Computational geometry and artificial neural networks: a hybrid approach to optimal sensor placement for aerospace NDE

Roberto Osegueda, Carlos Ferregut, Mary J. George,
Jose M. Gutierrez, and Vladik Kreinovich[1]

FAST Center for Structural Integrity of Aerospace Systems
University of Texas at El Paso
El Paso, TX 79968, USA

ABSTRACT

The ideal design of an airplane should include built-in sensors that are pre-blended in the perfect aerodynamic shape. Each built-in sensor is expensive to blend in and requires continuous maintenance and data processing, so we would like to use as few sensors as possible. The ideal formulation of the corresponding optimization problem is, e.g., to minimize the average detection error for fault locations. However, there are two obstacles to this ideal formulation:

- First, this ideal formulation requires that we know the probabilities of different fault locations etc., and there are usually not enough statistics to determine these probabilities.
- Second, even for a known distribution, finding the best locations is a very difficult computational problem.

To solve these problems, *geometric symmetries* are used; these symmetries enable to choose several possible sets of sensor locations; the best location is then found by using a *neural network* to test all these (few) selected locations.

Keywords: Non-destructive evaluation, aerospace structures, aging aircraft, futuristic aircraft, neural networks, symmetry groups, geometry.

1. Introduction

1.1. Checking structural integrity of aerospace systems is very important

Structural integrity is extremely important for airplanes, because in flight, the airframe is subjected to such stressful conditions that even a relatively small crack can be disastrous.

This problem becomes more and more important as the aircraft fleet ages.

At present, most airplanes do not have built-in sensors for structural integrity, and even those that have do not have a sufficient number of them, so additional sensors must be placed to test the structural integrity of an airframe.

[1]E-mail addresses: osegueda@utep.edu, ferregut@utep.edu, mjgeorge@utep.edu, jgutierr@utep.edu, and vladik@cs.utep.edu.

1.2. It is important to test structural integrity in-flight

Each integrity violation (crack etc.) starts with a small disturbance that is only detectable in stressful in-flight conditions. Therefore, to detect these violations as early as possible, we should complement on-earth testing by in-flight measurements. Hence, we need *sensors* for in-flight tests.

1.3. The problem of sensor placement

Where should we place sensors for in-flight tests?

Most *existing* airplanes do not have built-in sensors for testing structural integrity (or at least do not have a sufficient number of these sensors), so, to test these airplanes, we must place these sensors *outside* the airframe. Sensors attached outside the airframe interfere with the airplane's well-designed aerodynamics; therefore, we should use as few sensors as possible.

This limitation leads to the following problem:

GIVEN: the number of sensors that we can locate on a certain surface of an airframe,

FIND: the optimal placements of these sensors, i.e., locations that allow us to detect the locations of the faults with the best possible accuracy.

For *future* aircraft, we have a similar problem of sensor placement. The ideal design of a future airplane should include built-in sensors that are pre-blended in the perfect aerodynamic shape. Each built-in sensor is expensive to blend in and requires continuous maintenance and data processing, so again, we would like to use as few sensors as possible.

1.4. The problem of optimal sensor placement is difficult, because it requires optimization under uncertainty

In both cases, the *ideal* formulation of the corresponding optimization problem is to minimize, e.g., the average detection error for fault locations.

However, this ideal formulation requires that we *know* the *probabilities* of different fault locations and the probabilities of different aircraft exploitation regimes. In reality, we do not know these probabilities:

- for a *new* aircraft, we do not have this statistics; and
- for the *aging* aircraft, the statistics gathered from its earlier usage may not be applicable to its current state.

Therefore, instead of a well-defined optimization problem, we face a not so well defined problem of *optimization under uncertainty*.

Since the problem is not well defined, we cannot simply use standard numerical optimization techniques.

1.5. To solve the optimal sensor placement problem, we will use intelligent techniques

Since we cannot use *standard* numerical methods, we will use *intelligent* techniques:

1.5.1. First step: Geometric techniques

The problem of choosing an optimal sensor placement is formulated in *geometric* terms: we need to select *points* (sensor placements) on a *surface* of the given structure.

To solve this problem, we use the experience of solving similar geometric problems of optimization under uncertainty in image processing and image extrapolation[1–3]. Namely, astronomic image processing faces the problem of selecting the best family of images for use in extrapolation. It turns out that for every optimality criterion that satisfies the natural symmetry conditions (crudely speaking, that the relative quality of two image reconstructions should not change if we simply shift or rotate two images), the extrapolation shapes that are optimal with respect to this criterion can be described as orbits of the corresponding Lie transformation groups, which leads to exactly the shapes used in astronomy (such as spirals, planes, spheres, etc).

In this paper, we show that, since the basic surface shapes are symmetric, a similar symmetry-based approach can be applied to the problem of optimal sensor placement. For the simplest surfaces, this general approach describes several geometric patterns that every sensor placement, which is optimal with respect to reasonable (symmetric) optimality criterion, must follow.

Some of our results were announced in[4–5].

1.5.2. Second step: Neural networks

After we have selected several possible sensor locations, we then use neural networks:

- first, to *confirm* that these placement patterns indeed lead to better fault location, and
- second, to *select* a pattern that leads to the best results for each particular problem.

1.6. The results: in brief

As a result of this analysis, we get several possible optimal sensor placements.

A similar problem of optimal placement of NDE sensors is also important for *space* structures.

2. Preliminary analysis: we need the optimal sensor placement

2.1. It is desirable to find the optimal sensor placements

The quality of NDE essentially depends on the placement of the sensors: e.g., if all the sensors are concentrated in one area, and few are located in the remainder of the structure, then possible cracks and other faults in the under-covered area may go dangerously unnoticed. Therefore, it is important to choose a good sensor placement.

Currently, the choice of sensor placements is mainly made either *ad hoc*, or, at best, by testing a few possible placements and choosing the one that performs the best on a few benchmark examples. There are two drawbacks in this approach:

- In this approach, only a few possible placements are analyzed, so it is quite possible that we miss really good placements.
- Even when the placement that is good "on average" is indeed present among the tested placements, the very fact that we only test these placements on a few examples leads to the possibility that we will choose different placements, that work well for the tested examples, but that are, on average, much worse than the rejected placement.

In other words, often, the normal engineering *good enough* approach does not work for our problem.

It is, therefore, desirable to find the *optimal* (*best*) sensor placements.

2.2. "Optimal" in what sense?

2.2.1. Since we do not know the exact optimality criterion, we will try to describe sensor placements that are the best relative to all possible reasonable optimality criteria

If we knew the exact probabilities of different exploitation regimes and of different faults, then we could formulate the exact optimality criterion and look for the sensor placement that is optimal relative to this criterion. In reality, however, as we have already mentioned in the Introduction, we do not know the exact optimality criterion.

Since we do not know *the* exact criterion, the natural idea is to do the following:

- consider *all possible* optimality criteria that are reasonable for this problem;
- describe all the placements that are optimal relative to each of these criteria;
- and finally, depending on the exact situation, choose the best placement among the "possibly best" ones.

In this manner, we still face the problem of choosing between several possible placements (and we may still make a wrong choice), but we are, at least, guaranteed that we do not initially miss the best placement.

2.2.2. This general program sounds ambitious and computationally intractable, but it is actually doable

Even when we know the optimality criterion, finding the optimal sensor placement is extremely computationally difficult and time-consuming. According to the above program, we intend to describe sensor placements that are optimal relative to *all* possible reasonable optimality criteria. Since there are many such criteria, it may seem, at first glance, that we need to repeat the (already time-consuming) computations so many times that the resulting required computation time will make this problem computationally intractable.

Fortunately, we will see that this problem is quite doable: namely, it is possible to describe all possibly optimal placements without actually solving all possible optimization problems, but using *geometric* arguments instead.

Before we start describing and using these arguments, we must describe in precise terms what we mean by a "reasonable" optimality criterion.

2.2.3. Optimality criteria can be arbitrarily complicated

Traditionally, the quality of different alternatives is described by a *numerical* optimality criterion, in which the quality of each alternative a from the set A of all possible alternatives is characterized by a real number $J(a)$, and we choose the alternative a for which this value $J(a)$ is the smallest possible (i.e., $J(a) \to \min$). For example, for the problem of placing the given number n of sensors, A is the set of all possible placements of these sensors, and $J(a)$ is, e.g., equal to the mean square average detection error of fault location based on the data from these sensors.

Such numerical criteria are useful and often sufficient, but in many cases, we end up with *several* alternatives with the same smallest possible value of the average error $J(a)$. In this case, it makes sense to select, among them, an alternative a for which, e.g., the *worst-case* error $J'(a)$ is the smallest possible. This very natural idea leads to a non-numerical optimality criterion, according to which *two* different functions $J(a)$ and $J'(a)$ are given, and an alternative a is considered to be *better* than an alternative b if either $J(a) < J(b)$, or $J(a) = J(b)$ and $J'(a) < J'(b)$.

This more complicated criterion can also result in several "best" alternatives, in which case we will be able, simultaneously, to optimize a third characteristic, etc. As a result, we can have arbitrarily complicated non-numerical optimality criteria. Since we want to describe placements that are optimal with respect to *all* possible reasonable criteria, we have to consider all these criteria.

How can we describe them?

2.2.4. A general description of an optimality criterion

In general, when we say that an optimality criterion is given, this means that for every two alternatives a and b from the set A of all alternatives, one of the following four possibilities holds: either a is better than b according to this criterion (we will denote it by $a < b$), or b is better than a ($b < a$), or a and b are of the same quality (we will denote it by $a \sim b$), or, according to the given criterion, the alternatives a and b are incompatible (we will denote this case by $a \parallel b$).

So, we can describe the optimality criterion as a pair of relations $\langle <, \sim \rangle$.

These two relations must satisfy natural consistency conditions, e.g., if a is better than b and b is better than c, then a should be better than c, etc. A pair that satisfies these natural consistency conditions is called a *pre-ordering* relation. In these terms, an optimality criterion is a *pre-ordering relation* on the set of all alternatives.

There is also one additional requirement that we have used before, when we talked about the necessity for complicated optimality criteria: that there should be *exactly one* optimal alternative. Indeed, if there are several alternatives that are equally good according to some criterion, it means that we still need to choose between them; thus, the current optimality criterion is not final. We are interested in *final* criteria, i.e., in pre-ordering relations in which there exists exactly one best alternative.

Now, that we have a general definition of an optimality criterion, we must describe all sensor placements that are optimal relative to these criteria. For this description, as we have mentioned, we will use the geometric techniques.

3. Geometric techniques

3.1. Geometric transformations: a seeming complication

The idea of using symmetries first appeared not as a method for *solving* the problem, but rather as an additional unexpected complication that made its solution even harder. Namely, we started with simplified toy examples, and tried to use an optimization method to find the optimal placements for these toy problems. Since we were solving an extremely simplified problem, we expected that the optimization algorithm would soon give us a single optimal sensor placement. Instead, for each problem, different applications of the numerical algorithm, applications that started with different randomly chosen initial sensor placements, resulted in drastically different optimal sensor placements.

When we plotted these seemingly different solutions, we saw a simple explanation for this non-uniqueness: these "different" solutions turned out to be approximately one and the same solution, but differently rotated and/or shifted. How can we explain this behavior?

3.2. Symmetries of aerospace shapes explain the observed complication

There is a simple geometric explanation for the above-described behavior. This explanation is based on the fact that most surfaces that form an airframe can be described, within a good accuracy, in simple geometric terms.

3.2.1. Basic geometric shapes of aerospace structures and their symmetries

Let us first describe the geometric shapes of basic aerospace structures:

- The airplane cabin can be described as a *cylinder*.
- The surface of the wings can be approximately described as a *plane* (same, for the tail).
- Finally, the plane's "nose" can be approximately described as either a part of the *sphere* (to be more precise, a half-sphere), or as a piece of a *cone*.

Each of these geometric shapes has certain geometric *symmetries*, i.e., geometric transformations that leave this shape invariant:

- a *cylinder* is invariant with respect to shifts along its axis and rotations around this axis;
- a *plane* is invariant with respect to shifts in the plane, rotations in this plane, and dilations (similarities);
- a *sphere* is invariant with respect to arbitrary rotations around its center;
- finally, a *cone* is invariant with respect to rotations around its axis and dilations centered at its vertex.

3.2.2. For perfectly symmetric shapes, optimal placement is non-unique

Let us first consider the idealized situation in which the shape is *precisely* symmetric (e.g., a perfect sphere, that is invariant with respect to arbitrary rotation T around its center). Let $P = \{p_1, \ldots, p_n\}$ be a sensor placement for which the optimality criterion (e.g., the average fault location error) is the smallest possible, and let T be one of the symmetries. Since the shape is invariant with respect to this symmetry, locations $T(p_1), \ldots, T(p_n)$ also belong to this same shape. Since natural optimality criteria are also invariant with respect to these geometric symmetries, the quality of the rotated placement $T(P) = \{T(p_1), \ldots, T(p_n)\}$ is equal to the quality of the original placement and therefore, the rotated placement is also optimal.

Thus, if P is an optimal placement, then for every symmetry T of the geometric shape, the placement $T(P)$ is also optimal. This explains non-uniqueness of optimal sensor placement for *perfectly symmetric* shapes.

3.2.3. For approximately symmetric shapes, optimal placement is also non-unique

Since optimal placement is non-unique for *perfectly symmetric* shapes, it is natural to expect that a similar complication occurs for the shapes that are *close* to the perfectly symmetric shapes (e.g., for a slightly deformed sphere).

3.2.4. Optimal sensor placement is non-unique even when for geometric shapes that are only locally symmetric

As we have mentioned earlier in this section, the actual shapes of aerospace structures are indeed close to perfectly symmetric ones, but they are only *locally* close to the perfectly symmetric shapes:

- For example, the shape of a cabin is close to a cylinder. An (infinite) cylinder is invariant with respect to rotations *and* shifts; however, the shape of a cabin is only a *piece* of this infinite cylinder.
- Similarly, a *wing* is only a piece of a plane, a *nose* is only a piece of a sphere or of a cone, etc.

In other words, the actual shapes are not themselves symmetric, they are only locally close to the symmetric shapes. However, for sensors testing structural integrity, local is all we need: the very need for numerous sensors comes from the fact that the effects of each newly appearing small structural fault are so small that they can be only detected by a sufficiently close sensor. So, the interaction of a fault and of a nearby sensor on, e.g., a small piece of a spherical surface depends only on the *local* properties of this surface and practically does not depend on whether this surface is the whole sphere or a piece of it.

Thus, if a surface locally coincides with the symmetric one, the local quality of each sensor placement on this surface coincides with the local quality of their placement on the perfectly symmetric shape, and therefore, the *optimal* placement on the actual surface locally coincides with the optimal sensor placement on the ideal symmetric surface.

Since the optimal placement on an ideal surface is non-unique, the placement on its piece is also non-unique.

3.3. Due to non-uniqueness, we have not a single optimal sensor placement, but a family of different optimal sensor placements

We have shown, both experimentally and theoretically, that, due to symmetry of the basic shapes of airframes, optimal sensor placements are non-unique: for every optimal placement P and for every symmetry T, the placement $T(P)$ is also optimal. Therefore, we cannot find a *unique* optimal sensor placement. Instead, we must look for a *family* of optimal sensor placements (that correspond to different symmetries T).

3.4. So far, symmetries only made our problem more complicated, but symmetries can also help

So far, symmetries only made the optimal sensor placement problem more complicated. However, in general, symmetries are known to help in solving numerical problems.

For example, if we know that a solution $f(x, y, z)$ of a partial differential equation is invariant with respect to arbitrary rotations around 0, this means that the value of the desired function $f(x, y, z)$ depend only on a single parameter: distance $r = \sqrt{x^2 + y^2 + z^2}$ from 0. Therefore, instead of a *partial* differential equation that describes a function of three variables, we have a much easier-to-solve *regular* differential equation that describes an unknown function $f(r)$ of only one variable.

We will show that a similar simplification happens for the sensor placement problem.

3.5. Two-step approach

We start with a 2-dimensional geometric shape, and we want to find a finite number of points on this shape, i.e., in geometric terms, a 0-dimensional shape. Since moving directly from a 2D to 0D sets is complicated, we will do this transition in two, hopefully easier, steps – from 2D to 1D and then from 1D to 0D:

- On the first step, we will find a 1D curve or curves along which the optimal sensor placement will occur.
- Then, on the second step, we will find the optimal sensor placements on the chosen curves.

Let us first describe the first step.

3.6. First step: finding the optimal curve or curves on which sensors will be placed

Let us start with re-formulating our problem in precise mathematical terms.

3.6.1. What is given

We are given a geometric surface α that has several symmetries.

We can easily describe the set G of all these symmetries, i.e., in precise terms, the set of all geometric transformations (rotations, shifts, and dilations) that leave this surface invariant. Thus, we can assume that this set G is given as well.

3.6.2. An important comment about transformation groups

The set of transformations is very important because it is a particular case of a concept that plays a central role in modern theoretical physics: the concept of a *transformation group*.

Namely, it is easy to see that if transformations g and g' belong to this set G (i.e., leave the surface α invariant), then their composition $g \circ g'$ and the inverse transformation g^{-1} also leave the same surface invariant. A set of transformations that satisfies this property is called a *transformation group*.

3.6.3. The objective of the first step: from informal description to precise formulation

The goal of the first step is to find either a single curve or a family of curves that are optimal in some reasonable sense.

To describe this goal formally, we must explain which families of curves we will consider and what we mean by "optimal". Let is start with *families*.

In general, a *curve* can be described as a mapping that traces this curve, i.e., in more formal terms, a mapping $\vec{r}$ that maps real numbers t into points $\vec{r}(t)$ in 3D space. Correspondingly, a *family* of curves can be described as a family of such mappings characterized by one or several parameters $C_1, \ldots, C_p$, i.e., in more precise terms, as a mapping that maps tuples $(C_1, \ldots, C_p, t)$ of real numbers into points $\vec{r}(C_1, \ldots, C_p, t)$ of a 3D space. If we fix some values of p parameters $C_1, \ldots, C_p$, we get a curve from this family. (For example, the formula $\vec{r}(t) = (t, C_1 \cdot t + C_2)$ describes the family of all straight lines in a plane expect for the lines that are parallel to the y axis.)

How many parameters do we need in a sensor placement problem? In the simplest possible case of 1-parametric family ($p = 1$), the set of all points from all curves from this family already spans a 2D surface. Thus, we do not need more than one parameter to describe the lines of optimal sensor placements. So, in the following text, we will consider either single curves, or 1-parametric families of curves.

Now that we formalized the notion of a family, we must describe what "optimal" means. Here, the set of alternatives is the set of all curves (or of all 1-parametric families of curves) on the surface α. As before, the optimality criterion is a pre-ordering relation on this set for which there exists exactly one optimal curve (or family of curves).

We also want the optimality criterion to be *natural*, which means, in particular, that the relative quality of different placement curves should not change if we apply any transformation g that leaves the original surface α invariant. In precise terms, we require that the pre-ordering relation $\langle <, \sim \rangle$ that describes our optimality criterion satisfy the following two conditions:

- if $a > b$ and $g \in G$, then $g(a) > g(b)$;
- if $a \sim b$ and $g \in G$, then $g(a) \sim g(b)$.

A pre-ordering relation that satisfies these two conditions is called *invariant* with respect to the transformation group G.

So, we get the following precise formulation of the problem that correspond to the first step:

GIVEN: a surface α that is invariant with respect to a group G of geometric transformations.

WE KNOW: that on the set of all 1-parametric families of curves on a surface α, a pre-ordering relation is given that is invariant with respect to the transformation group G, and for which exactly one family is optimal.

FIND: the optimal 1-parametric family.

3.6.4. General solution to this problem

The problem, as formulated above, is a particular case of a *general problem* of finding optimal families of sets as formulated in the papers[1–3]. In these papers, we have actually solved this general problem by describing the *general solution* to it.

To formulate this general solution, we need to introduce two notions: of a *subgroup* and of an *orbit*.

- A *subgroup* G_0 of a transformation group G is a subset $G_0 \subseteq G_0$ that is itself a transformation group.

  For example, the set of all rotations around the x-axis is a subgroup of the group of all rotations.

- To describe an *orbit* of a transformation group G, we must fix a point $\vec{r}$. If we apply all transformation from G to this point $\vec{r}$, then the resulting set $\{g(\vec{r}) \,|\, g \in G\}$ is exactly the orbit.

  For example, for the group G of all rotations around the x-axis, depending on the choice of the point $\vec{r}$, we get either a point (if $\vec{r}$ is on this axis), or a circle circling around the axis.

In these terms, the above-mentioned solution is as follows: *Every set from the optimal family consists of one or several orbits of subgroups of the original transformation group.*

Thus, to apply this general solution to our problem, we must, for all the geometric shapes enumerated above:

- find all subgroups G_0 of the corresponding transformation groups G; and then
- find all orbits of these subgroups.

This is a (somewhat tedious but) doable task. The results are as follows:

3.6.5. The results of Step 1: Optimal curves for sensor placement

- For a *cylinder*, possibly optimal curves (i.e., orbits of subgroups) are:
  - *straight lines* parallel to the cylinder's axis;
  - *circles* orthogonal to the cylinder's axis; and
  - *cylindric spirals.*

    These spirals can be easily described in cylindric coordinates (z, ρ, φ), in which z is a coordinate along the cylinder's axis, ρ is a distance from this axis, and φ is an angle from some fixed direction orthogonal to the z-axis. In these coordinates, a spiral takes the form $\rho = \text{const}$ and $\varphi = k \cdot z$, for some constant k.

  Cylindric spirals are *generic* orbits; straight lines and circles can be viewed as their *degenerate* cases.

- For a *plane*, possibly optimal curves (i.e., orbits of subgroups) are:
  - *straight lines*;
  - *circles*;
  - *logarithmic spirals*, i.e., curves describe by the equation $\rho = C \cdot \exp(k \cdot \varphi)$ in polar coordinates.

  Here, logarithmic spiral is a generic shape.

- For a *sphere*, possibly optimal curves (i.e., orbits of subgroups) are *circles.*

- For a *cone*, possibly optimal curves (i.e., orbits of subgroups) are:
  - *straight line rays* going from the vertex of the cone;
  - *circles* that are orthogonal to the cone's axis; and
  - *conic spirals.*

    In cylindrical coordinates (z, ρ, φ), in which the cone is described by the equation $\rho = C \cdot z$, a conic spiral is described by the formula $\varphi = k \cdot z$ for some constant k.

  Conic spirals are the generic type of orbits.

Therefore, depending on the shape, sensors should be placed along one or several of these curves.

3.6.6. Important comments

1. If the optimal sensor placement is not along a single curve, but along *several* curves, then the same ideas of transformation groups can be used to choose appropriate families (as orbits of *discrete* subgroups). Let us give a few examples:

- If we have several straight lines on the cylinder, these straight lines must be *equidistant* in the sense that the angular distant between every two neighboring lines is the same.
- If we have several circles around the cylinder, then these circles should be equidistant.
- If we have several straight lines on a plane, then these straight lines should be:
  - either parallel and equidistant,
  - or parallel at distances that form a geometric progression, or
  - pass through the same point and form equidistant angles.
- If we have several circles on a plane, then these circles must be:
  - either parallel, equal, and equidistant,
  - or concentric, with their radii forming a geometric progression, etc.

2. In *space structures*, we face yet another shape: a paraboloid ($y = z = c \cdot x^2$). This structure is invariant with respect to rotations and *re-scalings* $x' = \lambda x$, $y' = \lambda^2 y$, $z' = \lambda^2 z$. For this group, we can also describe the resulting orbits as spirals ($\rho = C \cdot \exp(k \cdot \varphi)$ in cylindric coordinates).

3.7. Second step: finding the actual sensor placements (main idea)

The problem that corresponds to the second step, i.e., the problem of selecting a 0D subset from a 1D curve can be formulated and solve in a similar manner as the problem that we solved at the first step.

- We started with a surface α with a transformation group G.
- On the first step, optimal curves for sensor placements from orbits of subgroups G_0 of this group G.
- Similarly, on the second step, optimal sensor placements form orbits of subgroups G_1 of the corresponding groups G_0.

From the mathematical viewpoint, the main difference between these two steps is that on the second step, we start alreday with a 1D transformation group G_0 and thus, its subgroups G_1 are *discrete*. Thus, we face the problem of describing all orbits of discrete subgroups of the above groups.

Due to lack of space, we are not able to enumerate all possible orbits of this type here, but we will briefly enumerate the ones that correspond to generic curves: on a cylinder, we get equidistant points on a cylindric curve; on a plane and on a cone, we get points on the corresponding spiral whose distances from the center of this spiral form a geometric progression.

In all these families, there are still a few parameters whose choice depends on what exactly our goal is. The specific values of these parameters are determine by a neural network.

4. Neural networks: in brief

Ideally, we should test different sensor placements on different fault locations using the precise model an aerospace structure. However, such a precise model is very computationally intensive, so this direct way is intractable. Instead, we train a neural network to describe the structure, and then test different sensor placements by using this easier-to-compute neural model.

Acknowledgments

This research was supported by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

V. Kreinovich was also partly supported by NASA under cooperative agreement NCCW-0089, and by NSF grants No. DUE-9750858 and No. EEC-9322370.

References

1. A. Finkelstein, O. Kosheleva, and V. Kreinovich, "Astrogeometry, error estimation, and other applications of set-valued analysis", ACM SIGNUM Newsletter 31(4): 3–25, 1996.

2. A. Finkelstein, O. Kosheleva, and V. Kreinovich, "Astrogeometry: towards mathematical foundations", International Journal of Theoretical Physics 36(4): 1009–1020, 1997.

3. A. Finkelstein, O. Kosheleva, and V. Kreinovich, "Astrogeometry: geometry explains shapes of celestial bodies", Geombinatorics 6(4): 125–139, 1997.

4. R. Osegueda, C. Ferregut, M. J. George, J. M. Gutierrez, and V. Kreinovich, "Non-Equilibrium Thermodynamics Explains Semiotic Shapes: Applications to Astronomy and to Non-Destructive Testing of Aerospace Systems", *Proceedings of the International Conference on Intelligent Systems and Semiotics (ISAS'97)*, National Institute of Standards and Technology Publ., Gaithersburg, MD, 1997.

5. R. Osegueda, C. Ferregut, M. J. George, J. M. Gutierrez, and V. Kreinovich, "Maximum Entropy Approach to Optimal Sensor Placement for Aerospace Non-Destructive Testing", *The Seventeenth International Workshop on Maximum Entropy and Bayesian Methods*, August 3–7, 1997, Boise, Idaho, Abstracts (available at http://cme.idbsu.edu/conf); full text will appear in G. Erickson (ed.), *Maximum Entropy and Bayesian Methods 1997*, Kluwer, Dordrecht, 1997.