

# Smooth Map Deformation Using Integral Images

Molchanov Vladimir

Westfälische Wilhelms-Universität  
Münster  
Schlossplatz 2  
48149 Münster, Germany  
molchano@uni-muenster.de

Lars Linsen

Westfälische Wilhelms-Universität  
Münster  
Schlossplatz 2  
48149 Münster, Germany  
linsen@uni-muenster.de

## ABSTRACT

Deformation of geographical maps is a powerful tool used in geospatial data exploration systems. Virtual lenses help the user to retrieve local detail information without losing the global overview, while contiguous cartograms exploit map deformation for representing statistical data as areas of regions. In most applications, the map deformations should be smooth and computationally efficient. We propose a novel map deformation approach based on integral images computed for 2D density distributions. Depending on the prescribed density function, the resulting deformed map represents an approximation to a respective contiguous cartogram or may serve as a user-steered virtual lens. Our technique is suitable for use in highly interactive geospatial data exploration systems due to its algorithmic and computational efficiency. We test the proposed method on several artificial and real-world datasets and discuss directions for future work.

## Keywords

Contiguous cartogram, geospatial visualization, smooth map deformation, RadViz, generalized barycentric coordinates projection, integral image

## 1 INTRODUCTION

Visual data representation becomes increasingly challenging with growing complexity of modern datasets. The data complexity can be enhanced by including more detailed information as well as due to inhomogeneity of data components. However, effective visualizations should be simple, compact, and intuitive to remain practical. Therefore, a thorough static representation of modern data on standard computer displays is often impossible.

Advanced visualization systems solve the problem of displaying complex data in real-time by allowing the user to interact with the graphical data representations. Then, dynamical changes of views steered by the user disclose certain parts or aspects of the underlying data on demand. Hiding irrelevant and highlighting important information as well as a proper choice of the presented level of detail may significantly reduce visual clutter and information overload. Thus, interactive systems make it possible for the user to navigate through,

explore, transform, and analyze even very large and complex datasets.

Geo-spatial data visualization is an example of an application field with high demands on handling data at different spatial scales. General interfaces used in the interactive systems for working with multiple levels of details include *zooming*, *overview+detail*, and *focus+context* [CKB09]. Zooming may easily lead to either suppressing local information or to losing spatial awareness and the global overview [BDW<sup>+</sup>08]. Overview+detail approaches integrate different degrees of data granularity in one view at the cost of introducing spatial discontinuities between the scales. Finally, the focus+context metaphor is usually realized in the form of a lens, which deforms spatial domain by expanding a region of interest while compressing areas outside the region of interest. In this case, finer details of the region of interest become locally visible, while the global overview remains preserved by avoiding discontinuities and displaying the entire context.

*Cartograms* are a graphical representation of statistical data over geographical regions. E.g., scalar values can be visually encoded as areas of the regions, to which these values are assigned. Contiguous cartograms are computed by continuous deformation of the original map while conserving adjacencies (or neighborhoods) of the map regions. Thus, such cartograms are topologically accurate and therefore more intuitive for the user. Cartograms are usually not aimed at representing

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.

additional data at different spatial scales as in the focus+context approach. However, formally, contiguous cartograms can be considered as a simultaneous application of lenses to all map regions according to their scalar values that determine the regions' magnification factors.

The problem of contiguous cartogram construction can be formulated as follows: For a given non-negative scalar density function representing geospatial statistical data, find a smooth transformation of the map that equalizes the density distribution, i.e., integral density in any region becomes proportional to the region area. In our work, we propose a novel globally smooth map deformation method. The deformation parameters are computed based on integral images computed for a given density distribution. We allow the user to interactively change the discrete density function, thus, steering the local deformation of the map. The proposed method is algorithmically and computationally efficient. Thus, the method can be used in interactive geospatial data exploration systems.

Our contributions include:

- We demonstrate how integral images can be used for pseudo-cartogram construction and discover algorithmic equivalence of existing methods for pseudo-cartogram computation and the RadViz (Generalized Barycentric Coordinates) visualization algorithm.
- We improve the existing map deformation algorithm by considering tilted integral images and sliding anchor points, which results in novel smooth non-linear transformations.
- We modify the proposed map-deformation algorithm to function as a magnification lens of user-defined geometry in interactive geospatial data exploration applications.
- We perform numerical tests examining properties and efficiency of the proposed methods.
- We discuss possible application scenarios and future work directions.

## 2 RELATED WORK

*Virtual lenses* in visualization may act not only as pure magnification tools, but also to query detail information and to filter or modify data representations in a region of interest. A taxonomy of virtual interactive lenses according to the types of data and user tasks was recently presented in an extensive survey by Tominski et al. [TGK<sup>+</sup>17]. *Fisheye views* proposed by Furnas [Fur86] are based on local degree of interest and provide a smooth integration of two levels of details. While classical lenses have regular

shape (circle or rectangular), *JellyLenses* proposed by Pindat et al. [PPCP12] dynamically adapt their geometry to the content in the focus. Haurert and Sering [HS11] modified fish-eye projection applied to road networks by reducing the distortion in dense network regions. Wu et al. [WLLM13] developed a technique for content-aware resizing task. The method is based on an iterative optimization procedure to compute a deformation map.

While virtual lenses are commonly an example of local distortion-based visualizations, the construction of contiguous cartograms usually requires a global map deformation. The main idea of *rubber-sheet algorithms* is a deformation of a mesh superimposed to the map. Most grid-based methods are iterative and require a number of parameters to be correctly set up by the user. The *Carto3F* method by Sun [Sun13] is a rubber-sheet algorithm that guarantees topological integrity and computationally outperforms its predecessors when implemented on the GPU. The diffusion-based density-equalizing map by Gastner and Newman [GN04] is topologically correct as well but potentially impairs shape preservation of the map regions [Sun13].

*Pseudo-cartograms* proposed by Tobler [Tob86] provide approximations to contiguous cartograms by distorting a rectangular mesh along two coordinate axes  $x$  and  $y$  independently. Pseudo-cartograms may serve as an initial state for applying other iterative cartogram-construction algorithms. Keim et al. [KPS<sup>+</sup>03] proposed a real-time *HistoScale* method for computing pseudo-cartograms. The algorithm first computes 1D histograms of geospatial data in two Euclidian dimensions and then transforms the map by a proper rescaling of map regions in each histogram bin to fulfill the cartogram condition. *RadialScale* and *AngularScale* density-equalizing approaches developed by Bak et al. [BSS<sup>+</sup>09] distort a map in radial and angular directions and, thus, can be seen as an analogy of the pseudo-cartograms by Tobler in polar coordinates.

The cartogram quality is characterized by its statistical accuracy, the topological accuracy of the geometry as well as the polygonal complexity of the resulting layout. Numerical measures for cartogram quality were summarized by Alam et al. [AKV15]. A recent overview of the state of the art in cartograms was presented by Nusrat and Kobourov [NK16]. Aesthetic appearance is an important but hardly numerically measurable aspect of cartograms. Therefore, despite of existence of many fully automatic algorithms, there is a need in graphical interactive systems providing manual user control over cartogram creation. An example of such a system was recently presented by Kronenfeld [Kro18].

User interactions with graphical views are very practical in multidimensional data exploration systems. Recently, Sacha et al. [SZS<sup>+</sup>17] revealed seven common interaction scenarios of the user interactions with visual encodings of dimensionality reduction algorithm outcomes. *RadViz* proposed by Hoffman [HGM<sup>+</sup>97] is a well-known projection method which maps multidimensional samples in a 2D domain according to their values and the anchor points placed on a circle. The multidimensional coefficients act as attracting forces of the anchor nodes representing dimensions for the projected samples. Cheng et al. [CXM17] identified three types of errors in classical *RadViz* layout and minimized the errors by properly placing the anchor points. The authors also demonstrated equivalence of *RadViz* to the *Generalized Barycentric Coordinates* (GBC) interpolation method by Meyer et al. [MBLD02]. Nam and Mueller [NM13] used GBC for projecting multidimensional data describing geographical locations. The *RadViz++* method proposed by Pagliosa and Telea [PT19] addresses the scalability issue of the *RadViz* metaphor.

*Summed-area tables* or *Integral Images* (InIm) were originally introduced by Crow [Cro84] and applied to object detection in image analysis by Viola and Jones [VJ02]. Computation of InIm at arbitrary angles was studied by Chin et al. [CGT08]. Ehsan et al. [ECuRM15] addressed the problem of efficient parallel computation of InIm. Nowadays, many libraries (e.g., OpenCV [Bra00] used in our work) provide out-of-the-box functions for InIm computation.

We demonstrate how InIm can be used for pseudo-cartogram construction. By doing so, we discover algorithmic equivalence of existing methods for pseudo-cartogram computation and the *RadViz*/GBC algorithms. We improve these pseudo-cartogram computation algorithms by using tilted InIm. Finally, we show how to apply our method for virtual lenses and contiguous cartogram applications.

### 3 BACKGROUND AND NOTATIONS

Let an object  $\mathcal{O}$ , e.g., a continent or a country, be represented on a geographical map as a union of non-overlapping regions  $\mathcal{R}_i$ ,  $i = 1, \dots, n$ , e.g. states or provinces. Let  $a_i > 0$  be the area fractions of  $\mathcal{R}_i$  in  $\mathcal{O}$ . There may also be a special region  $\mathcal{B}$  representing the background, i.e., the part of the map that does not belong to  $\mathcal{O}$ . Let  $\{v_i\}$  be statistical data assigned to all regions  $\mathcal{R}_i$ . For cartograms, typical examples of  $v_i$  are population or voting results in  $\mathcal{R}_i$ . Then, density distribution  $\rho(x, y)$  is a scalar function, which satisfies the relation

$$\int_{\mathcal{R}_i} \rho(x, y) dx dy = v_i, \quad i = 1, \dots, n. \quad (1)$$

Note that the density functions used in our work are defined up to a global positive constant factor, which does not influence the results. A particular solution of Equation (1) is a piecewise constant  $\rho(x, y)$  that in each region  $\mathcal{R}_i$  is equal to a value  $v_i/a_i$  for a constant  $a_i$ . The density values in background  $\mathcal{B}$  can be arbitrary as long as no data are assigned to the background region. We discuss this aspect in Section 5.3.

In our work, we propose an efficient and simple algorithm for a smooth deformation of a geographical map given as a (one- or three-channel) 2D texture based on a scalar-valued discrete 2D density function. For simplicity of exposition, we assume that both map and density textures have the same resolution. All computations are performed in texture coordinates ranging between 0 and 1 in horizontal and vertical directions. The resolution of the textures is arbitrary and can be chosen by the user. Note that we use the same notations for both continuous and discrete functions, e.g., density  $\rho(x, y)$ , to simplify the exposition, as to not lead to confusion.

Our algorithm extensively uses InIm computed for the density texture and is defined as follows. The value of InIm at any location  $(x, y)$  is the sum of values in all the pixels that are both to the left and above location  $(x, y)$  [ECuRM15], which we denote by  $\alpha(x, y)$ . In fact, we need four integral values per pixel  $\alpha(x, y)$ ,  $\beta(x, y)$ ,  $\gamma(x, y)$ , and  $\delta(x, y)$  representing the sums of pixel values in all four rectangular domains as shown in Figure 1a. Whereas  $\alpha(x, y)$  is the conventional InIm and can be directly computed by a standard function call in many image processing libraries, the other three values can be easily expressed via  $\alpha(x, y)$  by

$$\beta(x, y) = \alpha(x, 1) - \alpha(x, y), \quad (2)$$

$$\gamma(x, y) = \alpha(1, 1) - \alpha(x, 1) - \alpha(1, y) + \alpha(x, y), \quad (3)$$

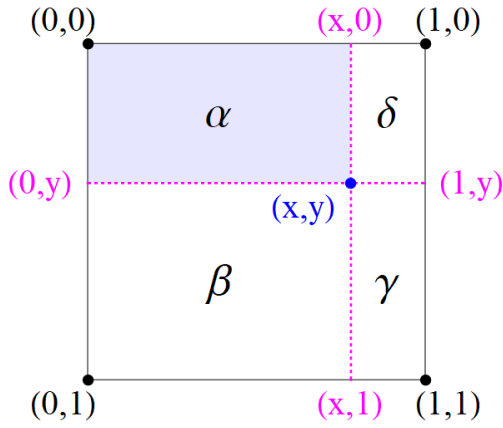
$$\delta(x, y) = \alpha(1, y) - \alpha(x, y). \quad (4)$$

Without loss of generality, we can assume that

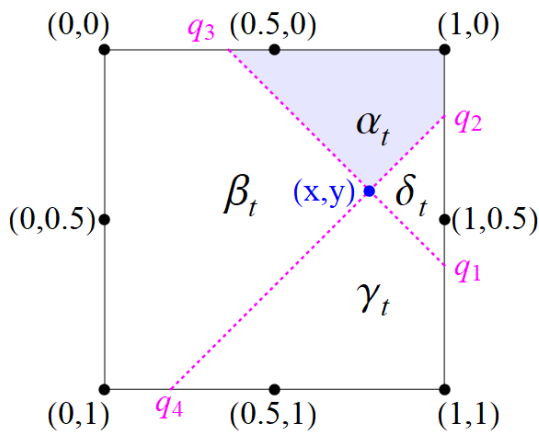
$$\alpha(x, y) + \beta(x, y) + \gamma(x, y) + \delta(x, y) \equiv \alpha(1, 1) = 1, \quad (5)$$

i.e., all integral coefficients sum up to unity, which can be reached by a proper scaling of density  $\rho(x, y)$ . In addition to the listed InIm values, we propose to use four integral values  $\alpha_t(x, y)$ ,  $\beta_t(x, y)$ ,  $\gamma_t(x, y)$ , and  $\delta_t(x, y)$  of a tilted InIm, see Figure 1b. Again,  $\alpha_t(x, y)$  can be computed using existing functionality (we use function `integral` of OpenCV [Bra00] in our implementation). For computing the other three values, one may call the same function for a tilted InIm of the density texture rotated by  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ , correspondingly. Obviously, tilted integral coefficients also sum up to unity, i.e., an analog of Equation (5) holds for  $\alpha_t(x, y)$ ,  $\beta_t(x, y)$ ,  $\gamma_t(x, y)$ , and  $\delta_t(x, y)$ :

$$\alpha_t(x, y) + \beta_t(x, y) + \gamma_t(x, y) + \delta_t(x, y) \equiv 1. \quad (6)$$



(a) integral coefficients at location  $(x,y)$



(b) tilted integral coefficients at location  $(x,y)$

Figure 1: InIm computations: (a) The four coefficients computed at location  $(x,y)$  stand for integrals of a density function over respective rectangular regions. (b) The four additional coefficients can be computed for the same location as integrals over tilted regions.

#### 4 EQUIVALENCE OF PSEUDO-CARTOGRAM CONSTRUCTION AND GBC PROJECTION

Contiguous cartogram construction algorithms aim at equalizing a given density distribution by smoothly deforming the geographical map. The deformation has to be smooth in order to preserve neighborhood relation of the map region and avoid region overlapping, thus, supporting readability of the result. Many cartogram methods iteratively deform a spatial mesh and then apply a texture mapping technique for computing cartograms.

The pseudo-cartogram algorithm by Tobler [Tob86] computes an exact cartogram for separable density functions, i.e., when  $\rho(x,y) = f(x) \cdot g(y)$  for some

functions  $f$  and  $g$ . In this particular case, Tobler proposed a transformation  $(x,y) \mapsto (x',y')$  given by

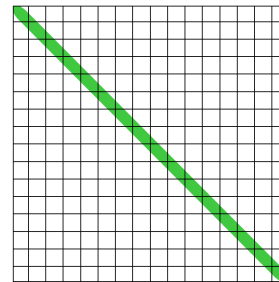
$$(x', y') = \left( \int_0^x f(t) dt, \int_0^y g(t) dt \right), \quad (7)$$

which constructs a perfect cartogram. For a non-separable density function, an approximate solution can be obtained by [Tob86]

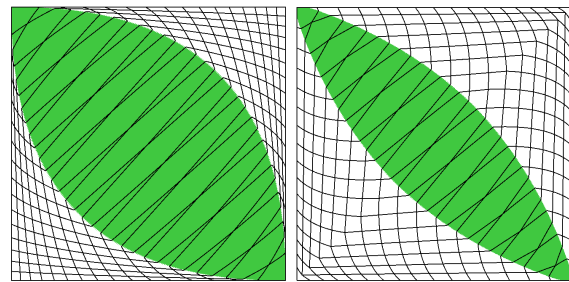
$$\begin{aligned} (x', y') &= \left( \int_0^1 \rho(x,y) dy, \int_0^1 \rho(x,y) dx \right) \\ &= (\alpha + \beta, \alpha + \delta), \end{aligned} \quad (8)$$

when using the notations from Figure 1a.

Limitations of the method by Tobler come from the fact that spatial dimensions are transformed separately. Then, in particular, lines of a rectangular mesh remain straight lines after transformation. The accuracy of the pseudo-cartogram highly depends on how close the given density function can be approximated by a separable function. E.g., if  $\rho(x,y)$  is constant along a diagonal of the texture and vanishes elsewhere, the pseudo-cartogram algorithm results in a trivial mapping, i.e., does not deform the map at all, see Figure 2a.



(a) Tobler's pseudo-cartogram



(b) 4 sliding anchors

(c) 8 sliding anchors

Figure 2: Synthetic dataset illustrating limitations of Tobler's approach. The density is set to 1 in the green diagonal stripe and to 0 elsewhere. While Tobler's pseudo-cartogram (a) does not deform the artificial map, our map deformations using 4 (b) or 8 sliding anchors (c) perform significantly better in enlarging the populated region and contracting the empty background area.

Before proposing our cartogram method, we first demonstrate that the mapping in Equation (8) is equivalent to a RadViz or GBC projection of the texture pixels. GBC maps a multidimensional sample  $(a_1, \dots, a_k)$  to a 2D domain according to formula [CXM17]

$$\mathbf{z} = \sum_{j=1}^k \frac{a_j}{\sum a_i} \cdot \mathbf{v}_j, \quad (9)$$

where  $\{\mathbf{v}_j\}$  are anchor points in the projection domain each representing a data dimension. The RadViz anchor points are usually placed in a circular layout. GBC generalizes the projection formula for the arbitrary configuration of the anchor-points distribution. Now, let a pixel in the undistorted map be characterized by InIm coefficients  $(\alpha, \beta, \gamma, \delta)$  and let the anchor points be  $\mathbf{v}_1 = (1, 1)$ ,  $\mathbf{v}_2 = (1, 0)$ ,  $\mathbf{v}_3 = (0, 1)$ , and  $\mathbf{v}_4 = (0, 0)$ . Then, according to Equation (9) and when taking into account that the InIm coefficients sum up to unity, see Equation (5), we obtain

$$\begin{aligned} \mathbf{z} &= \alpha \cdot (1, 1) + \beta \cdot (1, 0) + \gamma \cdot (0, 1) + \delta \cdot (0, 0) \\ &= (\alpha + \beta, \alpha + \delta), \end{aligned} \quad (10)$$

which is equal to the mapping in Equation (8) for  $\mathbf{z} \equiv (x', y')$ . Usually, the sample coefficients in GBC are interpreted as attracting force magnitudes acting between the sample and the respective anchor points. In our case, there are four forces, whose magnitudes are the InIm coefficients and which pull the pixel towards the respective texture corners. The resulting force induces the pixel position in the deformed map.

## 5 PROPOSED METHOD

### 5.1 Deformation Formulae

In order to alleviate limitations of the pseudo-cartogram method proposed by Tobler, we extend the deformation map in Equation (10) by taking into account the tilted InIm coefficients  $(\alpha_t, \beta_t, \gamma_t, \delta_t)$ . We test two options for assigning anchor nodes to the coefficients. First, in analogy to the procedure above, the nodes can be fixed and set to the midpoints of the boundary edges of the texture, resulting in a mapping

$$\begin{aligned} (x', y') &= \alpha_t(x, y) \cdot (0.5, 1) + \beta_t(x, y) \cdot (1, 0.5) + \\ &\quad \gamma_t(x, y) \cdot (0.5, 0) + \delta_t(x, y) \cdot (0, 0.5) \\ &= \left( \beta_t + \frac{\alpha_t + \gamma_t}{2}, \alpha_t + \frac{\beta_t + \delta_t}{2} \right). \end{aligned} \quad (11)$$

However, the resulting map is clearly suboptimal, see Figure 3c, since the mapping in Equation (11) does not cover the whole texture space, i.e., regions close to the texture corners remain empty for any possible integral coefficient. Thus, the transformation in Equation (11) is impractical and we list it only for the sake of complete exposition.

Instead, we propose to use anchor points which depend on the coordinates of the mapped pixel, namely

$$\begin{aligned} (x', y') &= \alpha_t(x, y) \cdot (x, 1) + \beta_t(x, y) \cdot (1, y) + \\ &\quad \gamma_t(x, y) \cdot (x, 0) + \delta_t(x, y) \cdot (0, y) \\ &= (\beta_t + x \cdot (\alpha_t + \gamma_t), \alpha_t + y \cdot (\beta_t + \delta_t)). \end{aligned} \quad (12)$$

The new adaptive anchor points are the intersection points of the dashed lines in Figure 1a with the sides of the domain boundary. We refer to this transformation as a *four sliding anchor map*. The mapping in Equation (12) is much more flexible when compared to Tobler's pseudo-cartogram. Note that the straight lines of the mesh become curved after the transformation, see Figure 2b.

Tobler's pseudo-cartogram is by construction accurate for separable density distributions but it may perform poorly otherwise. Our transformation in Equation (12) instead is not exact for separable  $\rho(x, y)$  but is more flexible. We further extend our four sliding anchor map by combining it with a flexible version of Tobler's pseudo-cartogram. We propose to modify Tobler's method by using sliding anchors instead of the four fixed texture corners used in Equation (9) and average the resulting mapping with our transformation in Equation (12), such that each pixel is mapped using *eight sliding anchors*. The proposed deformation formula has then the form:

$$\begin{aligned} (x', y') &= \frac{1}{2} \left( \alpha \cdot q_1(x, y) + \beta \cdot q_2(x, y) + \right. \\ &\quad \left. \gamma \cdot q_3(x, y) + \delta \cdot q_4(x, y) + \right. \\ &\quad \left. (\beta_t + x \cdot (\alpha_t + \gamma_t), \alpha_t + y \cdot (\beta_t + \delta_t)) \right), \end{aligned} \quad (13)$$

where the new adaptive anchor points  $q_i(x, y)$ ,  $i = 1, \dots, 4$ , are the intersection points of the dashed lines in Figure 1b with the sides of the domain boundary. The anchors can be computed according to these formulae:

$$\begin{aligned} \text{if } y < x : & \quad q_1(x, y) = (1, 1 + y - x), \\ & \quad q_3(x, y) = (x - y, 0); \\ \text{if } y \geq x : & \quad q_1(x, y) = (1 - y + x, 1), \\ & \quad q_3(x, y) = (0, y - x); \\ \text{if } x + y < 1 : & \quad q_2(x, y) = (x + y, 0), \\ & \quad q_4(x, y) = (0, x + y); \\ \text{if } x + y \geq 1 : & \quad q_2(x, y) = (1, x + y - 1), \\ & \quad q_4(x, y) = (x + y - 1, 1). \end{aligned} \quad (14)$$

The proposed deformation can be applied to every pixel of the undistorted map. However, in our tests, we map only nodes of a regular rectangular mesh according to Equation (13) and then use texture mapping technique for interpolating the map within the grid cells. For

better understanding of the deformation outcome, the sparse grid lines after transformation are shown in the results in Figure 2c.

## 5.2 Algorithm

The complete proposed deformation algorithm includes following five steps:

1. Prepare density texture  $\rho(x, y)$ , i.e., based on geostatistical data.
2. Compute textures containing integral coefficients.
3. Define a mesh over the geographical map, which will be deformed.
4. Apply deformation formula (13) for computing the new positions of the mesh's vertices.
5. Interpolate the map within each deformed mesh cell. This can be performed using a texture mapping technique.

Note that due to Equation (2), only one texture  $\alpha(x, y)$  is sufficient for computing  $\beta(x, y)$ ,  $\gamma(x, y)$ , and  $\delta(x, y)$ . Moreover, due to Equation (6), only three tilted InIms are needed. Therefore, in total, only four textures with InIms are required for computing the proposed deformations (12) and (13).

## 5.3 User Interactions with Density Texture

The density value for background region  $\mathcal{B}$  is typically not defined. However, its choice may significantly affect the resulting deformation map. If the assigned density value is high, object  $\mathcal{O}$  (the non-background regions) will generally be contracted by the transformation. If the background density is close to the average density of object  $\mathcal{O}$ , the size of object  $\mathcal{O}$  will tend to remain close to its original size. Also, its shape may remain close, but the the internal borders of the regions within  $\mathcal{O}$  will change. Finally, vanishing density in  $\mathcal{B}$  will result in mappings minimizing the background size while maximizing the total area of object  $\mathcal{O}$ . Thus, such a mapping can be interpreted as a global zooming transformation, which aims at efficient usage of available empty space in the map.

The map deformation algorithms proposed in the previous section can be executed at interactive rates, thus, they can be used in interactive applications. One possible application scenario of the methods is a user-steered magnification of an area of interest similar to a virtual lens. Since the resulting mapping is global and smooth, there is no need for a costly computation of the transition region between the magnified region and the rest of the map.

## 6 RESULTS

For our experiments we used geospatial data in the form of shape files provided at [www.gadm.org](http://www.gadm.org). All maps are given in a longitude-latitude coordinate system.

In our first numerical experiment, we compare the different proposed deformation formulae for exploring their character, best application cases, and limitations. Figure 3a shows the original map of the states of Italy and the resulting deformed maps. In all cases, the density distribution is piecewise constant being equal to 1.0 for all states and equal to 0.0 for background. Thus, ideally the cartogram would deform the map such that the background area vanishes. With  $r$  being the ratio of the background to country area, Tobler's pseudo-cartogram reduces the value of  $r$  from originally 3.18 to 1.44. The map for the tilted InIm approach with fixed anchors in Figure 3c was already discussed above. It does not make use of the whole texture space, since the four fixed anchors are placed in the centers of the texture sides. The deformation with four sliding anchors computed according to Equation (12) reduces the background area the most efficiently resulting in  $r = 1.06$ . The combination of this transformation with the flexible Tobler map (eight sliding anchors) gives  $r = 1.22$ , see Figures 3d and 3e. The fact that the four sliding anchors have a lower value  $r$  than the eight sliding anchors can be understood, if one considers that the shape of Italy is dominantly diagonal in the longitude-latitude coordinates. The non-linear shape of the deformed mesh lines is clearly visible in the proposed maps.

In our second test, we examine how the density value assigned to the background region affects the resulting deformation. Figure 4a shows undistorted map of the states of the Czech Republic. Constant density values are assigned to each region according to population data in 2011. The average density in the foreground regions is equal to unity. We test the proposed algorithms with four and eight sliding anchor nodes with background density  $\rho_0 = 1.0, 0.5, \text{ and } 0.0$ . As  $\rho_0$  decreases, the foreground regions occupy larger parts of the map, i.e., the background area diminishes. Note that in contrast to diagonally spread map of Italy in our first test, the Czech Republic map is horizontally elongated. Consequently, the method with eight sliding anchor nodes performs better in contracting the empty region, see Figures 4g and 4h. In Tobler's pseudo-cartogram in Figure 4b the most dense region of Prague (pink area in the north-west) leads to expansion the Praque region, but also of other regions that share the same longitude or latitude coordinates with Praque irrespective of their real population density. The expansion of the Praque region in our proposed methods instead has a more localized character as can be seen from the curved mesh lines, e.g., in Figure 4g. Therefore, shapes of the re-

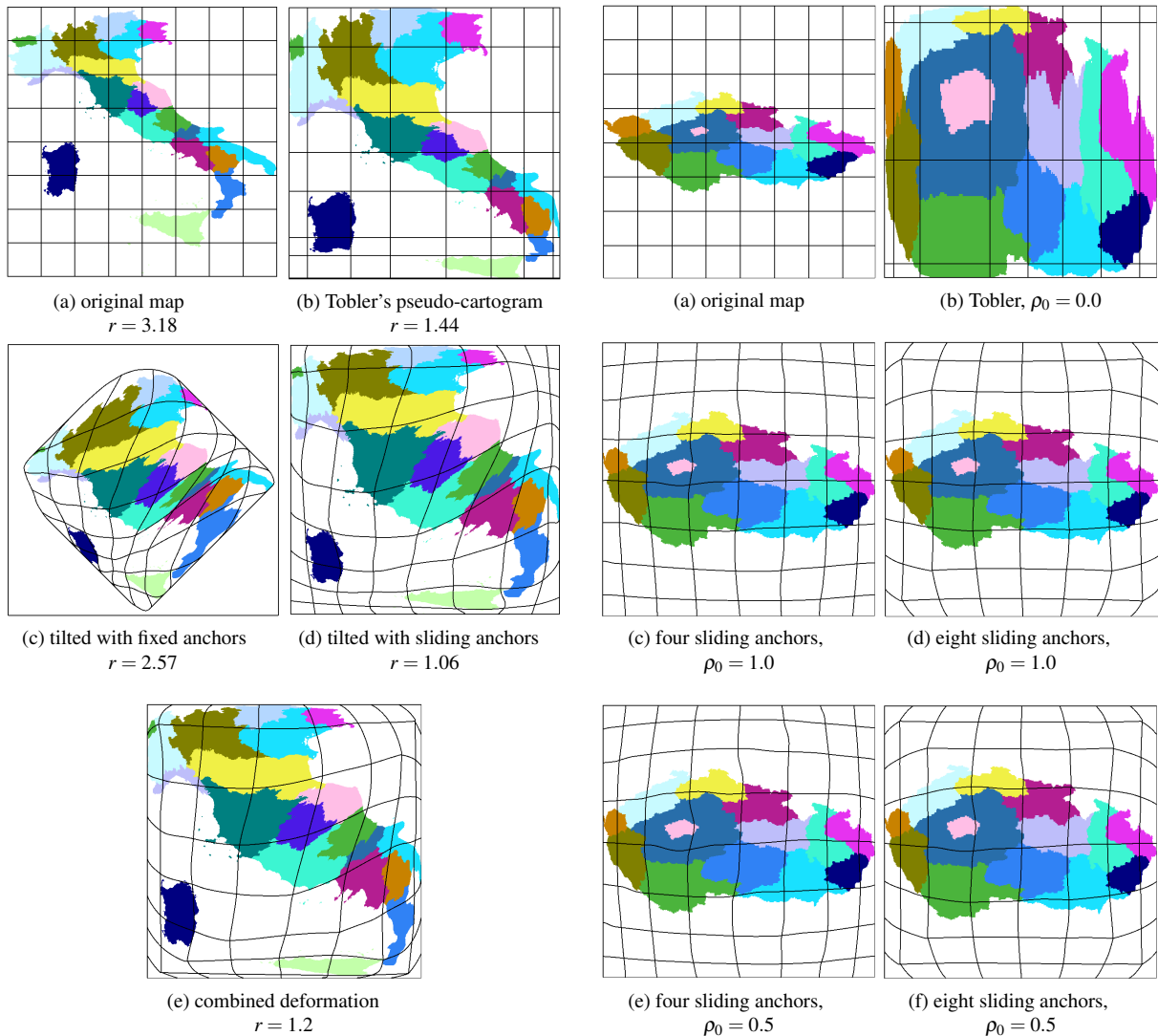


Figure 3: Comparison of different transformations of a map of Italy. (a) Original map with an overlaid regular mesh. Constant density values of 0.0 and 1.0 are assigned to the land and background regions, correspondingly. (b) Tobler’s pseudo-cartogram. (c) Deformations with fixed anchors according to Equation (11). (d) Deformations with 4 sliding anchors according to Equation (12). (e) Deformations with 8 sliding anchors according to Equation (13). Ratios of the background to country area  $r$  are presented for all maps.

regions in our maps are better preserved when compared to the Tobler map.

Next, we measured computational times required for the main steps of the proposed algorithms. We found that it takes, on average, 53ms for computing all eight InIm textures of size  $1024^2$  and 6.5ms for mapping  $128^2$  nodes of a regular mesh. The execution times were measured by averaging 1,000 runs of the respective steps. Integral images are computed using the OpenCV 3.3.1 library. The user interface is developed

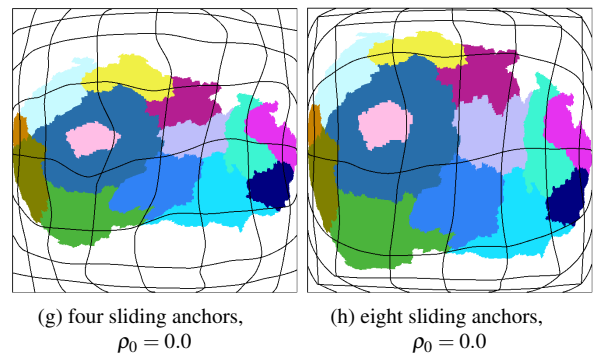


Figure 4: Dependence of the foreground region’s shape preservation in the deformed map on the background density value  $\rho_0$ .

in Qt 5.11. All relevant computations are performed on a PC with an i7-7700K 4.20GHz CPU. Texture mapping is executed using OpenGL 4.6.

Finally, we demonstrate how the proposed methods can be used in interactive exploration systems as a virtual lens. The user selections of regions of interest on a to-

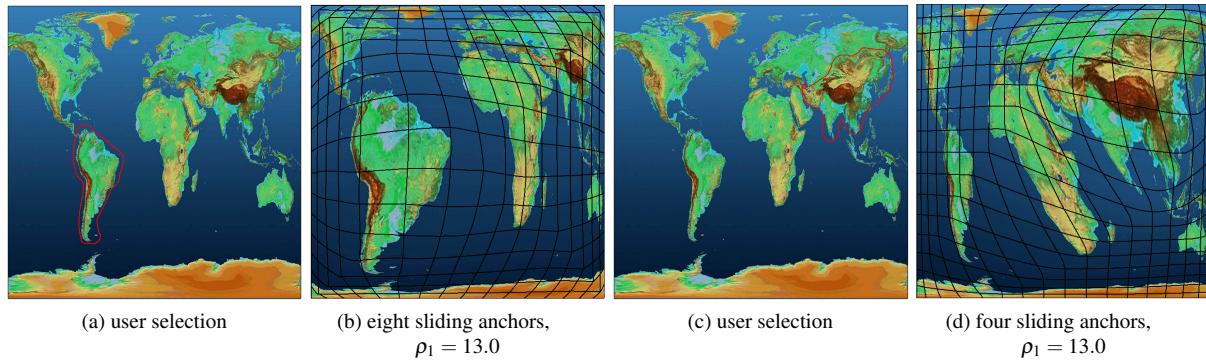


Figure 5: Focus+context zooming of a user-selected region: (a,c) User selections on the original map. (b,d) Transformed maps with density value  $\rho_1 = 13.0$  assigned to the selected regions, while the area outside the selected region keeps density value 1.0.

pographical world map are shown Figure 5a and 5c, respectively. The selections are performed interactively using a lasso tool, i.e., the region does not need to be predefined. Then, the user interactively increases the constant density value assigned to the selected area from an initial value of  $\rho_1 = 1.0$  to a new value of 13.0, while the background density (everything outside the selected region) remains equal to 1.0. The results of the transformation for the two selections are shown in Figures 5b and 5d. We observe that the selected regions appear as being zoomed in, while the rest of the original map is smoothly deformed.

In the accompanying video, we provide further test cases, demonstrate a prototype of our geospatial data exploration system, and explain possible user interactions with the tool.

## 7 CONCLUSION AND FUTURE WORK

In our work we showed how Tobler's pseudo-cartograms can be computed using InIm, i.e., an InfoVis task is solved by a projection method (RadViz or its generalization GBC) using an image processing tool. We illustrated the limitation of the method by constructing a simple synthetic dataset, for which Tobler's method is ineffective. We furthermore showed the equivalence of the pseudo-cartogram construction and the GBC projection method. Using tilted InIm, we proposed new deformation algorithms with four and eight sliding anchor nodes, correspondingly, which result in non-linear non-separable globally smooth mappings. Therefore, adjacency of the subregions of the original map is perfectly preserved in the transformed one, which is one of a desired properties of contiguous cartograms. We showed how these techniques can be used as virtual lenses for mapping regions of arbitrary shapes. Algorithmic simplicity and computational efficiency of the proposed algorithms allow for using them in highly interactive applications.

Many existing map deformation algorithms are iterative. Since the proposed mappings are explicit, disadvantages and limitations of iterative methods (i.e., need in setup of non-intuitive parameters, preventing overlapping of subregions, and computational burden) are avoided.

Future directions of our research include:

- Using InIm tilted by arbitrary angles (not only multiples of  $45^\circ$ ), we may arbitrarily increase the number of anchor nodes. The properties of the limiting deformation map as the number of anchor increases are of theoretical and practical interest.
- Since the positions of the anchor points are arbitrary, it may be possible to construct deformations defined for non-rectangular domains.
- All steps of the proposed algorithm allow for parallel computing. Thus, despite the fact that our computation times are already low and allow for interactive applications, computation times could further be significantly reduced by implementing our methods on the GPU.
- Application of the proposed algorithm to regularization of arbitrary distributed point clouds is technically possible. Certain adaptations for efficient computation of the density function are necessary.

## 8 ACKNOWLEDGMENTS

This work was supported by DFG grant MO 3050/2-1.

## 9 REFERENCES

- [AKV15] Md. Jawaherul Alam, Stephen G. Kobourov, and Sankar Veeramoni. Quantitative measures for cartogram generation techniques. *Computer Graphics Forum*, 34(3):351–360, 2015.



- [BDW<sup>+</sup>08] Thomas Butkiewicz, Wenwen Dou, Zachary Wartell, William Ribarsky, and Remco Chang. Multi-focused geospatial analysis using probes. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1165–1172, 2008.
- [Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [BSS<sup>+</sup>09] Peter Bak, Matthias Schaefer, Andreas Stoffel, Daniel A. Keim, and Itzhak Omer. Density equalizing distortion of large geographic point sets. *Cartography and Geographic Information Science*, 36(3):237–250, 2009.
- [CGT08] Tat-Jun Chin, Hanlin Goh, and Ngan-Meng Tan. Exact integral images at generic angles for 2D barcode detection. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, December 2008.
- [CKB09] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1), January 2009.
- [Cro84] Franklin C. Crow. Summed-area tables for texture mapping. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, pages 207–212, New York, NY, USA, 1984. Association for Computing Machinery.
- [CXM17] Shenghui Cheng, Wei Xu, and Klaus Mueller. RadViz deluxe: An attribute-aware display for multivariate data. *Processes*, 5(4), 2017.
- [ECuRM15] Shoaib Ehsan, Adrian F. Clark, Naveed ur Rehman, and Klaus D. McDonald-Maier. Integral images: Efficient algorithms for their computation and storage in resource-constrained embedded vision systems. *Sensors (Basel)*, 15(7):16804–16830, July 2015.
- [Fur86] G. W. Furnas. Generalized fisheye views. *SIGCHI Bull.*, 17(4):16–23, April 1986.
- [GN04] Michael T. Gastner and M. E. J. Newman. Diffusion-based method for producing density-equalizing maps. *Proceedings of the National Academy of Sciences*, 101(20):7499–7504, 2004.
- [HGM<sup>+</sup>97] Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. DNA visual and analytic data mining. In *Proceedings of the 8th Conference on Visualization, VIS '97*, pages 437–442, Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.
- [HS11] J. Haurert and L. Sering. Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2555–2562, December 2011.
- [KPS<sup>+</sup>03] Daniel A. Keim, Christian Panse, Matthias Schäfer, Mike Sips, and Stephen C. North. His-toScale: An efficient approach for computing pseudo-cartograms. In *14th IEEE Visualization 2003 (VIS 2003)*, Piscataway, N.J, 2003.
- [Kro18] Barry J. Kronenfeld. Manual construction of continuous cartograms through mesh transformation. *Cartography and Geographic Information Science*, 45(1):76–94, 2018.
- [MBLD02] Mark Meyer, Alan Barr, Haeyoung Lee, and Mathieu Desbrun. Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, 7(1):13–22, 2002.
- [NK16] Sabrina Nusrat and Stephen Kobourov. The state of the art in cartograms. *Computer Graphics Forum*, 35(3):619–642, June 2016.
- [NM13] J. E. Nam and K. Mueller. TripAdvisor<sup>{N-D}</sup>: A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):291–305, February 2013.
- [PPCP12] Cyprien Pindat, Emmanuel Pietriga, Olivier Chapuis, and Claude Puech. JellyLens: Content-aware adaptive lenses. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, pages 261–270, New York, NY, USA, 2012.
- [PT19] Lucas Pagliosa and Alexandru Telea. RadViz++: Improvements on radial-based visualizations. *Informatics*, 6:16, Apr 2019.
- [Sun13] Shipeng Sun. A fast, free-form rubber-sheet algorithm for contiguous area cartograms. *International Journal of Geographical Information Science*, 27(3):567–593, 2013.
- [SZS<sup>+</sup>17] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, January 2017.
- [TGK<sup>+</sup>17] C. Tominski, S. Gladisch, U. Kister, R. Dachsel, and H. Schumann. Interactive lenses for visualization: An extended survey. *Computer Graphics Forum*, 36(6):173–200, September 2017.
- [Tob86] Waldo R. Tobler. Pseudo-cartograms. *The American Cartographer*, 13(1):43–50, 1986.
- [VJ02] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002.
- [WLLM13] Yingcai Wu, Xiaotong Liu, Shixia Liu, and Kwan-Liu Ma. ViSizer: A visualization resizing framework. *IEEE Transactions on Visualization and Computer Graphics*, 19(02):278–290, February 2013.