Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2015) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

24th International Meshing Roundtable (IMR24)

A Robust Combinatorial Approach to Reduce Singularities in Quadrilateral Meshes

Chaman Singh Verma[a], Krishnan Suresh[b]

[a] *Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, USA*
[b] *Department of Mechanical Engineering, University of Wisconsin, Madison, WI 53706, USA*

Abstract

There are many automatic quadrilateral mesh generators that can produce high quality mesh with low distortion. However, they typically generate a large number of singularities that could be detrimental to downstream applications. This paper introduces *Minimum Singularity Templates (MST)* to reduce the number of singularities in an existing pure quad mesh. These templates are easy to encode with high-level grammar rules for complete automation, or interactive control. The MST exploits two important properties of quadrilateral meshes: (1) every submesh has even number of quad edges on its boundary, and (2) every submesh with 3, 4 or 5 topological convex corners on its boundary has at most two interior singularities. The MST (1) does not change the boundary edges of the patch, (2) avoids corner picking on a patch and solving NP hard internal matching algorithm to select divisions, (3) is extremely fast with time complexity of $O(1)$ in template creation, and (4) has low memory footprint and is robust. To illustrate the concepts, we consider quadrilateral meshes generated using Abaqus, Gmsh, and Cubit, and reduce the singularities within these meshes.

Peer-review under responsibility of organizing committee of the 24th International Meshing Roundtable (IMR24).

Keywords: Quadrilateral mesh generation; Singularities, Quad templates

1. Introduction

There are many applications especially in non-linear structural mechanics, higher order spectral methods, and texture mapping which are sensitive to the directions, curvatures, and features on the geometric models. In such applications, an all-quadrilateral and all-hexahedral mesh is often preferred over a triangle mesh. In linear elastic simulations, simplicial meshes (triangle and tetrahedral) can exhibit *locking phenomenon* [3]; therefore non-simplicial meshes (quads and hex) are preferred. Quadrilateral meshes are also de facto standard representation of geometric

* Corresponding author. Tel.: +0-608-698-4729
E-mail address: csverma@cs.wisc.edu

Peer-review under responsibility of organizing committee of the 24th International Meshing Roundtable (IMR24).

models used in animation for game and film production, since these meshes allow cage structures for Calmull-Clark subdivision, which are easier to retopologize compared to triangle meshes [9].

An ideal quadrilateral mesh is characterized by regular vertex distribution. An internal vertex is considered *regular* if it has four incident edges, otherwise it is a singular (or irregular) node. The famous Gauss-Bonnet theorem states that all surfaces with positive genus must have singular nodes. When singularities are present in a mesh, they lead to (1) numerical instability in CFD applications [24], (2) wrinkles in subdivision surfaces [13], (3) irrecoverable element inversions near concave boundaries, (4) helical patterns [4], (5) produce visible seams in texture maps, and (6) breakdown of structured patterns on manifolds. However, singularities are essential in controlling distortions near bifurcations, protrusions, cavities etc. and in abrupt shape transitions. Therefore, the major challenges in producing high quality quad and hex mesh generators are usually related to minimization and placement of singularities.

Recently many provably robust and efficient algorithms for automatic generation of quad meshes have been proposed. *Gmsh* [11] is an open source software, whereas *Cubit* [1] (and its many variants) are available in commercial products. All such automatic quadrilateral mesh generators produce high quality mesh with respect to quality criteria such as low distortion, good orientation, semantic alignment, size control etc, but they may have significant number of singularities. For example, in Figure 1, quadrilateral meshes obtained using *Gmsh* software have far too many singularities sprinkled over the mesh.

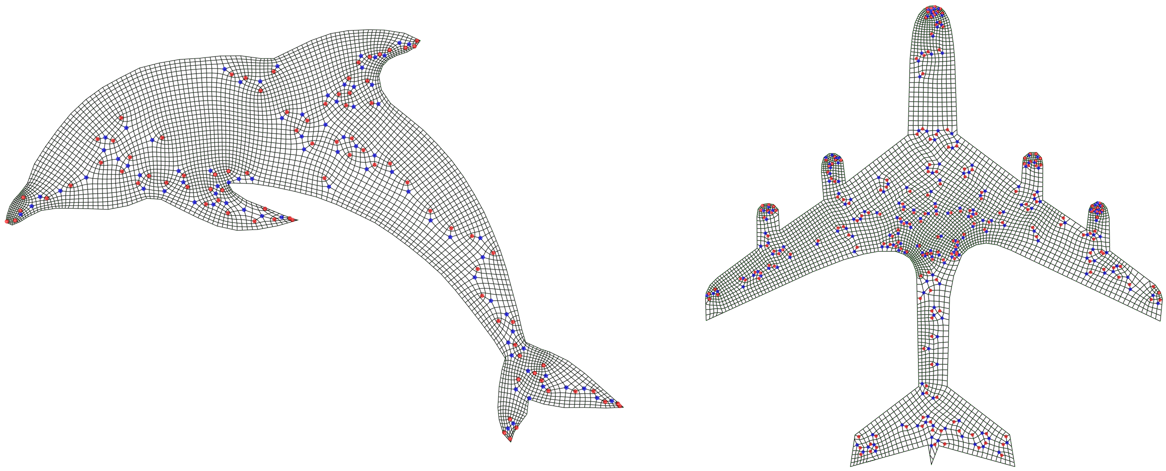


Fig. 1: Singularities over quad meshes generated using Gmsh software [11].

Further, unlike simplicial meshes, quad and hex meshes are inherently global in nature which can be easily understood by their *dual* representations [2]. A single modification to their topology can have a *domino effect* in that a large number of elements may have to undergo modifications to keep the mesh consistent. The non-localness compounds difficulties in the automation of quad mesh generation and editing as various quality criteria have non-linear dependencies which can be extremely hard to encode and solve. Furthermore, it is also impossible to refine, coarsen or edit a quad mesh with local operations as they create additional singularities [21]. For all these reasons, quad and hex meshing problems are often formulated as global optimization problems. Unfortunately, these optimizations are not only expensive, but also their parametric tweaking is non-intuitive. There is no direct intuitive connection between the user constraints and the resulting mesh topology

To address these issues, this paper describes a simple and robust method to reduce singularities in a given quad mesh. The main idea is to replace sub-meshes containing large number of singularities with *Minimum Singularity Templates*. This process is applied repeatedly while maintaining desired geometric quality.

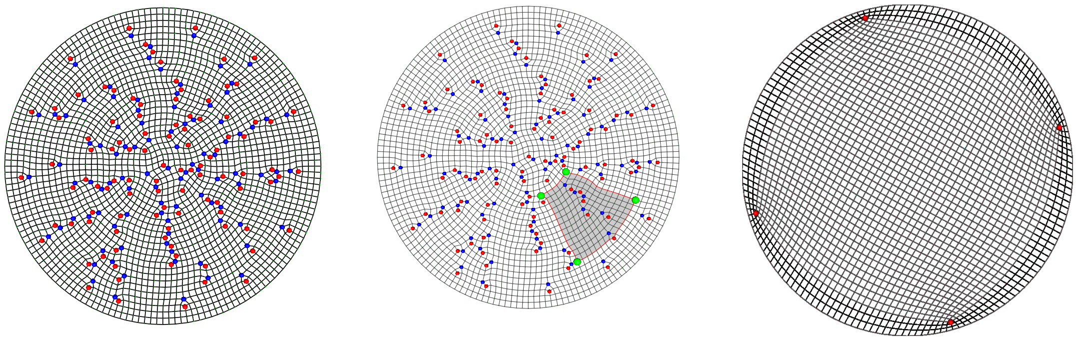
2. Related Work

Singularity control in a quad or hex mesh can be performed either during the mesh generation process or as a post-processing step. In the following we give a summary of some of the important work.

- **Ab-initio methods :** The Q-Morph algorithm [19] transforms a given triangle mesh into a quadrilateral mesh using an advancing front method. In this approach, quadrilaterals are formed using existing edges in the triangle mesh, by inserting additional nodes, or by performing local transformations to the triangles. The final mesh quality is improved by topological clean-up and local smoothing operations.
  Spectral methods [10,15] provide a novel approach for quadrangulating a manifold polygonal mesh using Laplacian eigenfunctions which are the natural harmonics of the surface. The surface Morse functions distribute their extrema evenly across a mesh, which connect via gradient flow into a quadrangular base mesh (known as a Morse-Smale Complex). An iterative relaxation algorithm refines this initial complex to produce a globally smooth parametrization of the surface. From this, well-shaped quadrilateral mesh with very few singularities are generated. Although very elegant, this approach poses many challenges: (1) The quality of mesh depends on an appropriate Morse function which is often heuristic and a poor choice may lead to huge numbers of singular nodes, (2) computing first few (about 40) eigenvectors is very expensive and sometimes impractical for a large mesh, and (3) tracing the curves from maxima to minima via saddle points may face geometric robustness issues [12].
- **Quad re-meshing** Quad re-meshing approaches attempt to improve an existing quad mesh to satisfy user specified properties. Many successful techniques are based on *parametrization* which can automatically find singularity positions by smoothing the principal curvature directions [5,18]. Global parametrization approaches directly generate all-quad meshes using a base-complex, but designing a base-complex is non-trivial, which is either semi-automated or based on the Morse theory.
  Kalberer et al.[16] generate a high quality quadrilateral mesh using global parametrization which is guided by a user-defined frame field (often the principal curvature directions). These frame fields simplify to vector fields on the covering spaces, so that the problem of parametrization with frame fields reduces to the problem of finding a proper integrable vector field on the covering surface. Similarly, Bommes et al. [4] formulated the quadrangulation problem as two step process (cross field generation and global parametrization), both formulated as a mixed-integer problems. This scheme allows placement of singularities at geometrically meaningful locations, and produces meshes with favourable orientation and alignment properties.
  Hormann et al. [14] presented an algorithm that converts an unstructured triangle mesh with boundaries into a regular quadrilateral mesh using global parametrization that minimizes geometric distortion.
- **Quad mesh decimation** Decimation approaches attempt to simplify a given mesh complex by using local or global transformations. In the area of topological clean-up operations, many methods have been presented. Canann [8] and Kinney [17] present very large numbers of operations (in 1000s) usually applied using an isomorphism approach. In contrast to those, Bunin [7] presents a very elegant technique for removing defect vertices based on patch replacement, whose results are better than those of the isomorphism approach while also being vastly simpler to implement. Peng etc. [21] provided connectivity editing operations for quadrilateral meshes explicitly control the location, orientation, type, and number of the irregular vertices in the mesh while preserving sharp edges by three fundamental operations to move and re-orient a pair of irregular vertices. Their operations are based on very low level atomic operations, therefore, many operations may be needed to produce the desired mesh topology.

3. Synopsis and Contributions

Given a quad mesh such as the one in Figure 2a, this paper describes a simple and robust technique to dramatically reduce the number of singularities. To begin with, the user provides a rule for ranking singularities within the mesh (example: distance of a singularity from the boundary). Thereafter, these singularities are sorted and incrementally removed as follows. Since it is not possible to remove a single or even a pair of singularities from a mesh [21], at least three singularities are needed to participate in the removal process. This leads to the idea of exploiting *patch*, which is a well known structure in mesh generation in various contexts. In our context, a patch is a subset of the mesh containing at least three singularities. Starting with three neighbouring singularities, additional elements are incrementally added to the patch, while maintaining certain desirable properties of the patch (to be described later on). One such property is that the boundary of the patch must contain at least 3 *topologically convex corners* [7]; Figure 2b illustrates a patch with 4 such corners. Once such a patch has been created, the elements within this patch are replaced by a new set of elements with fewer singularities, using the *Minimum Singularity Template (MST)*. A smoothing step is then carried out to improve the geometric quality. The process is then repeated, starting from three new singularities, until no new patches can be constructed, resulting in the quad mesh shown in Figure 2c. While the entire process can be automated, it is also amenable to user interactions in that specific singularities can be identified and removed at any stage. Each of the above steps is described in detail in the remainder of the paper.



(a) Input quad mesh (b) Patch replacement (c) Final mesh

Fig. 2: A brief overview of the singularity removal process

4. Basic Definitions and Propositions

Definition 1. *The valence of a vertex v_i is the number of edges incident on it. A vertex with "n" valence is denoted by $\mathcal{V}n$. An internal vertex with valence 4 is considered* regular*, otherwise it is an* irregular *or* singular *vertex. An internal vertex with valence 2 is a called* doublet.

In this paper, we consider only $\mathcal{V}3$ and $\mathcal{V}5$ singular nodes as all other high valence nodes can be converted into $\mathcal{V}3$ and $\mathcal{V}5$ nodes using standard atomic **face open** *or* **face close** *operation [2].*

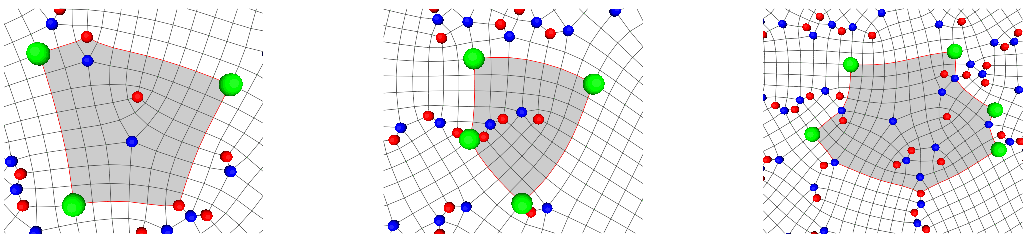
Definition 2. *A patch is a sub-mesh with disc topology.*

Definition 3. *The topological outer angle (TOA) [7] of a vertex on the boundary of a patch is defined as:*

$$TOA(V) = \#\,\textit{faces incident on the vertex that lie outside the patch} \; - 2 \qquad (1)$$

Definition 4. *A vertex on the boundary of a patch is a* convex corner *if its TOA is greater than or equal 1.*

Figure 3 illustrates three patches with 3, 4 and 5 convex corners. Observe that these corners naturally divide the boundary of the patch into 3, 4 and 5 segments, respectively.



(a) A 3-sided patch (b) A 4-sided patch (c) A 5-sided patch

Fig. 3: Convex patches containing singularities

Proposition 1. *The number of quad edges on the boundary of a patch is even.*

Proof. For a quad mesh $\mathcal{M}$, which is locally homeomorphic to an open subset of R^2, the number of faces and edges must satisfy $4F = 2E_i + E_b$, where E_i is the number of internal edges shared by two faces, and E_b is the boundary edges shared by one face. This equation implies that E_b must be even. □

The above proposition simply confirms the existence of a quad mesh in a patch. Our task is to provide a constructive algorithm to create an alternate quadrangulation with fewer singularities.

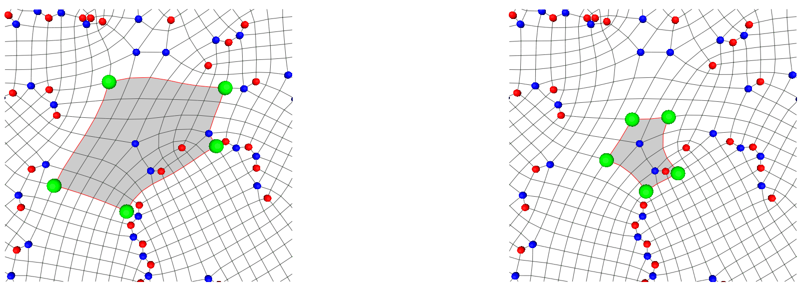
Proposition 2. *Any quadrangulation of a patch with k convex corners will have at least* $|k - 4|$ *interior singularities [22].*

Thus, for example, there are 3 interior singularities in Figure 3a, the above proposition asserts that, all quadrangulation of the patch must contain at least one singularity.

5. Patch construction

The first step in the proposed algorithm is to construct patches that satisfy certain desired properties. Towards this end, the user specifies a criterion to rank the singularities in the domain. After that patch identification process starts as follows:

1. Pick the highest ranking singularity which becomes the seed for the patch expansion.
2. Grow the patch around the seed. Bunin [7] used breadth first search (BFS) to expand the patch. Here we improve Bunin's method using one heuristic: We know that we need at least three singularities in a patch, so instead of searching in all directions (as in classical BFS), we search for one more singularity which is topological closest to the seed using Dijkstra's shortest path algorithm. All the nodes on the shortest path become seeds for the expansion to identify more singularities. This simple change creates a compact and thinner patch as shown in Figure 4.
3. The patch is expanded in a breadth-first order, until there are atleast three singularities in the patch.
4. The boundary nodes of the patch are identified and their topological angles are computed. If there are more than five convex corners, we pick only five of them.
5. If convex corners are not found, the entire patch is expanded by one layer. If the expansion reaches boundary, further expansion is halted.
6. If the expansion reaches boundary and no corners are found based on topological angle criterion, then we choose the corners from the uniformly distributed boundary nodes of the patch (In some cases, this may create *doublets* at the corners, which must be removed immediately after remeshing the patch).



(a) Breadth first Patch search

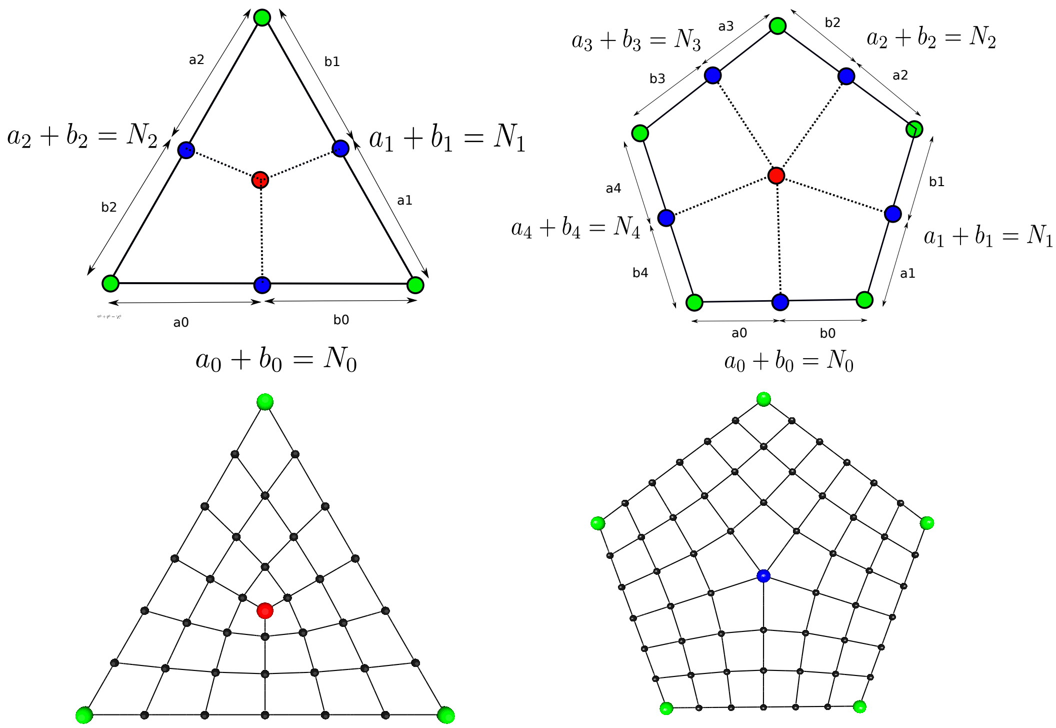
(b) A compact patch using shortest initial path

Fig. 4: Patch searching

6. Quadrangulation with Minimum Singularities

6.1. One singularity remeshing

Obtaining the least number of singularities in a patch is a constrained satisfying linear problem [7,25], for which a solution exists only for patches which have specific number of edges on its sides. Whereas for a quad patch, zero singularity is obtained when number of edges on opposite sides are equal, for other polygonal patches, we need to solve a linear system derived from *Interval matching* method.



(a) Triangle patch (b) Pentagon patch

Fig. 5: One singularity remeshing templates.

6.1.1. Triangle patch

The interval matching for a triangle patch leads to the following system of equations [25]. If there exists an integer solution to this system, then a triangle patch can be quadrangulated with a single $\mathcal{V}3$ singularity inside the patch as shown in Figure 5a. In this figure $a_0, b_0, ...$ denote the number of quad edges on the patch segments.

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{pmatrix} N_0 \\ N_1 \\ N_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \iff \begin{aligned} a_0 + b_0 &= N_0 \\ a_1 + b_1 &= N_1 \\ a_2 + b_2 &= N_2 \\ a_0 - b_1 &= 0 \\ a_1 - b_2 &= 0 \\ a_2 - b_0 &= 0 \end{aligned} \qquad (2)$$

6.1.2. **Pentagon patch**

The interval matching on a pentagon patch leads the following system of equations [25]. If there exists, an integer solution of this system then we can quadrangulate a pentagon patch with a $\mathcal{V}5$ singularity as shown in Figure 5b

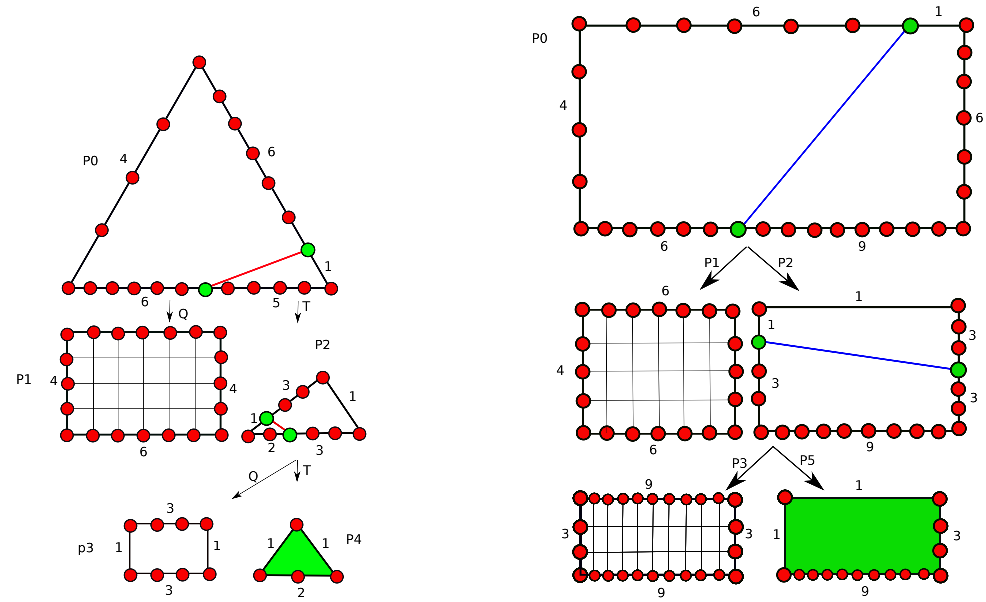
$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0
\end{pmatrix}
\begin{bmatrix} a_0\\ a_1\\ a_2\\ a_3\\ a_4\\ b_0\\ b_1\\ b_2\\ b_3\\ b_4 \end{bmatrix}
=
\begin{pmatrix} N_0\\ N_1\\ N_2\\ N_3\\ N_4\\ 0\\ 0\\ 0\\ 0\\ 0 \end{pmatrix}
\iff
\begin{aligned}
a_0 + b_0 &= N_0\\
a_1 + b_1 &= N_1\\
a_2 + b_2 &= N_2\\
a_3 + b_3 &= N_3\\
a_4 + b_4 &= N_4\\
a_0 - b_3 &= 0\\
a_1 - b_4 &= 0\\
a_2 - b_0 &= 0\\
a_3 - b_1 &= 0\\
a_4 - b_2 &= 0
\end{aligned}
\qquad (3)
$$

If solving linear equations 2, 3 lead to integer values to all the a_i and b_i then the patch can be remeshed with only one singularity in the interior (and vice versa). The next section addresses the quadrangulation when integer solutions do not exist.

The inversion of these matrices are pre-calculated and sparse matrix vector multiplication is hand optimized. Therefore, the decision of quadrangubility is trivially determined [25].

6.2. **Patch reduction**

Now the main objective is to find minimum number of singularities within a patch. An exhaustive search is infeasible except for small patch sizes. Therefore, we first divide a given patch into sub-patches. One of the sub-patches which is referred here as the *Maximally Reduced Patch(MRP)* has the property that it has one or more sides with only one edge and all other sub-patches are 4-sided with perfect quadmeshes (i.e. no singularity).



(a) Triangle patch reduction

(b) Quad patch reduction

Fig. 6: Examples of patch reduction

- **Triangle Patch Reduction:** Let a triangle patch be $P = P(N_0, N_1, N_2)$ where N_k denotes the number of quad edges on the k^{th} side. We rearrange the triangle so that $N_0 \geq N_1 \geq N_2$. In the first step we divide the the patch P_0 into P_1 and P_2 as follows (Figure 6a):

$$\begin{aligned} m_1 &= \min(N_0, N_1) - 1 \\ P_0(N_0, N_1, N_2) &= P_1(m_1, N_2, m_1, N_2) + P_2(N_0 - m_1, 1, N_2) \end{aligned} \tag{4}$$

The sub-patch P_1 leads to a perfect quad mesh, and the second side of the subpatch P_2 has only one edge. We can further reduce the subpatch P_2 into P_3 and P_4 subpatches as follows (Figure 6a);

$$\begin{aligned} m_2 &= \min(N_0 - m_1, N_2) - 1 \\ P_2(N_0 - m_1, 1, N_2) &= P_3(m_2, 1, m_2, 1) + P_4(N_0 - m_1 - m_2, 1, 1) \end{aligned} \tag{5}$$

With the second decomposition the subpatch P_3 will lead to a perfect quadmesh and P_4 has only edge on its second and third side. This forms the MRP for a given triangular P_0 patch. With these decompositions, finding the minimum number of singularities in the P_0 patch is reduced to finding minimum number of singularities in the sub-patch P_4. Quadrangubility of the sub-patch P_4 is guaranteed by the following proposition.

Proposition 3. *The number of edges on the boundary of the subpatch P_4 is even.*

Proof : Observe that the input patch P_0 has even number of edges.

$$N_0 + N_1 + N_2 = 2N \tag{6}$$

With the first decomposition

$$\overbrace{\{(N_0 - m_1) + m_1\}}^{first\ side} + \overbrace{\{m_1 + 1\}}^{second\ side} + \{N_2\} = 2N$$

Thus for the patch P_2

$$\overbrace{\{N_0 - m_1\}}^{first\ side} + \overbrace{\{1\}}^{second\ side} + \{N_2\} = 2N - 2m_1 = 2N'$$

Thus patch P_2 also has an even number of edges. Repeating this argument once more, we can show that P_4 also has an even number of edges, and therefore it can be quadrangulated. The precise nature of this quadrangulation is discussed in the next section.

- **Quad Patch Reduction** Let a quad patch be $P = P(N_0, N_1, N_2, N_3)$. We arrange the patch such that $N_0 \geq N_2$ and $N_1 \geq N_3$. We decompose the patch in two directions in two steps as follows(Figure 6b). The first decomposition is done with a vertical cut and we obtain two subpatches P_1 and P_2. While P_1 contains perfect quadmesh, the patch P_2 has only one edge on its top side.

$$\begin{aligned} m_1 &= \min(N_0, N_2) - 1 \\ P_0(N_0, N_1, N_2, N_3) &= P_1(m_1, N_3, m_1, N_3) + P_2(N_0 - m_1, N_1, 1, N_3) \end{aligned}$$

The second decomposition is performed with a horizontal cut on P_2 and we get two subpatches P_3 and P_4. The subpatch P_3 contains a perfect quadmesh and the patch P_4 has its left and top sides each containing one edge. This form the MRP of the original quad patch P_0.

$$\begin{aligned} m_2 &= \min(N_1, N_3) - 1 \\ P_2(N_0 - m_1, N_1, 1, N_3) &= P_3(N_0 - m_1, m_2, N_0 - m_1, m_2) + P_4(N_0 - m_1, N_2 - m_2, 1, 1) \end{aligned} \tag{7}$$

With these two decompositions, finding the minimum singularities in the original patch P_0 is reduced to finding minimum singularities in the patch P_4. The quadrangubility of P_4 is guaranteed as it contain the even number of edges.

Proposition 4. *The number of edges in the quad subpatch P_4 is even.*

Proof : Observe that the input patch P_0 has an even number of edges.

$$N_0 + N_1 + N_2 + N_3 = 2N \tag{8}$$

With the first decomposition

$$\overbrace{\{(N_0 - m_1) + m_1\}}^{first\ side} + \{N_1\} + \overbrace{\{(m_1 + 1\}}^{third\ side} + \{N_3\} = 2N$$

Thus for the patch P_2

$$\overbrace{\{N_0 - m_1\}}^{first\ side} + \{N_1\} + \overbrace{\{1\}}^{third\ side} + N_3 = 2N - 2m_1 = 2N'$$

which is even number. Similarly, we can prove the smallest quad patch P_4 will have even number of edges.

- **Pentagon Patch Reduction:** A pentagon can be decomposed into a quad and a triangle, or three triangles in various ways. Each triangle and quad patch can be quadrangulated with the methods described above. Therefore, we omit any discussions for this case.

6.3. Minimum Singularity Templates

Once a patch is decomposed into subpatches, due to the property discussed above, it is easy to show that the MRP will fall into one of the templates shown in the Figure 7 and 8. There are two templates for a triangular patch, and three for a quadrilateral patch.

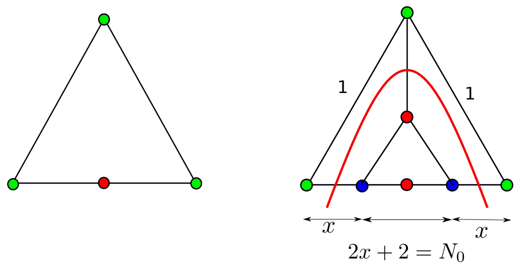


Fig. 7: MST for a triangle patch (a) T_1 type (b) T_2 Type

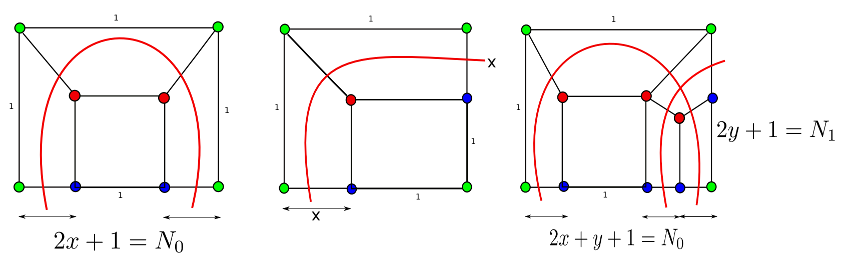


Fig. 8: MST for a quad patch (a) Q_1 type (b) Q_2 Type (c) Q_3 type

Corollary 1. *A triangular patch can be quadrangulated with at most two singularities and this occurs with T_2 template. These two singularities are shown in Figure 7 with red circles.*

Corollary 2. *A quadrilateral patch can be quadrangulated with at most two singularities and this occurs with Q_3 template. These singularities are shown in Figure 8 with the red circles.*

Corollary 3. *A pentagon patch can be quadrangulated with four singularities.*

Thus to summarize, given a patch with 3, 4, or 5 segments, and certain number of quad edges on each segment, template selection is automated as follows:

1. Check if the system of equations 2 or 3 give an integer solution. If so, then the patch is quadrangulable with only one singularity.
2. Else, if the patch is triangular, check the number of nodes in the subpatch P_4. If there are three nodes on its first side, apply $T1$ template, otherwise apply $T2$ template. With the $T1$ template, one singularity is introduced at the boundary of the patch and with $T2$, one interior and one boundary singularity is introduced.
3. If the patch is quadrilateral:

   - If the sides N_1, N_2, N_3 all have one edge, apply $Q1$ patch. This template will create two singularities in the interior of the patch.
   - If the number of edges on side N_0 and N_1 is equal and greater than one, apply $Q2$ patch. This template will create one singularity in the interior of the patch.
   - If the number of edges on side N_0 and N_1 differ, apply patch $Q2$. This template will create two singularities in the interior of the patch.

4. If the patch is pentagonal, decompose it into either quad-triangles pairs or in three triangular patches. Since there are many ways to decompose a pentagon, we check for each decomposition for the number of singularities and accept the one which gives the least number of singularities.

7. Patch replacement

Until now, all the steps were combinatorial and required only algebraic calculations, the final patch replacement involves geometric considerations. The shape of a patch could be arbitrarily complex (although it is always a topological disk). Therefore the mapping from template domain to physical domain can produce large distortions and inverted elements. The following steps are taken to replace a patch with the template mesh

1. Create a correspondence between corners of the template patch with the corners of the physical patch corners.
2. Calculate Floater's *Mean Value Coordinates* for each vertex of the template mesh with respect to its corners, therefore each vertex of the template mesh will have coordinates in n-dimensional space, where n is the number of corners.
3. Change the coordinates of the corners of the template mesh with the coordinates of the physical patch and calculate the new coordinates of each vertex of the template mesh.
4. Constraint the boundary nodes of the template mesh with the corresponding boundary nodes on the physical patch (i.e. Dirichlet boundary condition).
5. Optimize the patch using Lloyds relaxation and improve the shape of quad elements, for example, using *Mesquite* software [6].
6. Check for inversion of each face in the template mesh. If there is a inverted face, apply ***locally injective mapping*** of Schueller etc [23] to obtain a fold-free mesh.
7. If all the faces have positive Jacobian, improve mesh quality using *Mesquite* software, replace physical patch with the template mesh and update the mesh data structures.

8. Results

To evaluate the effectiveness of our approach, we created quad meshes using *Gmsh*, *Cubit* and commercial *Abaqus* codes. These meshes are shown in Figures 9 on the left side. The resulting meshes are illustrated in Figure 9, on the right side. The mesh qualities are evaluated with the *Verdict* software [20]. Various mesh quality values are presented in the Table 1. In the table L, H indicates whether lower value or higher value is preferred.

- In all cases, the number of quad faces remains almost constant.
- For the first example (plate with hole), there was a 92% reduction in singularities, and these singularities are almost symmetrical. In this case, all the mesh quality values have improved which reaffirms our initial hypothesis that there are often more singularities in some meshes than necessary to control the maximum distortions.
- In the Dolphin case, in the sharp concave region (near the fin) excessive singularity removal increased 'oddy' from 18 to 54. This can be significantly improved with an additional pillow-layer.
- The bird example has many concave regions, therefore only 82% singularities could be removed.
- For the 5-hole example, we could reduce only 89% singularities despite the fact that the model is geometrically simple. The small feature lengths played an important role in keeping large number of singularities in the domain. In this experiment, singularities are moved from the smaller circles first and then from the rest of the region to maintain high aspect ratio near the boundaries.

Although in all the experiments, the final mesh quality is within acceptable range (as defined by the Verdict Software), there are two ways to improve the results: (1) allow selective refinement near high aspect ratio elements, and (2) use L_∞ shape optimization methods instead of average quality improvement methods.

	Square hole	**Dolphin**	**Bird**	**5-holes**
Source	Abaqus	Gmsh	Gmsh	Cubit
Faces	4250/4015	5856/5586	2399/2271	9923/10085
Singularities	156/12 (92%)	474/38(92%)	272/47 (82%)	627/70(89%)
# Iterations	17	30	20	40
Aspect Ratio (L)	1.70/1.54	2.60/3.65	2.24/3.44	1.71/3.04
Condition number (L)	1.65/1.32	3.18/5.30	3.42/3.54	1.81/1.97
Distortion (L)	0.40/0.60	0.27/0.30	0.38/0.41	0.46/0.45
MinAngle (L)	48/57	32/24	25/22	46/37
MaxAngle (H)	140/126	155/162	158/159	140/145
Oddy (L)	3.39/1.5	18.31/54	21.46/23	4.56/5.78
Scaled Jacobian (H)	0.65/0.80	0.42/0.30	0.35/0.34	0.63/0.57
Skew (L)	0.58/0.40	0.74/0.86	0.83/0.88	0.66/0.68
Stretch (L)	1.04/1.26	15.03/23.2	12.81/18.0	1.45/2.06
Taper (L)	0.47/0.30	0.88/0.56	0.64/0.73	0.43/0.39

Table 1: Mesh qualities using *Verdict* software: (x/y) denotes the value of input and output mesh respectively

9. Conclusions and Future work

We have presented a simple, robust, and practical algorithm to reduce singularities in a pure quad mesh using *Minimum Singularity Template*. We have experimented with many models which were quad-meshed by *Gmsh*, *Cubit*, and *Abaqus*. Experiments have shown that some of the high quality quad mesh generators have significant number of removable singularities. These singularities can be removed with little impact on the mesh quality. Although algorithm does not guarantee least number of singularities, it is still very effective in reducing singularities.

Perhaps the most important future wotk that we shall pursue is to unify various optimization methods into single framework. This could be particularly used for singularity editing as proposed by Peng [21], and for removing the helical structures as noticed by Bommes [4] etc.

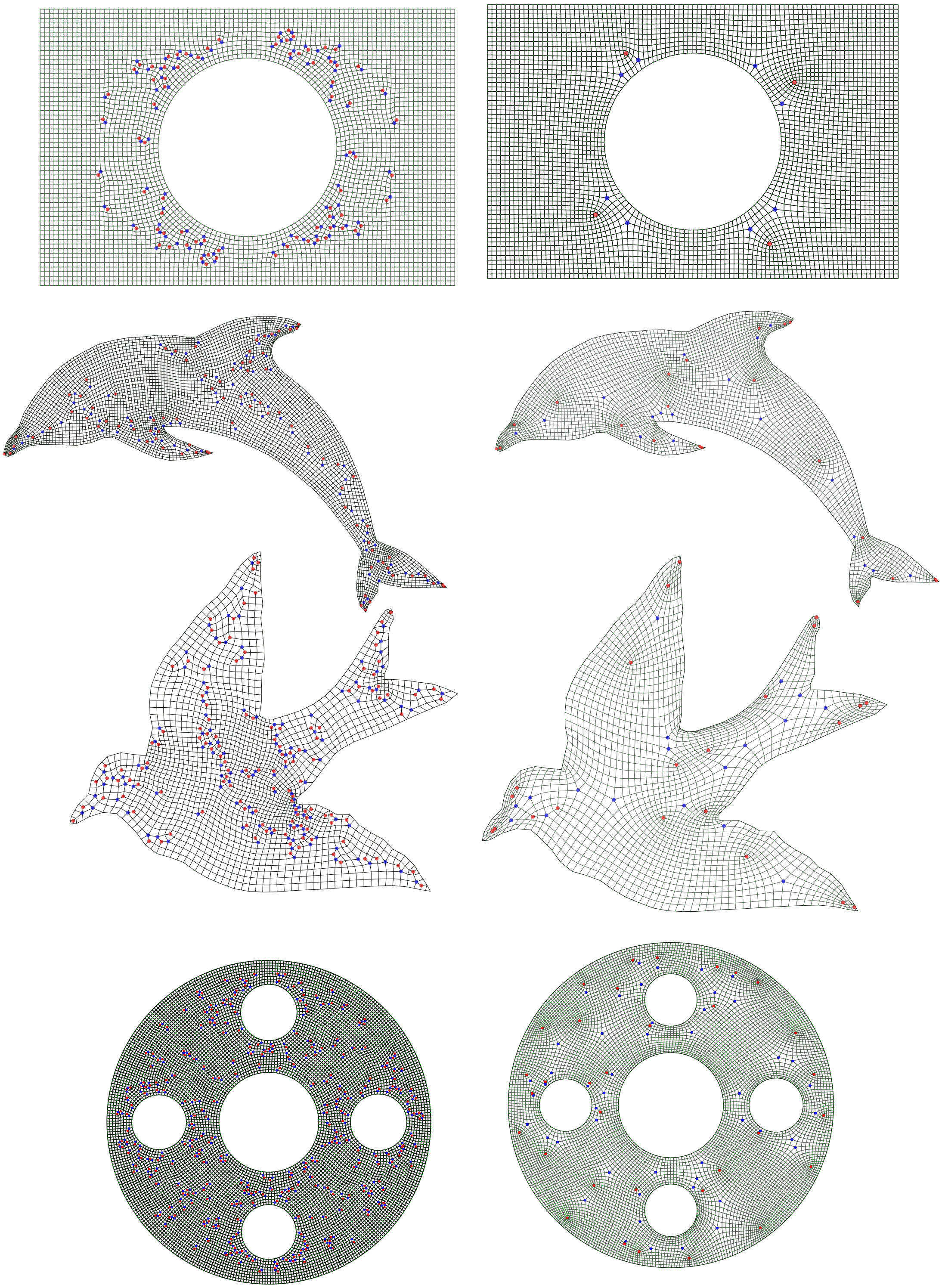


Fig. 9: Results: (A) Original mesh(left column) (B) Output mesh (right column)

References

[1] *Cubit*. https:://cubit.sandia.gov.

[2] Bret D. Anderson, StevenE. Benzley, and StevenJ. Owen. Automatic all quadrilateral mesh adaption through refinement and coarsening. In *Proceedings of the 18th International Meshing Roundtable*, pages 557–574. Springer Berlin Heidelberg, 2009.

[3] Ivo Babuska and Manil Suri. Locking effects in the finite element approximation of elasticity problems. *Numerische Mathematik*, 62(1):439–463, 1992.

[4] David Bommes, Timm Lempfer, and Leif Kobbelt. Global structure optimization of quadrilateral meshes. In *Computer Graphics Forum*, volume 30, pages 375–384. Wiley Online Library, 2011.

[5] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3):77:1–77:10, July 2009.

[6] Michael L. Brewer, Lori Freitag Diachin, Patrick M. Knupp, Thomas Leurent, and Darryl J. Melander. The mesquite mesh quality improvement toolkit. In *IMR*, 2003.

[7] Guy Bunin. Non-local topological clean-up. In PhilippeP. Pbay, editor, *Proceedings of the 15th International Meshing Roundtable*, pages 3–20. Springer Berlin Heidelberg, 2006.

[8] S.A. Canann, S.N. Muthukrishnan, and R.K. Phillips. Topological improvement procedures for quadrilateral finite element meshes. *Engineering with Computers*, 14(2):168–177, 1998.

[9] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 85–94, New York, NY, USA, 1998. ACM.

[10] Shen Dong, Peer-Timo Bremer, Michael Garland, Valerio Pascucci, and John C. Hart. Spectral surface quadrangulation. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1057–1066, New York, NY, USA, 2006. ACM.

[11] Christophe Geuzaine and Jean-Fran Remacle. *Gmsh*. http://www.geuz.org/gmsh, 1.12 edition, August 2003.

[12] Igor Guskov and Zoë J. Wood. Topological noise removal. In *Proceedings of Graphics Interface 2001*, GI '01, pages 19–26, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.

[13] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using catmull-clark surfaces. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 35–44, New York, NY, USA, 1993. ACM.

[14] Kai Hormann and Gnther Greiner. Quadrilateral remeshing. In *Proceedings of Vision, Modeling and Vizualization, 2000*, pages 153–162, 2000.

[15] Jin Huang, Muyang Zhang, Jin Ma, Xinguo Liu, Leif Kobbelt, and Hujun Bao. Spectral quadrangulation with orientation and alignment control. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 147:1–147:9, New York, NY, USA, 2008. ACM.

[16] Felix Kalberer, Matthias Nieser, and Konrad Polthier. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384, 2007.

[17] Paul Kinney. Cleanup: Improving quadrilateral finite element meshes. In *6th International Meshing Roundtable*, pages 449–461, 1997.

[18] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. Globally optimal direction fields. *ACM Trans. Graph.*, 32(4):59:1–59:10, July 2013.

[19] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal. Q-morph: an indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 44(9):1317–1340, 1999.

[20] Philippe P. Pebay, David Thompson, Jason Shepherd, Patrick Knupp, Curtis Lisle, Vincent A. Magnotta, and NicoleM. Grosland. New applications of the verdict library for standardized mesh verification pre, post, and end-to-end processing. In MichaelL. Brewer and David Marcum, editors, *Proceedings of the 16th International Meshing Roundtable*, pages 535–552. Springer Berlin Heidelberg, 2008.

[21] Chi-Han Peng and Peter Wonka. Connectivity editing for quad-dominant meshes. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, SGP '13, pages 43–52, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association.

[22] S. Schaefer, J. Warren, and D. Zorin. Lofting curve networks using subdivision surfaces, 2004.

[23] Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. Locally injective mappings. *Computer Graphics Forum (proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing)*, 32(5):125–135, 2013.

[24] Jos Stam. Flows on surfaces of arbitrary topology. *ACM Trans. Graph.*, 22(3):724–731, July 2003.

[25] Chaman Singh Verma and Tim Tautges. Jaal: Engineering a high quality all-quadrilateral mesh generator. In WilliamRoshan Quadros, editor, *Proceedings of the 20th International Meshing Roundtable*, pages 511–530. Springer Berlin Heidelberg, 2012.