NEW METHODS FOR REDUCING SIZE GRAPHS

Patrizio Frosini[1 2], Massimiliano Pittore[3]

[1] Department of Mathematics, University of Bologna, I-40127 BOLOGNA, ITALY
[2] C.I.R.A.M., BOLOGNA, ITALY
[3] Department of Informatics, University of Genoa, GENOA, ITALY

email: frosini@dm.unibo.it - pittore@disi.unige.it

Abstract. Two new methods for reducing complexity of size graphs are described. Some theoretical results are given together with various examples.

Keywords: shape, size graph, discrete size function.
1991 Mathematics Subject Classification: Primary 05C90, 68R10; Secondary 05C78, 05C85

1 Introduction: Size Graphs and the Reduction Problem

Size Theory is a new approach to the problem of comparing "shapes" of topological spaces, based on a mathematical transform named *size function*. Such a theory appears to be particularly useful in Computer Vision, when the topological spaces to be described and compared are images or parts of images. In fact, size functions have interesting properties such as resistance to noise and capability to be useful also in presence of occlusions (cf. [7]). Moreover, their modularity allows us to make them invariant under the transformation group we are interested in, e.g. the group of isometries or affine or projective transformations.

Size Theory has turned out to be useful for quite a lot of applications (see, e.g., [1], [9], [10], [11], [12], [13] and [14]).

In previous papers ([2], [3], [4], [5] and [6]) the base of Size Theory was given. For a survey of the subject we refer to [7].

Size functions are integer functions of two real variables, defined through "measuring functions". The idea underlying the concept of size function is that of setting metric obstructions to the classical notion of homotopy. Thus size functions convey information on both topological and metric properties of the viewed shape. In this paper we shall confine ourselves to the discrete aspect of the theory (cf. [3], [5] and [7]), since here we are interested in the algorithmic computation of size functions.

* This work was partially supported by MURST (Italy), ELSAG Bailey (Italy) and ASI (Italy).

In computing discrete size functions, we have to count the components of particular subgraphs of a graph labelled at its vertices, named *size graph*. Obviously, the smaller the graph, the faster is the computation. Moreover, size graphs are often big, so the problem of simplifying their structures without changing the associated discrete size functions is very important in order to use Size Theory successfully. This paper introduces two methods of reducing size graphs without changing the corresponding discrete size functions, that is $\mathcal{L}$-reduction and Δ-reduction. The main properties of these two methods are studied and some useful theorems about those are proved. Finally a comparison of such methods is given.

In this paper we only consider finite undirected graphs without loops or multiple edges. We stress that, although all graphs displayed in the following figures are planar and connected, the methods described in this paper can be applied to every size graph, also in case it cannot be embedded in the real plane or it is not connected.

2 Some basic definitions

In this section we recall the main definitions we need for dealing with discrete size functions. For our terminology about graphs we refer to [8].

Definition 1. Assume a finite graph G is given and call $V(G)$ and $E(G)$ the set of vertices and edges of G, respectively. Assume a function $\varphi : V(G) \to \mathbb{R}$ is fixed. Then the pair (G, φ) will be called *a size graph*.

Remark. If (G, φ) is a size graph and φ is injective, then G becomes an oriented graph $\vec{G}$ in a natural way, by giving to each edge e the orientation going from the vertex with higher value of φ to the other one.

In all figures we shall adopt the convention of representing φ as the height function with respect to the lowest vertex, so that $\varphi(v_a) > \varphi(v_b)$ if and only if v_a is higher than v_b in the picture.

For each size graph a discrete size function is defined:

Definition 2. For every $x \in \mathbb{R}$ we shall denote by $G\langle\varphi \le x\rangle$ the subgraph of G obtained by erasing all vertices of G at which φ takes a value strictly greater than x and all edges that connect those vertices to other vertices. If v_a, v_b are two vertices belonging to the same connected component of $G\langle\varphi \le x\rangle$, then we shall write $v_a \cong_{G\langle\varphi\le x\rangle} v_b$.

Definition 3. Call T^+ the set of the ordered real pairs (x, y) with $x \le y$. Consider the function $\ell_{(G,\varphi)} : T^+ \to \mathbb{N}$ defined by setting $\ell_{(G,\varphi)}(x, y)$ equal to the number of connected components of $G\langle\varphi \le y\rangle$ containing at least a vertex of $G\langle\varphi \le x\rangle$. We shall call $\ell_{(G,\varphi)}$ the *discrete size function of the size graph* (G, φ).

Example 1. In Figure 1 we give two size graphs and their respective discrete size functions. In each region of the half-plane $\{y \ge x\}$ the value taken by the discrete size function is displayed.

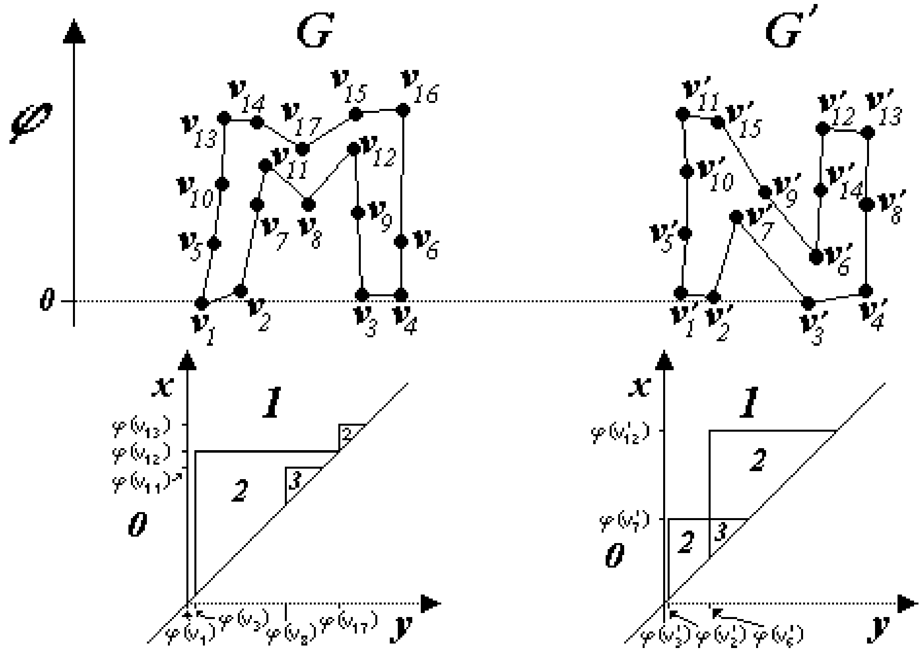

Fig. 1. Two size graphs and their discrete size functions.

3 A global method for reducing (G, φ): $\mathcal{L}$-reduction

In the following we assume that a size graph (G, φ) is given. We shall denote by $V = \{v_1, v_2, \ldots, v_n\}$ the (ordered) set of the vertices and by E the set of the edges of G.

Definition 4. We define a function $L : V \to V$ the following way. For every $v_i \in V$ call A_i the set containing v_i and all vertices of G that are adjacent to v_i. Define $B_i \subseteq A_i$ as the set whose elements are the vertices $w \in A_i$ for which the number $\varphi(v_i) - \varphi(w)$ takes the largest value. Finally, choose the vertex $v_k \in B_i$ for which the index k is minimum. Then we set $L(v_i) \stackrel{def}{=} v_k$. For obvious reasons, we shall call L the *single step descent flow function.*

Remark. Because of the definition of L and the finiteness of V, for every $v \in V$ there must exist a minimum index $m(v) \leq n$ such that $L^{m(v)}(v) = L^{m(v)+1}(v)$ (if $L(v) = v$ we set $m(v) = 0$).

The previous Remark justifies the following definition.

Definition 5. For every $v \in V$ we set $\mathcal{L}(v) \stackrel{def}{=} L^{m(v)}(v)$. This defines a function $\mathcal{L} : V \to V$: we shall call $\mathcal{L}$ the *descent flow operator.*

Remark. In plain words, the descent flow operator takes each vertex v_i to a local minimum with respect to the measuring function φ. During the descent, indexes are used to decide the path in case the set B_i contains more than one vertex.

As an example about our definitions, in the size graph (G, φ) displayed in Figure 1 we have $B_{16} = \{v_6\}$, $B_6 = \{v_4\}$, $B_4 = \{v_3, v_4\}$ and $B_{11} = \{v_7, v_8\}$. Hence $L(v_{16}) = v_6$, $L(v_6) = v_4$, $L(v_4) = \mathcal{L}(v_{16}) = v_3$ and $L(v_{11}) = v_7$.

Definition 6. Each vertex v for which $\mathcal{L}(v) = v$ will be called a *minimum vertex* of (G, φ). Call M the set of the minimum vertices of (G, φ).

Definition 7. Assume that v_{j_1}, v_{j_2} are two distinct minimum vertices of (G, φ). Suppose $v_{i_1}, v_{i_2} \in V$ are two adjacent vertices of G such that the following statements hold:

1. $\{\mathcal{L}(v_{i_1}), \mathcal{L}(v_{i_2})\} = \{v_{j_1}, v_{j_2}\}$
2. if $v_{i_3}, v_{i_4} \in V$ are two other adjacent vertices of G for which the equality $\{\mathcal{L}(v_{i_3}), \mathcal{L}(v_{i_4})\} = \{v_{j_1}, v_{j_2}\}$ holds then either $\max\{\varphi(v_{i_1}), \varphi(v_{i_2})\} < \max\{\varphi(v_{i_3}), \varphi(v_{i_4})\}$ or $\max\{\varphi(v_{i_1}), \varphi(v_{i_2})\} = \max\{\varphi(v_{i_3}), \varphi(v_{i_4})\}$ with (i_1, i_2) preceeding (i_3, i_4) in the lexicographic order.

We shall call the set $\{v_{i_1}, v_{i_2}\}$ *the main saddle adjacent to the minimum vertices* v_{j_1} *and* v_{j_2}. Call S the set of the main saddles of (G, φ).

Figure 1 displays some examples of minimum vertices and main saddles. In G the minimum vertices are v_1, v_3, v_8, v_{17} (we point out that $\varphi(v_3) = \varphi(v_4)$) and the main saddles are $\{v_{13}, v_{14}\}$ (adjacent to v_1 and v_{17}), $\{v_{15}, v_{16}\}$ (adjacent to v_{17} and v_3), $\{v_8, v_{12}\}$ (adjacent to v_8 and v_3), $\{v_8, v_{11}\}$ (adjacent to v_8 and v_1). In G' the minimum vertices are v'_2, v'_3, v'_6 and the main saddles are $\{v'_{11}, v'_{15}\}$ (adjacent to v'_2 and v'_6), $\{v'_{12}, v'_{13}\}$ (adjacent to v'_6 and v'_3), $\{v'_2, v'_7\}$ (adjacent to v'_2 and v'_3).

The previous definitions allow us to define the concept of $\mathcal{L}$-reduced graph:

Definition 8. Call $G^{\mathcal{L}}$ the graph whose vertices are the elements of the set $V^{\mathcal{L}} \stackrel{def}{=} M \cup S$ and whose adiacency relations are so defined: two vertices $u, w \in V^{\mathcal{L}}$ are adjacent if and only if one of them is a minimum vertex and the other one is a main saddle adjacent to it (in the sense of previous Definition 7). Then we define $\varphi^{\mathcal{L}} : V^{\mathcal{L}} \to \mathbb{R}$ this way: $\varphi^{\mathcal{L}}(u) \stackrel{def}{=} \varphi(u)$ if $u \in M$ and $\varphi^{\mathcal{L}}(u) \stackrel{def}{=} \max\{\varphi(v_{i_1}), \varphi(v_{i_2})\}$ if $u = \{v_{i_1}, v_{i_2}\} \in S$.

The size graph $(G^{\mathcal{L}}, \varphi^{\mathcal{L}})$ will be called *the* $\mathcal{L}$*-reduction of* (G, φ).

In Figure 2 two examples of $\mathcal{L}$-reduction are displayed.

The importance of the previous definition is shown by the following result:

Theorem 9. *For every* $x \leq y$ *the equality* $\ell_{(G^{\mathcal{L}}, \varphi^{\mathcal{L}})}(x, y) = \ell_{(G, \varphi)}(x, y)$ *holds.*

Theorem 9 allows to compute the discrete size function of (G, φ) by working on the $\mathcal{L}$-reduced size graph $(G^{\mathcal{L}}, \varphi^{\mathcal{L}})$, which has usually a much simpler structure.

In order to prove such a result we need the following lemma.

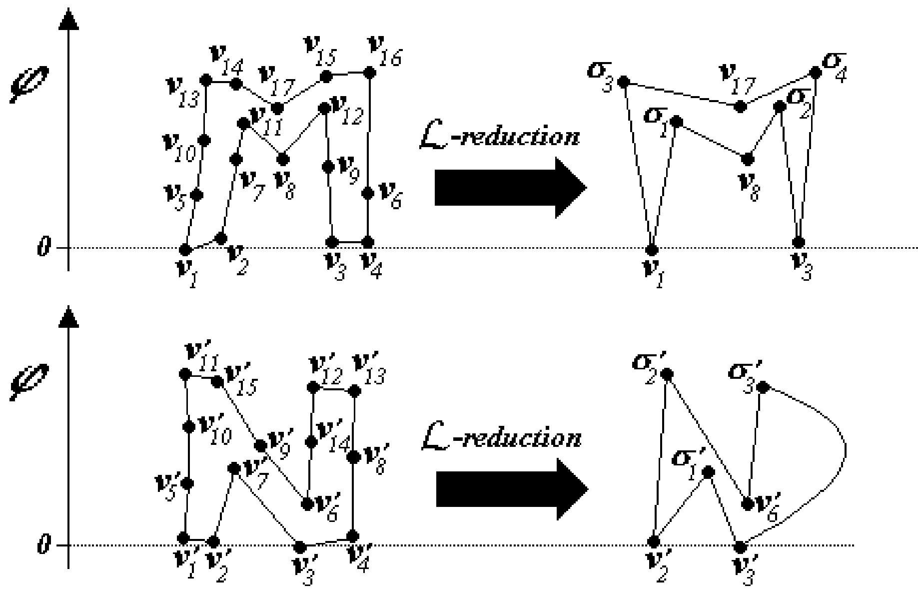

Fig. 2. Two examples of $\mathcal{L}$-reduction.

Lemma 10. *Consider two minimum vertices* $v_a, v_b \in M$. *Then* $v_a \cong_{G\langle\varphi\leq y\rangle} v_b$ *if and only if* $v_a \cong_{G^{\mathcal{L}}\langle\varphi^{\mathcal{L}}\leq y\rangle} v_b$.

Proof. We can confine ourselves to consider only the case $v_a \neq v_b$, since the case $v_a = v_b$ is trivial. Suppose $v_a \cong_{G\langle\varphi\leq y\rangle} v_b$. Then by definition there exists a sequence $(v_{i_1} = v_a, v_{i_2}, \ldots, v_{i_{r-1}}, v_{i_r} = v_b)$ of vertices of $G\langle\varphi \leq y\rangle$ such that for each index $j < r$ the vertex v_{i_j} is adjacent to $v_{i_{j+1}}$ in $G\langle\varphi \leq y\rangle$. Now, consider the sequence of minimum vertices $(\mathcal{L}(v_{i_1}) = v_a, \mathcal{L}(v_{i_2}), \ldots, \mathcal{L}(v_{i_{r-1}}), \mathcal{L}(v_{i_r}) = v_b)$. By substituting each subsequence of consecutive vertices by one representative in such a sequence, we obtain a new sequence $(u_1 = v_a, u_2, \ldots, u_s = v_b)$ (In plain words we mean that the sequence $(w_1, w_1, \ldots, w_1, w_2, w_2 \ldots, w_2, \ldots, w_h, w_h, \ldots, w_h)$ becomes the sequence $(w_1, w_2, \ldots, w_h)$). For each index $j < s$ there exists (exactly) one main saddle σ_j adjacent to u_j and u_{j+1}. Let us take the sequence $(u_1 = v_a, \sigma_1, u_2, \sigma_2, \ldots, u_{s-1}, \sigma_{s-1}, u_s = v_b)$: such a sequence proves that $v_a \cong_{G^{\mathcal{L}}\langle\varphi^{\mathcal{L}}\leq y\rangle} v_b$.

On the other side, suppose that $v_a \cong_{G^{\mathcal{L}}\langle\varphi^{\mathcal{L}}\leq y\rangle} v_b$. Then by definition there exists a sequence $(u_1 = v_a, \sigma_1, u_2, \sigma_2, \ldots, u_{s-1}, \sigma_{s-1}, u_s = v_b)$ of vertices of $G^{\mathcal{L}}\langle\varphi^{\mathcal{L}} \leq y\rangle$ where each u_j is a minimum vertex and each σ_j is a main saddle adjacent to u_j and u_{j+1}. Now, by modifying such a sequence we can construct the following new sequence: for each index $j < s$, between u_j and $\sigma_j = \{v_{i_j}, v_{k_j}\}$ insert the sequence $(L^{m(v_{i_j})-1}(v_{i_j}), L^{m(v_{i_j})-2}(v_{i_j}), \ldots, L^2(v_{i_j}), L(v_{i_j}))$ and between σ_j and u_{j+1} insert the sequence $(L(v_{k_j}), L^2(v_{k_j}), \ldots, L^{m(v_{k_j})-2}(v_{k_j}), L^{m(v_{k_j})-1}(v_{k_j}))$ (we are assuming $\mathcal{L}(v_{i_j}) = u_j$ and $\mathcal{L}(v_{k_j}) = u_{j+1}$). In plain words, we insert the vertices we go through by following the descent flow operator during the path from a main saddle to an adjacent minimum vertex. Finally, we substitute the two vertices v_{i_j}, v_{k_j} (taken in this order) for each main saddle σ_j. The new sequence we obtain proves that $v_a \cong_{G\langle\varphi\le y\rangle} v_b$. □

Now we can give the proof of Theorem 9.

Proof. Consider two values $x, y \in \mathbb{R}$ with $x \le y$. We have only to prove that there exists a bijection F from $G\langle\varphi \le x\rangle / \cong_{G\langle\varphi\le y\rangle}$ to $G^{\mathcal{L}}\langle\varphi^{\mathcal{L}} \le x\rangle / \cong_{G^{\mathcal{L}}\langle\varphi^{\mathcal{L}}\le y\rangle}$. For every equivalence class $C \in G\langle\varphi \le x\rangle / \cong_{G\langle\varphi\le y\rangle}$ we choose a minimum vertex $v \in C$. Obviously v is a vertex of $G^{\mathcal{L}}\langle\varphi^{\mathcal{L}} \le x\rangle$, too. Therefore in $G^{\mathcal{L}}\langle\varphi^{\mathcal{L}} \le x\rangle / \cong_{G^{\mathcal{L}}\langle\varphi^{\mathcal{L}}\le y\rangle}$ there is an equivalence class D containing v. We define $F(C) \stackrel{def}{=} D$. From previous Lemma 10 it follows that F is well defined and injective. Surjectivity of F is trivial, since each equivalence class in $G^{\mathcal{L}}\langle\varphi^{\mathcal{L}} \le x\rangle / \cong_{G^{\mathcal{L}}\langle\varphi^{\mathcal{L}}\le y\rangle}$ contains at least a minimum vertex of $G\langle\varphi \le x\rangle$. □

4 A local method for reducing (G, φ): Δ-reduction

$\mathcal{L}$-reduction is a global method for reducing size graphs, since the construction of main saddles requires the knowledge of all the size graph. Conversely, this section is devoted to a local method, called Δ-reduction, that requires only the knowledge of the local structure of the size graph. The basic idea is quite different from the one used for $\mathcal{L}$-reduction and is given by the following definition.

Definition 11. Assume v_1 is adjacent to v_2 and v_3 is adjacent to v_2 in G (in symbols: $v_1 \sim_G v_2$ and $v_3 \sim_G v_2$). Moreover, assume that $\varphi(v_3) \ge \varphi(v_1) > \varphi(v_2)$. Consider the new graph H obtained from G by erasing the edge connecting v_3 to v_2 and inserting the edge connecting v_3 to v_1 (unless it already exists in G). See Figure 3. We shall say that the size graph (H, φ) has been *obtained from* (G, φ) *by a simple Δ-move.* Every size graph obtained from (G, φ) by applying a finite sequence of simple Δ-moves will be called *a Δ-reduction of* (G, φ). Each Δ-reduction of (G, φ) for which no one Δ-move can be applied will be called *a total Δ-reduction of* (G, φ).

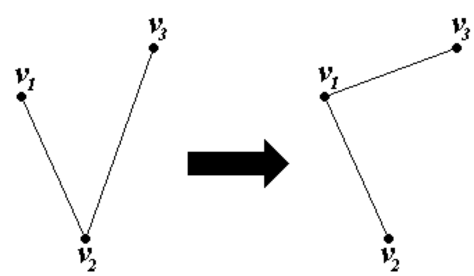

Fig. 3. A simple Δ-move.

First of all we point out the following property:

Lemma 12. *Assume that (H, φ) is a Δ-reduction of the size graph (G, φ). Then $v_a \cong_{G\langle \varphi \leq y \rangle} v_b$ if and only if $v_a \cong_{H\langle \varphi \leq y \rangle} v_b$.*

Proof. It is sufficient to point out that our thesis holds when (H, φ) is obtained from (G, φ) by using only one simple Δ-move. □

From Lemma 12 a useful theorem follows:

Theorem 13. *Assume (H, φ) is a Δ-reduction of (G, φ). Then for every $x \leq y$ the equality $\ell_{(H,\varphi)}(x, y) = \ell_{(G,\varphi)}(x, y)$ holds.*

Proof. It easily follows from the definitions and Lemma 12, much as Theorem 9 follows from Lemma 10. □

It is important to point out that a *a total Δ-reduction of G* always exists, that is the procedure of applying simple Δ-moves cannot proceed indefinitely.

Proposition 14. *A total Δ-reduction of G exists.*

Proof. Consider the complete graph G^* (i.e. the graph containing the edge $e \stackrel{def}{=} \{v_a, v_b\}$ if and only if $v_a, v_b \in V(G)$ and $v_a \neq v_b$). For every edge $e \in E(G^*)$ let us define $p(e)$ as the minimum between the values taken by φ at the vertices connected each other by e. Now, suppose that we can apply an infinite number of simple Δ-moves starting from (G, φ), and call $(G_1, \varphi), (G_2, \varphi), \ldots, (G_r, \varphi)$ the corresponding Δ-reductions. Since each Δ-move cannot make the number of edges greater, a positive integer q exists such that every Δ-move after the q-th one leaves the cardinality of the set $E(G_r)$ of edges of G_r constant. Because of its definition, each Δ-move after the q-th one increases the sum $\sum_{e \in E(G_r)} p(e)$ at least of the positive value $\delta \stackrel{def}{=} \min\{|\varphi(v) - \varphi(w)| : v, w \in V(G), \varphi(v) \neq \varphi(w)\}$, so that the previous sum can become arbitrarily large. This fact contradicts the boundedness of φ and the finiteness of the set E. Therefore our initial assumption is false and there must exist a total Δ-reduction of (G, φ). □

In Figure 4 two examples of total Δ-reduction are displayed.

The most useful property of Δ-reduction is given by the following result, showing that each total Δ-reduction of (G, φ) has a very simple structure. We recall that an *arborescence* is an oriented tree in which there do not exist pairs of edges directed to the same vertex. If (G, φ) is a size graph and G is an arborescence whose edges are all directed downwards with respect to φ, we shall say that such an arborescence is *decreasing*.

Proposition 15. *If φ is injective and (H, φ) is a total Δ-reduction of (G, φ) then H is a disjoint union of trees (i.e. a forest) and the oriented graph $\vec{H}$ is a disjoint union of decreasing arborescences.*

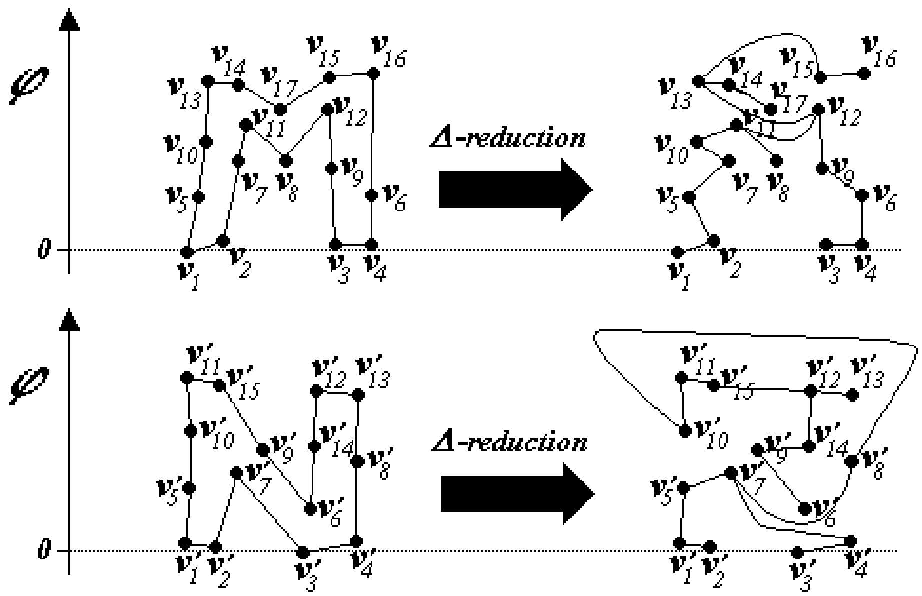

Fig. 4. Two examples of total Δ-reduction.

Proof. It follows immediately from the definition of Δ-reduction, implying that $\boldsymbol{H}$ cannot contain cycles. □

Obviously, the procedure of Δ-reduction is not unique, in the sense that at each step we have to choose the simple Δ-move to be applied. Hence a problem naturally arises about the relation between the total Δ-reductions of a size graph. As an example we refer to following Figure 5, where we can see that in general total Δ-reductions are neither equal nor isomorphic.

On the other hand, next Theorem 17 shows that under a reasonable hypothesis, uniqueness of total Δ-reductions can be proved. Before showing that, we need the following Definition 16.

Definition 16. A sequence $(w_1, w_2, \ldots, w_r)$ of vertices of G will be said to be an increasing path from w_1 to w_r in (G, φ) if and only if $w_i \sim_G w_{i+1}$ and $\varphi(w_i) < \varphi(w_{i+1})$ for $1 \leq i \leq r-1$.

Theorem 17. *If φ is injective then all total Δ-reductions coincide.*

Proof. Assume that (H, φ) and (K, φ) are two total Δ-reductions of the size graph (G, φ). We have to prove that $H = K$. Because of the definition of simple Δ-move the graphs H and K have the same vertices. Hence we have only to prove that two vertices are adjacent in H if and only if they are adjacent in K. So, suppose $v_a \sim_H v_b$, and assume $\varphi(v_b) > \varphi(v_a)$. We shall prove that $v_a \sim_K v_b$.

First of all, we have $v_a \cong_{H\langle\varphi \leq \varphi(v_b)\rangle} v_b$ and hence Lemma 12 implies that $v_a \cong_{K\langle\varphi \leq \varphi(v_b)\rangle} v_b$. Now, since each connected component of $\vec{K}$ is a decreasing

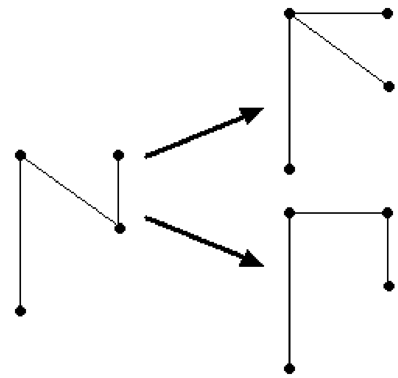

Fig. 5. Two non-isomorphic total Δ-reductions of the same size graph. We point out that φ is *not* injective.

arborescence (see Proposition 15), it follows that an increasing path exists from v_a to v_b in (K, φ): $(w_1 = v_a, w_2, \ldots, w_{r-1}, w_r = v_b)$. We want to prove that $w_2 = v_b$. Since $v_a \cong_{K\langle\varphi \leq \varphi(w_2)\rangle} w_2$ (in fact $v_a \sim_K w_2$ and $\varphi(v_a) < \varphi(w_2)$), by Lemma 12 we have that $v_a \cong_{H\langle\varphi \leq \varphi(w_2)\rangle} w_2$. But each connected component of $\vec{H}$ is a decreasing arborescence and hence an increasing path exists from v_a to w_2 in (H, φ): $(u_1 = v_a, u_2, \ldots, u_{s-1}, u_s = w_2)$. By recalling that $v_a \sim_H v_b$, $\varphi(v_a) < \varphi(w_2) \leq \varphi(v_b)$ and since each connected component of $\vec{H}$ is a decreasing arborescence, the equalities $s = 2$ and $w_2 = v_b$ must hold, and so $v_a \sim_K v_b$ follows from $v_a \sim_K w_2$.

Analogously, $v_a \sim_K v_b$ implies $v_a \sim_H v_b$. □

Remark. It is important to point out that the ways to obtain the same unique total Δ-reduction can be quite different from each other (see example displayed in Figure 6).

5 Comparison of $\mathcal{L}$-reduction and Δ-reduction. Examples.

In this section we shall compare $\mathcal{L}$-reduction and Δ-reduction from a computational point of view. The natural question is the following one: which is the best way to simplify a size graph? As regards the two methods we give, the answer depends on the graph we are studying. As an example, consider the size graph displayed on the left of Figure 7. Such a graph cannot be changed by Δ-reduction, while $\mathcal{L}$-reduction produces the simpler graph on the right (the same phenomenon happens for every disjoint union of decreasing arborescences). On the other side, the size graph displayed on the left of Figure 8 cannot be simplified by using a $\mathcal{L}$-reduction, while Δ-reduction allows to obtain the simpler graph

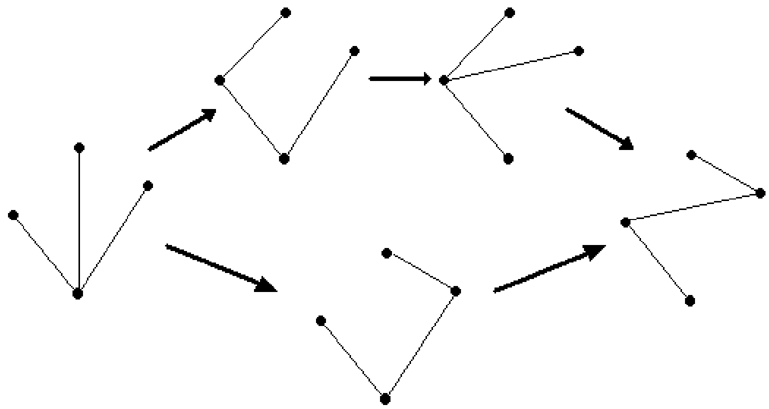

Fig. 6. Two different ways to obtain the same total Δ-reduction.

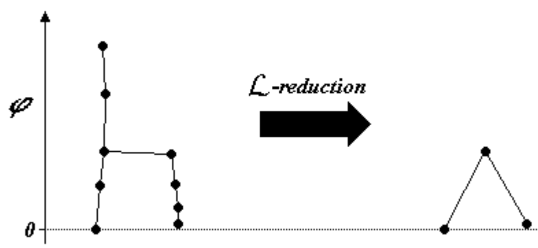

Fig. 7. Δ-reduction is useless for reducing the size graph on the left. $\mathcal{L}$-reduction gives the size graph on the right.

on the right. In general a combined use of both such techniques is suggested, but a theoretical study of this problem is still to be addressed.

Remark. We point out that in some cases the use of $\mathcal{L}$-reduction may make the size graph we are studying worse (see Figure 9). On the other side, the main lack of Δ-reduction is that in general we have to apply an unknown and large number of Δ-moves before obtaining the total Δ-reduction.

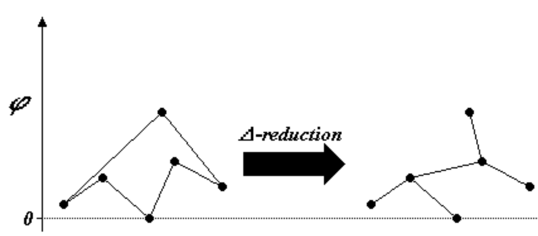

Fig. 8. $\mathcal{L}$-reduction is useless for reducing the size graph on the left. Δ-reduction gives the size graph on the right.

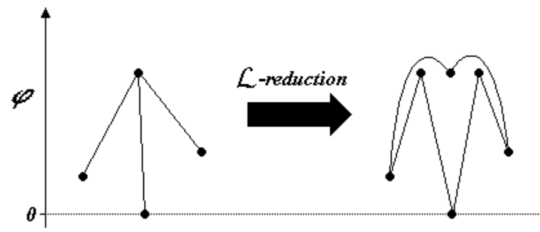

Fig. 9. Sometimes $\mathcal{L}$-reduction may make the size graph we are studying worse.

Finally, we give the next two useful results. With a slight abuse of notation, we shall call $M(G)$ the set of minimum vertices of (G, φ).

Proposition 18. *Assume* $(G^{\mathcal{L}}, \varphi^{\mathcal{L}})$ *is the* $\mathcal{L}$*-reduction of* (G, φ)*. Then we have* $M(G^{\mathcal{L}}) = M(G)$, $|E(G^{\mathcal{L}})| \leq |M(G)| \cdot (|M(G)| - 1)$ *and* $|V(G^{\mathcal{L}})| \leq \frac{|M(G)| \cdot (|M(G)| + 1)}{2}$.

Proof. Trivial.

Example in Figure 9 shows that previous statement is sharp.

Proposition 19. *Assume* (H, φ) *is a total* Δ*-reduction of* (G, φ)*. Then we have* $M(H) = M(G)$, $V(H) = V(G)$ *and* $|E(H)| \leq |E(G)|$*. Furthermore, if* φ *is injective we have* $|E(H)| = |V(G)| - h$*, where* h *is the number of connected components of* G*. In case* G *is a tree (or a forest) we have* $|E(H)| = |E(G)|$.

Proof. The first three statements are trivial. The last two follow by observing that $|E(K)| = |V(K)| - h$ for every graph K which is the disjoint union of h trees, and that each Δ-move takes a tree into a tree without changing the number of edges.

Acknowledgements. The authors wish to thank M. Ferri for his helpful advice. Anyway, the authors are the sole responsible for any mistakes.

References

1. Ferri, M., Lombardini, S., and Pallotti, C.: *Leukocyte classification by size functions*, Proceedings of the second IEEE Workshop on Applications of Computer Vision, IEEE Computer Society Press, Los Alamitos, CA (1994), 223–229.
2. Frosini, P.: *A distance for similarity classes of submanifolds of a Euclidean space*, Bull. Austral. Math. Soc. **42**, 3 (1990), 407–416.
3. Frosini, P.: *Discrete computation of size functions*, J. Combin. Inform. System Sci. **17**, 3-4 (1992), 232–250.
4. Frosini, P.: *Connections between size functions and critical points*, Math. Meth. Appl. Sci., **19** (1996), 555-569.
5. Frosini, P.: *Measuring shapes by size functions*, Proc. of SPIE, Intelligent Robots and Computer Vision X: Algorithms and Techniques, Boston, MA **1607** (1991), 122–133.

6. Frosini, P.: *Natural size distances and size functions for studying shapes of manifolds*, (to appear).
7. Frosini, P.: *Size Theory as a Topological Tool for Computer Vision*, (to appear).
8. Harary, F.: *Graph Theory*, Addison-Wesley, 1969.
9. Uras, C., Verri, A.: *Studying shape through size functions*, Shape in Picture, NATO ASI Series F **126**, Y. O, A. Toet, D. Foster, H. Heijmans, P. Meer (editors), Springer-Verlag, Berlin Heidelberg (1994), 81–90.
10. Uras, C., Verri, A.: *On the recognition of the alphabet of the sign language through size functions*, Proc. XII IAPR International Conference on Pattern Recognition, Jerusalem (Israel) **II**, IEEE Computer Society Press, Los Alamitos, CA (1994), 334–338.
11. Verri, A., Uras, C.: *Invariant size functions*, Applications of invariance in computer vision. Lecture Notes in Computer Science **825**, J. L. Mundy, A. Zisserman, D. Forsyth (editors), Springer-Verlag, Berlin Heidelberg (1994), 215–234.
12. Verri, A., Uras, C.: *Metric-topological approach to shape representation and recognition*, Image Vision Comput., **14** (1996), 189-207.
13. Verri, A., Uras, C.: *Computing size functions from edge maps*, Internat. J. Comput. Vision (to appear).
14. Verri, A., Uras, C., Frosini, P., Ferri, M.: *On the use of size functions for shape analysis*, Biol. Cybern. **70** (1993), 99–107.