GPGPU-based Highly Parallelized 3D Node Localization for Real-Time 3D Model Reproduction

Kazuki Hirosue
Osaka University
Suita, Osaka, Japan
hirosue.kazuki@ist.osaka-u.ac.jp

Shohei Ukawa
Osaka University
Suita, Osaka, Japan

Yuichi Itoh
Osaka University
Suita, Osaka, Japan
itoh@ist.osaka-u.ac.jp

Takao Onoye
Osaka University
Suita, Osaka, Japan
onoye@ist.osaka-u.ac.jp

Masanori Hashimoto
Osaka University
Suita, Osaka, Japan
hasimoto@ist.osaka-u.ac.jp

ABSTRACT
This paper proposes a highly parallelized 3D node localization method based on cross-entropy method for the 3D modeling system. Cross-entropy localization statistically estimates node positions from node-to-node distance information by sampling, and each sample evaluation and internal computation of objective function can be processed in parallel. Experimental results show our GPGPU-based implementation achieved 5,163x and 61.5x speed up compared to a single processor and 80-processor implementations. In addition, for enhancing model reproduction accuracy, this work introduces a penalty function to mitigate flip ambiguity.

ACM Classification Keywords
H.5.2. User Interfaces: Input devices and strategies

Author Keywords
3D modeling; Node localization; Wireless sensor network; Cross-entropy method; Parallel computing

INTRODUCTION
With advancement in hardware and IT technology, 3D models are used to improve user experience not only in professional services/contents but also semi-professional or even amateur ones. For supporting 3D modeling, a number of commercial 3D modeling software is available. On the other hand, since 3D modeling is first established for professional use, those software is designed for professionals in terms of performance and usability, and hence it is not suitable for most of people. An intuitive scheme of 3D modeling is demanded by non-expert people.

IUI 2017, March 13-16, 2017, Limassol, Cyprus

DOI: http://dx.doi.org/10.1145/3025171.3025183

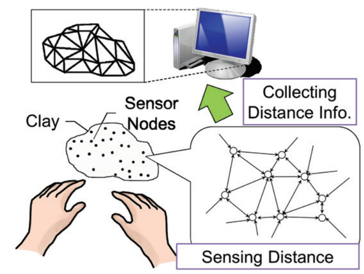

Figure 1. iClay system for 3D modeling.

For providing an intuitive modeling environment, Shinada et al. proposed a concept of three-dimensional shape reproduction system that builds a sensor network by embedding sensor nodes into deformable clay, where this system was named "iClay" [1]. Figure 1 illustrates iClay system. A number of 1mm^3-class tiny sensor nodes are distributed in the clay and they construct a wireless sensor network. This system measures distances between sensor nodes located in the vicinity, gathers the distance information through the sensor network, and reproduces the clay shape based on the estimated relative positions of the sensor nodes. The expected advantages of iClay system, compared to 3D scanner and laser range finder, are occlusion-free shape acquisition and real-time object reproduction, which enables us to use iClay as a tangible user interface and develop new human computer interaction. We are working for actualizing this system in terms of wireless power transmission [2], wireless signal transmission [3] and node localization [4].

To implement a prototype system of iClay, one of major difficulties is node localization for real-time 3D model reproduction since the number of nodes is large and the given distance estimates include large errors. To overcome these issues, Ukawa et al. proposed to apply cross-entropy method for 3D node localization [4]. This method statistically estimates node positions based on many samples generated according to probability distributions, where a sample is a set of node localization result. The probability distributions indicate prospective node locations, and they are iteratively updated by selecting top samples that are the most consistent with the given distance information, which is helpful to overcome large distance error problem. In addition, the sampling based approach is highly compatible with parallel computing, and [4] showed multiprocessor-based implementations achieved 22x CPU time reduction. However, this processing speed is not enough for real-time node localization.

In this paper, we propose GPGPU-based implementation that can exploit not only sample-level but also node-level parallelism in objective function computation. As the number of processing elements increases from 80 (multi-processor) to 2048 (GPGPU), the throughput of node localization is improved by 61.5x and now the node positions can be updated 16 times per second. A significant progress was made for the real-time clay shape reconstruction. Furthermore, we improved the node localization accuracy by introducing a penalty term to the objective function for preventing flip-induced accuracy degradation.

POTENTIAL APPLICATIONS

The advantage of iClay system is that users can construct 3D model by touching an actual object in real-time. This makes it possible for users to concentrate on the shape construction without being bothered by software manipulation. With this advantage iClay system is intended to be used for, for example, kids interactive education and rehabilitation training for brain disease patients. In one of the firstly mentioned applications, a computer shows an object shape from several different viewpoints on a 2D screen and asks kids to construct the corresponding 3D shape for developing the capability of object recognition in 3D space. By introducing game property, we expect kids are willing to do exercises with enjoyment. Related the second application, [5] showed that 5cm$\times$5cm block devices whose connected structure can be reproduced on a computer are helpful to diagnose Alzheimer disease and [6] showed about developmental coordination disorder.

We will further explore potential applications that take advantage of the real-time interaction with iClay system, and investigate application-wise requirements for future iClay system development.

PROBLEM FORMULATION AND CROSS-ENTROPY BASED LOCALIZATION

Problem formulation

For a sensor network consisting of N nodes, a set of node-to-node distance information is given, where d_{ij} is the distance between the i-th node to j-th node. This distance information includes measurement error. Also, the measurable range is limited. When the i-th node is distant from j-th node, d_{ij} cannot be measured. In this case, d_{ij} is set to 0. Based on this distance information, the location of q_i is estimated for N nodes .

Here, the distance information includes measurement error, and hence the exact location cannot be computed. What we

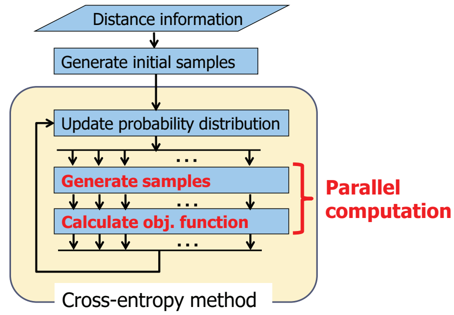

Figure 2. Cross-entropy based localization method.

can do is to compute the most probable locations. In this context, the sum of localization errors (dRMS) between the true node position and estimated node position should be minimized. The dRMS error is computed by comparing all the pairwise distances, and is defined as

$$dRMS^2 = \frac{1}{N^2}\sum_i^N\sum_j^N (||p_i - p_j|| - ||q_i - q_j||)^2, \quad (1)$$

where p_i is the true position of the i-th node in the absolute three-dimensional coordinates. q_i is also the estimated position in the absolute three-dimensional coordinates. $||p_i - p_j||$ means the Euclidean distance between p_i to p_j.

On the other hand, the true node positions are, of course, not available during localization, and hence dRMS cannot be computed. Instead, [4] proposed to use the given node-to-node distance information d_{ij} instead of $||p_i - p_j||$ while d_{ij} includes measurement error.

$$eval^2 = \frac{1}{n^2}\sum_i^N\sum_{j,d_{ij}\neq 0}^N (d_{ij} - ||q_i - q_j||)^2. \quad (2)$$

Now, the node localization problem corresponds to finding q_i which minimizes $eval$.

Localization with cross-entropy method

[4, 7] showed that cross-entropy method [8] works well for the localization. Here, the localization method proposed by Ukawa et al. [4], which is illustrated in Fig. 2, is briefly explained since the sample-level parallel computing is explicitly considered in [4]. First, a set of initial solutions are generated referring to [9], and they are given to the cross-entropy based iterative estimation.

The iteration loop mainly consists of the following two steps.

- Generate N_{gen} samples, each of which is a set of prospective node positions, according to the present probability distributions of individual node positions, and compute the objective function of Eq. (2) for evaluating the consistency between the set of node positions and the given distance information.

- Choose N_{top} samples that have better consistency and update the probability distributions of node positions based on the selected samples.

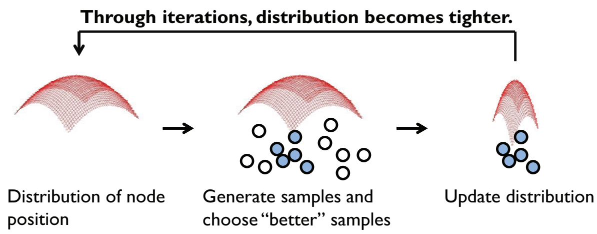

Figure 3. Solutions converge through iteration.

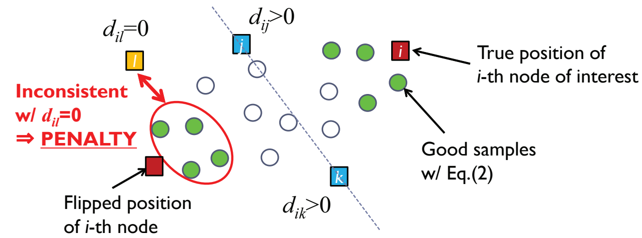

Figure 4. Flip ambiguity and penalty.

This iteration is expected to converge to the globally optimal solution as illustrated in Fig. 3. Initially, the distributions are wide, and the solution space is widely explored. As the iteration proceeds, the distributions of node positions become tighter, and finally the local optimization is carried out. The node position distributions are assumed to follow Gaussian distribution.

Here, it should be noted that the computation of Eq. (2) for each sample is totally independent. This means that it can be computed in parallel. This sample-level parallelism is exploited in [4] on multi-processor computers to reduce CPU time of cross-entropy based localization.

This cross-entropy based iteration repeats unlimitedly for coping with the model transformation by an iClay user. [4] expands the distributions once the model transformation is detected to explore wider solution space. In this case, the number of iterations per second is important to reflect the model transformation in real-time. However, in case of the multi-processor based implementation in [4], one iteration needs 3.77 seconds and the real-time model reproduction is impossible.

Another problem we found in [4] is the poor accuracy of the model reconstruction due to the flip ambiguity [10]. A mechanism to mitigate the flip ambiguity must be introduced in the node localization.

PROPOSED LOCALIZATION METHOD

This section describes the proposed method that overcomes the two problems pointed out in the previous section; accuracy and computation time per iteration. The following subsections present countermeasures to these two problems.

Accuracy improvement with penalty term

In the objective function of Eq.(2), the information of which node-to-node distance d_{ij} is larger than 0 is used. On the other hand, as mentioned in Section 3.1, the node-to-node distance which is larger than the measurable range D is set to 0. Therefore, d_{ij}=0 includes the information that the distance between i-th and j-th nodes is larger than D.

We use this information to mitigate the flip ambiguity. Let us show an illustrative example in Fig. 4. Suppose we are localizing the i-th node, where $d_{ij} > 0, d_{ik} > 0, d_{il} = 0$. Circles in the figure represent generated samples for i-th node. With Eq. (2), green samples are regarded as good samples. You can see the green samples in the left bottom are much distant from the true i-th node position. This is because we cannot determine whether the true position and the flip position across the dotted $j - k$ line should be selected only with the information of $d_{ij} (> 0)$ and $d_{ik} (> 0)$. This is the problem called flip ambiguity. This ambiguity often arises in wireless sensor network localization, and also in iClay system it arises near the model surface since the nodes are aligned on a plane and it is easier that the flip across the plane arises. On the other hand, we have more information of $d_{il} = 0$. With this information, we can know the green samples in the left bottom are not good samples since the distance to l-th node is too small. To exploit this information, we give penalty to these green samples not to be selected as good samples.

Then, we improve the objective function so that it includes a penalty term shown below.

$$new\ eval = eval + \frac{1}{N} \sum_{i} \sum_{j, d_{ij}=0, i \neq j} \left(exp \left(\frac{D - r_{ij}}{D} \right) \right)^2, \quad (3)$$

where D is the measurable range. This penalty term is 1 when $r_{ij} = D$, and it increases rapidly as r_{ij} decreases below D. Due to the measurement error, the i-th node in Fig. 4 can exist within the measurable range D from l-th node. However, as the i-th node and l-th node becomes closer, its possibility becomes smaller. This tendency can be considered in the penalty term of Eq. (3).

Parallel computation with GPU

As discussed in Section 3.2, the cross-entropy method has a good property of sample-level parallelism. When we implement the cross-entropy based localization for a multi-processor computer, the sample-level parallelism is good enough to fully utilize the computing power since the number of processors is generally smaller than the number of samples. On the other hand, when we implement it with GPGPU, the number of samples can be smaller than the number of processing elements. To fully exploit the computing power of GPGPU, we need to find further parallelism in the localization algorithm.

In the proposed implementation, we decompose the computation of the objective function into N threads, where N is the number of nodes. This decomposition can be achieved by unrolling the *for* loop of summation in the objective function, and it provides node-level parallelism. Figure 5 shows the conventional sample-level parallel computation with multi-processor computer and the proposed sample-level and node-level parallel computation with GPGPU. In

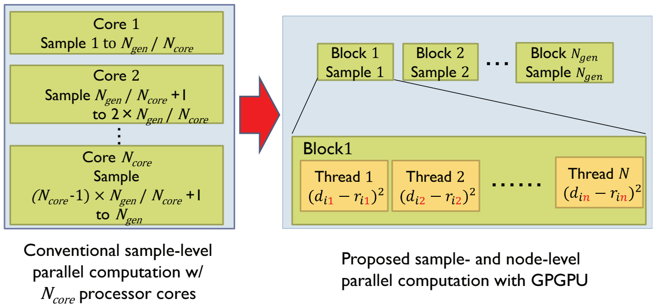

Figure 5. Proposed sample- and node-level parallel computation with GPGPU.

the multi-processor computer, N_{gen}/N_{core} samples are assigned to each processor core, where N_{gen} is the number of generated samples and N_{core} is the number of processor cores. On the other hand, in the GPGPU implementation, each sample is assigned to individual blocks, and each block includes N threads. All the threads can be processed in parallel, and hence massively parallel computation becomes possible. Note that if the product of N_{gen} and N exceeds the number of available processing units N_{PE}, the computation is folded sequentially into $(N_{gen} \times N)/N_{PE}$ steps.

EXPERIMENTAL RESULTS

This section evaluates the performance improvement in terms of node localization accuracy and computation time in comparison to [4]. The proposed method is implemented with MATLAB and its parallel computing toolbox for using GPGPU.

The experimental setup for clay and nodes is determined referring to [4] as follows. The clay volume is $50 \times 50 \times 50$ mm^3, and $343 (= 7^3)$ nodes (N=343) are distributed in the clay. It is assumed that each node can sense the distance to other nodes in a range of 20 mm, which means $D = 20$. The distance information given to this experiment includes the measurement error according to [1]. The number of generated samples N_{gen} is 1,000 and the number of good samples N_{top} is 100.

Accuracy improvement

We localized the nodes in a static object with Eq.(3), where a static object means its shape is unchanged. The original shape is a rectangular solid. Figure 6 (a) shows the shape reconstructed by the true node positions.

Figure 6 (b) and (c) show the shapes reconstructed by the node locations estimated by [4] and the proposed method, respectively. The proposed method contributes to more accurate shape reconstruction compared to [4], which is especially found at the edges of the rectangular shape. As explained in Section 4.1, the nodes near the surface suffer from the flip ambiguity, and hence the model of the conventional method (b) fails to reproduce the edges. In addition, the volume of the rectangular model (b) is much smaller than that of (a). On the other hand, the localization accuracy of nodes near the surface is improved in (c) by the improved objective function. The model volume of (c) becomes closer to that of (a).

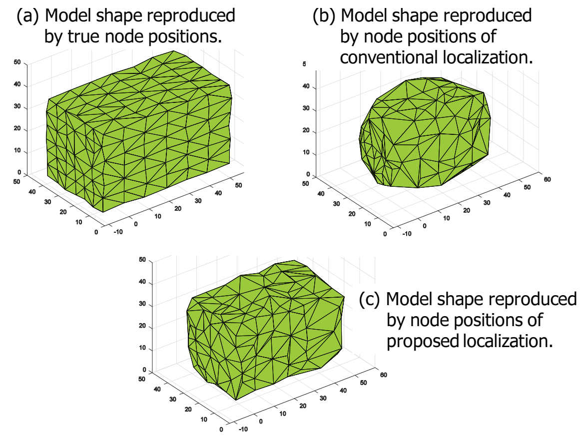

Figure 6. Reproduced models (Rectangular).

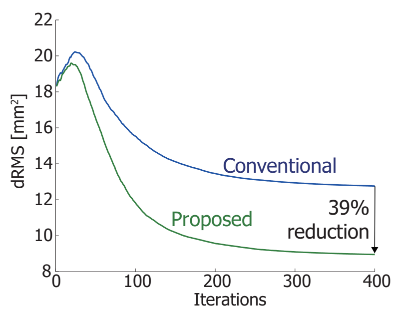

Figure 7. dRMS errors through cross-entropy iteratinos.

The overall dRMS error is evaluated. Figure 7 shows dRMS errors of the proposed method and [4] through the cross-entropy iteration. We confirmed that the proposed method reduced dRMS error after 400 iterations by 39% compared to [4], which confirms that the penalty function help mitigate flip ambiguity and improve the accuracy.

We carried out similar experiments for the shapes of tetrahedron and cylinder. Figures 8 (a), (b) and (c) show the reconstructed tetrahedron shapes. With the conventional localization, the vertexes in (b) are rounded and the reconstructed shape is much different from tetrahedron. On the other hand, the vertexes are reproduced by the proposed method in (c). The dRMS reduction is 36%. The remaining shape deterioration in Figure 8 (c) should be improved by increasing the number of nodes in the clay. The similar observation was found the reconstructed cylinder shapes in Figure 9. The dRMS reduction is 43%.

CPU time reduction

We used NVIDIA GTX980 GPGPU with CUDA-7.5 for the experiment. The number of processing units N_{PE} is 2,048. We implemented the parallelized localization method with CUDA and it is called from MATLAB. As for the conventional method of [4], we implemented it with MATLAB and it was executed at a computer that has four Xeon E7-8870 2.4GHz CPUs. The number of available processor cores is 80.

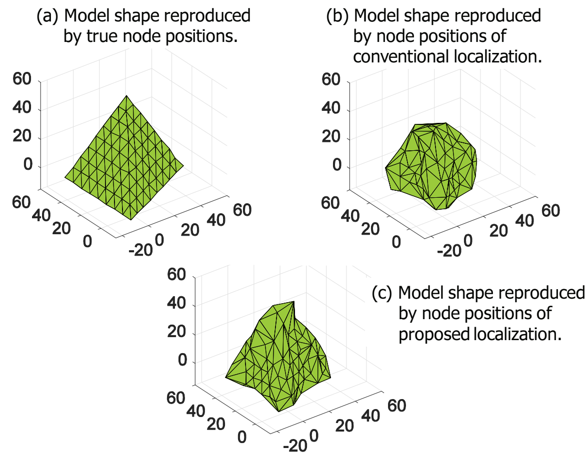

Figure 8. Reproduced models (Tetrahedron).

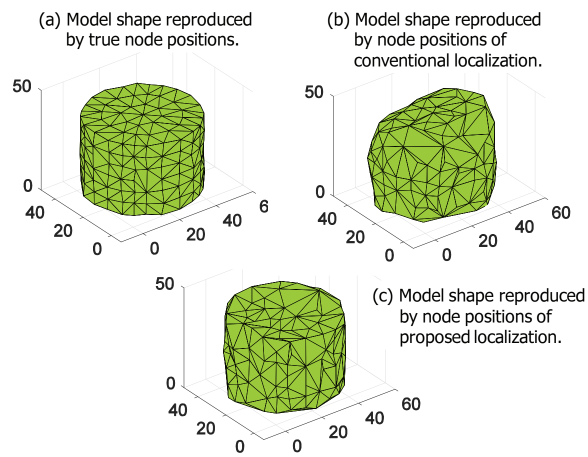

Figure 9. Reproduced models (Cylinder).

Figure 10 shows the computation times per iteration of [4] and the proposed method, where the iteration corresponds to an iteration in the cross-entropy method. The computation times per iteration of the conventional multi-processor implementation and the proposed GPGPU implementations are 3.77 and 6.10×10^{-4} seconds, respectively. We achieved 61.5x speed-up. Compared to the single processor implementation, 5,163x speed-up was attained. The proposed implementation performs 16.4 iterations per second, which could reflect the user shape modification in real-time, whereas only 0.3 iterations can be performed in a second by the conventional implementation.

RELATED WORKS

Reed presented a similar approach that embeds sensors into a deformable object [11]. It embedded locators that used magnetic field for sensing in clay and reproduced the digital model based on information of the absolute coordinate and the direction of each locator. However, the size of locators was $7\times4\times2\text{cm}^3$, and hence the clay shape could not be accurately captured. In addition, the real-time interaction was not possible.

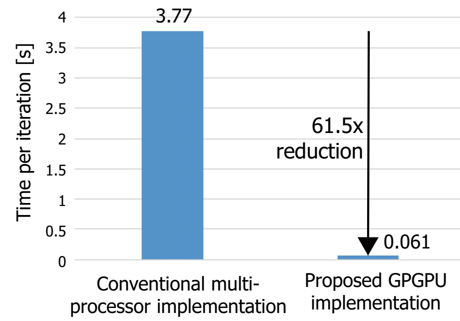

Figure 10. Computation times in a cycle.

Wireless sensor networks are expected to provide a variety of novel monitoring applications and are intensively studied. In some applications, node locations are needed to provide services, while the absolute locations cannot be directly sensed and only the distances to neighboring nodes are measurable. For such applications, node localization based on the node-to-node distance information is necessary. In most sensor networks, this node localization problem is approximated to a two-dimensional localization problem, since the height variation is much smaller than the node-to-node distance[12][13].

On the other hand, three-dimensional localization is also studied [9]. This approach sequentially estimates three-dimensional node position based on the distance information between nodes from the nodes that are already localized. This sequential localization, however, has a problem. Once a flipping mistake happens due to indeterminacy, it is likely that nodes localized after this mistake have a larger localization error or localization itself becomes impossible. To minimize this problem, [9] proposes an approach that a robust set of nodes which are less likely to cause fatal mistakes are first searched, and they are used for node localization. This concept of robust node set is first proposed by Moore for 2D problem [10] and it is extended to 3D problem in [9]. To find a robust node set, volume test and ambiguity test are performed before coordinate assignment and for each step of sequential node localization. The robust node test improves the node localization accuracy, but searching the next robust node at every sequential step spends tremendous computation time. In addition, as the node density becomes sparse, some nodes cannot satisfy the robust node test and they cannot be localization. Furthermore, this approach is a sequential approach and is not totally suitable for parallel computing.

CONCLUSION

This work presented a GPGPU-based localization method that exploits sample-level and node-level parallelism in the cross-entropy based node localization. Compared to conventional multi-processor implementation that uses only sample-level parallelism, 61.5x speed-up was attained by exploiting not only sample-level but also node-level parallelism in GPGPU implementation. This speed-up enables real-time model reconstruction in iClay system. We also improved the estimation accuracy by mitigating flip ambiguity. The model reproduced after the node localization was clearly improved especially near the edges. The overall RMS error was reduced by 39% and the fully worked on any other object.

ACKNOWLEDGMENT
This work was supported by JSPS KAKENHI Grant Number JP15H01679.

REFERENCES
1. Shinada, T., Hashimoto, M., and Onoye, T. Proximity Distance Estimation based on Electric Field Communication between 1mm^3 Sensor Nodes . In *Analog Integrated Circuits and Signal Processing* (2013).
2. Masuda, S., Hirose, T., Akihara, Y., Kuroki, N., Numa, M., and Hashimoto, M. Highly-Efficient Power Transmitter Coil Design for Small Wireless Sensor Nodes. In *Proc. of International Symposium on Antennas and Propagation (ISAP)* (2015).
3. Kono, J., Hashimoto, M., and Onoye, T. Feasibility Evaluation of Near-Field Communication in Clay with 1-mm^3 Antenna. In *Proc. of Asia-Pacific Microwave Conference (APMC)* (2013), 1121–1123.
4. Ukawa, S., Shinada, T., Hashimoto, M., Itoh, Y., and Onoye, T. 3D Node Localization from Node-to-Node Distance Information using Cross-Entropy Method. In *Proc. of Virtual Reality Conference (VR)* (2015), 303–304.
5. Sharlin, E., Itoh, Y., Watson, B., Kitamura, Y., Sutphen, S., and Liu, L. Cognitive cubes: a tangible user interface for cognitive assessment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2002), 347–354.
6. Jacoby, S., Gutwillig, G., Jacoby, D., Josman, N., Weiss, P. L., Koike, M., Itoh, Y., Kawai, N., Kitamura, Y., and Sharlin, E. Playcubes: Monitoring constructional ability in children using a tangible user interface and a playful virtual environment. In *2009 Virtual Rehabilitation International Conference*, IEEE (2009), 42–49.
7. Azim, M. A., Aung, Z., Xiao, W., and Khadkikar, V. Constrained cross entropy localization technique for wireless sensor networks. *International Journal of Distributed Sensor Networks 2015* (2015), 4.
8. Rubinstein, R. Y., and Kroese, D. P. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.
9. Mautz, R., Ochieng, W., Brodin, G., and Kemp, A. H. 3D Wireless Network Localization from Inconsistent Distance Observations. *Ad Hoc & Sensor Wireless Networks 3*, 2-3 (2007), 141–170.
10. Moore, D., Leonard, J., Rus, D., and Teller, S. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ACM (2004), 50–61.
11. Reed, M. Prototyping digital clay as an active material. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, ACM (2009), 339–342.
12. Savvides, A., Park, H., and Srivastava, M. B. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, ACM (New York, NY, USA, 2002), 112–121.
13. Doherty, L., pister, K., and El Ghaoui, L. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3 (2001), 1655–1663 vol.3.