09) was similar to the other apes but also had no difference with the other species.

6 CONCLUSION

The base sequences, which contain the DNA information, are data in character string form. To check such data visually, related research based upon previous visualization methods of two- and three-dimensional walk plots is actively ongoing. Many methods of comparison between two difference sequences have been proposed based on the walk plot characteristics.

This visualization method using base sequences can visualize the sequence in a short amount of time, and the characteristics of the base sequences can be intuitively analyzed from the visualization result without the use of complicated algorithms such as the alignment method. Because previous research usually focused on short base sequences with special functions such as insulin and the β -globin base, these methods are inapplicable to long base sequences of 10 Mbp.

We proposed a new three-dimensional visualization method for long sequences of more than 10 Mbp. The advantages of our proposed visualization method are as follows:

- It can visualize quite long sequence data such as chromosomes.

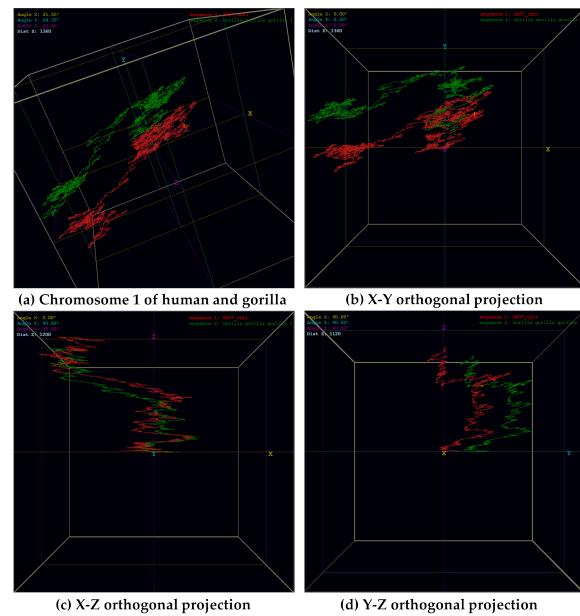


Figure 10: The comparison result of the chromosome 1 of human and gorilla (a). And the result of orthogonal projection on plane of XY(b), XZ(d), YZ(d). Compared with human, in the case of gorilla, the front part of sequence is weighted towards y axis but the general form is very similar to each other. Those characteristics are more revealed at the result of orthogonal projection on 2 dimensional plane.

- If there are no changes to the screen size or sequence data, it can visualize the sequence in a short time by reducing the number of points that are generated with each base point and that need to be visualized through a preprocess.
- It can reveal more specific characteristics than previous visualization methods by using the z axis for bases such as AT (TA) and GC (CG); such information is lost with the WS-curve method.
- It can compare massive amounts of base sequences like a chromosome in a short amount of time through our proposed beta shape comparison.

We expect that our proposed method can be used to grasp the relation between long sequences in a short amount of time without comparison-based alignment, which would reduce the computation cost and contribute to evolutionary research. Because DNA information necessarily contains mutation, deletion, and metastasis, the following requires further study:

- Searching for partial similarity in the visualization results of base sequences.
- Determining high levels of similarity by comparing visualization results at various scales.
- Separating parts with high and low levels of similarity when visually comparing two sequences.

	s01	s02	s03	s04	s05	s06	s07	s08	s09	s10
s01	1.000	0.135	0.187	0.024	0.051	0.130	0.236	0.104	0.164	0.180
s02		1.000	0.190	0.083	0.061	0.129	0.201	0.138	0.163	0.112
s03			1.000	0.032	0.127	0.179	0.220	0.159	0.207	0.162
s04				1.000	0.080	0.073	0.140	0.060	0.171	0.053
s05					1.000	0.345	0.143	0.299	0.192	0.174
s06						1.000	0.181	0.535	0.267	0.214
s07							1.000	0.206	0.198	0.200
s08								1.000	0.249	0.223
s09									1.000	0.166
s10										1.000

Table 6: The comparative result for chromosome 1 of each sort, when $\beta = 10$. The similarity between apes(s05, s08, s09) and human(s06) is relatively high.

7 ACKNOWLEDGEMENT

This research was supported by a grant from Marine Biotechnology Program(PJT200620, Genome Analysis of Marine Organisms and Development of Functional Applications) Funded by Ministry of Oceans and Fisheries, Korea.

8 REFERENCES

- [1] A. B. Alencar, F. V. Paulovich, R. Minghim, M. G. Andrade, and M. C. F. Oliveira. Similarity-based visualization of time series collections. *12th Int. Conf. on Information Visualisation*, pages 280–286, 2008.
- [2] M. Krstajic, E. Bertini, and D. A. Keim. Cloud-lines: Compact display of event episodes in multiple time-series. *IEEE Trans. Visualization and Computer Graphics*, 17(12):2432–2439, Dec 2011.
- [3] E. Graells and A. Jaimes. Lin-spiration: using a mixture of spiral and linear visualization layouts to explore time series. *Proc. ACM Int. Conf. on Intelligent User Interfaces*, pages 237–240, 2012.
- [4] S. Thakur and A. J. Hanson. A 3d visualization of multiple time series on maps. *Proc. IEEE 14th Information Visualization*, pages 336–340, 2010.
- [5] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.
- [6] B. Langmead, C. Trapnell, M. Pop, and SL Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol.*, 10(3):R25, 2009.
- [7] H. Li and R. Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.
- [8] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.
- [9] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler. The human genome browser at ucsc. *Genome Res.*, 12(6):996–1006, 2002.
- [10] UCSC. Genome browser. <http://genome.ucsc.edu/>.
- [11] I. Milne, M. Bayer, L. Cardle, P. Shaw, G. Stephen, F. Wright, and D. Marshall. Tablet—next generation sequence assembly visualization. *Bioinformatics*, 26(3):401–402, 2010.
- [12] M. A. Kim, E.J. Lee, H. G. Cho, and K. J. Park. A visualization technique for dna walk plot using k-convex. *Proc. Fifth Int. Conf. on Central Europe ComputGraphVisualization*, pages 10–14, 1997.
- [13] B. Liao and K. Ding. A 3d graphical representation of dna sequences and its application. *Theor. Comput. Sci.*, 358(1):56–64, 2006.
- [14] F. Bai, Y. Liu, and T. Wang. A representation of dna primary sequences by random walk. *Math. Biosci.*, 207(1):282–291, 2007.
- [15] N.W. Lo, H. T. Chang, S. W. Xiao, C. H. Li, and C. J. Kuo. Global visualization and comparison of dna sequences by use of three-dimensional trajectories. *J. Inf. Sci. Eng.*, 23(6):1723, 2007.
- [16] G. Xie and Z. Mo. Three 3d graphical representations of dna primary sequences based on the classifications of dna bases and their applications. *J. Theor. Biol.*, 269(1):123–130, 2011.
- [17] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theor.*, 29(4):551–559, Jul 1983.
- [18] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, E. P. Mucke, , and C. Varela. Alpha shapes: definition and software. *Int. Comput. Geom. Softw. Workshop*, 1995.
- [19] NCBI. National center for biotechnology information. <http://www.ncbi.nlm.nih.gov/>.

Gender Prediction using Individual Perceptual Image Aesthetics

Samiul Azam

University of Calgary
Dept. of Computer Sci.
Calgary, Alberta, Canada
samiul.azam@ucalgary.ca

Marina Gavrilova

University of Calgary
Dept. of Computer Sci.
Calgary, Alberta, Canada
mgavrilova@ucalgary.ca

ABSTRACT

Images have rarely been used for psychological behavior analysis or for person identification in the information technology domain of research. In this paper, we present one of the first methods that allows to accurately predict gender from a collection of person's favorite images. We select 56 image aesthetic features, and propose a mixture of expert models consisting of support vector machine, K-nearest neighbor and Decision tree. Final decision is taken based on the weighted combination of probability generated by individual classifiers. We introduce a genetic algorithm based method to improve the prediction accuracy of the model, which allows us to find the best combination of feature subset in 56D binary search space. Moreover, feature dimension is reduced significantly that decreases the testing time. Finally, three weights of the prediction model are adjusted using genetic algorithm in 3D real-number search space. Experimental results conducted on a true image database of 24000 images provided by 120 Flickr users. The experimental results demonstrate superiority of the proposed method over other approaches for gender prediction from perceptual image aesthetics preferences.

Keywords

Perceptual image features; Gender prediction; Image aesthetic features; Ensemble of classifiers; Probability; Genetic Algorithm

1 INTRODUCTION

Traditionally in computer graphics and image processing domains, images are used for classification [Iiv03, Lee03], visual data exploration [Mol14], landmark recognition [Pri13], pose estimation [Tew15] or image reconstruction [Ska13]. However, images have rarely been used for psychological behavior analysis or for person identification in the information technology domain. In this paper, we address this gap and study effects of human aesthetic perception, expressed through choice of favorite images, onto behavior and gender recognition of a person. Recently, it has been shown that a person's visual preferences can be measured using image aesthetic features and his or her favorite images [Lov14]. Moreover, there are differences between male and female neural correlation of aesthetic preferences [Cel09]. A study on website appearance concurred with the fact that males and females have differences in aesthetics perception [Mos06, Mos07]. This research motivates us to look deeper into the possibility of gender identification from a set of individual's favorite images.

Preliminary research on gender recognition was recently conducted in the Biometric Technologies Lab at the University of Calgary. It was relying on aesthetic preferences, tested on a database of 120 Flickr [Fli04] users, and has been accepted for publication

to ICCI*CC 2016 [Aza16]. The main novelty of the current work is in proposing to use the genetic algorithm to improve the prediction accuracy. While both the preliminary and the current research use the same set of aesthetic features tested on the Flickr image database, the newly proposed method uses genetic algorithm (GA) for best feature subset selection, as well as choosing the best weighted combination of the three classifiers. This, in turn, and allows to achieve a higher accuracy of a gender recognition, compared both to similar research and the recently developed algorithm [Aza16].

This paper is organized as follows. Section 2 presents the literature review on social behavioral biometric and gender prediction research. The proposed methodology of gender prediction is described in Section 3. Section 4 presents the experiment conducted on Flickr users. Finally, discussions and future directions are outlined in Section 5.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2 LITERATURE REVIEW

In the area of biometric, most of the research on person identification and gender estimation is conducted through processing of images or videos. A person's walking pattern (or gait) is a popular trait for identification and gender estimation where video data is collected using conventional surveillance camera or KINECT depth camera [Ahm15a, Ahm15b, Gur11]. Another large domain is the processing of face image for recognition and gender estimation [Sul15, Dan16, kuk04]. Besides that, social activities of a person can be used for identification and gender prediction which are known as social behavioral biometrics [Mad14]. In the recent years, with the rise of popularity of on-line social networks (OSN), such as Pinterest [Pin10] and Flickr [Fli04], more and more users sharing their views, choices and preferences in the form of images and videos. In the OSN Flickr [Fli04], people share their favorite images that contain a person's visual aesthetic preferences. A 2012 research proved that it is indeed possible to establish an identity of a person experiment from Flickr user's image preferences [Lov12]. They extracted contextual and perceptual image aesthetic features and generated a template for each Flickr user based on those features using LASSO regression. An improved version of this method was introduced two years later [Lov14]. By incorporating more distinctive image aesthetic features, they reported 96% accuracy at rank 20. Instead of LASSO regression, authors in [Seg14] applied counting grid model and support vector machine (SVM) to generate template, which resulted in 98% accuracy in Flickr user identification experiment conducted on the same database.

Gender is one of the common demographic features used as a soft trait in the area of human authentication biometric [Gav13, Dan16]. Gender prediction from the context of social behavioral biometric has (if ever) rarely been explored in literature. Very recently, authors in [Qua14] used image based OSN Pinterest [Pin10] to predict gender from user's image posting behavior and image contextual features. They applied a bag of visual word model to identify the difference between male and female users. They conducted their experiment on 160 users (80 male and 80 female) from Pinterest, and depicted 72% accuracy in gender prediction. One of the shortcomings of the approach is that it used 33 board categories of Pinterest (posting behavior) as features for gender prediction, which makes the method very limited to a specific OSN. In this paper, we propose a gender prediction method where gender is predicted from a person's favorite list of images only. To make the method OSN independent, user's posting behavior is ignored as a feature. For experiment, we use the 200 Flickr users database (contains 40000 images) provided by one of the authors from the article [Lov12]. We have done a preliminary research on gen-

der prediction using image aesthetics [Aza16] which shows 76.65% accuracy over the same database. In this paper, we present a different methodology which results in the improvement over the preliminary work. We use genetic algorithm for feature selection, as well as weight adjustment of the prediction model, which allows to reach approximately 83% accuracy of gender prediction. This is higher than all of the current state of the art methods (by 6% to 12%). Also, it's worth noting that unlike typical biometric identification based on much more concretely defined data (i.e. ear, palm, face, gait etc), gender identification is based on the soft biometric features which makes it a much harder problem.

3 METHODOLOGY

The detail description of the proposed gender prediction method is provided in the following subsections.

3.1 Collection of Aesthetic Features

The proposed gender prediction method uses a person's aesthetics as a cue to his or her gender prediction. Different types of aesthetic features were introduced by researchers for the purpose of automatic image ranking [Ayd15, Mar11, Rit06, Jia10], image classification [Xia13, Jan10] and person identification [Lov12, Lov14]. After a comprehensive review, we identify five categories of image aesthetic features that are mostly found in existing articles: 1) image content; 2) composition; 3) texture; 4) color and 5) image parameters. Detail description of all these features can be found in the previous works. For simplicity of implementation, we use a subset of the above features [Aza16] composed of image composition, texture, color and parameter features in our proposed model. The length of the features vector is 56. Brief description of the features are provided in Table 1 with assigned feature number.

Performance of a machine learning based model depends on the feature vector used in their training and testing. Convergence of decision boundary relies on the features. Some features are highly distinctive, and are sufficient to describe the model efficiently. On the other hand, some features are unnecessary which increase the training and testing time, as well as move the decision boundary away from the best position. So, the feature selection is a crucial step for our prediction model also. In the subsection 3.3, we describe the feature selection step used in the proposed prediction model. After the selection step, we identify a set of distinctive features for each classifier (in the mixture of expert model) that maximizes their classification accuracy, and speeds up the testing time by reducing dimension of the feature space. In the supervised learning phase, we train each classifier of our ensemble using the selected features. We group the images into two labels or classes: male

Feature	Brief description
f_1	Average intensity of V channel in HSV image
f_2	Average intensity of S channel in HSV image
f_3	Standard deviation of V channel in HSV image
f_4	Standard deviation of S channel in HSV image
f_5	Entropy of RGB image
f_6	Aspect ratio of the image
f_7	Rule of thirds in H channel
f_8	Rule of thirds in S channel
f_9	Rule of thirds in V channel
f_{10}	Hue Circular Variance
f_{11}	Canny edge pixel count
f_{12}	Emotion based: Pleasure
f_{13}	Emotion based: Arousal
f_{14}	Emotion based: Dominance
f_{15}	Colorfulness
f_{16}	Tamura directionality
f_{17}	Tamura contrast
f_{18}	Wavelet Textures in H channel: level 3
f_{19}	Wavelet Textures in H channel: level 2
f_{20}	Wavelet Textures in H channel: level 1
f_{21}	Sum of f_{18}, f_{19}, f_{20}
f_{22}	Wavelet Textures in S channel: level 3
f_{23}	Wavelet Textures in S channel: level 2
f_{24}	Wavelet Textures in S channel: level 1
f_{25}	Sum of f_{22}, f_{23}, f_{24}
f_{26}	Wavelet Textures in V channel: level 3
f_{27}	Wavelet Textures in V channel: level 2
f_{28}	Wavelet Textures in V channel: level 1
f_{29}	Sum of f_{26}, f_{27}, f_{28}
f_{30}	low depth of field: H channel
f_{31}	low depth of field: S channel
f_{32}	low depth of field: V channel
f_{33}	GLCM texture features in H channel: Contrast
f_{34}	GLCM texture features in H channel: Correlation
f_{35}	GLCM texture features in H channel: Energy
f_{36}	GLCM texture features in H channel: Homogeneity
f_{37}	GLCM texture features in S channel: Contrast
f_{38}	GLCM texture features in S channel: Correlation
f_{39}	GLCM texture features in S channel: Energy
f_{40}	GLCM texture features in S channel: Homogeneity
f_{41}	GLCM texture features in V channel: Contrast
f_{42}	GLCM texture features in V channel: Correlation
f_{43}	GLCM texture features in V channel: Energy
f_{44}	GLCM texture features in V channel: Homogeneity
f_{45}	Color pixels in HSV image: Black
f_{46}	Color pixels in HSV image: White
f_{47}	Color pixels in HSV image: Gray
f_{48}	Color pixels in HSV image: Red
f_{49}	Color pixels in HSV image: Orange
f_{50}	Color pixels in HSV image: Yellow
f_{51}	Color pixels in HSV image: Green
f_{52}	Color pixels in HSV image: Cyan
f_{53}	Color pixels in HSV image: Blue
f_{54}	Color pixels in HSV image: Purple
f_{55}	Color pixels in HSV image: Magenta
f_{56}	Color pixels in HSV image: Pink

Table 1: All the image aesthetic features [Aza16] considered in our prediction model.

and female. During training each classifier, we ignore the user information (only consider it as a two class classification problem). Later, these trained classifiers are used in the proposed model to predict gender from a person’s bag of favorite images. Figure 1 shows the steps of feature selection and training phase. In the fig-

ure, S_F and S_M means set of images selected by female and male persons respectively.

3.2 Prediction Model

In this paper, we use the same prediction model that we proposed in [Aza16] for gender prediction using perceptual image aesthetic features. The model is a mixture of experts [Mik11, Dym05] where decision (probability of being female) of three well known binary classifiers: support vector machine (SVM), decision tree (D-Tree) and k-nearest-neighbor (KNN) are combined to make the final prediction [The08]. Here, each individual classifier is trained based on different feature spaces (having different dimensions) which make them distinct from each other. So combining their results minimizes the final prediction error. Figure 2 shows the block diagram of the prediction model. In the model, the probability of a person being female (P_f^{mix}) is calculated based on the weighted (w' , w'' and w''') combination of individual probabilities (P_f' , P_f'' and P_f''') generated by each classifier. The equation for P_f^{mix} is as follows

$$P_f^{mix} = w' P_f' + w'' P_f'' + w''' P_f''' . \quad (1)$$

The positive weight values multiplied with each classifier’s prediction define the influence of individual classifier. Assigning higher weight to a classifier means it is contributed more than others. Moreover, negative weight value is possible in the case where one classifier needs to minimize the error of higher weighted classifier. In the subsection 3.4, we describe the process of adjusting values of w' , w'' and w''' using genetic algorithm. Finally, the decision of gender is taken using P_f^{mix} and P_m^{mix} . If $P_f^{mix} > P_m^{mix}$ then the person is female. If $P_m^{mix} > P_f^{mix}$ then the person is male. The model takes random decision in the case of equal probability. In our experiment, we consider this case as “undecided”, and include it in the classification error.

3.3 Searching Best Feature Subset

Initially, we select all the aforementioned 56 features to train each classifier. We use the fine-tuned classifiers to ensure maximum accuracy as individual, as well as in the mixture of expert model. The way of fine-tuning is described in details at the experiment section. Table 2 shows the classification accuracy, number of selected features and overall testing time. Among them decision tree shows highest performance having 72.50% classification accuracy.

Instead of using all features, we need to find a subset of features that maximizes the prediction. One of the naive approach can be the brute force algorithm: checking all 2^N combinations of features where N is the number of features. In our case, $N = 56$ and each checking means

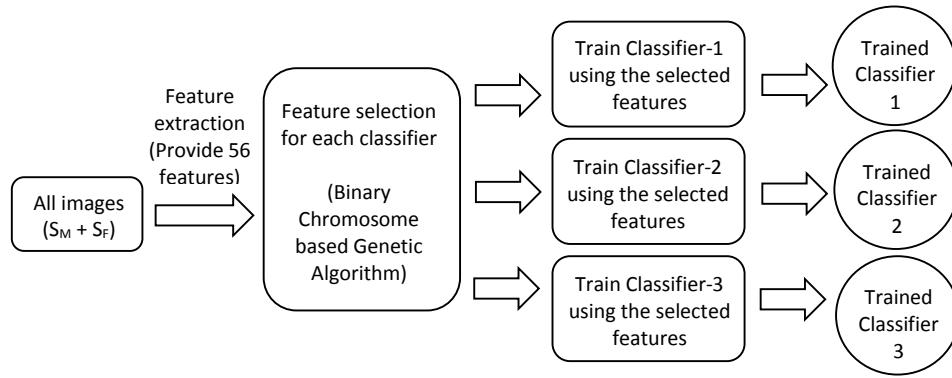


Figure 1: High level view of the feature selection and training phase.

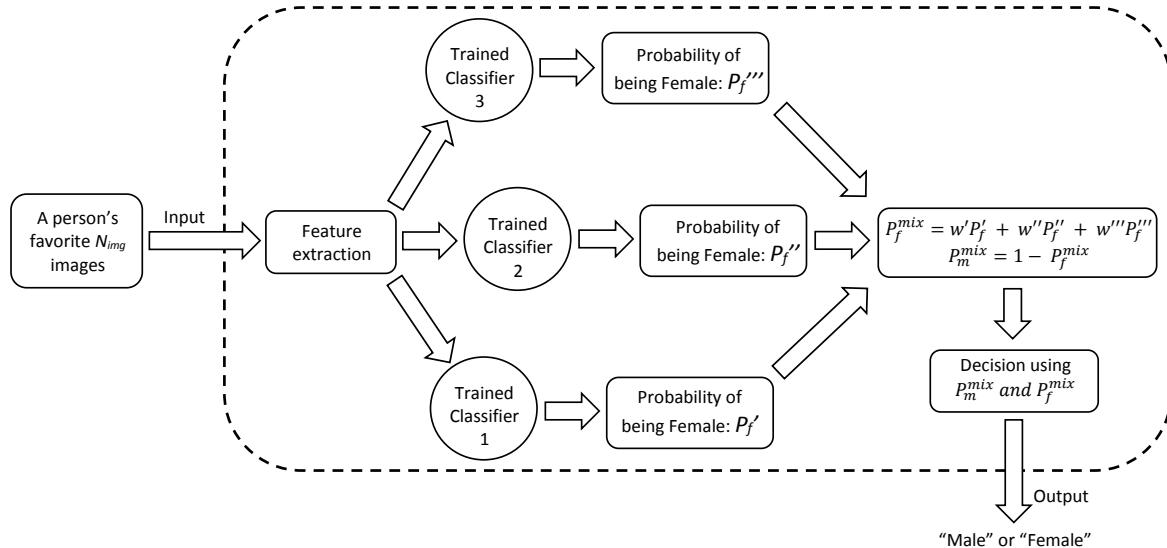


Figure 2: Block diagram of the gender prediction model.

Classifiers	Number of selected Features (out of 56)	Gender prediction accuracy (%)	Overall testing time (in seconds)
D-tree	All 56	72.50	1.43
SVM	All 56	70.83	14.48
KNN	All 56	66.67	33.28

Table 2: Performance of individual classifier when all 56 features are selected.

2 fold cross validation using $24000 \times N$ feature matrix. So it is not feasible, even impossible to run the brute force algorithm. One of the best way of feature selection is binary chromosome based genetic algorithm (GA) [Ray00]. Genetic algorithm is a stochastic search process for an optimal solution to a given problem. It can find the optimal or near optimal solution within a reasonable GA generations [Eng05]. For feature selection, we use binary chromosome of length 56 (56 dimensional binary search space) as an individual in the population where one bit represents one gene. Binary 1 means the corresponding feature is selected, binary 0 means the corresponding feature is not selected. We

run the GA algorithm for each classifier for 50 generations considering gender prediction error as fitness function. The parameter settings of the GA algorithm is provided in Section 4. Figure 3 shows the graph of GA generations (x-axis) vs prediction error (y-axis) for each classifiers. A black dot is the best individual (having minimum error), and a red dot is the average of fitness value of all individuals in a specific generation. From these graphs, we see that as the generation passes it minimizes the fitness values among all the individual in the population. We take the chromosome of the best individual and select the features according to the chromosome bit pattern. Table 3 shows all the features selected by GA for each classifiers. Here, the feature number is according to Table 1.

Next, we apply the selected features (using GA) to train and test each classifier individually. Table 4 shows that the classifier's performance increases in terms of feature reduction, testing time, as well as prediction accuracy after selecting features by GA. In Figure 4, a horizontal bar chart depicts the improvement for each classifier. The prediction accuracy of KNN, SVM and

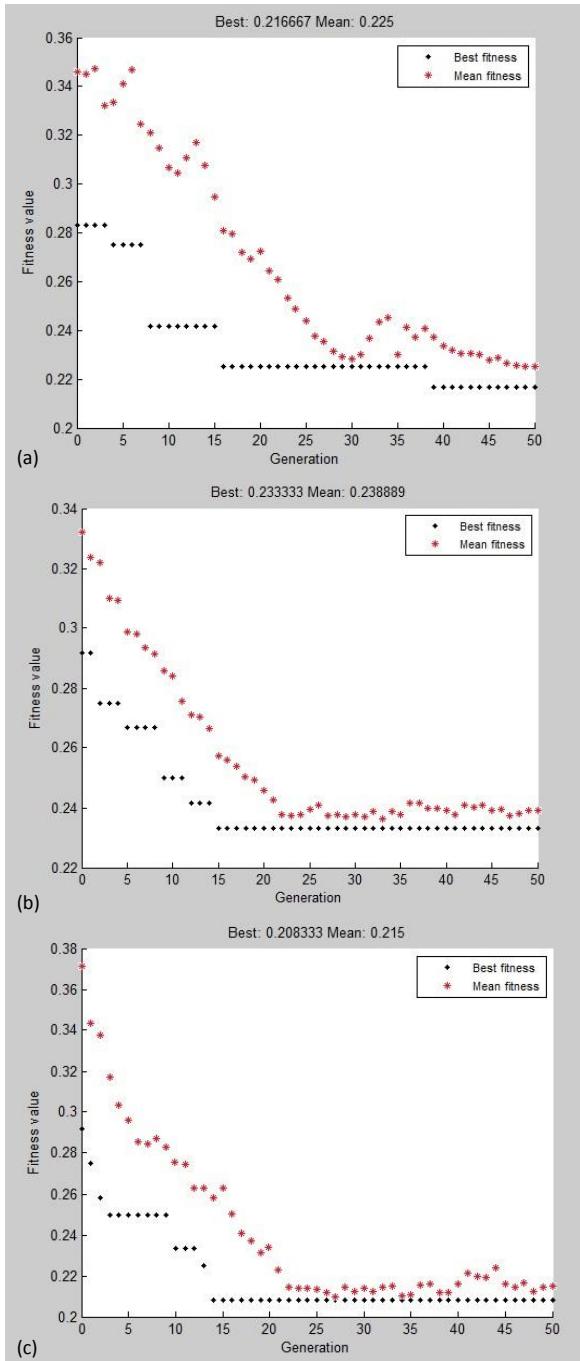


Figure 3: Graph shows the best fitness values (black dots) and the mean of population fitness values (red dots) over 50 generations. We see that best fitness value minimizes as the generation passes. Here the fitness value is the prediction error scaled to the range 0-1. Three graphs for the (a) Decision tree, (b) Support vector machine, and (c) K-nearest neighbor. At the top of each graph, the best fitness value and the mean fitness value of 50th generation are also provided.

D-tree are improved by 12.5%, 5.84% and 5.83%, respectively. Also the dimension of the feature spaces

and the overall testing times are minimized noticeably (see Table 2 and 4).

Classifiers	Selected features by GA for each classifier.
D-tree	1, 2, 3, 6, 7, 10, 11, 13, 16, 17, 18, 20, 24, 26, 27, 30, 31, 35, 36, 37, 38, 39, 41, 42, 44, 46, 47, 49, 51, 52, 53, 55, 56
SVM	3, 5, 6, 7, 11, 12, 14, 15, 16, 17, 19, 20, 23, 24, 29, 31, 33, 34, 41, 43, 44, 46, 47, 51, 52, 53, 54, 55
KNN	3, 4, 5, 6, 7, 10, 14, 18, 19, 20, 21, 22, 24, 28, 29, 30, 32, 34, 37, 39, 43, 44, 46, 47, 48, 49, 50, 52, 54, 55, 56

Table 3: Selected features by GA for each classifier.

Classifiers	Number of selected Features (out of 56)	Gender prediction accuracy (%)	Overall testing time (in seconds)
D-tree	34	78.33	1.13
SVM	28	76.67	13.10
KNN	31	79.17	20.48

Table 4: Showing the performance of individual classifier when features are selected by GA.

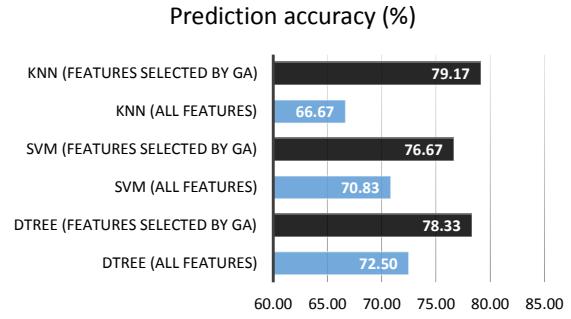


Figure 4: Bar chart showing the significant improvement of prediction accuracy after using GA based feature selection

3.4 Weight Adjustment Using GA

The key contribution of this paper is in the use of genetic algorithm (GA) for feature selection and weight assignment. In the mixture of expert models, we combine three classifiers by using simple weighted sum of the individual prediction. Before taking the final decision, the weighted probability is rescaled to 0 to 1. Assigning appropriate weights gives the benefit of using ensemble. Each weight controls the influence of individual prediction, as well as overall ensemble performance. Finding appropriate weights is a crucial step. In the experiment section, we show that assigning weights based on the partial participation and separate performance doesn't help to improve the model performance. Even it reaches only the maximum among classifiers. Moreover, empirically assigning weight values is not a trivial task because of large search space and highly non-linear function (ensemble accuracy). Due to three weights w' , w'' and w''' , our search space become 3D

floating number. To find the best values of weights that minimizes prediction error, we use genetic algorithm where a chromosome consists of three genes (floating point number) [Fli04]. Parameter settings of GA are provided in the experiment section. After 50 generations, GA gives 82.50% prediction accuracy at weight vector $(w', w'', w''') = (-1, 2.33, 0.88)$ where w' , w'' and w''' are the weights associated with D-tree, KNN and SVM respectively. Figure 5 shows the GA graph for weight adjustment. The axis setup of the graph is same as graphs in Figure 3.

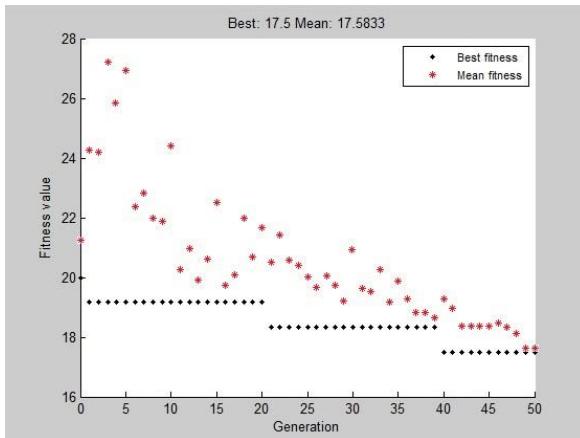


Figure 5: GA Graph showing Generation vs Fitness value (prediction error) for the purpose of weights adjustment.

4 EXPERIMENT

In our experiment, we consider a real database of 40000 color images collected from 200 Flickr users along with their profile reference. Each user provided 200 images from his or her favorite picture collection in OSN Flickr. Out of 200, we collect the gender information of 120 users: 60 male and 60 female. We conduct experiment on these 120 user's gender information and their 24000 favorite images. This is the same database used in [Aza16] for gender prediction experiment. According to [Lov14], duplicate images across users are less than 0.05%. The images are in different file format and resolution. Before using them into our experiment, we convert all images into JPEG file format. Then we extract all 56 features from these 24000 images, and make a data matrix of size 24000×57 . The 57th column contains the gender information of the Flickr users considering real-number 0 as male and 1 as female. Starts from row 1, consecutive 200 feature vectors are from a single user's 200 images. For implementation, we use MATLAB 2012 with image processing, machine learning and global optimization toolbox [Mat94], and a workstation having AMD A8-7410 APU 2.2 GHz processor with 8 GB RAM. Experimental setups, results and analysis are described in the following subsections.

4.1 Experimental Setup

In every stages of the proposed method, we evaluate each classifier and the prediction model based on average accuracy of 2 fold cross validation [Cro16]. Moreover, the fitness function in the GA is average misclassification error of 2 fold cross validation. We partition the whole image database into two folds, where fold 1 contains 12000 images from 30 male and 30 female persons, and fold 2 contains rest of the 12000 images from remaining 30 male and 30 female persons. There is no overlapping between these two folds. In any training and testing phase, we first train the model with fold 1, and test with fold 2. Then again train with fold 2 and test with fold 1. Finally, the average accuracy of fold 1 and fold 2 is considered as overall accuracy of a single classifier or the prediction model.

Before using three classifiers: SVM, KNN and D-tree in the mixture of expert models, we fine-tune them to ensure maximum performance as individual. For fine-tuning we apply iterative approach. In KNN classifier, one of the sensitive parameter is number-of-neighbor. We evaluate KNN for a range of number-of-neighbor values, and choose the number-of-neighbor where KNN shows maximum accuracy. Figure 6 shows the line graph of KNN for the number-of-neighbor range 1 to 50. Similarly, we iteratively fine-tune SVM (RBF kernel) and D-tree for their sensitive parameters sigma and minimum-leaf, respectively.

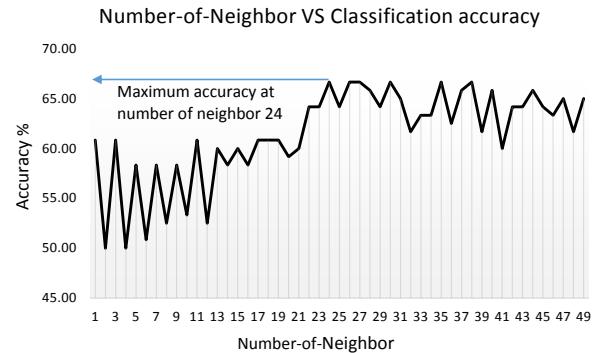


Figure 6: Line graph showing the iterative fine-tuning of KNN for the parameter number-of-neighbor.

In the proposed prediction method, we use genetic algorithm (GA) for feature selection and weight adjustment. In GA, several parameters are associated with it [Gen16]. Fine-tuning some of them may produce more optimal result, as well as speed up the GA execution. For simplicity, we keep most of the parameters to its default value as per MATLAB function documentation. Moreover, iterative approach is not a good way to tune GA because of stochastic nature. For same reason, we run GA multiple times for feature selection and weight adjustment, and keep the best result. Table 5 shows all the parameter values of three classifiers and the genetic

algorithm used in the experiment. Other parameters are set to their default value as per MATLAB function documentation.

Classifiers / GA	Parameter settings
KNN	Number of neighbors: 24.
D-tree	Minimum-leaf: 34.
SVM	Non-linear kernel: Radial Basis Function (RBF); Sigma value of RBF: 2.9; Maximum iteration: 30000.
GA	Population type: bit-string (during feature selection) and double-vector (during weight adjustment), Population size: 20, Fitness limit: 0.05; Generations: 50.

Table 5: Parameter settings for each binary classifier and GA.

4.2 Experimental Results

At first, we assign different combination of weights in the equation (1) considering either zero or one as weight value. For weight combination (0, 0, 1), (0, 1, 0), and (1, 0, 0), the prediction model actually shows individual performance of SVM, KNN and D-tree. As separate classifier, KNN performs maximum (79.17%) among them. Table 6 shows the prediction accuracy, male and female counting, as well as count of undecided cases for different weight combinations. In the table, we see that pair combinations (1, 1, 0), (0, 1, 1) and (1, 0, 1) show degraded performance of 77.50% which is less than the maximum performed classifier within the pair. After observing each case, we found that in few cases when a well formed classifier gives correct prediction with marginal probability (0.51 to 0.55), the other one gives incorrect prediction with low probability (0.45 to 0.49). So weighted probability drops below 0.5. Due to similar reason, the prediction model shows more degraded performance for the weight vector (1, 1, 1). Another weight assignment can be based on ranking of individual performance. According to classification accuracy, we rank D-tree, KNN and SVM as 2, 3, and 1. After assigning these rankings as weights in the prediction model, we get 79.17% accuracy which is equal to the best individual performance by KNN. Even after increasing the weight value of top two classifiers, it doesn't show any improvement in accuracy (see the Table 6 for weight vector (3, 4, 1)). Overall, weight assignment based on performance and participation doesn't show good results. To find the best combination of weights for this non-linear function, we apply genetic algorithm. After 50 generations, it gives the weight vector (-1, 2.33, 0.88) which brings 82.50% of prediction accuracy (99 users out of 120) with zero undecided situation. Here, the noticeable weight value is negative 1 for D-tree, which actually minimizes the female prediction error done by KNN.

For comparing the proposed method, we select our previous image aesthetic based gender prediction method

[Aza16] as the best result reported so far in the literature on the subject. We applied greedy feature selection and ranking based weight adjustment which reported 76.65% of accuracy over the same database. The overall testing time (all 120 users) of the method [Aza16] is 12 seconds, whereas the proposed method takes 21.6 seconds. The number of features (after GA selection process) used in the proposed method is higher than the method in [Aza16]. Experimentally we find that the testing time of KNN increases with the number of features. The required memory for both of these methods is approximately 13.62 MB, because they using the same prediction model. The memory effect of increased number of features in the proposed method is insignificant. Another image based gender prediction method is [Qua14], which considered favorite images and posting behavior of OSN Pinterest [Pin10] users, and reported accuracy is 72% [Qua14]. The required time and memory for the system is not reported by the authors. Figure 7 shows the performance of [Aza16] with ranking based weight assignment, and the proposed method with same weight assignment, as well as after weight adjustment using GA. The proposed method shows further close to 83% gender prediction accuracy and thus proves the superiority of the currently proposed method over the existing approaches.

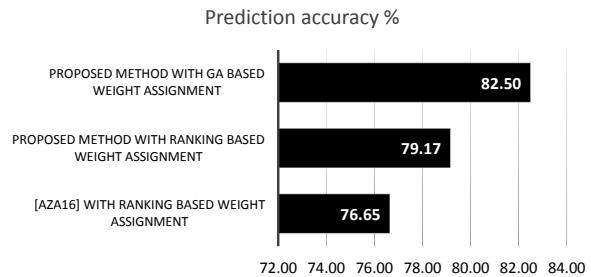


Figure 7: Bar chart showing the accuracy of the proposed gender prediction method and the method in article [Aza16].

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a new method to predict gender from a person's favorite images. We consider 56 image aesthetic features from existing literatures, and a mixture of expert models consisting of SVM, KNN and D-tree. Final decision is taken based on the weighted combination of probability generated by individual classifiers. To improve the prediction accuracy of the model, we find the best combination of feature subset using genetic algorithm in 56D binary search space. Moreover, feature dimension is reduced significantly that decreases the testing time. Finally, three weights of the prediction model are adjusted using genetic algorithm in 3D real-number search space.

Weights ($W_{dtree}, W_{knn}, W_{svm}$)	No. of Undecided situation (out of 120)	No. of correct male prediction (out of 60)	No. of correct female prediction (out of 60)	Accuracy (%)
(0,0,1)	0	41	51	76.67
(1,0,0)	2	48	46	78.33
(0,1,0)	2	53	42	79.17
(1,1,0)	0	50	43	77.50
(1,0,1)	1	44	49	77.50
(0,1,1)	0	47	46	77.50
(1,1,1)	3	46	46	76.67
(2,3,1)	0	49	46	79.17
(3,4,1)	0	49	46	79.17
(-1, 2.33, 0.88)	0	53	46	82.50

Table 6: Prediction performance for different weighted combination of three binary classifiers.

Experiment is conducted on a real image database of 24000 images provided by 120 Flickr users. The proposed method shows 82.50% accuracy in gender prediction. As our future work, we will incorporate contextual image aesthetic features to improve the prediction accuracy. Investigation will be needed to see the performance of other machine learning algorithms to make sophisticated and well performed prediction model. Moreover, fine-tuning of GA parameters can be applied to hope for better weight adjustment.

6 ACKNOWLEDGMENTS

We would like to acknowledge NSERC Discovery Grant RT731064, as well as NSERC ENGAGE and URCG for partial funding of this project. Our thanks to all the members of BTLab, Department of Computer Science, University of Calgary, Calgary, AB, Canada for providing their valuable suggestions during conducting this research.

7 REFERENCES

- [Ahm15a] Ahmed, F., Paul, P.P., Gavrilova, M., “DTW-based Kernel and Rank Level Fusion for 3D Gait Recognition using Kinect,” Visual Computer, vol. 31, no. 6-8, pp. 915-924, 2015.
- [Ahm15b] Ahmed, F., Paul. P.P., Gavrilova. M., “Kinect-Based Gait Recognition Using Sequences of the Most Relevant Joint Relative Angles,” journal of WSCG, pp. 147-156, vol. 23, no. 2, 2015.
- [Ayd15] Aydin, T.O., Smolic, A., Gross, M., “Automated Aesthetic Analysis of Photographic Images,” IEEE Transactions on Visualization and Computer Graphics, vol. 21, no. 1, pp. 31-42, Jan 01 2015.
- [Aza16] Azam, S., Gavrilova, M., “Soft Biometric: Give Me Your Favorite Images and I Will Tell Your Gender,” in 15th IEEE International Conference on Cognitive Informatics and Cognitive computing (ICCI*CC), Stanford University, USA, Aug 2016. [Accepted]
- [Cel09] Cela-Conde, C. J., Ayala, F. J., Munar, E., Maestu, F., Nadal, M., Capo, M. A., Marty, G., “Sex-related similarities and differences in the neural correlates of beauty,” Proceedings of the National Academy of Sciences, vol. 106. no. 10, pp. 3847-3852, 2009.
- [Cro16] “Cross-validation,” [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) [Online accessed: 19 January 2016].
- [Dan16] Dantcheva, A., Elia, P., Ross, A., “What Else Does Your Biometric Data Reveal? A Survey on Soft Biometrics,” in IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 441-467, March 2016.
- [Dym05] Dymitr, R., Bogdan, G., “Classifier selection for majority voting,” Information Fusion, Elsevier, Vol. 6, Issue. 1, pp. 63-81, March 2005.
- [Eng05] Engelbrecht, A., “Fundamentals of Computational Swarm Intelligence,” John Wiley and Sons Canada, ISBN: 1-55860-595-9, 2005.
- [Fli04] “Flickr”, <https://www.flickr.com/> [Online accessed: 19 January 2016].
- [Gav13] Gavrilova, M., Monwar, M., “Multimodal Biometrics and Intelligent Image Processing for Security Systems”, IGA book, 250 pages, 2013.
- [Gen16] “Genetic Algorithm”, <http://www.mathworks.com/help/gads/genetic-algorithm.html> [Online accessed: 25 February 2016]
- [Gur11] Gurbuz, E., Senyer, N., “Gender estimation according to Gait by using ellipse fitting and static body parameter approaches,” IEEE 19th Conference on Signal Processing and Communications Applications (SIU), Antalya, pp. 162-165, 2011.
- [Iiv03] Iivari, K., Leena, L., Juhani, R., Ari, V., “Binary Histogram in Image Classification for Retrieval Purposes,” Journal of WSCG, WSCG 2003, vol. 11, no. 1, pp. 269-273, Feb 2003.
- [Jan10] Jana, M., Allan, H., “Affective image classification using features inspired by psychology and art theory,” In Proceedings of 18th ACM international conference on Multimedia (MM), ACM, New York, NY, USA, pp. 83-92, 2010.

- [Jia10] Jiang, W., Loui, A.C., Cerosaletti, C.D., "Automatic aesthetic value assessment in photographic images," in IEEE International Conference on Multimedia and Expo (ICME), pp. 920-925, 19-23 July 2010.
- [kuk04] Kukharev, G., Nowosielski, A., "Visitor Identification - Elaborating Real Time Face Recognition System," WSCG 2004 SHORT Communication papers proceedings, Feb. 2004.
- [Lee03] Leena, L. Iivari, K., Juhani, R., Ari, V., "Rock Image Classification Using Non-Homogenous Textures and Spectral Imaging," in Short Paper Proceedings of 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2003.
- [Lov12] Lovato, P., "Tell me what you like and I'll tell you what you are: Discriminating visual preferences on Flickr data", Proc. Asian Conf. Comput. Vis, pp. 45 -56, 2012.
- [Lov14] Lovato, P., Bicego, M., Segalin, C., Perina, A., Sebe, N., Cristani, M., "Faved! Biometrics: Tell Me Which Image You Like and I'll Tell You Who You Are," IEEE Transactions on Information Forensics and Security, vol. 9, no. 3, pp. 364-374, March 2014.
- [Mad14] M. Sultana, P. P. Paul and M. Gavrilova, "A Concept of Social Behavioral Biometrics: Motivation, Current Developments, and Future Trends," 2014 International Conference on Cyberworlds (CW), pp. 271-278, 2014.
- [Mat94] "Mathworks", <http://www.mathworks.com/products/matlab/> [Online accessed: 01 January 2016]
- [Mar11] Marchesotti, L., Perronnin, F., Larlus, D., Csurka, G., "Assessing the aesthetic quality of photographs using generic image descriptors," in IEEE International Conference on Computer Vision (ICCV), pp. 1784-1791, 6-13 Nov 2011.
- [Mik11] Mikel, G., Alberto, F., Edurne, B., Humberto, B., Francisco, H., "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," Pattern Recognition, Elsevier, vol. 44, no. 8, pp. 1761-1776, August 2011.
- [Mol14] Molchanov, V., Linsen, L., "Visual Exploration of Patterns in Multi-run Time-varying Multi-field Simulation Data Using Projected Views," Proceedings of WSCG, The 22nd International Conference on Computer Graphics, Visualization and Computer Vision, vol. 21, pp. 49-58, 2014.
- [Mos06] Moss, G., Gunn, R. and Heller, J., "Some men like it black, some women like it pink: consumer implications of differences in male and female website design," Journal of Consumer Behavior, vol. 5, pp. 328-341, 2006.
- [Mos07] Moss, G., Gunn, R., "Gender differences in website design: Implications for education," Journal of Systems, Cybernetics and Informatics, vol. 5, no. 6, pp. 38-43, 2007.
- [Pin10] "Pinterest", <https://www.pinterest.com/> [Online accessed: 19 January 2016].
- [Pri13] B., Priyadarshi, Gavrilova, M., "Spatial consistency of dense features within interest regions for efficient landmark recognition," Visual Computer, vol. 29, no. 6-8, pp. 491-499, June 2013.
- [Qua14] Quanzeng, Y., Bhatia, S., Tong, S., Jiebo, L., "The Eyes of the Beholder: Gender Prediction Using Images Posted in Online Social Networks," in IEEE International Conference on Data Mining Workshop (ICDMW), pp. 1026-1030, 14 Dec 2014.
- [Ray00] Raymer, M., Punch, W., Goodman, E., Kuhn, L., Jain, A., "Dimensionality reduction using genetic algorithms," IEEE Transactions on Evolutionary Computation, vol. 4, no. 2, pp. 164-171, 2000.
- [Rit06] Ritendra, D., Dhiraj, J., Jia, L., James, Z. W., "Studying aesthetics in photographic images using a computational approach," In Proceedings of the 9th European conference on Computer Vision (ECCV), Springer-Verlag, Berlin, Heidelberg, pp. 288-301, 2006.
- [Ron97] Ron, K., George, H. J., "Wrappers for feature subset selection," Artificial Intelligence, Elsevier, vol. 97, no. 1-2, pp. 273-324, Dec 1997.
- [San07] Sanchez-Marono, N., Alonso-Betanzos, A., Tombilla-Sanroman, M., "Filter methods for feature selection: a comparative study," In Intelligent Data Engineering and Automated Learning (IDEAL), pp. 178-187, Springer Berlin Heidelberg, 2007.
- [Seg14] Segalin, C., Perina, A., Cristani, M., "Personal Aesthetics for Soft Biometrics: A Generative Multi-resolution Approach," In Proceedings of the 16th International Conference on Multimodal Interaction (ICMI '14), ACM, New York, NY, USA, pp. 180-187, 2014.
- [Ska13] Skala, V., "Fast reconstruction of corrupted images and videos by radial basis functions," International Conference on Control, Automation and Information Sciences (ICCAIS), pp. 267-271, 2013.
- [Sul15] Sultana, M., Paul, P.P., Gavrilova, M. "Occlusion Detection and Index-based Ear Recognition, Journal of WSCG, pp. 121-129, vol. 23, no. 2, 2015.

- [Tew15] Tewari, A., Taetz, B., Stricker, D., Grandidier, F., "Using Mutual Independence of Slow Features for Increased Information and Improved Hand Pose Classification," Journal of WSCG, vol. 23, no. 1, pp. 35-44, 2015.
- [The08] Theodoridis and Koutroumbas, "Pattern Recognition," 4th Edition, Academic Press, pages 984, 2008.
- [Xia13] Xiaohui, W., Jia, J., Jiaming, Y., Lianhong, C., "Interpretable aesthetic features for affective image classification," in 20th IEEE International Conference on Image Processing (ICIP), pp. 3230-3234, 15-18 Sep 2013.

Compression Artifacts Removal Using Convolutional Neural Networks

Pavel Svoboda

Michal Hradis

David Barina

Pavel Zemcik

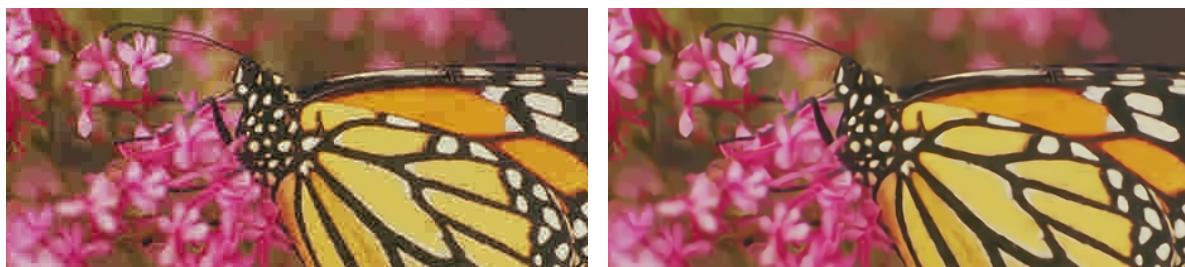
Faculty of Information Technology

Brno University of Technology

Bozatechova 1/2, Brno

Czech Republic

{isvoboda,ihradis,ibarina,zemcik}@fit.vutbr.cz



ABSTRACT

This paper shows that it is possible to train large and deep convolutional neural networks (CNN) for JPEG compression artifacts reduction, and that such networks can provide significantly better reconstruction quality compared to previously used smaller networks as well as to any other state-of-the-art methods. We were able to train networks with 8 layers in a single step and in relatively short time by combining residual learning, skip architecture, and symmetric weight initialization. We provide further insights into convolution networks for JPEG artifact reduction by evaluating three different objectives, generalization with respect to training dataset size, and generalization with respect to JPEG quality level.

Keywords

Deep learning, Convolutional neural networks, JPEG, Compression artifacts, Deblocking, Deringing

1 INTRODUCTION

This work presents a novel method of image restoration using convolutional networks that represents a significant advancement compared to the state-of-the-art methods. We study the direct approach [12] in which a fully convolutional network accepts a degraded image as input and outputs a high quality image. By making a number of important improvements regarding the network architecture, initialization, and training, we are able to train large and deep networks for JPEG compression artifact reduction which surpass the state-of-the-art in this task. The networks predict a residual image [16] describing changes to be applied

to the input image, and they incorporate skip connections [18] which allow information to bypass the middle layers. We reduce the "saturation" of ReLU units in deeper layers by centering filters during network initialization which allows us to use significantly faster learning rates.

Lossy image compression achieves high compression ratios through elimination of information that does not contribute to human perception of images, or contributes as little as possible. Due to the limitations of the human visual system, such loss of information may be acceptable in many scenarios but the introduced visual artifacts become unacceptable at higher compression ratios. The primary methods currently used for lossy image compression include JPEG and JPEG 2000. This paper focuses on the JPEG compression method [13] and the degradation it causes. The JPEG compression chain consists of a block-based discrete cosine transform (DCT), followed by a quantization step utilizing a quantization matrix, and an entropy coding. The decompression follows this process in reverse order.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Blocking, blurring, and ringing artifacts are typical examples of image degradation caused by the lossy compression methods. Considering the JPEG method, the degradation is the result of information loss in the DCT coefficient quantization step. More specifically, the blocking artifacts are caused by the grid segmentation into 8×8 cells employed in the JPEG standard and the resulting discontinuities at the cell edges. The ringing artifacts (or the Gibbs phenomenon) are induced oscillations caused by removal of high frequencies during the quantization. The removal of high frequencies causes blurring as well, but the blurring is less noticeable compared to the ringing artifacts. Blocking is mostly noticeable in low-frequency regions, while the ringing artifacts are especially well noticeable around sharp edges.

The convolutional networks have to learn to recognize the compression artifacts and fill them appropriately with respect to the neighboring image content. In this sense, the networks incorporate both the data term and prior regularization term of standard image restoration techniques, and they can make use of correlations between image content and the image degradation.

Convolutional networks have been successfully used in many image restoration tasks including super resolution [4, 16], denoising [15], structured noise removal [6], non-blind deconvolution [21, 30], blind deconvolution in specific image domains [12, 24], and sub-tasks of blind deconvolution [20]. Our work was mostly inspired by the large deblurring networks of Hradis *et al.* [12], and by Kim *et al.* [16] who showed that residual learning together with good weight initialization enabled training of large convolutional networks for super resolution. We extend the work of Dong *et al.* [4] who achieved state-of-the-art compression artifact reduction even with very small convolutional networks. However, they were not able to scale up their networks due to problems with training convergence.

2 RELATED WORK

A large number of methods designed to reduce compression artifacts exist ranging from relatively simple and fast hand-designed filters to fully probabilistic image restoration methods with complex priors [29] and methods which rely on advanced machine learning approaches [4].

Simple deblocking and artifact removal postprocessing filters are included in most image and video viewing software. For example, the FFmpeg framework includes the simple postprocessing (spp) filter [19] which simply re-applies JPEG compression to the shifted versions of the already-compressed image, and averages the results. The spp filter uses the quantization matrix (compression quality) of the original compressed image as the matrix has to be stored with the image

to allow for decompression. Pointwise Shape-Adaptive DCT (SA-DCT) [7, 8], in which the thresholded or attenuated transform coefficients are used to reconstruct a local estimate of the signal within the adaptive-shape support, is currently considered the state-of-the-art deblocking method. However, similarly to other deblocking methods, SA-DCT overly smooths images and it is not able to sharpen edges. In video compression domain, advanced in-loop filters (deblocking and SAO filters) known from video compression standards like H.264 or H.265 are obligatorily applied. A completely different deblocking approach was presented in [31], where the authors applied DCT-based lapped transform on the signal already in the DCT domain in order to undo the harm done by the DCT domain processing. However, the video in-loop deblocking methods, SA-DCT deblocking (only to estimate parameters), and methods derived from the lapped DCT rely on the cognizance of the DCT grid. Unlike these methods, the method proposed in this paper is able to process images without such knowledge.

This work focuses on application of convolutional networks to reconstruction of images corrupted by JPEG compression artifacts. Convolutional networks belong to an extensively studied domain of deep learning [2]. Recent results in several machine learning tasks show that deep architectures are able to learn the high level abstractions necessary for a wide range of vision tasks including face recognition [25], object detection [9], scene classification [17], pose estimation [26], image captioning [27], and various image restoration tasks [4, 16, 15, 6, 21, 30, 12, 24, 20]. Today, convolutional networks based approaches show the state-of-the-art results in many computer vision fields.

Small networks were historically used in image denoising and other tasks. On the other hand, deep and large fully convolutional networks have become only recently important in this field. Burger *et al.* [3] used feed forward three layer neural network for image denoising. While there were attempts to use neural networks for denoising before, Burger *et al.* showed that this approach can produce state-of-the-art results when trained on a sufficiently large dataset.

A non-blind deconvolution method of Schuler *et al.* [21] uses a regularized inversion of the blur kernel in Fourier domain followed by a multi-layer perceptron (MLP) based denoising step. The shortcoming of the approach is that a separate MLP models have to be trained for different blur kernels, as a general models trained for multiple blur kernels provide inferior reconstruction quality. Schuler *et al.* [20] introduced a learning based approach to blind deconvolution. They perform a regression from the blurred image towards the source blur kernel. The neural network itself is trained to extract image features useful for estimation

of the blur point spread function. Sun *et al.* [23] presented CNN-based approach for non-uniform motion blur removal which classified image patches into closed set of blur kernel types. The local classification outputs were used as input to a Markov random field model which estimates the dense non-uniform motion blur field over the whole image. Hradis *et al.* [12] trained CNNs composed of only convolutional layers and rectified linear units (ReLU) to directly map blurred and noisy images of text images to high quality clean images. The approach was extended by Svoboda *et al.* [24] who demonstrated high quality deblurring reconstructions for car license plates in a real-life traffic surveillance system. Their results show that a single CNN can be trained for a full range of motion blurs expected to appear in a specific traffic surveillance camera resulting in a robust and fast system.

Dong *et al.* [4] introduced super-resolution convolutional neural network (SRCNN) to deal with the ill-posed problem of super-resolution. The SRCNN is designed according the classical sparse coding methods – the three layers of SRCNN consist of feature extraction layer, a high dimensional mapping layer, and a final reconstruction layer. The very deep CNN based super-resolution method proposed by Kim *et al.* [16] builds on the work of Dong *et al.* [4] and it shows that deep networks for super-resolution can be trained when proper guidelines are followed. They initialized networks properly and they used so-called residual learning in which the network predicts how the input image should be changed instead of predicting the desired image directly. Residual learning appears to be very important in super-resolution. The resulting 20 layers deep networks trained with adjustable gradient clipping significantly outperform previous approaches. However, it is unclear how effective residual learning would be in other image processing tasks where the networks inputs and outputs are not correlated that strongly as in super-resolution. We follow this approach in our work on JPEG reconstruction.

Convolutional networks have previously been used for suppressing compression artifacts by Dong *et al.* [5], who proposed a compact and efficient CNN based on SRCNN – artifacts removing convolutional network (AR-CNN). AR-CNN extends the original architecture of SRCNN with feature enhancement layers. The network training consist of two stages – a shallow network is trained first and it is used as an initialization for a final 4 layer CNN. As reported in the paper, this two stage approach improved results due to training difficulties encountered when training the full 4 layer network from scratch. The authors also state that they aim to achieve feature enhancement instead of just making the CNN deeper. They argue that although the deeper SRCNN introduces a better regressor between the low-level features and the reconstruction, the bottleneck lies

on the features. Thus the extracting layer is augmented by the enhancement layer which together may provide better feature extractor.

We adapt the idea of residual learning [16] for the JPEG compression artifact removal based on CNN. We follow the assumption "deeper is better" and we try to learn our deep residual CNNs in a single step by creating a new recipe including initialization, network architecture, and high learning rates. The resulting networks significantly outperform the classical JPEG compression artifact removal methods, as well as, the AR-CNN [5] on common dataset measured by PSNR, specialized deblocking assessment measure PSNR-B, and SSIM.

3 CNN IMAGE ENHANCEMENT

In computer vision, CNNs are most extensively studied in the context of classification, semantic class segmentation, object detection, and captioning where the networks are often constrained to a fixed input size. This is due to the fully connected layers which are used as the final layers in order to aggregate information from a whole image. In low level image processing (but not limited to it), the so-called fully convolutional neural networks [18] (FCN) are preferred as they behave as non-linear convolutional operators – they process each image position the same way and they can be applied to images of arbitrary size.¹ The architecture of fully convolutional networks is limited to convolutional operations (linear convolution, so-called deconvolution, local response normalization, and local pooling) and element-wise operations. Most image processing networks use only convolutions and element-wise non-linearities (ReLU, sigmoid, tanh) [12, 24, 5, 16, 4, 21, 20]. In the case that no pooling and no deconvolution layer is used, the size of the input is reduced only by size of the convolution layer kernels (by the size of receptive field).

The fully convolutional networks F used in our work consist of an input data layer F_0 , convolutional layers F_ℓ , where $0 < \ell \leq L$ with F_ℓ weights represented as convolutional kernels W_ℓ with their biases b_ℓ , and element-wise max operations (ReLU) as follows:

$$\begin{aligned} F_0(y) &= y \\ F_\ell(y) &= \max(0, W_\ell * F_{\ell-1}(y) + b_\ell) \\ F(y) &= W_L * F_{L-1}(y) + b_L \end{aligned} \quad (1)$$

Where y is the distorted input image and $F(y)$ is the restored output image.

¹ In practice, the minimum size of processed images is constrained by the receptive field size of the network.

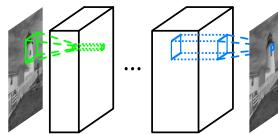


Figure 1: Illustration of a network with direct architecture.

We use the standard mean squared error (MSE) objective function

$$\frac{1}{n} \sum_{i=1}^n \|F(y_i) - x_i\|_2^2, \quad (2)$$

which is often used for general image enhancement. It is computed on a training data represented as pairs (y_i, x_i) , $0 < i \leq n$, where y_i represents the reconstructed image and x_i its corresponding clean image.

Direct mapping objective. In direct mapping shown in Figure 1, the networks learn to transform corrupted images directly to clean images. This approach leads to high quality results in specific low level image processing tasks i.e. in blind and non-blind deconvolution for text denoising or motion deblurring [12, 24], in super-resolution [4] or JPEG compression artifacts reduction [5]. Direct mapping forces the network to transfer the whole image through all its layers until it reaches the output. The learning of such autoencoder-like mapping in situations where the input images are highly correlated with the desired outputs may be wasteful especially for large and deep networks. It may be one of the main reasons why Dong *et al.* [5] were not able to scale up their networks and why they required approximately 10^7 iterations to train their AR-CNN. Similar problems were reported by Kim *et al.* [16].

Residual objective. The residual objective was originally introduced for super-resolution [16] where the input and output images are highly correlated. Instead of learning to predict the output image, the network in residual learning learns the changes which should be applied to the input image – it predicts the residual image $r = y - \hat{x}$ between the distorted y and latent high-quality image \hat{x} . The residual learning scheme is depicted in Figure 2. Kim *et al.* [16] were able to speed up

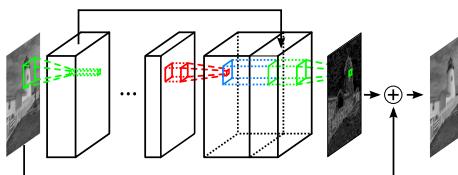


Figure 2: Illustration of a network with skip architecture and residual loss.

the training by large factor of (up to $10^4 \times$) with residual learning and it allowed them to learn much deeper networks – 20 layers vs. 3 in [4] and 4 in [5].

Edge emphasized objective. Mean square error used in many image restoration methods does not necessarily correlate well with the image quality perceived by humans. With convolutional networks, it is relatively easy to use more perceptually valid error measures as objective functions, as long as they can be efficiently differentiated (e.g. SSIM). We decided to add partial first derivatives of the image to the loss function in a form of vertical and horizontal Sobel kernels. This is achieved by adding the objective function computed on image derivatives calculated by Sobel kernels G as

$$\frac{1}{n} \sum_{i=1}^n \|G * F(y_i) - G * x_i\|_2^2. \quad (3)$$

Our assumption is that the addition of the first derivatives should force the network to focus specifically on high frequency structures such as edges, ringing artifacts, and block artifacts and it could lead to perceptually better reconstructions. The combined edge emphasized loss can be easily implemented in all existing convolutional network frameworks by defining the derivative Sobel kernels as a convolutional layer with predefined fixed filters. The network utilizing such objective function is shown in Figure 3.

Symmetric weight initialization. Weights in convolutional networks are usually initialized by sampling from some simple distribution (e.g. Gaussian or uniform) with mean equal to 0. The zero mean is desirable as it prevents mean offsets of activations to propagate through the layers. In case the mean was not zero, any mean offset in input values would result in non-zero mean of output activations which could force the ReLU non-linearities to get fully stuck either in the positive linear interval or, even worse, in the negative interval where gradients are not propagated rendering the unit useless.

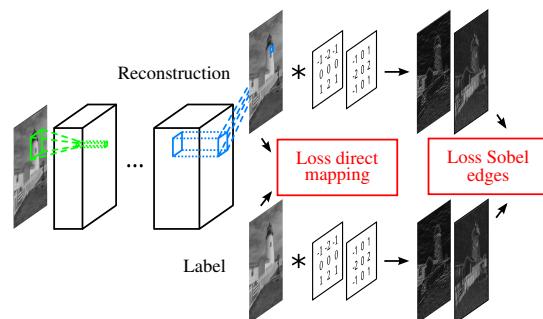


Figure 3: Illustration of a network with edge preserving loss.

Layer	1	2	3	4(+1)	5	6(+1)	7	8
Filter size	11×11	3×3	3×3	3×3	1×1	5×5	1×1	5×5
Channels	32	64	64	64(+32)	64	64(+32)	128	1

Table 1: L8 architecture – filter size and number of channels for each layer.

Although the weights are sampled from a distribution with zero mean, the means of individual convolutional filters are not zero due to the fact that they are a finite sample from the distribution. These random offsets together with the positive offset of ReLU activations cause units in deeper layers to become more likely to be either permanently turned off or turned on, which increases sparsity of the activations and increases effective mean offsets of the deeper layers. The result is that that majority of units in deep layers become almost useless right after the initialization.

Some activation normalization methods, such as "batch normalization" [14], can eliminate the saturation problem, but the normalization introduces noise during training which is not desirable for image restoration networks.

We eliminate this problem by explicitly forcing individual filters to have zero mean during initialization. Such initialization allows us to use significantly higher initial learning rates, especially together with residual learning, and it results in trained networks with significantly fewer saturated neurons.

We could explicitly force all filters to have zero mean during the whole training. Such constraint almost entirely eliminates any potential for unit saturation, but it prevents networks to utilize the DC component of input signals. Although we were able to achieve reasonably good results with this constraint in our preliminary experiments, we did not find it necessary and it was not used in the experiments presented in this paper.

Skip architecture. Deeper networks may have problems with exploding and vanishing gradients and they may take a long time to learn to efficiently propagate information through large number of layers. The problems with the gradients can be eliminated by proper initialization [10]. The problems with propagating information through many layers can be alleviated by bypassing some layers [18] or by letting layers to learn residual of their inputs [11]. The skip architecture with the residual objective function is shown in Figure 2.

We employ a skip architecture similarly to Long *et al.* [18]. We feed activations of the first convolutional layer to some deeper layers bypassing the layers in-between. Unlike Long *et al.* [18] who add the activations together, we concatenate them. The goal of the skip architecture is to allow the network to pass geometric information easily from the input to the output,

Layer	1	2	3	4
Filter size	11×11	3×3	3×3	5×5
Channels	48	64	64	1

Table 2: L4 architecture – filter size and number of channels for each layer.

and to allow for more complex reasoning about the image content in the middle layers (e.g. what is an artifact and what local context information should be used to repair the artifacts).

Network architectures. This paper presents two different FCN architectures which use only convolutional units and ReLU non-linearities. The first architecture denoted as L4 is relatively small with four layers defined in Table 2. The second network, denoted as L8, has eight layers and it utilizes the skip architecture by concatenating activation of the first layer with activations of the fourth and sixth layers. The exact definition of L8 is in Table 1. The receptive fields of L4 and L8 are 19×19 and 25×25 , respectively.

4 EXPERIMENTAL RESULTS

All the experiments were computed on images from BSDS500 [1] and LIVE1 [22] datasets. The networks were trained solely on the merged train and test part of BSDS500 which contain 400 images. The images were transformed to gray-scale using the YCbCr color model by keeping the luma component – Y only. Although the networks can process color images, we evaluate on gray-scale images because we focus on the ringing and blocking artifacts and not on the chromatic distortions. The gray-scale images were compressed with the MATLAB JPEG encoder into six disjoint sets according the JPEG quality. Specifically, we use images compressed with the quality 10, 20, 30, 40, 50, and 60.

The networks were evaluated on the test set from BSDS500 which includes 100 high quality compressed images and on the LIVE1 dataset containing 29 color images (uncompressed BMP format). All the evaluation images were transformed to gray-scale the same way as the training images and compressed using the same encoder.

Several metrics for objectively assessing perceptual quality of images exist. We use PSNR, PSNR-B, and SSIM. Generally, the most commonly used quality metric is the mean squared error (MSE). This quantity is computed by averaging squared intensity differences

method	Q10			Q20		
	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	27.77	25.33	0.791	30.07	27.57	0.868
spp	28.37	27.77	0.806	30.49	29.22	0.877
SA-DCT	28.65	28.01	0.809	30.81	29.82	0.878
AR-CNN	28.98	28.70	0.822	31.29	30.76	0.887
L4 Residual	29.08	28.71	0.824	31.42	30.83	0.890
L8 Residual	—	—	—	31.51	30.92	0.891

Table 3: Image reconstruction quality on LIVE1 test dataset for JPEG quality 10 and 20.

method	Q10			Q20		
	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	27.58	24.97	0.769	29.72	26.97	0.852
spp	28.13	27.49	0.782	30.11	28.68	0.859
AR-CNN	28.74	28.38	0.796	30.80	30.08	0.868
L4 Residual	28.75	28.29	0.800	30.90	30.13	0.871
L8 Residual	—	—	—	30.99	30.19	0.872

Table 4: Image reconstruction quality on BSDS500 test dataset for JPEG quality 10 and 20.

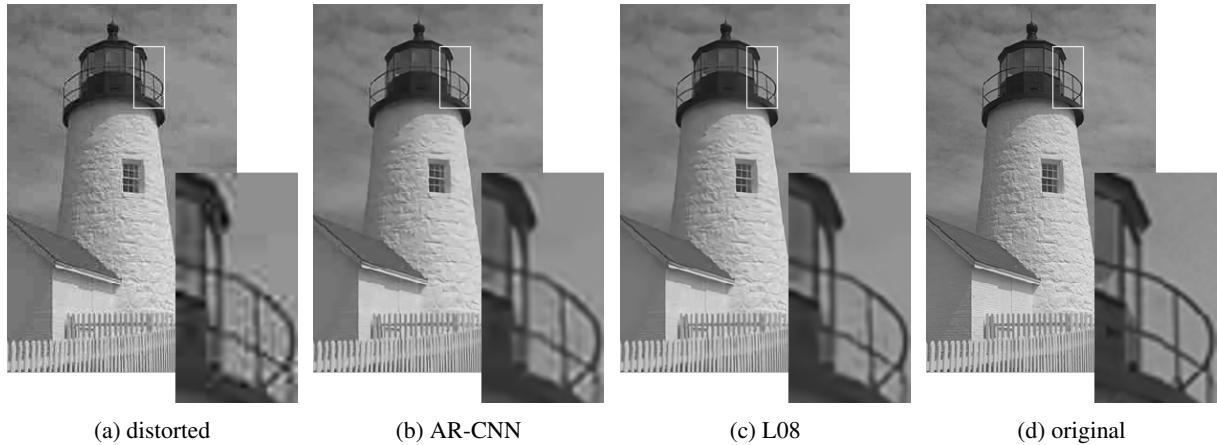


Figure 4: Illustrative comparison of reconstruction quality on lighthouse3 image from LIVE1 dataset, for JPEG quality 20.

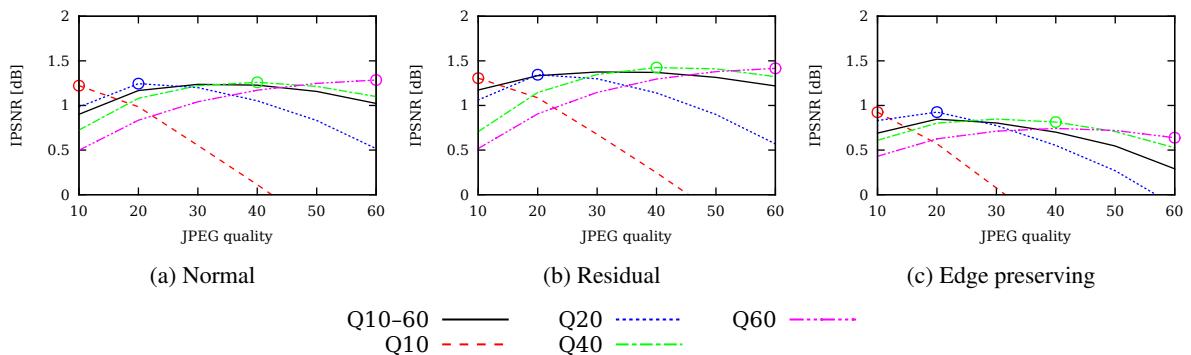


Figure 5: Generalization ability of L4 networks trained with Normal, Residual, and Edge preserving objectives for different JPEG quality levels.

of the distorted image and the reference image. The quantity is often expressed in a logarithmic scale as the peak signal-to-noise ratio (PSNR). Unfortunately, PSNR and MSE are not necessarily correlated well with perceptual quality. The structural similarity index (SSIM) [28] that compares local patterns of pixel intensities should be better correlated with perceptual quality. Since we focus on JPEG artifacts which include blocking artifacts, a block-sensitive metric referred to as the PSNR-B [32] should provide additional insights. PSNR-B modifies the original PSNR by including an additional blocking effect factor (BEF). Some experiments report IPSNR which is a PSNR increase compared to PSNR of the degraded image. IPSNR is more stable across different dataset and it directly reflects the quality improvement.

We compare our results to AR-CNN [5], to the widely regarded deblocking oriented SA-DCT [7, 8], and to a simple postprocessing filter spp included in the FFmpeg framework [19]. While L4 was used in most experiments and it was trained for various compression quality levels, L8 was trained only for quality 20. If not stated otherwise, the residual version of networks was used.

The L4 and L8 networks were trained on mini-batches of 64×64 patches and $4 \times 128 \times 128$ patches respectively. The patches were randomly sampled from training images. The number of training iterations was fixed to 250 K which is significantly less compared to AR-CNN's 10^7 iterations. The learning rate was scaled down by factor of 2 every 50 K iterations. The networks were initialized by the Xavier initialization [10] in the first three layers, and a Gaussian initialization with lower variation was used in the final layer. The learning rate of the last layer was set ten times smaller than for the other layers.

Artifacts reduction quality. The results of artifacts removal on LIVE1 dataset with JPEG quality 10 and 20 are shown in Table 3. The results on the BSDS500 test set are presented in Table 4. L8 outperforms all other methods with significantly higher scores in all three quality measures. L4 which performs worse compared to L8, still surpasses the other methods in most cases even though it is much small and computationally efficient compared to L8. Examples of resulting images are presented in Figure 4.

JPEG quality generalization. We evaluated the ability of the trained networks to generalize to a different compression quality by training L4 on one quality and evaluating on other qualities (L4Q10 trained for quality 10, L4Q20 for quality 20, etc up to L4Q60). To assess the ability of CNNs to handle multiple compression qualities in a single model, we trained a single L4 network on all the qualities together (L4Q10-Q60). The results in

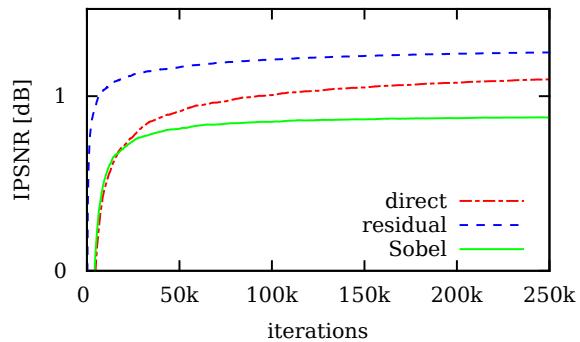


Figure 6: Training development of L4 with different training objectives.

Figure 5 show that L4Q10-Q60 provides stable results across the quality range. However, the quality-specific networks perform better for their respective qualities. The quality-specific networks generalize only to similar qualities. In practice, a single network should easily be able to handle smaller quality ranges (e.g. 10–20 quality points wide) when trained on data from the whole range.

Impact of learning objective. We compare L4 networks trained for direct mapping, residual, and edge preserving loss. Although the architecture and initialization of all the L4 networks were the same, we had to select suitable learning rates (lr) and weight decay coefficients (wd) by performing grid search for each learning objective separately. The chosen values are for direct mapping $lr 0.4$, $wd 5 \times 10^{-7}$, for residual learning $lr 8$, $wd 5 \times 10^{-7}$, and for edge preserving objective $lr 0.05$, $wd 5 \times 10^{-4}$.² The values were chosen on JPEG quality 10 and they were used for all other qualities.

The progress of learning is shown in Figure 6. The residual network converges much faster compared to the direct mapping network. The results on LIVE1 measured by PSNR, PSNR-B and SSIM are in Table 5.

Figure 7 shows 1st layer filters of the networks during different stages of training. All the networks formed reasonable-looking filters. The residual network

Objective	PSNR	PSNR-B	SSIM
Distorted	27.58	24.97	0.769
Direct mapping	28.99	28.66	0.820
Edge preserving	28.69	28.40	0.813
Residual learn.	29.08	28.71	0.824

Table 5: Results of L4 networks with different objectives on LIVE1 dataset with quality 10.

²In our experiments, the loss was normalized by the number of output pixels. This scaling influences the scale of gradients and results in relatively high learning rates and low weight decay coefficients.

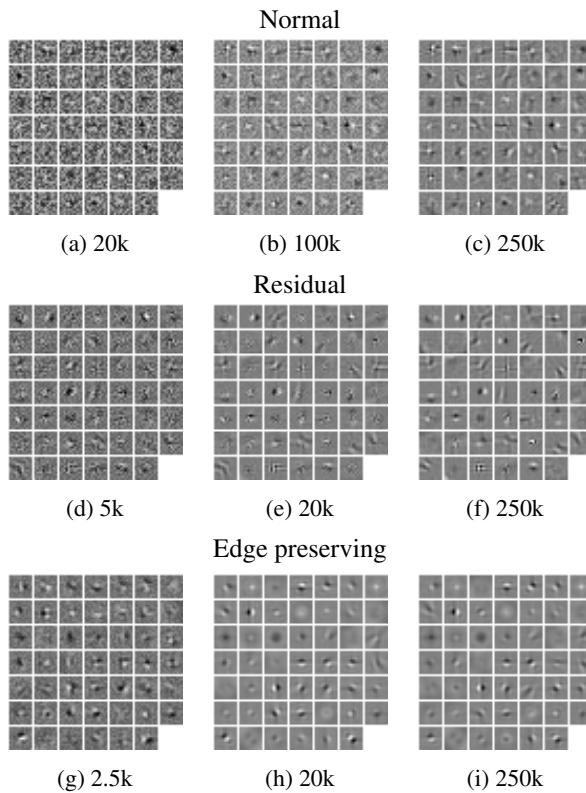
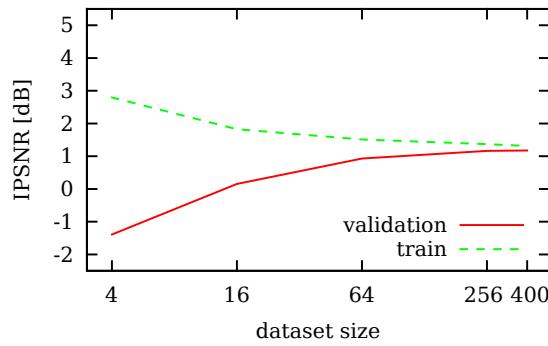


Figure 7: Filters from the first layer of L4 networks with normal/residual/edge preserving objective at different stages of training. Iterations are showed below the images.

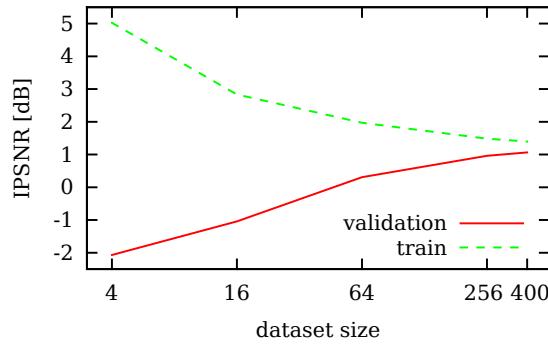
formed more complex higher frequency filters compared to the other networks. The edge preserving network learned a number of low-pass filters which are probably needed to transfer the general image appearance through the network – these filters are missing in the residual network. The filters of the normal direct mapping network remain noisy, which could be due to different weight decay coefficient the low learning rate, or their combination.

The results show that the residual learning is beneficial for JPEG artifact reduction in terms of resulting reconstruction quality and training speed. On the other hand, the edge preserving objective does not improve resulting quality noticeably in the case of L4.

Dataset size. The quality of reconstruction achieved by larger networks may suffer due to inadequate size of a training set. In order to asses how the L4 and L8 behave with respect to training set size, we trained the residual versions of the networks on 4, 16, 64, 256, and 400 images from the training set. The L4 and L8 networks contain approx 70 K and 220 K learnable parameters respectively which suggests that L8 should require larger training set for the same generalization. Figure 8 shows results on the different training sets and corresponding results on the independent test set. Both networks clearly overfit on the smaller datasets. L8 overfits



(a) L4



(b) L8

Figure 8: Generalization for different sized train set.

significantly more and it would require more images to reach proper generalization, while L4 seems to reach perfect generalization already on the relatively small dataset of 400 images.

Speed. Using cuDNN v3 implementation of convolutions on GeForce GTX 780, we were able to process 1 Mpx images in 220 ms with network L4 and in 1052 ms with L8. The L4 and L8 networks require approximately 140 K and 440 K floating point operations per pixel, respectively.³

5 CONCLUSIONS

In this work, we show that it is possible to train large and deep networks for JPEG artifacts removal which outperform previous state-of-the-art results of smaller networks. We combine the residual learning by Kim *et al.* [16], skip architecture [18], and symmetric weight initialization which allowed us to successfully train networks with 8 layers.

We compare networks with three different objectives – direct mapping, residual learning, and edge preserving. The best reconstruction results are provided by the residual learning.

³ The networks, processed images, and implementations are available at <http://www.fit.vutbr.cz/~ihradis/CNN-Deblur/>.

We further investigate the network ability to generalize across different compression JPEG quality levels. Our results show that it is possible to use one network trained for several qualities as an acceptable trade-off.

Finally, we evaluate generalization of the networks with respect to training set size. The results suggest that small networks similar to L4 (20 K parameters) can be safely trained on the BSD dataset. However, the generalization of L8 (100 K parameters) and larger networks is not guaranteed on this small dataset and a larger common dataset should be compiled to allow fair and consistent evaluation in the future.

In a future work, we intend to apply convolutional networks to other compression methods, for example, JPEG 2000, JPEG XR, or WebP. Next, we would like to train convolutional networks to reconstruct images directly from the JPEG coefficients which should provide the networks with significant clues as to which image elements are and which are not artifacts. The receptive field even of the L8 network is still relatively small and we expect that it should be possible to reach higher reconstruction quality by increasing the receptive field or by providing context information by other means.

Acknowledgements

This work has been supported by the ARTEMIS joint undertaking under grant agreement ALMARVI (no. 621439), the Technology Agency of the Czech Republic (TA CR) Competence Centres project V3C – Visual Computing Competence Center (no. TE01020415) and the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (no. LQ1602).

REFERENCES

- [1] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33 (5):898–916, May 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.161.
- [2] Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2 (1):1–127, 2009. doi: 10.1561/2200000006. Also published as a book. Now Publishers, 2009.
- [3] Burger, H. C., Schuler, C. J., and Harmeling, S. Image denoising: Can plain neural networks compete with BM3D? In *Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, June 2012. doi: 10.1109/CVPR.2012.6247952.
- [4] Dong, C., Loy, C. C., He, K., and Tang, X. Learning a deep convolutional network for image super-resolution. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science (LNCS) 8692, pages 184–199. Springer International Publishing, Sept. 2014. ISBN 978-3-319-10593-2. doi: 10.1007/978-3-319-10593-2_13. Part IV.
- [5] Dong, C., Deng, Y., Loy, C. C., and Tang, X. Compression artifacts reduction by a deep convolutional network. In *International Conference on Computer Vision (ICCV)*, pages 576–584, 2015.
- [6] Eigen, D., Krishnan, D., and Fergus, R. Restoring an image taken through a window covered with dirt or rain. In *International Conference on Computer Vision (ICCV)*, pages 633–640, 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.84.
- [7] Foi, A., Katkovnik, V., and Egiazarian, K. Pointwise Shape-Adaptive DCT for high-quality deblocking of compressed color images. In *European Signal Processing Conference (EUSIPCO)*, Sept. 2006.
- [8] Foi, A., Katkovnik, V., and Egiazarian, K. Pointwise Shape-Adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing*, 16 (5):1395–1411, May 2007.
- [9] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [10] Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.
- [11] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. to appear in CVPR 2016.
- [12] Hradis, M., Kotera, J., Zemcik, P., and Sroubek, F. Convolutional neural networks for direct text deblurring. In Xie, X., Jones, M. W., and Tam, G. K. L., editors, *British Machine Vision Conference (BMVC)*, pages 6.1–6.13. BMVA Press, Sept. 2015. ISBN 1-901725-53-7. doi: 10.5244/C.29.6.
- [13] International Telegraph and Telephone Consultative Committee. CCITT recommendation T.81: Terminal equipment and protocols for telematic services : Information technology - digital compression and coding of continuous-tone still im-

- ages - requirements and guidelines, 1993.
- [14] Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Blei, D. and Bach, F., editors, *International Conference on Machine Learning (ICML)*, pages 448–456. JMLR Workshop and Conference Proceedings, 2015.
- [15] Jain, V. and Seung, S. Natural image denoising with convolutional networks. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, pages 769–776. Curran Associates, 2009.
- [16] Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015.
- [17] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, 2012.
- [18] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *Computer Vision and Pattern Recognition (CVPR)*, Nov. 2015.
- [19] Nosratinia, A. Embedded post-processing for enhancement of compressed images. In *Data Compression Conference (DCC)*, pages 62–71, 3 1999. doi: 10.1109/DCC.1999.755655.
- [20] Schuler, C., Hirsch, M., Harmeling, S., and Scholkopf, B. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2481418.
- [21] Schuler, C. J., Burger, H. C., Harmeling, S., and Scholkopf, B. A machine learning approach for non-blind image deconvolution. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [22] Sheikh, H. R., Wang, Z., Cormack, L., and Bovik, A. C. LIVE image quality assessment database release, 2015.
- [23] Sun, J., Cao, W., Xu, Z., and Ponce, J. Learning a convolutional neural network for non-uniform motion blur removal. In *Computer Vision and Pattern Recognition (CVPR)*, pages 769–777, June 2015. doi: 10.1109/CVPR.2015.7298677.
- [24] Svoboda, P., Hradis, M., Marsik, L., and Zemcik, P. CNN for license plate motion deblurring. *CoRR*, abs/1602.07873, 2016.
- [25] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. DeepFace: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, June 2014. doi: 10.1109/CVPR.2014.220.
- [26] Toshev, A. and Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, June 2014. doi: 10.1109/CVPR.2014.214.
- [27] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [28] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 4 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.
- [29] Wong, T. S., Bouman, C. A., Pollak, I., and Fan, Z. A document image model and estimation algorithm for optimized JPEG decompression. *IEEE Transactions on Image Processing*, 18(11):2518–2535, Nov. 2009. ISSN 1057-7149. doi: 10.1109/TIP.2009.2028252.
- [30] Xu, L., Ren, J. S. J., Liu, C., and Jia, J. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014.
- [31] Yang, S., Kittitornkun, S., Hu, Y.-H., Nguyen, T. Q., and Tull, D. L. Blocking artifact free inverse discrete cosine transform. In *International Conference on Image Processing*, volume 3, pages 869–872, 2000. doi: 10.1109/ICIP.2000.899594.
- [32] Yim, C. and Bovik, A. C. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, Jan. 2011. ISSN 1057-7149. doi: 10.1109/TIP.2010.2061859.