Scale Dependent Definitions of Gradient and Aspect and their Computation

Iris Reinbacher

Marc van Kreveld

Tim Adelaar

Marc Benkert

Department of Information and Computing Sciences, Utrecht University

Technical Report UU-CS-2006-011

www.cs.uu.nl

ISSN: 0924-3275

Scale Dependent Definitions of Gradient and Aspect and their Computation*

Iris Reinbacher† Marc van Kreveld† Tim Adelaar† Marc Benkert‡

Abstract

Contour maps, where each contour represents constant height values, are probably the most common kind of maps used for terrain visualization. Similar maps with curves representing constant gradient or areas representing constant aspect are also useful for applications in visualization and geomorphometry. In order to compute such maps of a terrain, we introduce scale dependent local gradient and aspect for a neighborhood around each point of a terrain. We present four different definitions for local gradient and aspect, and give efficient algorithms to compute them for a TIN terrain. We have implemented the algorithms for grid data and compare the results for all methods and different sizes of the neighborhood.

1 Introduction

Geomorphometry is concerned with the precise measurement and quantitative description of the shape of landforms. Given a terrain, the most important measures to classify landforms are slope, as well as profile curvature and plan curvature [19]. The value for slope at each point of the terrain is usually divided into *gradient*, i.e. the steepness of the slope, and *aspect*, the cardinal direction in which the slope faces. Using such measures and classifications, the goal is for example to derive drainage maps, specify areas in mountains that have high danger of avalanches, or study how a certain area has been formed.

Using some numerical value for gradient, and the classification convex or concave for plan and profile curvature, it is possible to identify landforms like convergent and divergent shoulders, footslopes, or crests, swales, and plains (see e.g. [9, 14, 15]). Slope can also be used to compute shaded relief maps, for irradiance mapping [4], and for parametric terrain classification.

Contour maps of terrains where each curve represents constant height values are very common. Especially when the original data is not available, they are used for example for digital terrain modelling [5, 7, 17]. However, as contour maps lack morphometric information between the contour lines, the outcome may not be satisfactory. Maps with curves representing constant gradient values — for simplicity we will call them *isogradients* — or areas of constant aspect (*isoaspects*) can aid in digital terrain modelling.

Other geomorphological features in terrains are critical lines. Critical lines are features where the slope or curvature changes abruptly, like ridges and valleys. They can be determined by identifying critical points such as maxima, minima, and saddle points of the terrain, and connecting them [16]. Critical points can be detected from their slope and curvature values [22], or using drainage network and catchment area delineation [23]. For applications like extracting volcanotectonic features it is necessary to find lines that define a break of slope without being a ridge or a valley. The terrain can also be partitioned into areas having the same curvature, and the critical lines are identified as the boundaries of these areas [10]. For this application, isogradient lines are needed as well.

*This research was partially supported by the Netherlands Organisation for Scientific Research (NWO) through the BRICKS project GADGET.

†Institute of Information and Computing Sciences, Utrecht University, {iris,marc,tadelaar}@cs.uu.nl

‡Department of Computer Science, Karlsruhe University, mbenkert@ira.uka.de

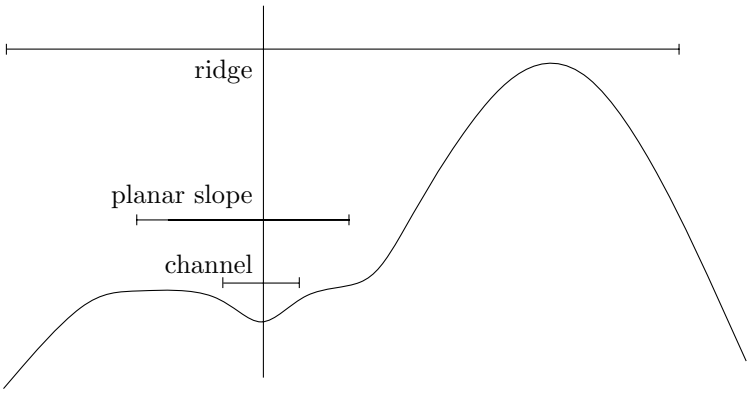

Figure 1: The same point can be classified differently at different scales, based on [12].

An important factor of geomorphometry is the scale at which we study the terrain. Visualization of a terrain at a large scale provides the most detail, and the smaller the scale, the more information is lost due to generalization. In his thesis, Wood [22] gave an extensive overview of the characterisation of geomorphological properties. He also argued that properties should be analyzed at the appropriate scale, which is not necessarily the scale of the data. For example, a potential problem for the classification of landforms with respect to scale is that the morphometric class may change with different scales; see Figure 1, taken from Fisher et al. [12]. The scale of the terrain model also influences for example the area of a lake or the length of a seashore. This influence can be crucial when the investigated spatial object already has fuzzy boundaries [3, 6]. The book edited by Tate and Atkinson [20] presents research on scale related issues in GIS.

As a new method that yields isogradient lines and isoaspect maps, we introduce *local gradient* and *local aspect* for each point of a terrain. The basic idea is to define the local gradient for a point based on some neighborhood of the point. The size of the neighborhood can be chosen, which makes the definition scale dependent. Our definitions yield a continuously changing value of the local gradient value on for example TINs (Triangular Irregular Networks), whereas the standard definition on a TIN does not yield continuity of the gradient. Continuity is important for the generation of isogradients. Our methods complement the research of Wood [22], who provides scale-dependent definitions of slope and aspect for gridded digital elevation models based on quadratic surface fitting. Our definitions apply to any terrain representation.

This paper is structured as follows. In Section 2 we introduce four different scale dependent definitions for local gradient and local aspect for every point on a terrain. In Section 3 we give corresponding algorithmic results for a TIN, assuming a square shaped neighborhood. The first two definitions for local gradient (aspect) use an average of all gradient (aspect) values in the neighborhood around any point. We can either apply uniform or non-uniform weighing for the averaging. These two definitions lead to algorithms to compute local gradient (aspect) for every point on a terrain with a running time of $O(mn)$. Here, m denotes the maximum number of edges intersected by the neighborhood of any point. In the third definition we choose the local gradient at a point to be the maximum gradient value to any other point inside the neighborhood. Local aspect is then defined by the vector to the point that realizes this maximum. We can compute local gradient and local aspect for the whole terrain in $O(n \cdot m^{3+\epsilon})$ time, where $\epsilon > 0$ is an arbitrarily small constant. The fourth definition for local gradient (aspect) at a point is the maximum average gradient (aspect) over a diameter of the neighborhood. This definition does not lead to an algorithm that can analytically derive a solution on a TIN, therefore we only present a heuristic to compute approximate isogradients and isoaspects in $O(nm)$ time. We have implemented all four methods for grid data and we compare the results for different sizes of the neighborhood in Section 4. Conclusions and an outlook on future research can be found in Section 5.

2 Definitions for Local Slope

Assume we are given a terrain T where every point has well defined values for slope. By default, slope is defined by a plane tangent to the surface at any given point. Since the tangent plane may be different for different points on the terrain, slope is a function in x and y. It consists of two components: the gradient, which is the maximum rate of change of altitude, and the aspect, the compass direction of the maximum rate of change. We will refer to these definitions as the *standard definitions* of gradient and aspect. The standard gradient and aspect can be seen as functions in x and y, because the values of x and y determine the standard gradient and aspect. Therefore we denote them by $\hat{F}_g(x, y)$ and $\hat{F}_a(x, y)$, respectively. Usually, gradient is given in percent, in degrees, or by a nonnegative real, and aspect is given in degrees, which is often converted to a compass bearing. The slope in the standard definition is denoted by $\hat{F}_s(x, y)$, and gives a vector in $\mathbb{R}^3$ whose length is irrelevant. It will be convenient to use the (upward) normal vector of the terrain at each point, and normalize the z-component of this vector to 1 as the representation of $\hat{F}_s(x, y)$.

With a suitable representation of the gradient and aspect functions we can generate scale dependent maps with lines or areas representing constant values of gradient or aspect. Throughout the paper, we will call these constant value lines and regions *isogradients* and *isoaspects*, respectively.

We introduce the notion of *local slope*, composed of *local gradient* and *local aspect*, for each point p of the terrain T, using some neighborhood around p. They are denoted by $F_s(p)$, $F_g(p)$, and $F_a(p)$. A natural choice for such a neighborhood can be derived from a disk in the xy-plane with some prespecified radius r, centered at p, which we denote by D_r. It is obvious that the choice of the size of the neighborhood influences the resulting isogradients and isoaspects and is therefore very important. If we choose a small radius r, taking only points close to p into account, we expect to get many, detailed isogradients. If we choose a large value for r, we expect to get few, more smooth isogradients.

The basic idea is to define for every point p that is the center of a disk D_r the local gradient and aspect depending on the points that lie inside D_r. We can do this in the following four ways:

1. Uniform weighing over the neighborhood D_r.
2. Non-uniform weighing over the neighborhood D_r.
3. Maximum value in the neighborhood D_r.
4. Uniform weighing over a diameter in the neighborhood D_r.

The local gradient value to be defined at any point will be given by a function $F_g(x, y) : \mathbb{R}^2 \to \mathbb{R}$. Note that the standard gradient value need not be defined everywhere on certain terrain representations. For example, for a TIN, the values are not well-defined on the edges and vertices. All four definitions for local gradient that we introduce lead to continuous functions $F_g(x, y)$ on the whole terrain if it is represented by a TIN.

The local aspect value at each point is given by a function $F_a(x, y) : \mathbb{R}^2 \to [0°, 360°) \cup \{\text{Flat}\}$, where the degree value is usually divided into a number of discrete classes (like North, East, West, South, or the eight cardinal directions). The additional value Flat is assigned to horizontal parts of the terrain, where the aspect is undefined. In this context, the local aspect function is continuous at p if the aspect at p is Flat, or if for every point in a sufficiently small ϵ-neighborhood of p, the aspect value changes only by a small amount δ_ϵ. The first two options to define aspect given above will lead to continuous functions $F_a(x, y)$ for the local aspect, the last two will not give continuity.

We will give the basic definitions and properties of local gradient and local aspect for a circular neighborhood D_r with radius r at any point p of the terrain in the remainder of this section.

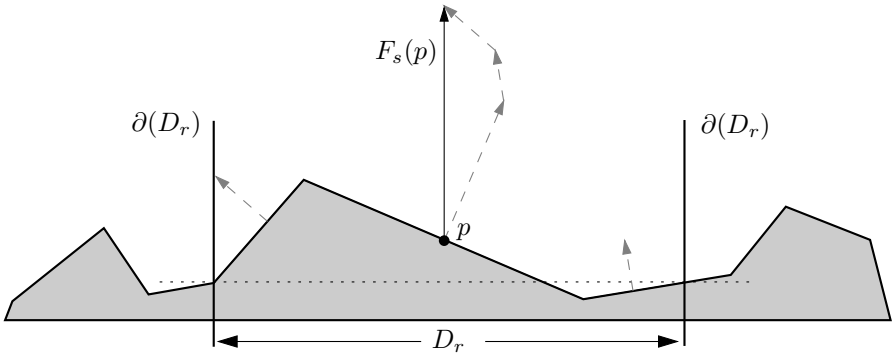

Figure 2: When the terrain is cut off at the same height at the boundary of D_r (∂D_r), the resulting gradient vector should point vertically upwards, indicating a value zero.

2.1 Uniform weighing over a neighborhood

2.1.1 Gradient

In the uniform weighing method over the given neighborhood D_r, we compute the average gradient sum of all points in D_r. The local gradient at a point p is defined by the following equation:

$$F_g(p) = \frac{1}{area(D_r)} \int_{p' \in D_r} \hat{F}_g(p') \quad dxdy \tag{1}$$

Here, $p' = (x, y)$ is some point in the neighborhood D_r, and $\hat{F}_g(p')$ is the gradient according to the standard definition.

Gradient is usually considered to be a scalar value. However, in the standard definition, the gradient at p is derived from the tangent plane at p, or the slope vector. Therefore, we have the choice whether we use the scalar gradient $\hat{F}_g(p')$ in Equation (1), or use the vector slope $\hat{F}_s(p')$ and derive the local gradient from the resulting local slope vector $F_s(p)$. The following example shows that this makes a difference. If we have two equal-size, adjacent regions with the same standard gradient value, say 10, and their outer normals pointing in opposite directions, we will get as local gradient the value 10, when we use the standard gradient $\hat{F}_g(p')$ in Equation (1). If we use the standard slope $\hat{F}_s(p')$ in Equation (1), the local slope $F_s(p)$ is vertically upward. Hence, the local gradient $F_g(p)$ at p, derived from $F_s(p)$, will be zero.

When integrating the slope vector, and the neighborhood cuts off the terrain at the same height everywhere at its boundary, the resulting gradient should give a local gradient zero. In case of a piecewise linear terrain in two dimensions, we can achieve this by normalizing the vertical component of the normal vectors to 1, before weighing it with the projected length of the terrain. The local gradient at p is derived from the weighted vector sum of all normal vectors. It is easy to show that in case of a piecewise linear terrain in two dimensions, the local gradient is a vector that points vertically upwards (see also Figure 2). This is the reason for normalizing the vertical component of the vector at every point before integration.

The uniform weighing leads to a continuous function F_g for local gradient (independent of whether we use $\hat{F}_g$ or $\hat{F}_s$). Hence, isogradients will generally be closed loops or end at the boundary of T.

2.1.2 Aspect

To define local aspect so that it corresponds to the local slope, we will use the vectors $\hat{F}_s(p')$ in the integration of Equation (1). We can derive the local aspect $F_a(p)$ at p directly from the resulting vector $F_s(p)$. This involves projecting the vector $F_s(p)$ into the xy-plane and taking its direction. If $F_s(p)$ projects to the null-vector, the aspect has the value Flat.

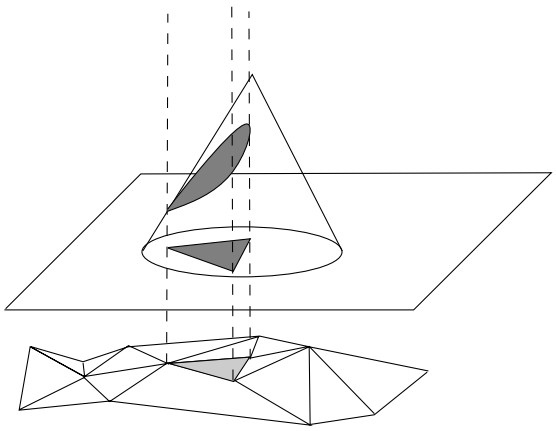

Figure 3: Computing the non-uniform weight for one triangle of a TIN is equivalent to computing the intersection of the weight cone with the prism erected on the triangle.

It is easy to see that for example on a TIN, the standard definition of aspect may jump from one of the possible values to any other one as a point passes an edge of the terrain. However, with our method of uniform weighing, that is, averaging over the neighborhood D_r, the local aspect function F_a becomes continuous, which means that it can only change to adjacent values (or Flat) with respect to the circular scale of directions.

2.2 Non-uniform weighing over a neighborhood

2.2.1 Gradient

In the non-uniform weighing method we give a higher importance to points that are closer to p and a lower importance to points that are closer to the rim of the disk D_r. We do this by a weight that decreases linearly with the distance to p. The image of the weight function is a cone with its apex at p. On the rim of D_r and outside D_r, the weight is zero.

Computing the weighted average over all points p' inside the disk D_r is equivalent to computing the height of the point on the cone C directly above p' times the standard gradient value at the point p'. See Figure 3 for an illustration of this definition on a TIN.

Stated more formally, we get the following integral representing the local gradient at point p:

$$F_g(p) = \frac{1}{vol(C)} \int_{p' \in D_r} (h - \frac{h}{r}\sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2}) \cdot \hat{F}_g(p') \quad dxdy \tag{2}$$

Here, $p = (p_x, p_y, p_z)$, $p' = (p'_x, p'_y, p'_z)$, h is the height of the cone, r the radius of the disk, and $\hat{F}_g(p')$ is the gradient at p' according to the standard definition. We may choose $h = 1$ because its value does not influence the relative weights of points in D_r.

Again, we can integrate over the standard gradient $\hat{F}_g(p')$ or the standard slope $\hat{F}_s(p')$. In the latter case we compute the local gradient from the local slope, as in the uniformly weighted case. The non-uniform weighing method also gives a continuous function F_g for local gradient.

2.2.2 Aspect

For the local aspect vector at a point p, weighted non-uniformly over the disk, we can use the same model with the linearly decreasing weight that corresponds to the local gradient. We determine the local aspect from the local slope vector at p, by projection into the xy-plane and taking its direction. The method of non-uniform weighing also gives a local aspect function F_a that is continuous, which means that its values can only change to adjacent values in the classified situation.

2.3 Maximum value in a neighborhood

2.3.1 Gradient

In the maximum value method for a neighborhood D_r, we set the local gradient at a point p to be the absolute maximum gradient value from p to any other point p' inside D_r. The gradient between two points $p = (p_x, p_y, p_z)$ and $p' = (p'_x, p'_y, p'_z)$ is defined as their z-distance divided by their Euclidean distance in the xy-plane, which leads to the following definition for the local gradient at a point p:

$$F_g(p) = \sup_{p' \in D_r \setminus \{p\}} \frac{|p_z - p'_z|}{\sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2}} \tag{3}$$

Note that this definition gives local gradient values that are at least as large as the standard gradient definition. Note that the point p' that gives the maximum gradient value may not be unique. This model leads to a continuous function F_g for local gradient.

2.3.2 Aspect

As the local aspect at p we choose the vector between p and the point that gives the maximum gradient and project it into the xy-plane. When there is more than one point p' giving the maximum gradient, the aspect is generally not well-defined. A possible solution could be to choose the point p' that has the largest z-distance to p, but this is an arbitrary choice, and can still give more than one point p' for the maximum gradient.

Note that this definition does not lead to a continuous local aspect function F_a, because whenever the point of maximum gradient changes, the local aspect vector jumps to correspond with the point of the new maximum.

2.4 Uniform weighing over a diameter of the neighborhood

2.4.1 Gradient

The uniform weighing over a diameter method is in a sense a combination of the first and the third methods presented in the previous subsections. As the standard slope value is realized in one direction, it is natural to take points in only that one direction into account for the averaging, in a local slope definition. We define the local gradient to be the maximum average gradient over all line segments ℓ that are diameters of D_r. Any such line segment ℓ has length $2r$ in case of a circular neighborhood. The local gradient at a point p is defined by the following integral:

$$F_g(p) = \max_{\phi \in [0,180)} \frac{1}{2r} \int_{p' \in \ell(p,\phi)} \hat{F}_g(p') dx dy \tag{4}$$

Here, ϕ denotes the angle of the line segment ℓ and the x-axis, and $\hat{F}_g(p')$ denotes the gradient at p' according to the standard definition.

As we are again taking an average value for local gradient, we can use the gradient value $\hat{F}_g(p')$ or the slope vector $\hat{F}_s(p')$ in Equation (4). In the first case we compute the local gradient value by integrating over all values and dividing by the length of ℓ inside the neighborhood. In the second case, the local gradient value is obtained from the local slope vector $F_s(p)$. The uniform weighing over a diameter method yields a continuous function F_g for the local gradient.

2.4.2 Aspect

For local aspect, we use the corresponding definition as for gradient, that is, we take the line segment $\ell(p, \phi)$ that leads to the (maximum) local gradient value, and compute the local aspect using the slope vectors of all points that lie on $\ell(p, \phi)$. We project the resulting vector into the xy-plane and take its direction.

This definition of local aspect does not lead to a continuous function F_a. The diameter leading to the maximum average gradient may change abruptly, and therefore the local slope vector can change abruptly as well (but note that the local gradient function is continuous even if it is derived from the local slope vector).

3 Algorithms for Local Slope and Isolines

In this section we will describe efficient algorithms to compute an explicit representation of the local gradient and local aspect on the whole terrain. This will allow us to determine isogradients and isoaspects in a simple way. On a grid terrain model, the computation of local gradient and aspect at every grid cell is relatively straightforward. We first compute the standard gradient, aspect, and slope for all grid cells using any of the existing methods (usually based on 3×3 subgrids). Then we approximate the neighborhood D_r by a window of grid cells, and use it to compute the local gradient and aspect at any grid cell according to any of the definitions given in the previous section. We note that Wood provides alternative definitions and computation of the local gradient and aspect on grid terrain models [22].

We next concentrate on the development of efficient algorithms for a TIN, which is considerably more involved. For the first two methods, where we compute the weighted average, we need to know the area of each triangle (in the triangulation in the xy-plane, not on the terrain surface) that is intersected by the neighborhood D_r. This area of intersection is given by a function in the coordinates of p. In case of a circular neighborhood, this function may consist of up to a linear number of terms, all involving square roots. Such functions generally cannot be simplified, and hence, the equations describing isogradients are too complex to be used. Therefore, we restrict ourselves to the case of a square neighborhood with side length $2r$, denoted by D_r. In this case, the area of intersection is given by a quadratic function in x and y, and the problems mentioned above do not occur. We observe that all algorithms can easily be adapted from square neighborhoods to regular polygons, if a better approximation to a circular neighborhood is desired. The area of intersection remains a quadratic function.

Note that on the edges and vertices of a TIN, the standard gradient and aspect are not defined. However, in our definitions of local gradient and aspect we assume that every point inside the neighborhood has a value for gradient and aspect. We can overcome this problem by either excluding these points from the neighborhood and hence from the computation, or by assigning the value from any neighboring triangle.

As mentioned in the last paragraph, we want to determine a function $F_g(x, y) \to \mathbb{R}$ that expresses the local gradient at each point (x, y). It is therefore natural to subdivide each triangle of the TIN into smaller cells such that for each point inside a cell, the function $F_g(x, y)$ is determined by the same features, that is, the same edges or vertices of the TIN. Whenever the boundary of such a cell is crossed, the list of features influencing $F_g(x, y)$ changes. When this function is known in each cell, it can be evaluated in constant time to compute the value for local gradient at any point p. Also, for any chosen local gradient value we can compute the isogradient in one cell in constant time by setting the function of that cell equal to the chosen value. The resulting equation gives the isogradient in that cell. We use $f_g^c(x, y)$ to denote the function $F_g(x, y)$ inside one cell c only. All functions $f_g^c(x, y)$ for all cells together define $F_g(x, y)$. We will see that for all four definitions of local gradient, the function $f_g^c(x, y)$ has constant description size. The descriptive or combinatorial complexity of $F_g(x, y)$ is the number of cells, or functions $f_g^c(x, y)$ needed to form $F_g(x, y)$.

Van Kreveld et al. show in [21] how to compute the *placement space* for a general subdivision S and a square Q, such that for each cell of the placement space, a reference point of the square can be placed such that the sides of the corresponding square intersect only fixed sets of edges of the subdivision. They obtain:

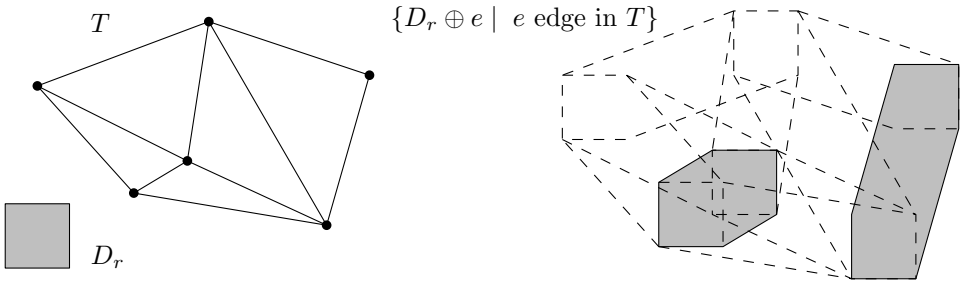

Figure 4: The triangulation T (left) and the Minkowski sums of its edges with a square D_r (right). For two out of ten edges, the Minkowski sums are shown in grey.

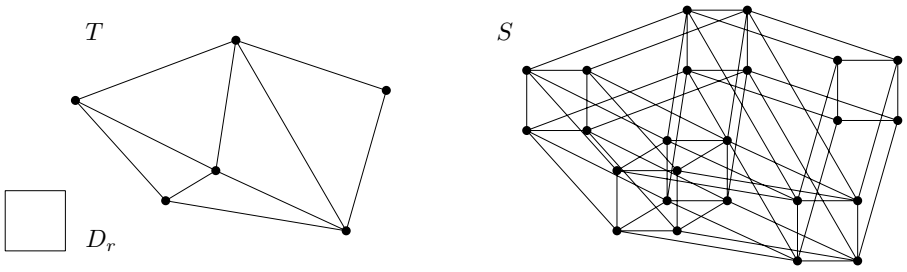

Figure 5: The placement space S of a square D_r in a triangulation T.

Theorem 1 *(From [21]) Given a subdivision S with n edges and a square Q, the arrangement representing all distinct placements of Q with respect to S can be constructed in $O(n \log n + k)$ time, where $k = O(n^2)$ is the number of distinct placements.*

Let T be a TIN, a planar triangulation that represents a terrain. We will use T to denote both the planar triangulation and the piecewise linear surface in three dimensions. Since the TIN and three-dimensional surface are in one-to-one correspondence, there is no ambiguity. Let n be the number of edges of T and let m be the maximum number of edges of T that are intersected by any placement of a square D_r. This number can be as large as $O(n)$, but typically it is much smaller. We will show that the arrangement representing the distinct arrangements has complexity $O(nm)$ in the worst case, and can be constructed in $O(nm)$ time.

Analyzing the number of distinct placements of a square D_r with fixed size and orientation in a triangulation T can be achieved by considering the Minkowski sum of D_r and the edges of T. The Minkowski sum of two sets in $\mathbb{R}^2$ is defined as

$$S_1 \oplus S_2 := \{p + q \mid p \in S_1 \,,\; q \in S_2\}, \tag{5}$$

where $p + q$ denotes the vector sum of the vectors from the origin to p and q. The Minowski sum of any edge and D_r is a hexagonal polygon; this is shown for two edges in Figure 4. Any point in the plane (in the triangulation) can be covered by at most m Minkowski sums of D_r and an edge, otherwise there must be at least one square intersecting more than m edges of T. Furthermore, the boundaries of every pair of Minkowski sums can have only two proper intersection points [8, 13]. With these observations, we can use a result of Sharir [18], which gives us an upper bound of $\Theta(mn)$ on the number of combinatorially distinct placements of D_r on T.

To compute the placement space of D_r on T, we compute the Minkowski sums of the triangulation T with each corner of a square D_r with side length $2r$, centered at the origin. This gives four equivalent triangulations, each translated by $+r$ or $-r$ in x- and y-direction. As these triangulations are simply connected and planar, we can use an algorithm of Finke and Hinrichs [11]

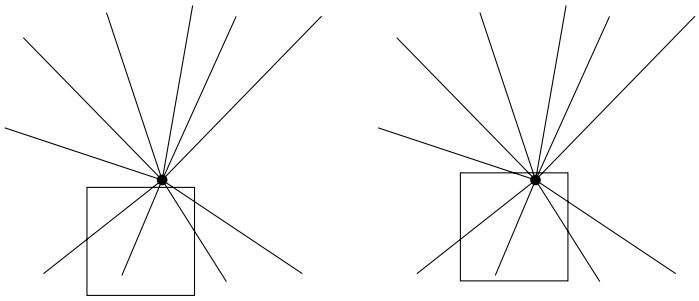

Figure 6: The set of features changes when an edge of D_r moves over a vertex of the terrain.

three times to compute their overlay in $O(n + mn) = O(mn)$ time. To get the final placement space S (see Figure 5), we also need to insert the square centered at each original vertex of the triangulation. We separately insert the vertical and horizontal sides of D_r, which each connect two of the translated vertices. We do so by traversing the cells of the subdivision S between the two vertices that are connected, and adding a new edge and a vertex for every intersection point of a side of a square with an existing edge of the subdivision. Observe that every cell of S has constant complexity because T is a triangulation. This property remains valid when horizontal and vertical sides of the squares are inserted. Hence, any side of a square can be inserted in time linear in the added complexity of the subdivision, that is, in the number of intersection points of the new edge. There can only be $O(m)$ intersected edges per inserted side, therefore all edge insertions take $O(nm)$ time altogether. We summarize this in the following lemma:

Lemma 1 *Let E be the set of n edges of a triangulation T, and let D_r be a square of fixed size and orientation. Let m be the maximum number of edges intersected by any placement of D_r. There are $O(mn)$ distinct subsets of edges of E intersected by different placements of D_r, and the subdivision representing all distinct placements of D_r with respect to T can be constructed in $O(nm)$ time.*

The combinatorially distinct placements of a square D_r in the triangulation T gives a subdivision S into smaller cells, see Figure 5. When placing the reference point (the center of D_r) anywhere inside such a cell, the sides of D_r will intersect the same set of at most m features (edges or vertices) of T.

3.1 Uniform weighing method

In the uniform weighing method, the placement space S is the subdivision such that in each of its cells, the local gradient is given by a function $f_g^c(x, y)$ that is a quadratic function in the coordinates (x, y) of the center of D_r. Clearly, the quadratic function has constant description size. The local gradient function $f_g^c(x, y)$ in a cell c depends on the $O(m)$ terrain features the square D_r will intersect. We need to determine the local gradient function for each cell of the subdivision S. A straightforward way to do this takes $O(nm^2)$ time. However, by not explicitly computing the $O(m)$ features for each cell of the subdivision, we can improve this. The idea is to precompute the *change* of $f_g^c(x, y)$ across all cell boundaries. Since $f_g^c(x, y)$ is a quadratic function, we can store the change with each edge of S in $O(1)$ space, while taking $O(nm)$ time overall.

We do this as follows, see Figure 6. Imagine placing D_r over a vertex of the triangulation such that the upper side lies directly above the vertex, and compute the $O(m)$ features and the gradient function for this placement of D_r. Then we move D_r downwards until the upper side lies directly below the vertex. We compute the change of the set of intersected features and of the gradient function between these two placements of D_r. We store the change of the gradient function with the corresponding $O(m)$ horizontal edges of the subdivision S. Note that it does not matter where the vertex of T crosses the upper side of D_r, the same set of intersected features

changes and hence, the cahnge of the local gradient function is the same everywhere. Therefore, we need to compute the change of the gradient function only once. We do this for all four sides of D_r and every vertex of the triangulation. Furthermore, for all other edges of S, the change of the gradient function is easy to determine in $O(1)$ time, as only one feature of T changes when these edges are crossed. The whole precomputation takes $O(nm)$ time.

Afterwards, we traverse the placement space S to determine the gradient function $f_g^c(x, y)$ of each cell. We choose a starting cell of S, in which we compute the $O(m)$ features and the local gradient function. From there, we traverse S to an adjacent cell. Whenever the reference point crosses a boundary of a cell, there are two possibilities: Either a corner of the square D_r crosses an edge of the triangulation, or a side of D_r crosses a vertex of the triangulation. As the change of the gradient function is stored with each cell boundary, both types of events can be handled in constant time. A depth first search traversal of S determines all gradient functions $f_g^c(x, y)$ in all cells c of S in $O(nm)$ time overall. We summarize:

Lemma 2 *Given a subdivision S representing the combinatorially distinct placements of a square D_r with fixed orientation and side length on a triangulation, we can determine the gradient functions $f_g^c(x, y)$ for all cells c of S in $O(nm)$ time.*

If we use the uniformly weighted local slope to determine the local gradient, we need to determine functions $f_a^c(x, y)$ rather than $f_g^c(x, y)$. These functions give a 3-dimensional vector of which all three components are quadratic functions. Hence, the same technique can be used for this version of uniform weighing. It also follows that the local aspect function can be computed efficiently, in $O(nm)$ time.

The gradient function $F_g(x, y)$ for the uniform weighing method is continuous over the whole terrain, but not differentiable at the boundaries of the cells. To determine the isogradients for any given value, we set the gradient function $f_g^c(x, y)$ in each cell c to this value and the equation gives the curve in one cell. We can do this for each cell of S in constant time. As the local gradient function $F_g(x, y)$ is continuous, the isogradients are curves on the terrain, which are either closed loops or end at the boundaries of the terrain.

In a similar fashion, we can compute the local aspect value for each point p. It is important to note that the cells c of the subdivision S, giving a function $f_a^c(x, y)$ for the local aspect, do not correspond to the isoaspect areas. The boundaries of the isoaspect areas depend on the classification and can be determined by setting $f_a^c(x, y)$ to the value in degrees of the boundary between adjacent aspect values in the compass rose (e.g. 22.5° for N-NE) for each cell c. Wherever $f_a^c(x, y)$ is undefined, we assign the value Flat. We summarize:

Theorem 2 *Let T be a TIN terrain with n triangles, let D_r be a square neighborhood with side length $2r$, and let m be the maximum number of edges of T intersected by any placement of D_r. We can compute in $O(nm)$ time a subdivision S of T that has $O(nm)$ cells, and for each cell c the local gradient $f_g^c(x, y)$. In each cell c, $f_g^c(x, y)$ is a quadratic function for the uniformly weighted local gradient. We can compute the corresponding subdivision for the uniformly weighted local aspect in the same time.*

3.2 Non-uniform weighing method

In the non-uniform weighing method, we give a higher importance to points that lie close to p and a lower importance to points that lie at the boundary of the neighborhood. The weight decreases linearly with the distance to p. As our neighborhood D_r is now a square, it is natural to use the L_∞ distance instead of the Euclidean distance. This way, the weight values form a pyramid with its apex at p. Computing the weight for each point inside a triangle is equivalent to computing the intersection volume of the pyramid P centered at p with a prism A, which has a triangular base and edges parallel to the z-axis. The volume of one such prism can be computed as

$$V_A = A_{xy} \cdot \frac{a + b + c}{3}, \tag{6}$$

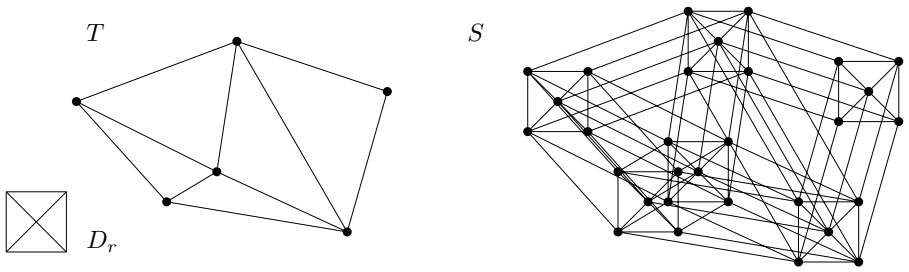

Figure 7: The placement space S of a square with diagonals D_r in a triangulation T.

where A_{xy} denotes the area of the triangle (in the xy-plane) and a, b, c depend on the coordinates of p and denote the side lengths of the three sides of the prism.

The algorithm to compute local gradient on the whole terrain is as follows: We construct the placement space S of the triangulation T with D_r as before, except that D_r is partitioned into four triangles by its diagonals, see Figure 7. This increases the number of cells only by a constant factor, so S has $O(mn)$ cells. Due to the non-uniform weighing, the local gradient function is a cubic function $f_g^c(x, y)$ in each cell c of S. We can use the same technique to compute the cubic functions $f_g^c(x, y)$ for all cells c of S as in the uniformly weighted case. The gradient function $F_g(x, y)$ is continuous over the whole terrain, but not differentiable at the boundaries of the cells.

To determine the isogradients for any given value, we set the gradient function in each cell to this value and the resulting equation gives the curve in the cell. We can do this for each cell of S in constant time. The local aspect function $F_a(x, y)$ and isoaspects are computed analogously. We conclude:

Theorem 3 *Let T be a TIN terrain with n triangles, let D_r be a square neighborhood with side length $2r$, and let m be the maximum number of edges of T intersected by any placement of D_r. We can compute in $O(nm)$ time a subdivision S of T that has $O(nm)$ cells, and for each cell c the local gradient $f_g^c(x, y)$. In each cell c, $f_g^c(x, y)$ is a cubic function for the uniformly weighted local gradient. We can compute the corresponding subdivision for the non-uniformly weighted local aspect in the same time.*

3.3 Maximum value method

In the maximum value method, the local gradient at p is defined by the maximum absolute gradient from p to any other point p' in D_r. We observe that on a TIN, the maximum gradient will occur between p and a vertex or an edge of the terrain or the boundary of D_r.

Computing the gradient from p to any other point p' inside D_r is straightforward. One can show that for an arbitrary line ℓ and any point p that is not on ℓ, the gradient between p and ℓ can have only one maximum. That means that the maximum gradient between a point p and an edge e of the terrain T either lies in the interior of e or at one of its endpoints. Note that when only part of an edge e lies inside D_r, we only consider this part for the computation of the maximum gradient. The point of maximum gradient on ℓ can be determined by applying analytical methods using Equation (3), the equation for ℓ and the coordinates of p. This takes constant time for each line and thus for each edge e of the terrain.

The algorithm to compute local gradient on the whole terrain is as follows: We generate the placement space S from the projected terrain and D_r as described in the beginning of Section 3, and overlay it with the original triangulation T to assure that p is inside a single triangle if p is in a cell of this new subdivision S'. Within one cell of S', two points p_1 and p_2 can still realize the maximum gradient to different features, causing different definitions of the local gradient function. Therefore, we need to subdivide cells of S' further, so that the maximum gradient is realized to the same feature within each cell. Only then can we define constant size functions $f_g^c(x, y)$ that

are valid in the whole cell. So we further subdivide each of the cells of S' to obtain a subdivision S'' such that in each cell of S'', exactly one vertex or edge of the TIN or part of the boundary of D_r defines the maximum gradient.

We determine the subdivision S'' as follows. For any cell c of S' we compute the gradient function from every point p to each of the $O(m)$ features inside each cell in $O(nm^2)$ time for all cells of S'. This will yield $O(m)$ surface patches in three dimensional space given by the fraction of a quadratic function and the square root of a quadratic function in x and y, derived from Equation (3). To find the one feature that determines the maximum for a given point, we need to find the pointwise maximum of all surface patches inside the cell. The pointwise maximum of m surfaces is called the *upper envelope*. It has complexity $O(m^{2+\epsilon})$ and can be computed in $O(m^{2+\epsilon})$ time [2], where $\epsilon > 0$ is an arbitrarily small constant. The intersection curves of two surface patches on the upper envelope are projected onto the triangulation to obtain a refined subdivision S''. In each cell of S'', there is only one feature such that the gradient from each point in the cell to this feature is maximal, and the local gradient function $f_g^c(x, y)$ itself can easily be determined from this feature and the triangle of T that (x, y) lies in.

The local gradients functions $f_g^c(x, y)$ for all cells c of S'' form the overall local gradient function $F_g(x, y)$, which is continuous, but not differentiable at the boundaries of the cells. For each cell, we can determine the isogradients as before in constant time, and joined, they give the isogradients of T.

For the local aspect function, we determine the vector from p to the point with maximal gradient and convert it to the aspect value. The subdivision S'' is the same as the subdivision for local gradient. Note that the local aspect function $F_a(x, y)$ is not continuous.

Theorem 4 *Let T be a TIN terrain with n triangles, let D_r be a square neighborhood with side length $2r$, let m be the maximum number of edges of T intersected by any placement of D_r, and let $\epsilon > 0$ be an arbitrarily small constant. We can compute in $O(n \cdot m^{3+\epsilon})$ time a subdivision of T that has $O(n \cdot m^{3+\epsilon})$ cells, and for each cell c the local gradient $f_g^c(x, y)$. The maximum value local gradient $f_g^c(x, y)$ is a fraction of a quadratic function and the square root of a quadratic function. We can compute the corresponding subdivision for the maximum value local aspect in the same time.*

3.4 Uniform weighing over diameter method

In the uniform weighing over diameter method, the local gradient is defined to be the maximum average gradient over all diameters of D_r. The computation of local gradient and aspect with this method on a TIN has two potential difficulties. Firstly, note that a diameter may contain an interval that coincides with an edge of the TIN. This causes the local gradient on that diameter to be undefined, because the standard gradient is undefined for a large part of the diameter, or even the whole diameter. We will exclude these diagonals from the ones over which we maximize. Secondly, for each point p on the terrain T, the local gradient is determined by a function in (x, y, ρ), where (x, y) are the coordinates of p and ρ denotes the angle of the diameter with the x-axis. To compute the maximum average gradient at any point p, we have to determine the value of ρ that maximizes the gradient function. This cannot be done analytically, not even for a given point p, because it requires solving a polynomial equation in ρ that can be of degree up to m.

To deal with the latter problem, we will not compute the exact value of the local gradient according to Equation (4), but an approximation of it. We do this by computing the average value of the gradient function for a constant number of predefined values of ρ, and then taking the maximum of these average values. To deal with the former problem, we make sure that we choose the predefined values of ρ such that none of them has the orientation of any edge of T. Observe that for a square neighborhood D_r, different values of ρ give different diameter lengths to be used in Equation (4).

The heuristic to compute an approximate local gradient on the whole terrain is as follows. Let some constant k be given. Compute k diameters of D_r using angles $\phi = 180/k$ degrees. The union of these diameters is a star $Z_{k,r}$ with $2k$ points. If any of the diameters is parallel to any edge of

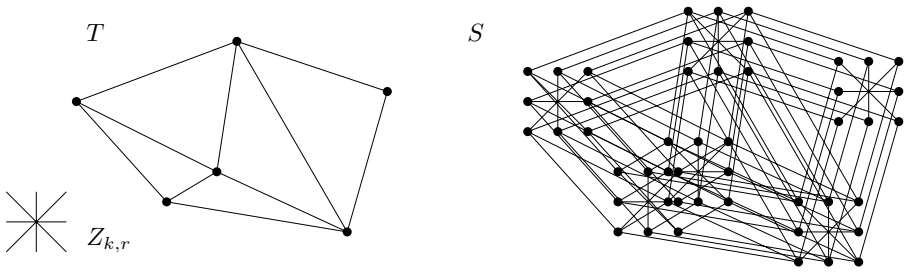

Figure 8: The placement space S of a star $Z_{k,r}$ with $k = 4$ in a triangulation T.

T, then we rotate the star $Z_{k,r}$ slightly to avoid the problem that the gradient is not well-defined for some diameters. We compute the placement space S from the triangulation T and $Z_{k,r}$ with the same technique as before, see Figure 8. Since k is assumed to be constant, S has complexity $O(nm)$. We can further subdivide each of the $O(nm)$ cells of S, such that in each cell of the refined subdivision S', the local gradient for each point is defined by the same value of ϕ. Inside each cell c of S', the local gradient will be defined by a linear function $f_g^c(x, y)$ corresponding to the average gradient over one diameter only.

We can determine the boundaries between the cells of S', where the local gradient is defined by the same value of ϕ, as follows. For each cell c of the subdivision S, and for each of the k predefined values for ϕ, we compute the average gradient of any point (x, y) for that value of ϕ as a function of x and y. We get k functions $f_{g,1}^c(x, y), \ldots, f_{g,k}^c(x, y)$ defining the average gradient over the diameters. These functions are linear in x and y. We can find the maximum average gradient in each cell of S by computing the upper envelope of all k linear functions in $O(k \log k)$ time, which is constant for constant k. The intersection of two functions of the upper envelope gives the boundaries of the refined subdivision S', which can have $O(k)$ cells for each cell of S.

As before, the upper envelope of all linear functions over the whole terrain is the representation of the gradient of every point of the terrain. The resulting gradient function $F_g(x, y)$ is continuous, but not differentiable at the boundaries of the cells.

The local aspect value is computed in a similar fashion. The subdivision S' is the same as the subdivision for the local gradient. The representation of the local aspect of every point is not continuous, as the diameter leading to the maximum average gradient may change abruptly whenever a cell boundary is crossed.

Theorem 5 *Let T be a TIN terrain with n triangles, let D_r be a square neighborhood with side length $2r$, and let m be the maximum number of edges of T intersected by any placement of D_r. We can compute in $O(nm)$ time a subdivision of T that has $O(nm)$ cells, and for each cell c a function $f_g^c(x, y)$ that approximates the local gradient for the uniform weighing over diameter method. $f_g^c(x, y)$ is a linear function. We can compute the corresponding subdivision for the local aspect, uniformly weighted over a diameter, in the same time.*

We observe that, as we only average over a number of diameters of the neighborhood D_r, the shape of D_r does not directly influence the computations. Therefore, in our approximation of the uniform weighing over a diameter, a circular neighborhood can be used as well. The lengths of the diameters forming the star $Z_{k,r}$ would be the same in this case.

4 Experimental Results for Grid Data

We have implemented our methods for DEM data in Java and compared them for data of different types of terrain. We downloaded the data set used for the visualizations shown here from [1]. It is a 358 by 468 pixel grid representing an area of approximately 11.3 by 14.5 kilometers northwest of

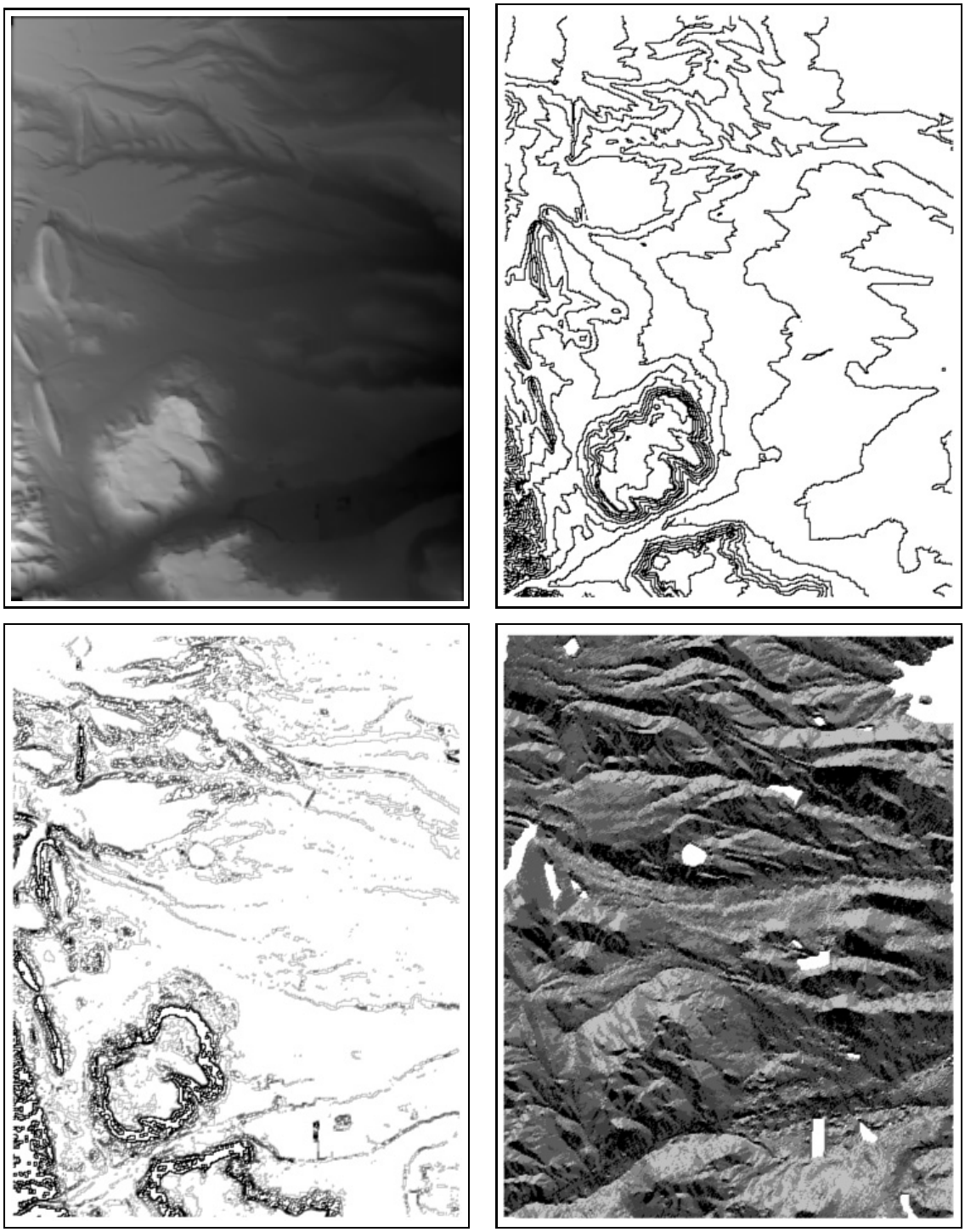

Figure 9: From top to bottom and left to right: The original terrain as a DEM, the 100 m contour map of the original terrain, the standard gradient, and the standard aspect.

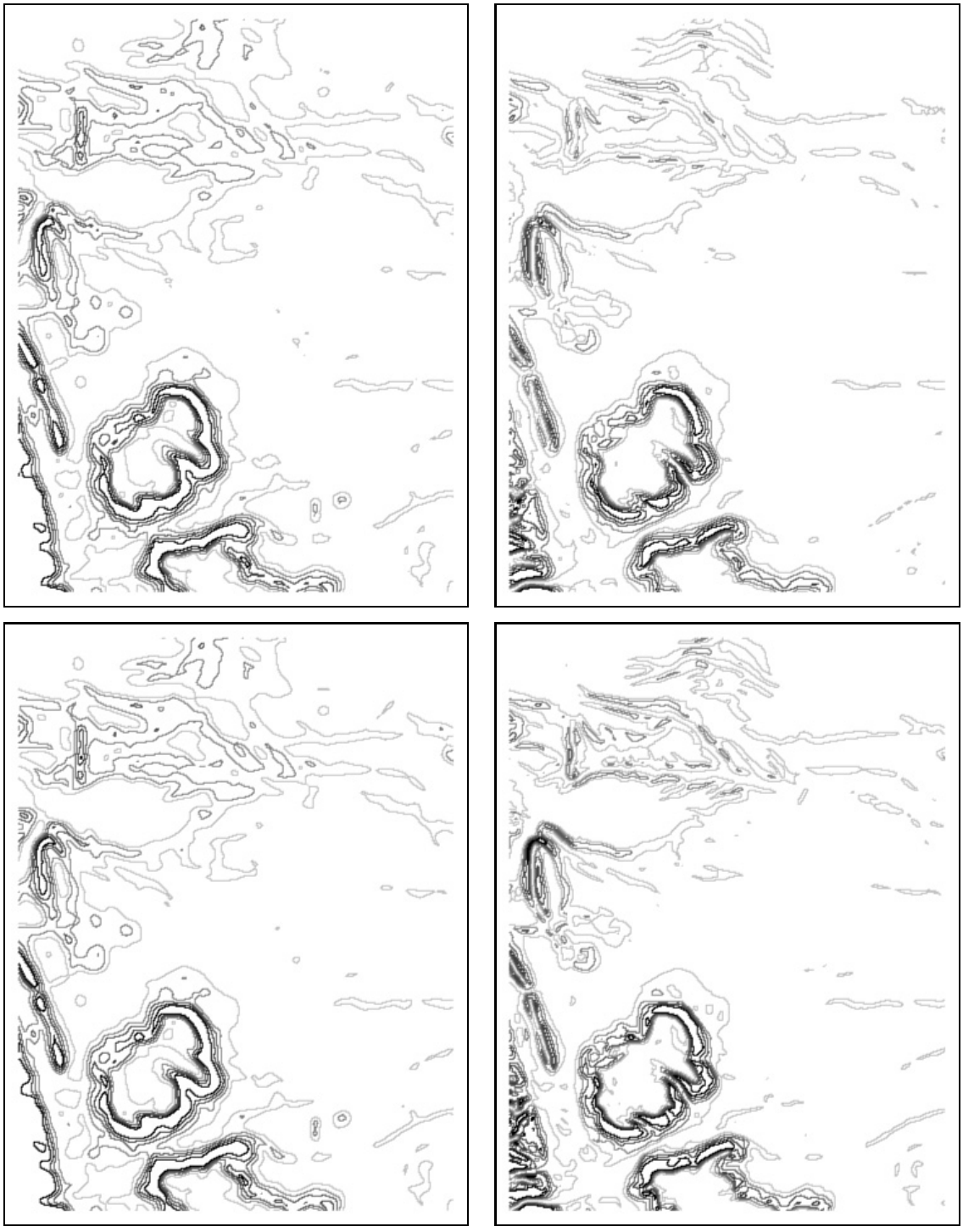

Figure 10: Comparison of different gradient methods for radius $r = 5$. From top to bottom and left to right: uniform weighing (by standard gradient), uniform weighing (by standard slope), non-uniform weighing (by standard gradient), non-uniform weighing (by standard slope).

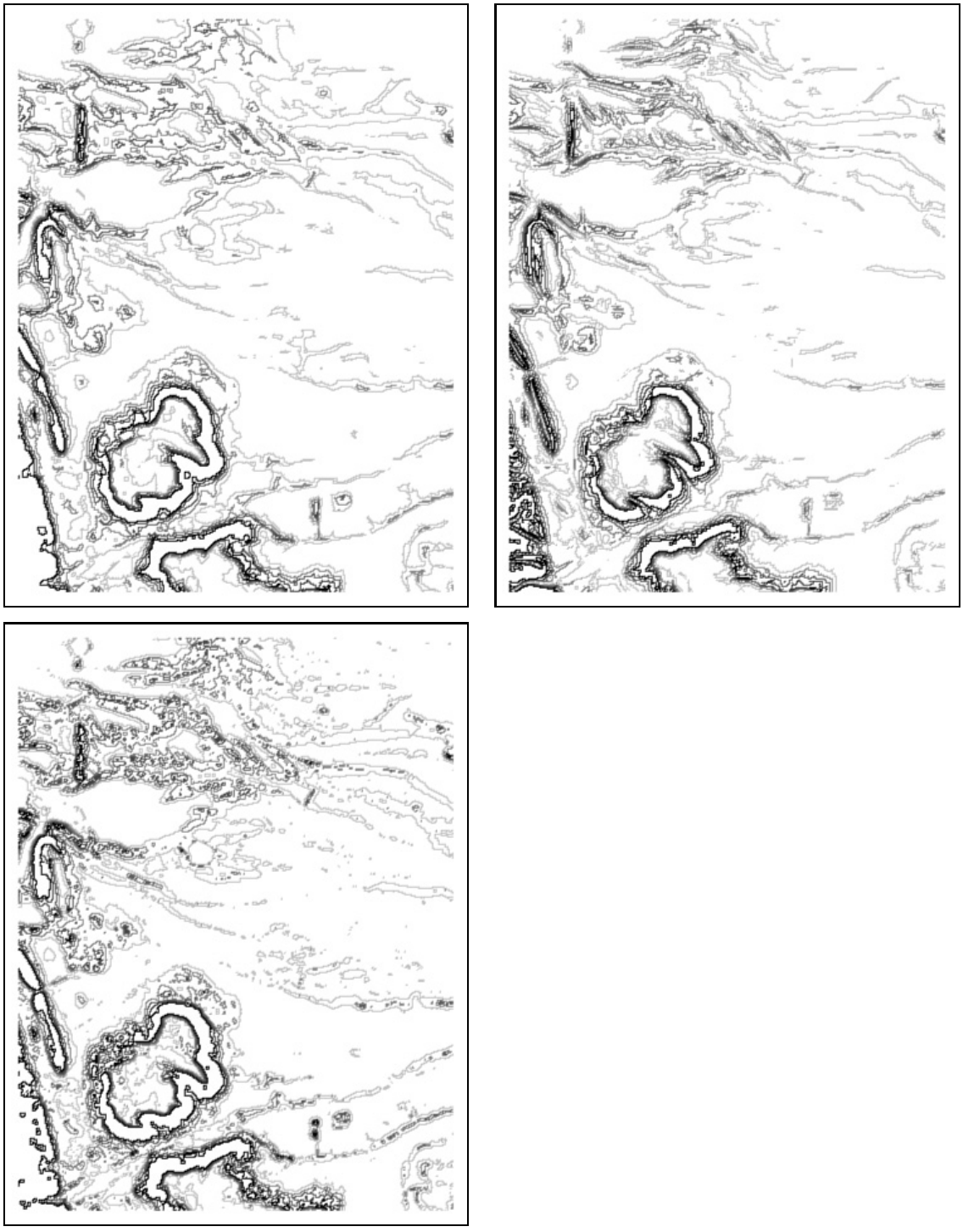

Figure 10 (cont.): Comparison of different gradient methods for radius $r = 5$. From top to bottom and left to right: uniform weighing over diameter (by standard gradient), uniform weighing over diameter (by standard slope), maximum value.

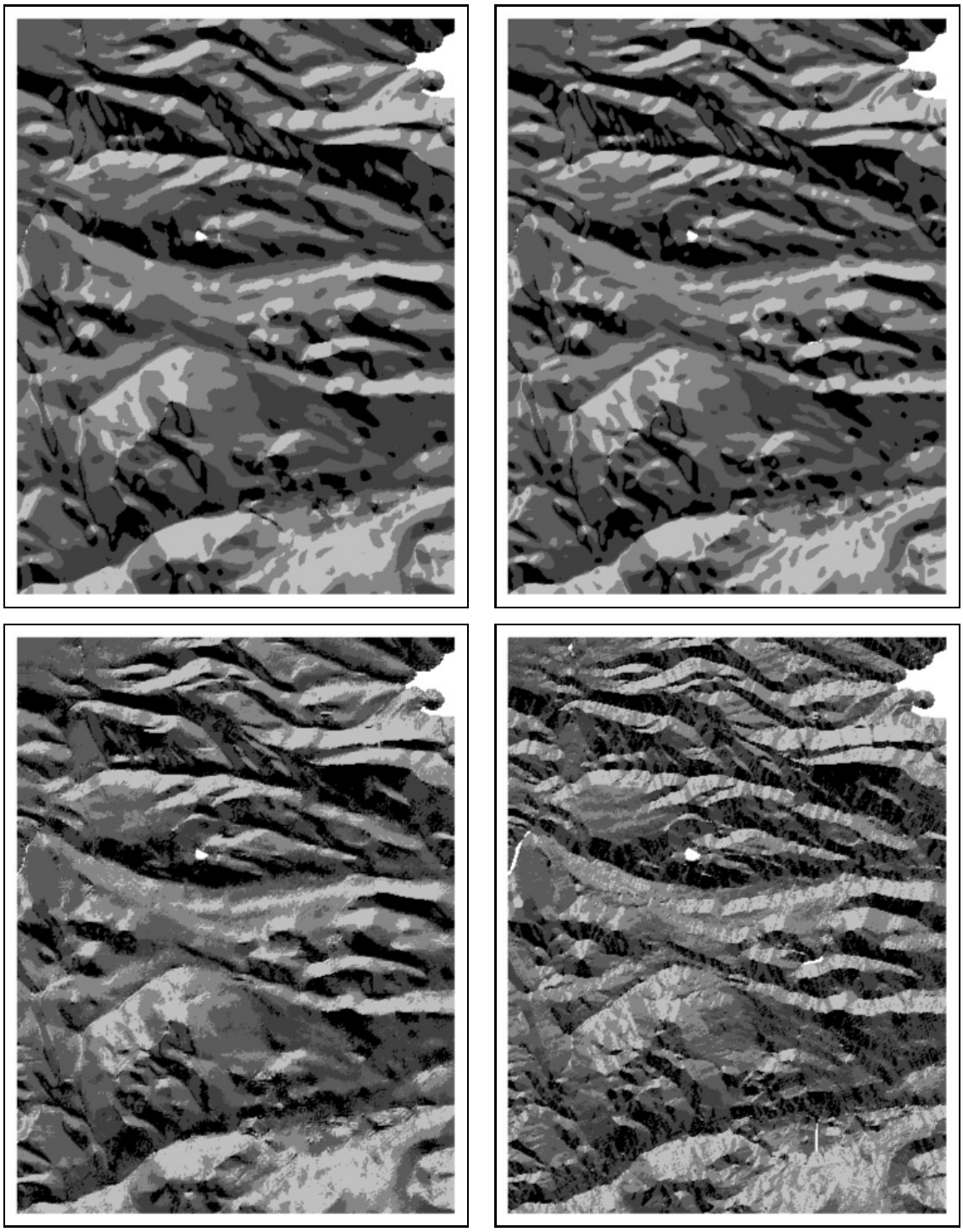

Figure 11: Comparison of four different aspect methods for radius $r = 5$. From top to bottom and left to right: uniform weighing, non-uniform weighing, uniform weighing over diameter, maximum value.

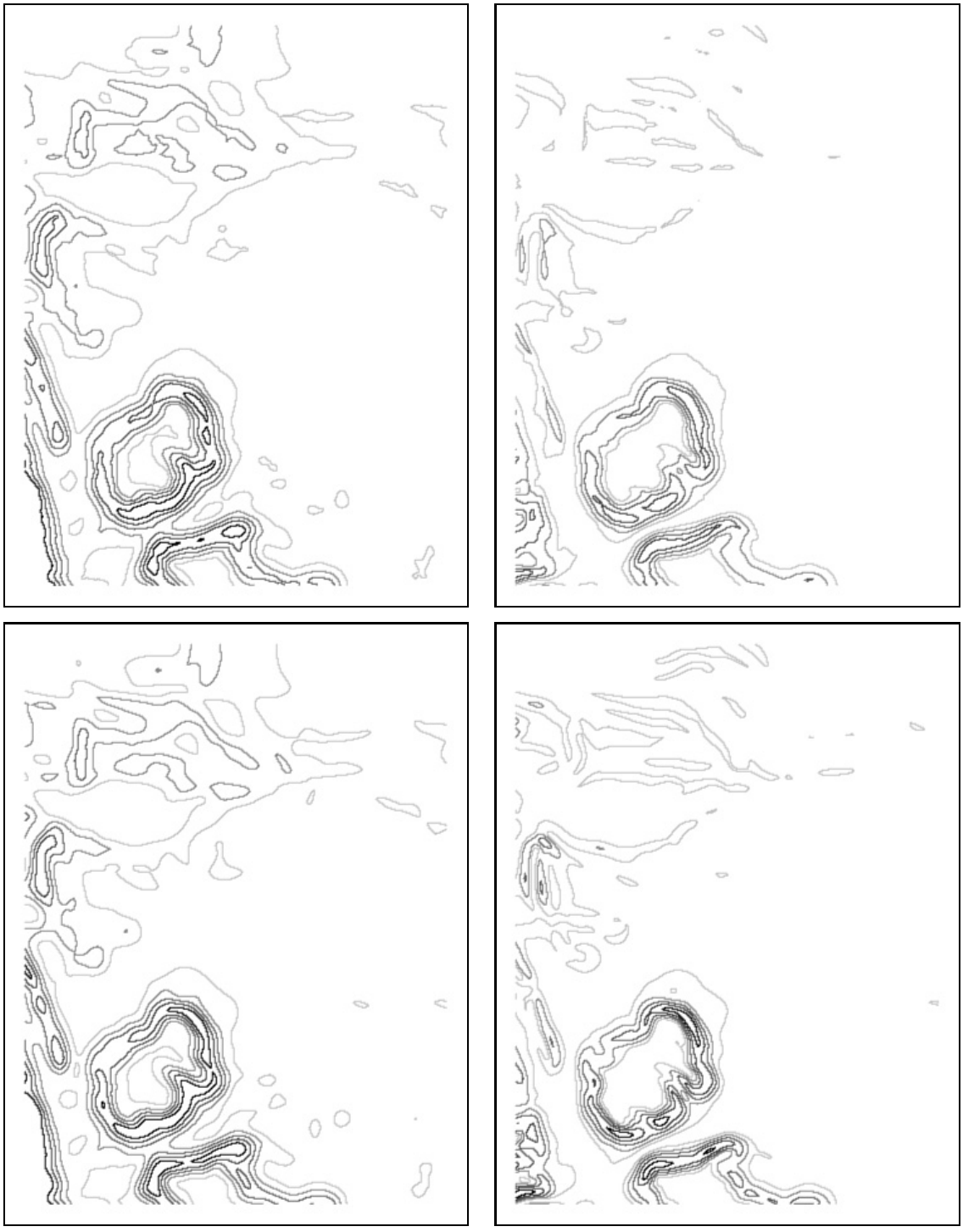

Figure 12: Comparison of different gradient methods for radius $r = 10$. From top to bottom and left to right: uniform weighing (by standard gradient), uniform weighing (by standard slope), non-uniform weighing (by standard gradient), non-uniform weighing (by standard slope).

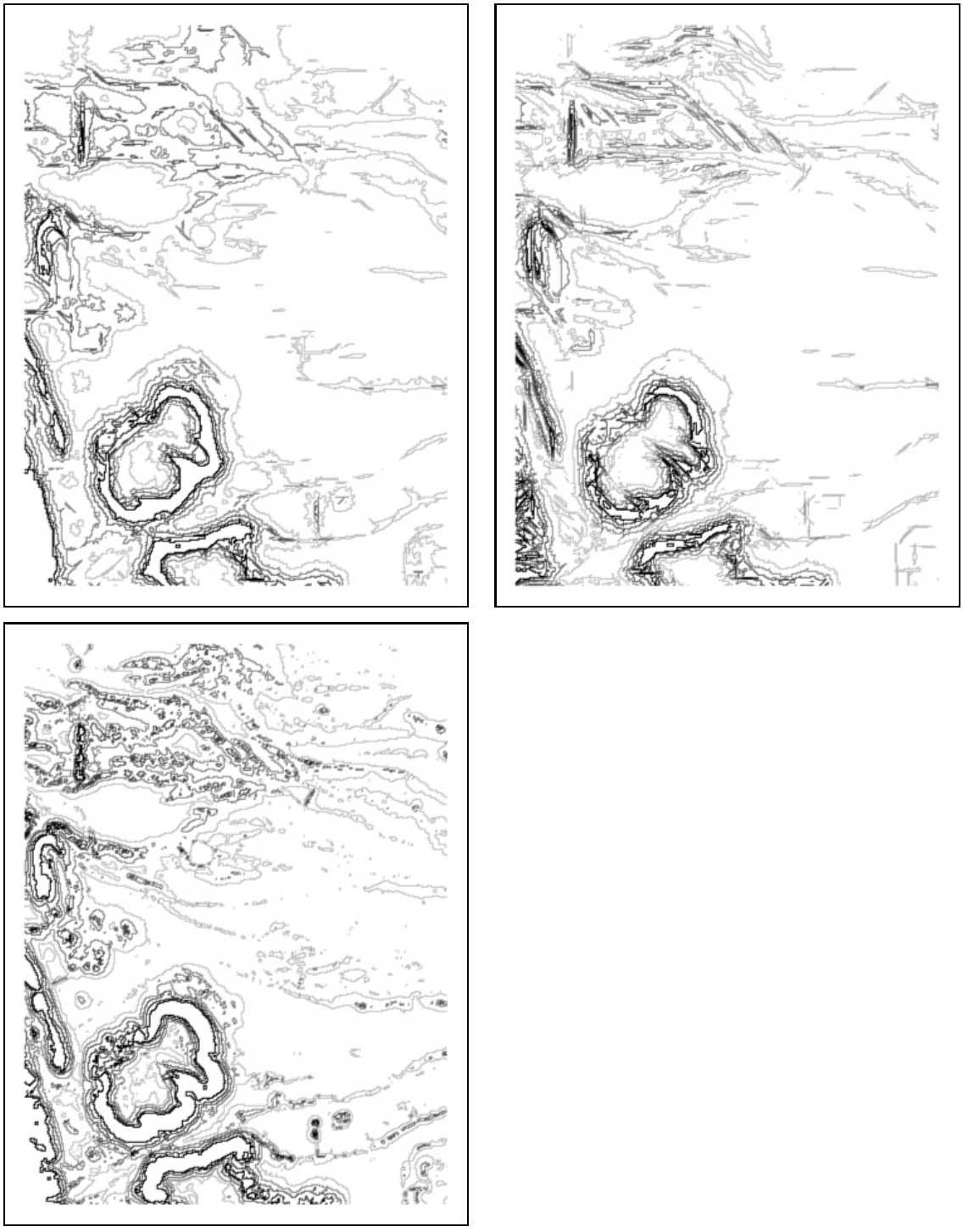

Figure 12 (cont.): Comparison of different gradient methods for radius $r = 10$. From top to bottom and left to right: uniform weighing over diameter (by standard gradient), uniform weighing over diameter (by standard slope), maximum value.

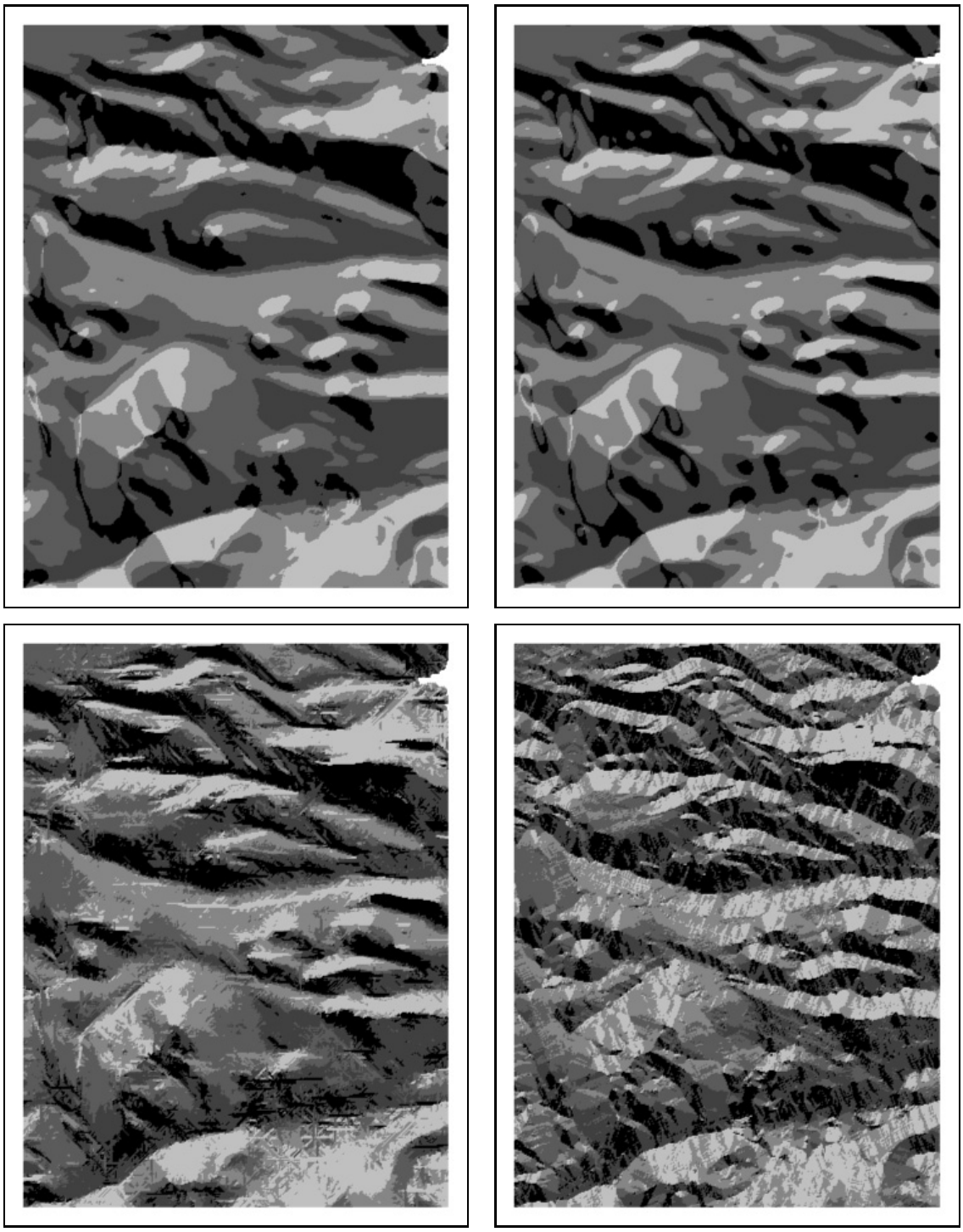

Figure 13: Comparison of four different aspect methods for radius $r = 10$. From top to bottom and left to right: uniform weighing, non-uniform weighing, uniform weighing over diameter, maximum value.

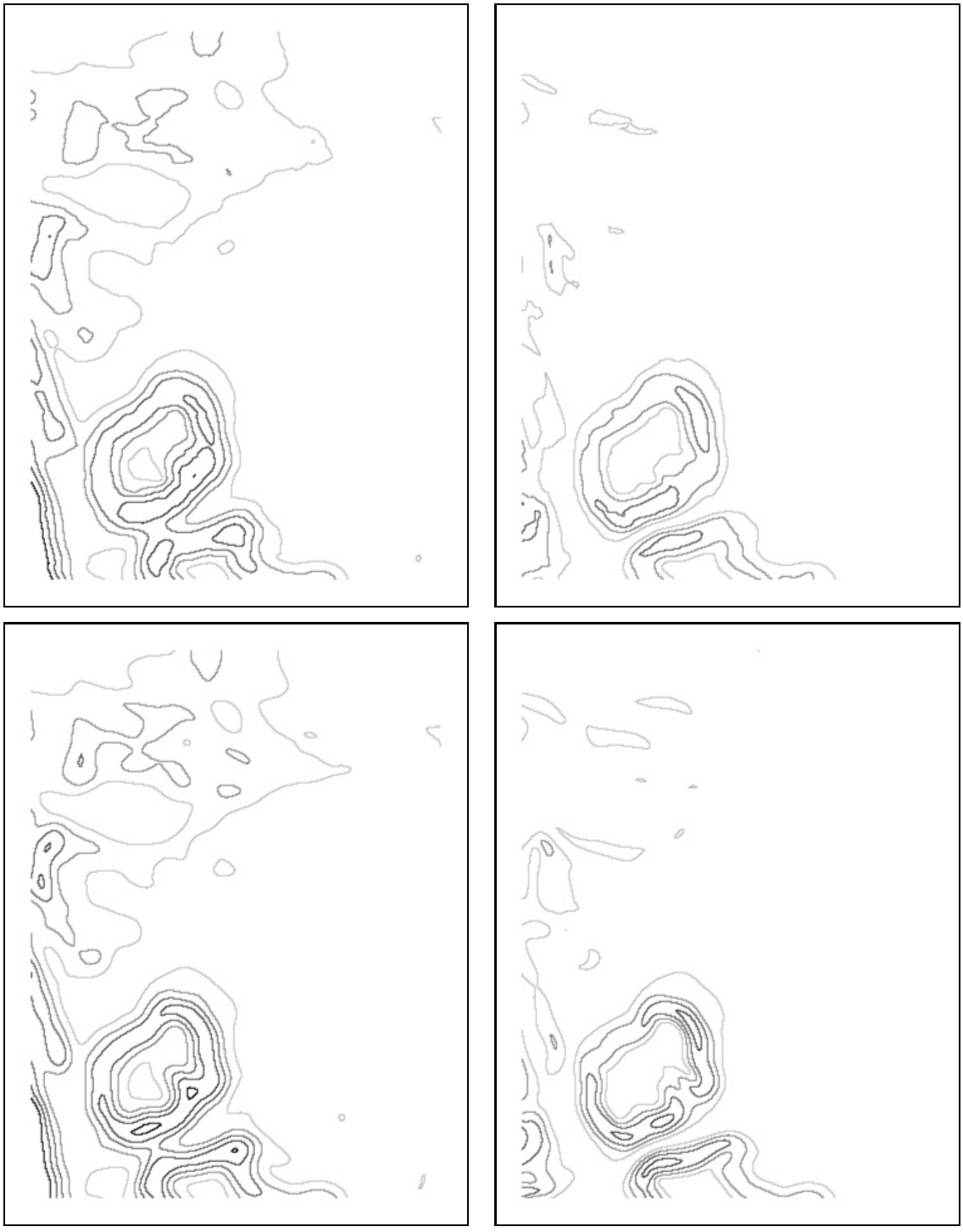

Figure 14: Comparison of four different gradient methods for radius $r = 15$. From top to bottom and left to right: uniform weighing (by standard gradient), uniform weighing (by standard slope), non-uniform weighing (by standard gradient), non-uniform weighing (by standard slope).

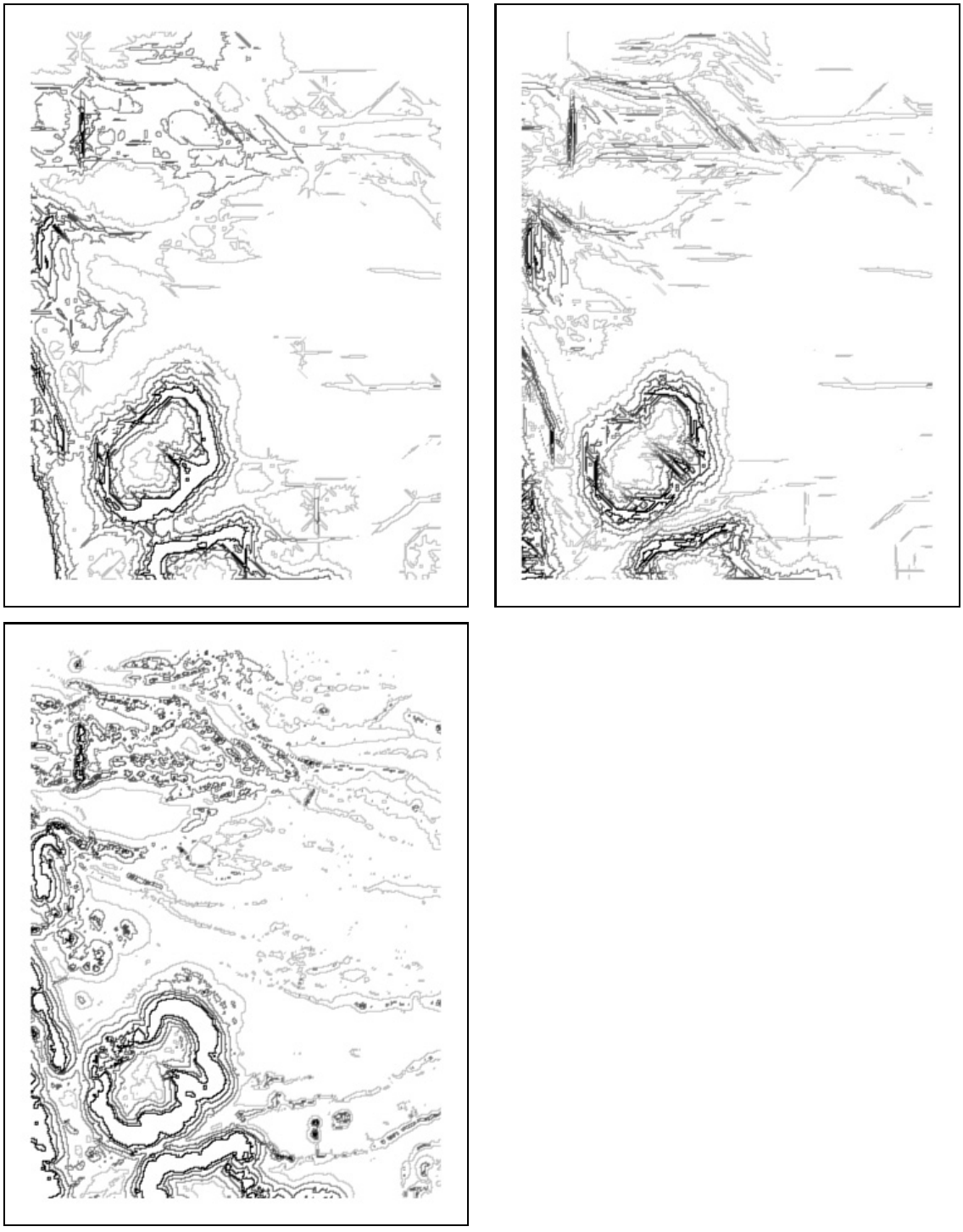

Figure 14 (cont.): Comparison of four different gradient methods for radius $r = 15$. From top to bottom and left to right: uniform weighing over diameter (by standard gradient), uniform weighing over diameter (by standard slope), maximum value.

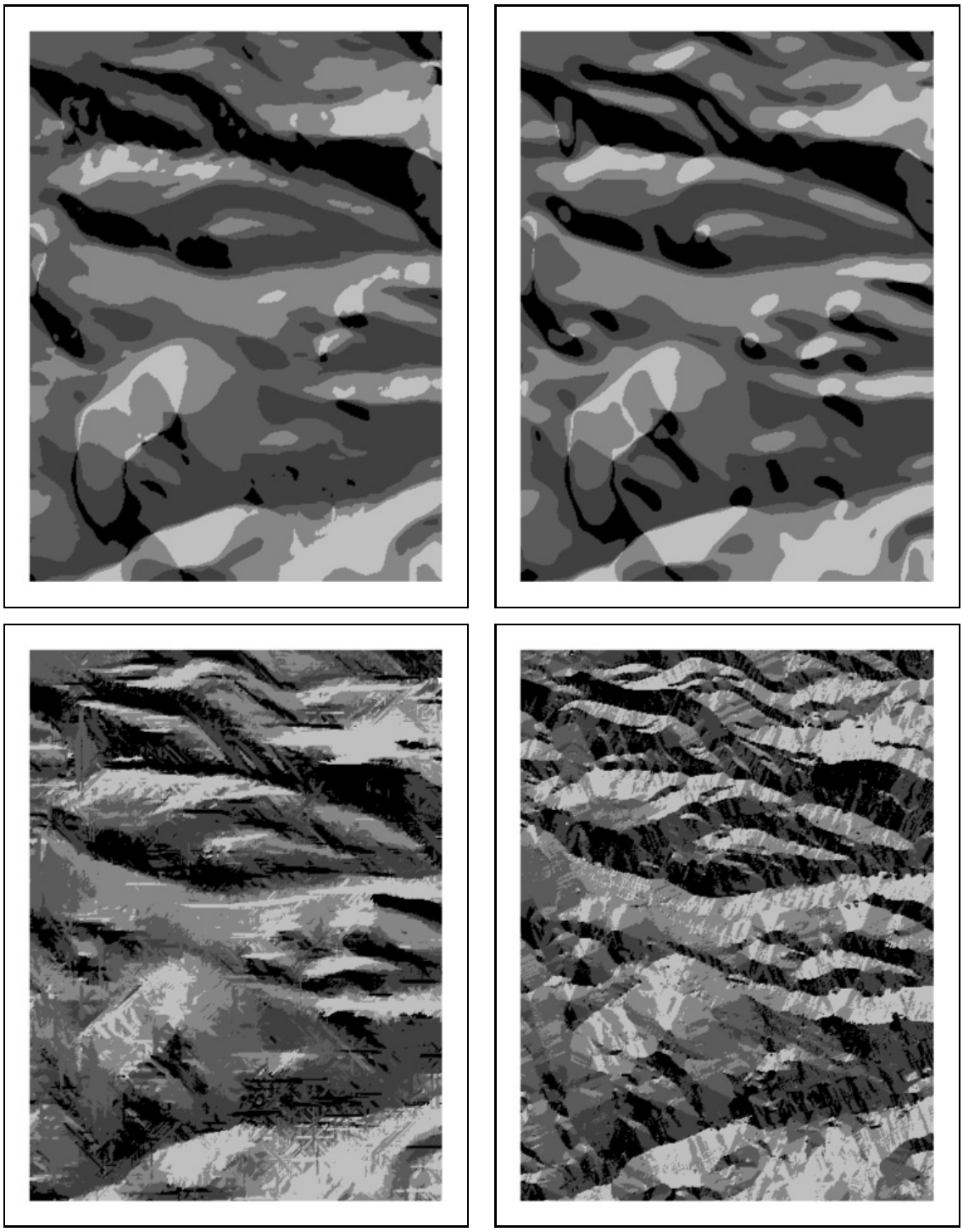

Figure 15: Comparison of four different aspect methods for radius $r = 15$. From top to bottom and left to right: uniform weighing, non-uniform weighing, uniform weighing over diameter, maximum value.

Denver, USA, with a grid spacing of approximately 30 meters. In Figure 9, we show the original elevation data (black is low and white is high elevation) and its 100 meter contour map at the top. For the presented data set and a radius of $r = 5$ pixels, our experiments on an iBook G4 with 1.2 GHz CPU and 384 MB memory took only a few seconds for all methods.

In all figures that we will discuss, we have used a circular neighborhood of given radius r around p. We approximate the disk D_r on the grid as follows: Every grid cell whose center is closer to p than r pixel sides is part of D_r, all other grid cells are not. The chosen values for the isogradient lines were 0.3 (light gray), 0.6, 0.9, 1.2, and 1.5 (black). The aspect maps are computed similar to the gradient maps. In the aspect maps, white represents a flat area, black has aspect facing South and light gray is aspect North. For readability of the aspect maps, no distinction was made between the grey toning of East and West. The bottom half of Figure 9 shows the standard gradient map and the standard aspect map of the terrain, based on a 3×3 neighborhood of each pixel. The white, flat areas of the standard aspect map depict lakes in the chosen area.

Note that when the chosen neighborhood does not fully cover the terrain, that is, at the edges of the terrain, there is no proper neighborhood to define local gradient and aspect, so we set the value for local gradient and aspect to zero. This is the reason for having a frame of width r around each of the maps.

4.1 Comparison of different methods

We first compare the outcome of the different methods for local gradient and aspect. The following observations are valid for all three investigated values of radius $r = 5$, 10, and 15 pixels, which corresponds to approximately 150, 300, and 450 meters.

For the local gradient, we see in Figures 10, 12, and 14, that the maximum value method gives the most detailed map. This is as expected, as this method does not perform any averaging. As expected, the level of detail of the uniform weighing over a diameter method lies in between the maximum value method and the averaging methods that use all of D_r, which give the smoothest maps. Furthermore, we see that the methods that compute the local gradient based on the standard gradient provide more detail than the methods that compute the local gradient based on the standard slope. This can be explained by the additional smoothing due to the use of slope. There appears to be more detail in the non-uniform weighing methods than in the uniform weighing methods.

In general, it seems that the uniform weighing method (in both versions) provides the best output when looking for a generalization of the isogradients. When high detail is desired, the maximum value method is preferred. This is also the only method that preserves the highest isogradient class well. It gives a smoothing of a different character than the other methods.

The situation for the methods for local aspect is similar, see Figures 11, 13, and 15. Again, the maximum value method shows the most detailed map of all methods, followed by the uniform weighing over a diameter method. The uniform and the non-uniform weighing method give a similar result for this terrain; the non-uniform method produces a slightly more detailed aspect map.

4.2 Comparison of different radii

Next we discuss the influence of different radii on the outcome of isogradients and isoaspects. Comparing Figures 10, 12, and 14, we can see that the smaller the radius, and therefore the area of influence, the more detailed the gradient map becomes. We get many isolines with highly detailed boundaries, many of which are relatively short. Also the spacing between isolines with different values is small. When increasing the radius, we get less detailed maps, the isolines show less detail and become more smooth, and also the distance between isolines with different values gets larger.

The situation for aspect is similar, see Figures 11, 13, and 15. Again, the aspect map with smallest radius of the neighborhood shows the highest detail, and the map with largest radius

has the largest and smoothest areas. Note the gradual decrease in size of the flat white lake area in the upper right corner of the aspect map. As an area is flat only if the aspect points exactly in z-direction, even one cell inside the neighborhood that is outside the flat area will change the outcome of the averaging and cause the local aspect to be non-flat. The effect of decreasing detail with increasing radius is smallest in case of the maximum value method and larger, but similar, in all other methods.

5 Conclusions and Future Research

In this paper we have introduced the notion of scale dependent slope for a terrain. The scale parameter is given by a radius of the neighborhood of influence. We suggested four different definitions of the slope that are scale dependent, and presented efficient algorithms to compute a representation of it on a TIN for each of them. Once we have a representation of the slope at every point of the terrain, it is straightforward to compute maps with lines of constant gradient or areas of constant aspect.

The results of the implementation on a gridded DEM show the expected smoothing behavior compared to the slope values computed by the standard method. The maximum value method gives the most detailed maps for all investigated radii, followed by the uniform weighing over a diameter. The output of the unweighted and weighted methods are similar. It is unclear to us whcih of the methods is preferable. The uniform weighing method is the easiest one to implement, it may be the method of choice. However, which method to prefer for a certain application may also depend on the desired level of detail of the result.

Directions for future research includes more extensive experimentation, for example to compare the length of isogradients and the areas of isoaspects for different methods and radii. Furthermore, it is interesting to develop similar methods as investigated in this paper to compute scale dependent maps that show plan and profile curvature, and other measures used in geomorphometry.

Acknowledgements: We would like to thank Peter Lennartz for valuable discussions.

References

[1] Mapmart homepage, `http://www.mapmart.com/samples/samples.htm`.

[2] P. K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.

[3] P. Burrough and A. Frank, editors. *Geographic Objects with Indeterminate Boundaries*, volume II of *GISDATA*. Taylor & Francis, London, 1996.

[4] P. A. Burrough and R. A. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, 1998.

[5] A. Carrara, G. Bitelli, and R. Carla. Comparison of techniques for generating digital terrain models from contour lines. *International Journal of Geographical Information Systems*, 11(5):451–473, 1997.

[6] T. Cheng, P. Fisher, and Z. Li. Double vagueness: Effect of scale on the modelling of fuzzy spatial objects. In *Developments in Spatial Data Handling, Proc. 11th Int. Symp. on Spatial Data Handling*, pages 299–314, 2004.

[7] M. Dakowicz and C. M. Gold. Extracting meaningful slopes from terrain contours. *Int. J. Comput. Geometry Appl.*, 13(4):339–357, 2003.

[8] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.

[9] I. S. Evans. The morphometry of specific landforms. *International Geomorphology*, Part II:105–124, 1986.

[10] B. Falcidieno and M. Spagnuolo. A new method for the characterization of topographic surfaces. *International Journal of Geographical Information Systems*, 5(4):397–412, 1991.

[11] U. Finke and K. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 119–126, 1995.

[12] P. Fischer, J. Wood, and T. Cheng. Where is Helvellyn? Fuzziness of multi-scale landscape morphometry. In *Transactions of the Institute of British Geographers, 29(1)*, pages 106–128, 2004.

[13] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

[14] C. W. Mitchell. *Terrain Evaluation*. Longman, London, 2nd edition, 1991.

[15] D. J. Pennock, B. Zebarth, and E. de Jong. Landform classification and soil distribution in hummocky terrain, Saskatchewan, Canada. *Geoderma*, 40:297–315, 1987.

[16] S. Rana, editor. *Topological Data Structures for Surfaces*. Wiley, 2004.

[17] B. Schneider. Geomorphologically sound reconstruction of digital terrain surfaces from contours. In *Proc. 8th Int. Symp. on Spatial Data Handling*, pages 657–667, 1998.

[18] M. Sharir. On k-sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.

[19] J. G. Speight. Parametric description of land form. In G. A. Stewart, editor, *Land Evaluation papers of a CSIRO Symposium 1968*, pages 239–250, 1968.

[20] N. Tate and P. Atkinson, editors. *Modelling Scale in Geographical Information Science*. John Wiley & Sons, Chichester, 2001.

[21] M. van Kreveld, E. Schramm, and A. Wolff. Algorithms for the placement of diagrams on maps. In *Proc. 12th Int. Symp. ACM GIS (GIS'04)*, pages 222–231, 2004.

[22] J. Wood. *The Geomorphological Characterisation of Digital Elevation Models*. PhD thesis, University of Leicester, 1996.

[23] S. Yu, M. van Kreveld, and J. Snoeyink. Drainage queries in TINs: From local to global and back again. In *Proc. 7th Int. Symp. on Spatial Data Handling*, pages 13A.1 – 13A.14, 1996.