HOMOGENIZATION OF FACADE STRUCTURES

R. Guercke[a, *], M. Sester[b]

[a] Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany - guercke@ipi.uni-hannover.de
[b] Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany - monika.sester@ikg.uni-hannover.de

KEY WORDS: Generalization, 3D Models, Homogenization, Facade Structures

ABSTRACT:

In this paper, we introduce the generalization operator of *homogenization* as the replacement of k different objects by k references to a single template object. This approach can reduce the size of the data set and the complexity of location- and orientation-invariant processing tasks consisiderably because those tasks only have to be applied to the template object instead of all original objects. The task is to identify similar objects that can be replaced by a common template, i.e. be treated as identical. As exact identity of the objects' parameters can only be enforced by changing parameters, which, in turn, incurs costs in terms of a generalization objective, this leads to an optimization problem. As an example, we investigate the homogenization of objects within rectangular blocks of cells in a grid-like facade layout as an optimization problem and derive a Mixed Integer Pramming (MIP) representation for this problem. We present results of solving the resulting MIP problems for a set of real-world facade structures.

1. INTRODUCTION

We define the generalization operator of *homogenization* (we use homogenization as a preliminary name; finding a perhaps more fitting and less ambiguous name may be the subject of further discussion) as the replacement of k different objects (instances of a class with a set of parameters) by k references to a template object.

Even though we do not reduce the number of objects or parameters per object in the scene, the total amount of data to be stored and processed can be reduced considerably in a digital environment by this operator because the set of parameter values has to be stored only once for the template object, while the references need to store only the ID of the template object and a geometric transformation that puts them in the right position and orientation in space.

In addition to this reduction in memory footprint, location- and orientation-invariant processing tasks like volume calculation, basic geometric simplification approaches or force resistance parameters for different angles of impact have to be performed only once on the template object instead of all instances. Conversely, this also means that we can treat objects at a much more fine-grained level with a reduced level of prcessing requirements: We can process thousands of instances of a template object at once by processing the template, so instead of processing coarse models for each instance of, say, the IKEA Billy shelf (appearing quite frequently in central Europe) we can process one high-quality instance (with every nut and bolt) once and use the result for all instances, at once saving effort and *increasing the quality of the result*.

Note that homogenization can be applied to objects of any given class Whole 3D building models, 2D shapes on maps or even individual objects within larger ones like shelves, power plugs or door handles within (and across!) buildings.

In order to perform homogenization, we have to find a set of parameter values for the template object on which all original objects can agree – meaning that their parameters can be replaced by the common value without exceeding the postulated generalization accuracy tolerance (e.g. in a Hausdorff sense).

Further opportunities for reductions of storage space and complexity as well as additional options for further simplifications occur if the objects are distributed in a regular structure. In this paper, for example, we consider facade openings (windows and doors) arranged in a logical grid of rectangular cells.

In such a setting, we only have to store the row and column index for the different references while all other parameters like the width and height of the cell, the location of the opening in the cell and the dimensions of the opening only have to be specified for the template object.

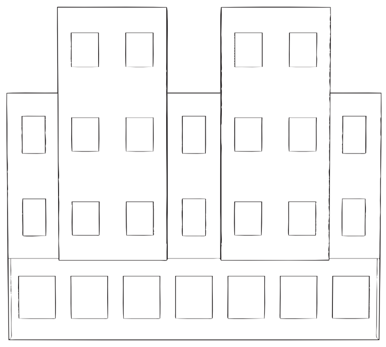
Taking this one step further, if we know that all cells within a rectangular region within a grid structure are covered by instances of the same template, we only need to store the indices of the start and end rows and columns of the rectangular region instead of the template ID, row and column indices for all cells.

We refer to blocks of cells that are replaced by the same template as *clusters* or *tiles*. An assignment of all cells in a grid to a set of clusters is called a tiling of the structure. Note that we use the term cluster also for groups of objects that are replaced by the same template if the distribution does not happen to be a regular grid.

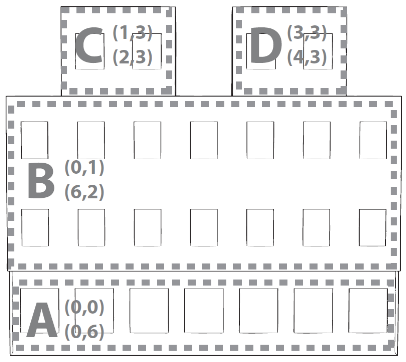
The concept of homogenization can be extended by introducing exception or difference parameters that may be specified for different references to the same template object. This makes the concept considerably more flexible and versatile. If, for example, we have a large rectangular facade of a shopping center with 5 floors with 20 almost (within the generalization accuracy) identical windows in each and a single door in the 5^{th} column of the ground floor, we can represent this as a 20x5 grid of instances of the template window object with an exception in cell (0, 4) which contains our door object.

The compactness of such a description makes it also very suitable for multiple-LoD data sets: Just by dropping the exception, simplifying the template and employing typification, we can easily derive a low-LoD version of our model. Intermediate LoDs can

*Corresponding author.



(a) Original facade.



(b) A tiling with four tiles.

Figure 1: A tiling for a facade structure.

be produced by independently simplifying the individual door element and the template element (and optionally the distribution, e.g. through a typification) of the windows.

In the second part of this paper, an algorithm is presented for an automatic clustering of grid-structured facades into rectangular clusters because the compactness and regularity of the rectangular shape of the resulting homogenized blocks ensures that further generalization operators like typfication can be employed with maximum ease and effect at later generalization stages.

As an introductory example, consider Figure 1. It shows the result of homogenization, where mainly the depth differences between the different parts of the facades have been aligned, leading to a large, homogeneous main facade.

The combinatoric optimization problem underlying this setting consists of the decision which original objects should be clustered, i.e. replaced by the same template. The objective function is a linear combination of the number of resulting regions and the sum-of-squares of the changes in the parameter values for all objects. In (Guercke, 2014), this problem is shown to be NP-hard even if there is only one parameter for each cell.

In the following section, we first give a more detailed definition of the homogenization operator and compare it to other generalization operators, followed by a review of the Mixed Interger Programming approach introduced in (Guercke, 2014) for regular facade structures. In section 4, examples show the potential of the method for facade structures of different complexity. A summary and outlook on future work conclude the paper.

2. HOMOGENIZATION AS A GENERALIZATION OPERATOR

In this section, we give a more detailed definition of the homogenization operator and show the differences to the similar classical generalization operators.

At the end of this section, we put the contents of this paper into the context of the current state of research on generalization operators and building model generalization.

We assume that the digital model is given as a collection of objects of semantic classes like buildings, roads, roofs etc., where each class has a number of properties that assume different values for the different instances. This model can be hierarchical, meaning that objects may contain other objects: A facade object, for example, may contain several window, door, and balcony objects.

Homogenization is the replacement of several objects of the same class with potentially different parameter values (for example, different heights) by instances of a template with fixed values for all parameters. This means that for all objects that are to be replaced by the same template, a common value for all parameters has to be found.

In the following, we compare this operator to the classical cartographic generalization operators described, for example, in (Hake et al., 1996):

Simplification An object is represented by a simplified version of itself.

Enlargement, Exaggeration – In many cases, objects are too small to be recognizable at the target scale andhaveto be enlarged; objects of higher importance may be exaggerated beyond the necessity of being visible in order to make them more prominent.

Displacement – In order to avoid cluttering the resulting map, objects may have to be shifted away from their original position.

Aggregation – Objects of the same class are merged into one bigger object of the same class.

Selection of relevant or elimination of less relevant features to avoid cluttering the resulting map.

Classification, Symbolization – Objects of more special classes are treated as objects of more general classes and may be represented by a symbol for this more general class.

Typification – n objects of similar classes are represented by $k < n$ objects of the same or a more general class.

Homogenization is not a simplification of a given object in a strict cartographic sense because the complexity of the object itself, i.e. the complexity of its shape and the number of parameters describing a given object, remains unchanged.

This is the reason why homogenization is a *digital* generalization tool: While it offers huge potential for reduction of complexity in a digital setting, the resulting objects on a rendered map have the same complexity as before, and the effects are mainly aesthetic (giving a more regular impression) if no further generalization steps are employed.

An object is not explicitly enlarged by a homogenization step – some object may, of course, be enlarged, but others will be smaller afterwards. Neither are objects displaced with the purpose of a cleaner distribution of the objects on a map.

After homogenization, all objects that are replaced by references to the same template still remain distinct entities, while aggregated objects are merged into a single larger object.

Homogenization is neither selection nor typification because the number of objects is not reduced, but it can be employed before a typification step to identify sets of features with similar properties.

The pattern of homogenization before typification can be especially useful for multiscale datasets: Homogenization (especially with explicit handling of exceptions) can be employed to produce the multiscale data set from which simplified models can be derived by employing typification to the different homogenized regions in the original data set.

Since homogenization itslef does not imply a change of the class of an object, it is not a classification. If we model the class of an object as a parameter of the homogenization, class changes may occur, but they are inherent to the concept. We may also perform a classification before the homogenization in order to avoid having to explicitly model class changes if the specific classes of the objects are not of major importane to the application at hand.

Including semantic aspects into a homogenization implementation can be achieved by introducing label parameters for the class of the features. In this case, the mapping from the different parameters of the classes involved to the parameters to be homogenized has to be known. In the simplest case, the homogenization is performed on the parameters of a common superclass of all classes involved (plus a parameter for the class label). Then we can define penalties for class label changes due to the homogenization (i.e. assigning a common class label to all instances in a cluster) or forbid certain class label changes.

Note that the operator of homogenization is independent of the spatial distribution of the features to be homogenized; this spatial distribution may be chosen according to the problem at hand. This distribution may be known to follow a certain pattern (like the regular facade structures in the example or buildings along a road) or be inferred statistically from the original distribution to maintain maximum similarity with the original.

The question if homogenization has to be regarded a generalization operator can only be decided if a very precise definition is agreed upon within the community of what a generalization operator is – which is not the case yet (as far as the authors are aware). Especially a distinction is needed between a generalization *operator* and the additional aspects a genralization *algorithm* has to take into account that uses or implements an operator (or more than one operator).

Some of the most important of these aspects are (among many others): How to partition the data set (either to apply different operators to different subsets or to cluster the objects in a subset, e.g. in the context of aggregation or homogenization – this, if it is necessary, is often the computationally hardest problem), how to simplify the spatial distribution of the objects, and how to reassemble composite objects from simplified part objects in a hierarchical model.

In (Anders, 2005), an approach for the typification of regular structures in spatial data sets is described for road patterns and facade structures.

(Ripperda, 2008) uses a sophisticated facade model based on shape grammars to generate model proposals for the reconstruction of facade structures in an approach based on reversible jump Markov chain Monte Carlo (rjMCMC) simulation.

(Sester and Klein, 1999) propose a rule based simplification of facades for different Levels of Detail, based on a semantic building model. In (Fan et al., 2009), an approach for the generalization of models in the CityGML data format (Kolbe et al., 2005) is decribed. In this paper, simplification approaches for the geometry of the facades and indiviual windows are presented as well as a user survey on different approaches for the layout of objects in a regular grid after typification.

3. A MIXED INTEGER PROGRAMMING REPRESENTATION FOR THE FACADE STRUCTURE HOMOGENIZATION PROBLEM

After having introduced the operator of homogenization, we present, as a second contribution of this paper, an approach for a specific homogenization scenario: the homogenization of regular facade structures.

In this scenario, the task is to partition a facade structure consisting of rectungular cells arranged in a regular grid into a set of clusters in which all facade elements are replaced by the same template object.

We also require the clusters themselves to be rectangles in order to produce results with maximum regularity that are best suited for subsequent typification steps. Although, on the first glance, this seems to simplify the problem, this rectangularity constraint turns out to make the problem *much* harder. While it can be incorporated in the MIP-based approach through a few additional constraints as shown in this section, it makes the design of alternative probabilistic approaches based on Monte Carlo techniques extremely difficult (at least the authors of this paper could not develop a convincing one so far) because the notion of similar solutions or a neighborhood of solutions which is critical for Monte Carlo methods is most difficult to define for rectangular tilings.

We state this problem as an optimization problem defined by a set of constraints and an objective function. The degree of generalization is controlled by thresholds imposed on the change of the parameter values through hard constraints; the connectedness and rectangularity of the clusters is also ensured through hard constraints.

The objective function (in our definition, to be minimized) is a weighted sum of two groups of objective terms

1. the total number of clusters and
2. a sum of squares (weighted by an impact factor for the parameter) of the parameter changes.

In our setting, we assign a large weight to the number of clusters and a lower weight to the parameter changes, so tha minimization of the number of clusters takes precedence over the minimization of the parameter changes. Note that the combination of the minimzation of the number of clusters and the thresholds on the parameter changes define the core of the optimization problem, while the second objective chooses the “geometrically most similar” of the minimum-size tilings and causes the parameters within the clusters to be set to the mean of the original values (if the thresholds allow this).

The approach introduced in this paper is similar to the graph-based approaches used in (Haunert, 2009) for the aggregation of land parcels in map generalization and in (Guercke et al., 2011) for the aggregation of LoD 1 building models.

As in these cases, one member is defined as the representative (also referred to as *center* in (Haunert, 2009) for each cluster in the result. In the case of our facade homogenization problem, this means that one cell will be designated as the representative of a given tile. Note that this representative does not have to be a kind of geometric center of the tile as the term *center* suggests – in fact, we will define the center to be the lower left corner of each tile to avoid ambiguities.

We define a set of variables $X_{(a,b),(i,j)}$ that are TRUE if a cell (i,j) is assigned to center (a,b). In our case, the underlying neighborhood graph is implicitly given by the logical grid structure of the facade.

Because the center has no other function than to represent the tile, the constraints defined in the rest of this section are designed in such a way that the center of each tile is the lower left cell in the tile. This definition has the advantage that it reduces the ambiguity of optimal solutions (and therefore the number of possible solutions the solver has to explore) considerably and that it simplifies the constraints for ensuring the rectangular shape of the clusters significantly. Additionally, it saves variables because the variable $X_{(a,b),(i,j)}$ only has to be defined for $i \geq a$ and $j \geq b$.

For this reason, iterating over all variables would be denoted in the form: $\forall(a,b) \in \text{Cells}, \forall i \in \{a, \ldots, N_R - 1\}, \forall j \in \{b, \ldots, N_C - 1\})$. As a short form for this expression we write: $\forall(a,b),(i \geq a, j \geq b) \in \text{Cells}$, meaning that i and j will assume all valid values in the given array greater than a or b, respectively.

In the following, the constraints and objectives of this optimization problem are described. We will number constraints and objectives separately, so, for example, C2 is the second constraint and O1 the first objective term (the whole objective function is a weighted sum of the objective terms).

C 1 Cells can only be assigned to centers:

The first constraint states that cells can only be assigned to centers where a center is a cell that is assigned to itself:

$$\forall(a,b),(i \geq a, j \geq b) \in \text{Cells} : X_{(a,b),(i,j)} \leq X_{(a,b),(a,b)}.$$

Figure 2 shows the $X_{(a,b)(i,j)}$ variables that were assigned the value TRUE for a tiling of our illustration facade. The black lines connect the cells in the tiles to their centers; the circular lines at the centers illustrate that for each center $c = (a,b)$, the variable $X_{c,c}$ is TRUE. Constraint 3. ensures that a cell can only point to a center: Each black line terminates in a center in figure 2.

C 2 A cell is assigned to exactly one center:
Additionally, all cells will belong to exactly one tile (center) if (i,j) is part of the facade and to none if it is a background cell:

$$\forall(i,j) \in \text{Cells} : \sum_{a=0}^{i}\sum_{b=0}^{j} X_{(a,b),(i,j)} = 1 - \text{BG}(i,j),$$

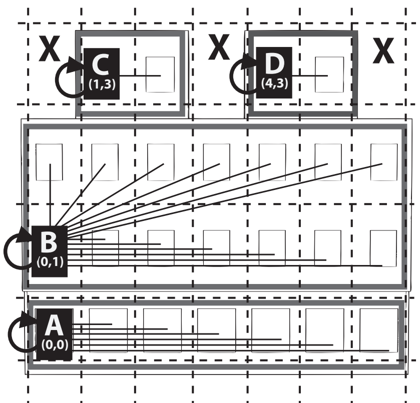


Figure 2: Visualization of the TRUE $X_{(a,b)(i,j)}$ variables for a tiling of a facade.

where $\text{BG}(i,j)$ is TRUE(=1) if (i,j) is a background cell and FALSE(=0) if not. The black crosses in the unoccupied cells in figure 2 illustrate that none of the X variables is set to TRUE for the background cells. For the other cells, this constraint makes sure that only one of the outgoing X variables is set: there is only one black line from each occupied cell in figure 2.

O 1 Now we can define the *objective* of using a minimum number of tiles:

$$\text{MIN} \quad \left(W_{\#f} \sum_{(a,b)\in\text{Cells}} X_{(a,b),(a,b)} \right),$$

where $W_{\#f}$ is a constant factor that expresses the relative weight of the number of tiles compared to the other objective terms. The sum counts the number of centers because a center is defined as a cell assigned to itself.

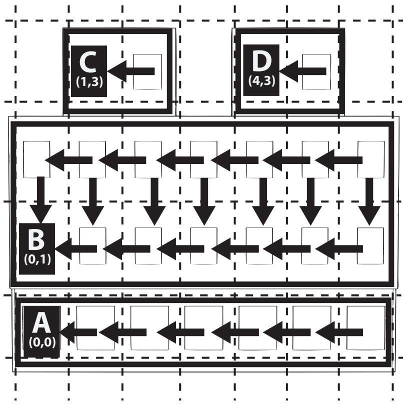


Figure 3: Constraint set 3. for the tiling in figure 2.

C 3 Cells in rectangle between a cell and its associated center must also be assigned to the center:

The following constraint set ensures that the center of a tile will be its lower left corner and that the rectangle between a given tile and its associated center must be part of the tile defined by the center:

$$\forall (a,b),(i \geq a, j \geq b) \in \text{Cells}:$$
$$X_{(a,b),(i,j)} \leq X_{(a,b),(i-1,j)} \quad \text{if } i > a$$
$$X_{(a,b),(i,j)} \leq X_{(a,b),(i,j-1)} \quad \text{if } j > b.$$

A cell (i,j) can only be assigned to center (a,b) if its predecessors in x (second constraint) and y (first constraint) direction are also assigned to the same center. Note that if cell (i,j) is already in the same row or column as the center (a,b), the corresponding predecessor must not be forced to be assigned to the same center. For this reason, the constraints for the predecessors in y direction are only defined if $i > a$ and the constraints for the predecessors in x direction are only defined if $j > b$.

The transitive effect of these constraints on the predecessors ensures that for all tiles (i,j) all cells in the rectangle spanned by (a,b) and (i,j) must be assigned to (a,b) if (i,j) is assigned to (a,b).

Figure 3 illustrates this effect for a valid tiling: The horizontal arrows show the implications enforced by the first constraint (in x direction), the vertical arrows the implications in the y direction. Since the center is the bottom left cell in the tile, all arrows have to converge towards the center and no predecessor of a cell in the tile can be left out.

This does, however, not yet ensure that each tile will be a rectangle.

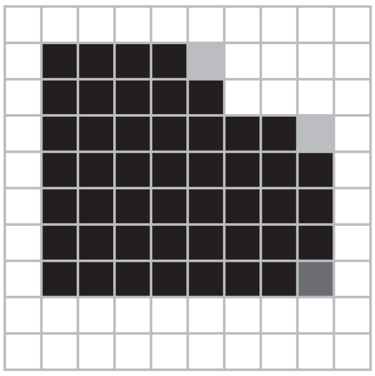


Figure 4: Valid non-rectangular tile under constraint 3..

Figure 4 shows that in constraint 3., there is a "loophole" that allows non-rectangular tiles: In the upper right corners, fragments may be cut out of the rectangle without violating the constraint. The light gray cells that define the two basic rectangles of the tile shown in the figure may also be left out; the dark gray cell in the bottom line, for example, must, however, be part of the tile because it is "covered" by the cell above it.

C 4 Close the rectangle to upper right:
In order to ensure that the tiles are rectangles, we define a constraint that works in the opposite direction compared to constraint 3.:

$$\forall (a,b),(i > a, j > b) \in \text{Cells}:$$
$$X_{(a,b),(i-1,j)} \wedge X_{(a,b),(i,j-1)} \Rightarrow X_{(a,b),(i,j)}$$
$$\longrightarrow \; X_{(a,b),(i,j)} \geq X_{(a,b),(i-1,j)} + X_{(a,b),(i,j-1)} - 1.$$

This closes the rectangle in the direction of its upper right corner: If both predecessors of a cell in x and y direction point to the same center, the cell itself has to point to this center as well. Note the ">" in the $\forall$-clause: This constraint will only concern cells that are located neither on the same row nor on the same column of a given possible center.

This constraint only works in context with constraint 3. because it only performs the closing in the upper right direction. A more generic implementation in which the center may be an arbitrary member of the cluster would require a more sophisticated set of constraints.

In the linear form given in the second row, the right side of the constraint is 1 if both predecessors point to the same center (a,b), so $X_{(a,b),(i,j)}$ has to be TRUE as well in this case. Otherwise, the right side is 0 or -1, so the constraint is relaxed.

The parameters are handled by assigning tile parameter values to the centers of the tiles. Since we do not know which cells are going to be centers, there have to be parameter placeholders $V_{(i,j),p}$ for each parameter p for all cells (i,j). For cells that are not centers, we force this variable to equal the value of the center and use it to calculate the difference $D_{(i,j),p}$ between the original value $v_{(i,j),p}$ of parameter p and the new value $V_{(i,j),p}$ defined by the covering tile for each cell (i,j).

C 5 Ensure homogeneous parameter values within the clusters:
First, we make sure that all cells in a tile have the same value for each parameter:

$$\forall (a,b),(i > a, j > b) \in \text{Cells}, \forall p \in P:$$
$$V_{(i,j),p} \geq V_{(a,b),p} - M_p \left(1 - X_{(a,b),(i,j)}\right)$$
$$V_{(i,j),p} \leq V_{(a,b),p} + M_p \left(1 - X_{(a,b),(i,j)}\right),$$

where M_p is an adequate "big-M" value for parameter p, for example the maximum difference between two values of p in the original facade data set. The purpose of the $M_p(1 - X_{(a,b),(i,j)})$ term is to relax the constraint for all centers to which p is not assigned (otherwise, the value of p would have to be same in *all* cells). For this reason, M_p is defined to be greater than any possible difference between $V_{(i,j),p}$ and $V_{(a,b),p}$, and $M_p(1 - X_{(a,b),(i,j)})$ evaluates to M_p if (i,j) is not assigned to (a,b).

This means that the first constraint becomes $V_{(i,j),p} \geq V_{(a,b),p} - M_p$ which will be satified for all sensible values because $V_{(a,b),p}$ and $V_{(i,j),p}$ will not leave the interval in which the original paramters were distributed: With $V_{(a,b),p}$ in the original interval, $V_{(a,b),p} - M_p$ will be smaller than (or equal to) *any* value from the valid interval which we may assign to $V_{(i,j),p}$, so the constraint is relaxed (fulfilled for any sensible value of $V_{(i,j),p}$) if (i,j) is not assigned to the center (a,b). By the same argument, the second $\leq$ part of the constraint is relaxed if (i,j) is not assigned to the center (a,b).

If (i,j) is assigned to (a,b), then the term $M_p \left(1 - X_{(a,b),(i,j)}\right)$ becomes 0 and the two inequalities can be merged to the equation $V_{(i,j),p} = V_{(a,b),p}$. So all cells within a tile have the same values for each parameter as the center which means, by the transitivity of the "=" relation, that all cells have the same value.

C 6 Introduce variable $D_{(i,j),p}$ for the change of parameter value due to homogenization:

Now we set the variables $D_{(i,j),p}$ to hold the (absolute value of the) difference between the value $V_{(i,j),p}$ of parameter p

in the tile and the original value $v_{(i,j),p}$ of p in cell (i,j). Since we will use the variables $D_{(i,j),p}$ only in contexts where a minimum value is desired, we do not need to define constraints that enforce an upper bound on $|D_{(i,j),p}|$:

$$\forall (i,j) \in \text{Cells}, \forall p \in P : D_{(i,j),p} \geq \left|V_{(i,j),p} - v_{(i,j),p}\right|$$

In order to capture the *absolute* value of $D_{(i,j),p}$, we split this relation into two inequalities for values greater or less than zero:

$$D_{(i,j),p} \geq V_{(i,j),p} - v_{(i,j),p}$$
$$D_{(i,j),p} \geq -V_{(i,j),p} + v_{(i,j),p},$$

We introdude the variables $D_{(i,j),p}$ because their value is used both in constraint C7 with a hard threshold to define the desired level of generalization and in objective O2 where it is used to make the solver determine the values $V_{(i,j),p}$ to be the mean of the original values $v_{(i,j),p}$ of all cells in the same cluster (if the hard threshold C7 allows it).

C 7 Threshold for $D_{(i,j),p}$ (degree of generalization):

Now we can easily implement the constraint of bounding the homogenization error for each parameter in all cells:

$$\forall (i,j) \in \text{Cells}, \forall p \in P : D_{(i,j),p} \leq \Delta_p$$

O 2 Minimize sum of squares of all $D_{(i,j),p}$:

Finally, we add the penalties for the parameter errors to the objective function:

$$\text{MIN} \left(\sum_{(i,j) \in Cells} \sum_{p \in P} W_p \, D^2_{(i,j),p}. \right)$$

This final objective term ensures that the resulting value for each parameter p for a given tile is the *average* of the values in the cells of the tile – if the difference threshold allows the parameter to assume this value.

Had we used the sum of the oridinary (non-squared) parameter differences in this objective, the result would have been the median of the values instead of the mean – which, in a typical adjustment setting, is what we would expect. The squared values in the objective function make the whole problem considerably harder to solve; furtunately, the current version of CPLEX could handle this problem for the instances we used in our tests.

Note that is easy (while not actually done in the experiments) to introduce semantic aspects into the problem using a similar mechanism by defining numeric variables that encode the class of the object in the tile, for expamle 0 for WINDOW, 1 for DOOR, and 2 for ORNAMENT. Of course, we would, in this case, not use constraint C7 and objective O2 in the same way as for continuous parameter variables.

Instead, we would define mutual exclusion tables for the semantic classes and introducde constraints that forbid cells with mutually exclusive classes to appear in the same cluster (semantic class equivalent of hard constraint C7). Additionally, we can define a semantic objective that minimizes the sum of class label change costs (defined in a between-classes transition cost table) corresponding to a soft constraint or objective term similar to objective term O2 penalizing semantic class change.

4. EXAMPLES AND RESULTS

The effects of the homogenization process are illustrated with the help of examples in this section. We can see that the generalization process provides visually convincing results for simple as well as non-trivial setups.

For all experiments, we use the commercial MIP solver CPLEX (IBM ILOG, 2011) which can be used without fee in academic contexts because it offers high performance and could solve all problem instances of our test problems (including the quadratic versions).

Figure 1 shows an example of a small facade part modeled after a facade from a typical townhouse. In this simple case, the protrusions were pushed back into the main facade plane and originally slightly different measures of the windows were set to identical values.

Because tiles are supposed to be rectangular, the windows in the gables of the dormers on top of the protrusions are not part of the main tile covering the upper floors. For this reason, they are left in their initial protruding position because changing the depth of a tile incurs a penalty in the optimization function.

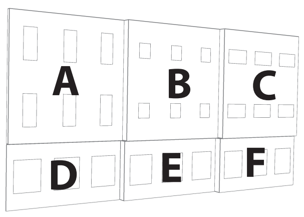
In (Guercke, 2014), the impact of different weights for different parameters is discussed at length for an example. Figure 5 shows the different homogenization results for the case in which the difference in window height is considered more harmful (and therefore receives a higher penalty) than the difference in window width (Fig. 5(b): $W_{height} > W_{width}$ in objective O2), and vice versa (Fig. 5(c): $W_{height} < W_{width}$).

Note that the simplification of the top and bottom rows are independent in this example, and the width and height parameters as well as the depth differences between regions D and E and regions E and F are equal. For this reason, it can happen that the mergings of the regions in the top and bottom rows are not aligned. In the case shown in figure 5, one would, for example, intuitively have merged regions E and F rather than regions D and E in order to achieve a more homogeneous appearance.

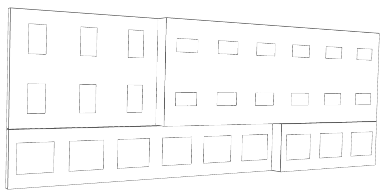
Another issue is the fact that small differences in the depth of cells that are not part of the same homogenization tile are preserved. If we brought them to the same depth, we could save the wall surfaces that are needed to connect the tiles in order to close the facade. For this reason, it makes sense to add a term to the objective function that rewards bringing adjacent tiles to the same depth. The perspective view in figure 5 shows the different depths of the facade parts and how they are changed within the homogenization process.

Figure 6 shows a complex facade structure covering one side of a block of townhouses in Hanover (Schneiderberg). The figure shows a screenshot of a part of a point cloud acquired using the Riegl VMX 250 mobile laser scanning device available at the Institute of Cartography and Geoinformatics (IKG).

In Figure 7, an abstract representation of the facade structure is shown in the first row. This representation was ceated manually from measurements within the point cloud; ornaments above the



(a) Original facade.


(b) Result with higher penalty for window height.

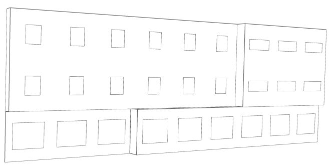

(c) Result with higher penalty for window width.

Figure 5: Homogenization results with different penalties for window width and height.

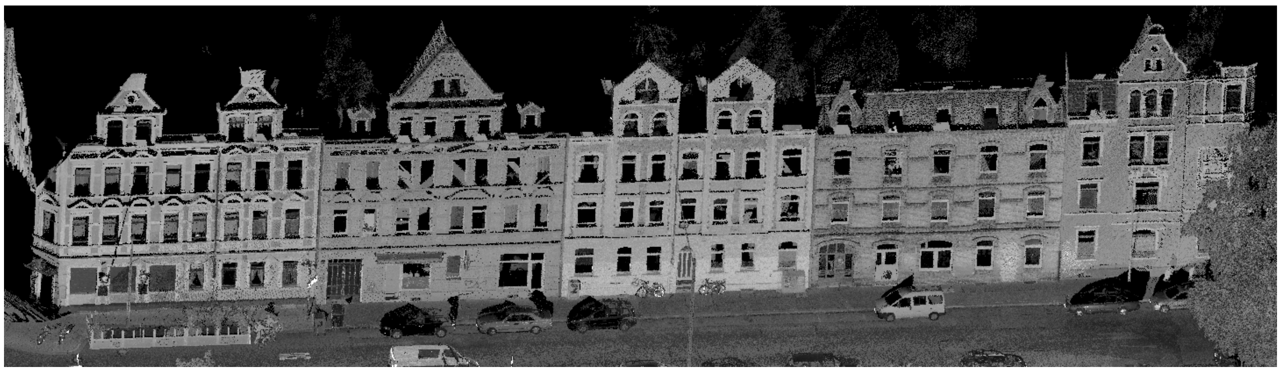
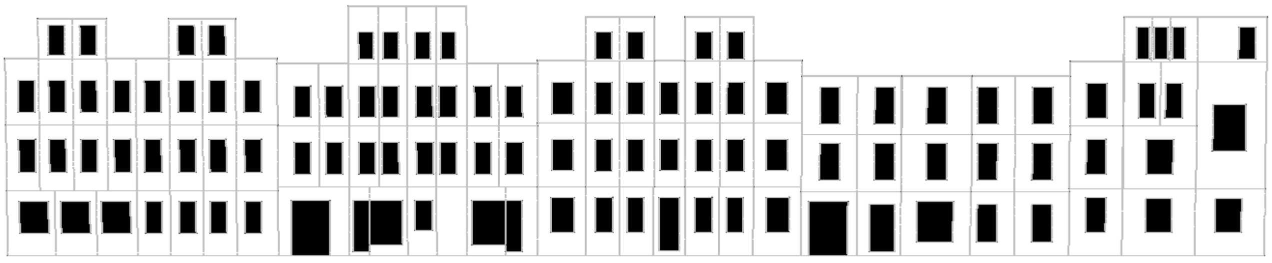
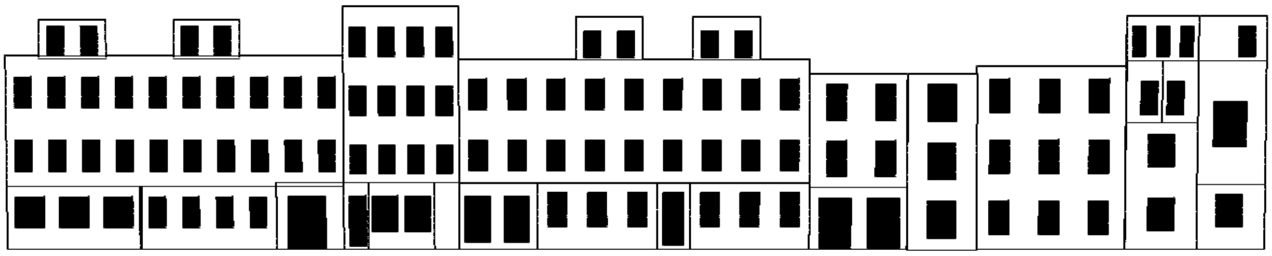


Figure 6: A facade structure covering a part of a block.


(a) Original


(b) Target resolution $\varrho = 0.3m$

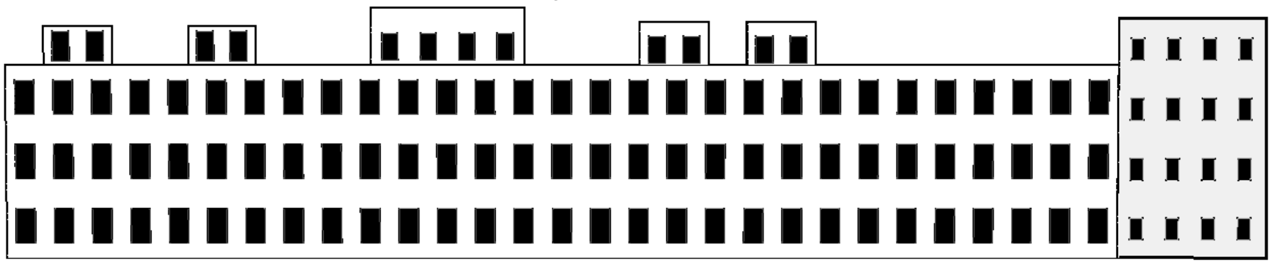

(d) $\varrho = 5.0m$

Figure 7: Cells for the facade structure in figure 6.

windows were ignored in this process (by decision of the operator) and windows and doors were treated in the same way. The lines dividing the facade are the boundaries of the underlying cells.

In the following rows, results of the homogenization process are shown for a large scale (desired resolution $\varrho = 0.3m$) and a small scale ($\varrho = 5.0m$) generalization request; the lines in the figure mark the boundaries of the homogeneous regions. Note that even for the large scale scenario, many large homogenization groups could be formed because the parameters of the openings in the tiles were very similar – especially in the upper floors.

Especially the slightly generalized version in the second row shows that in the current version, an alignment of the heights is only performed within the tiles, not along a whole row of windows – although the windows were almost aligned in the original model.

The rightmost building in the facade string (shaded in the bottom row of the figure) is similar in its structure to the example illustrating possible advantages of combinations of aggregation and homogenization. Because in the current version, only homogenization is applied, the maximum number of four columns of windows (from the four windows in the top row) are generated for the homogenized facade part.

5. CONCLUSION AND OUTLOOK

In this paper, the generalization operator of homogenization was introduced. It was established that, for data compression, homogenization in itself is a means of generalization if efficient structures representing grids of identical features are available.

An optimizing approach to solve the problem for facade objects arranged in a regular grid structure was introduced and evaluated. The examples show the potential of the approach, in that it allows to define abstract constraints and objectives, which are then optimized.

The elegant effect of such an approach is that the quality of the result can be evaluated with respect to a well-defined objective function subject to the given constraints; also, the choice of different penalty values leads to different results. Furthermore, the algorithm can be applied sequentially, leading to a generalization hierarchy in terms of different LoDs.

The facade homogenization problem was examined in an isolated fashion. Even though facade homogenization is in itself a generalization step, it is, however, strongly related to aggregation and typification; it may, in fact, be interpreted as a degenerated case of a typification in which a set of k features is replaced by a set of k' identical symbolic features where, for the homogenization, $k' = k$ and for a typification, $k' < k$.

As shown in (Guercke, 2014), performing an aggregation step before a homogenization step can increase the number of options for the homogenization. The decision which of the generalization operators are best applied to which parts of a facade or if an optimal result can be achieved by different sequences of the operators for the different parts of a facade is a most involved optimization problem (model selection) on top of the optimization of the homogenization process introduced in this paper.

An important aspect to be integrated in future versions of the facade simplification is the special role of the (relative) depth of the facade tiles: Even if they do not have anything else in common, we can save several wall, roof and floor features if two adjacent tiles have the same depth. In the current implementation, this aspect is not yet considered, so in some cases, adjacent wall tiles with only very small offsets were not merged to form a single wall surface because the features in the cells of the tiles were not similar enough.

The overlap between the first and second floor in the facade structure in figure 7b) is due to the fact that the current version does not include constraints that force the floors below a given tile to end exactly at the same height level in order to form a sensible platform for the tile to rest on. Implementing this constraint set is the next step for the improvement of the facade homogenization approach.

The generalization operator of homogenization is widely applicable beyond simple facade elements – one may even compose complete suburbs by a few references to parts lists of prefabricated building vendors.

Note that CityGML supports managing objects as template instances through the concept of XML XLink elements. In order to exploit the full capacity of this approach, data structures to model slight modifications of a referenced object in CityGML would be helpful.

REFERENCES

Anders, K.-H., 2005. Level of detail generation of 3d building groups by aggregation and typification. In: Proceedings of the 22nd International Cartographic Conference of the ICA.

Fan, H., Meng, L. and Jahnke, M., 2009. Generalization of 3D Buildings Modelled by CityGML. In: Advances in GIScience: Proceedings of 12th AGILE Conference on GIScience, Lecture Notes in Geoinformation and Cartography, Springer, pp. 387–405.

Guercke, R., 2014. Optimization Aspects in the Generalization of 3D Building Models. Dissertation, Leibniz Universität Hannover. Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften. Reihe C. Dissertationen. Heft Nr. 723.

Guercke, R., Götzelmann, T., Brenner, C. and Sester, M., 2011. Aggregation of lod 1 building models as an optimization problem. ISPRS Journal of Photogrammetry and Remote Sensing 66(2), pp. 209 – 222. Quality, Scale and Analysis Aspects of Urban City Models.

Hake, G., Grünreich, D. and Meng, L., 1996. Kartographie. Visualisierung raum-zeitlicher Informationen. Walter de Gruyter & Co., Berlin, Germany.

Haunert, J.-H., 2009. Aggregation in Map Generalization by Combinatorial Optimization. Dissertation, Leibniz Universität Hannover. Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften. Reihe C. Dissertationen. Heft Nr. 626.

IBM ILOG, 2011. CPLEX, Optimization, Operations Research, Mathematical Programming, Linear Programming, Integer Programming, Mixed Integer Programming, Quadratic Programming, Modeling and solving optimization problems. Website, URL: `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`.

Kolbe, T. H., Gröger, G. and Plümer, L., 2005. CityGML – Interoperable Access to 3D City Models. In: Proceedings of the first International Symposium on Geo-Information for Disaster Management, Springer Verlag, pp. 21–23.

Ripperda, N., 2008. Grammar based facade reconstruction using rjmcmc. Photogrammetrie Fernerkundung Geoinformation (PFG) 2, pp. 83–92.

Sester, M. and Klein, A., 1999. Rule based generalization of buildings for 3d-visualization. In: Proceedings of the 19th International Cartographic Conference of the ICA.