Research Statement

Li-Yi Wei

Repetition is an integral part of nature. Hardly can we go by our daily lives without encountering repetitive structures or activities. Being such a ubiquitous and fundamental phenomenon, repetition has been an important subject of study in a variety of science and engineering disciplines. In a sense, computer science existed primarily for dealing with repetitive computations. However, most prior methods can handle only *artificial repetitions* which may lack proper variations commonly see in nature. Thus, one of my main research directions is to design general and fundamental algorithms that can compute **natural repetitions** within a variety of application contexts. This endeavor can not only help us understand how the nature works but also build better and more powerful computational tools, involving knowledge from and applications to multiple disciplines such as computer graphics, vision, multi-media, sampling theory, human computer interaction, and parallel computing.

In addition to algorithm design, my industry experience also enabled me to apply research to products, both internally with NVIDIA and Microsoft and externally to other companies such as Weta Digital and Bank of America or open source projects such as CUDPP, the CUDA Data Parallel Primitives Library.

1 Computing Natural Repetitions

A variety of natural phenomena consist of repetitions within distinctive structures. Some common examples include random numbers (e.g. physical measures of temperature or conductivity), surface textures (e.g. carpets, wall papers and tiled floors of a building), geometry elements (e.g. food piles, pebble beach, grass lawn, tree leaves, cityscapes, and biological cells), and motions (e.g. bipedal walking, fluid turbulence, and crowd/traffic flows). Being such a ubiquitous and fundamental phenomenon, natural repetition has been an important subject of study in several art, science, and engineering disciplines.

For both the generative and descriptive purposes, it is usually preferable for the users to fully specify the distinctive structures while only tersely denote the repetitions. The existing generative tools and descriptive languages can already handle the distinctive structures quite well, but less so for repetitions due to their versatility and complexity. For example, even though current CAD tools can be used to design almost any building or machinery one has in mind, it is still difficult to use a single tool to specify all detailed repetitions. Often, users have to resort to either tedious manual work or different specialized tools designed for individual effects.

One of my primary research interests is to design general, simple, efficient, and easy to use algorithms that can compute a variety of natural repetitions with different properties and complexities that also integrate well with the overall domain contexts. Below, I will describe more details, including my past and current work as well as future plan. For clarify, I will divide the approaches into major categories: *procedural*, *data-driven*, and *hybrid*.

1.1 Procedural Approach

Several fundamental classes of natural repetitions have widespread and important usages across multiple disciplines, such as white noise (i.e. uniform random numbers) and blue noise (i.e. random but domain-uniform distributions). Such fundamental distributions also have very simple properties and definitions and thus could be generated by purely procedural methods, in contrast to data-driven methods that require user exemplars.

White noise Pseudo (uniform) random number generators are essential for any computation tasks that require simulating randomness in nature. They are part of almost all standard programming environments and a variety of different methods have been proposed over the years in the hope to achieve high randomness and fast computation [Knuth 1998]. However, most existing methods are sequential in which the output numbers have to be produced one after another. This presents issues in not only efficiency (not parallelizable) but also usage (not random-accessible). Furthermore, some of the prior methods have quality issues where the generated outputs might not be random enough.

I have proposed a parallel pseudo random number generator [Wei 2004a; Tzeng and Wei 2008] that can compute each number independently while maintaining high aggregate randomness. The basic idea is to apply cryptographic hash to individual numbers of an input sequence that can be prepared trivially in parallel (e.g. a linear ramp). Due to the properties of cryptographic hash, the output sequence is guaranteed to have high randomness. I obtained this idea as a synergy between graphics hardware design [Wei 2004a] and cryptography, a subject that I happened to be studying as a hobby around the same time I was architecting NVIDIA chips.

Blue noise A blue noise distribution contains samples that are randomly located but remain spatially uniform. This is a general category, and contains several well known methods used in different disciplines, such as Poisson disk distribution [Cook 1986] for stochastic sampling and image synthesis, as well as Lloyd relaxation [Lloyd 1983] for quantization and image processing.

Despite its importance, previous blue noise methods often have limited capabilities, e.g. applicable to only one class of samples, only to isotropic distribution, or only to Euclidean sample space. I have designed algorithms to generalize various aspects of blue noise distribution, including multiple classes of samples [Wei 2010], samples with anisotropic distribution [Li et al. 2010], and analysis + synthesis on Riemannian domains such as surfaces [Bowers et al. 2010]. These generalizations extend the applicability of blue noise into wider application domains. Inspired by the presence of multiple classes of cone and rod cells in human retinas, multi-class blue noise sampling [Wei 2010] can be applied for placing multiple categories of objects (e.g. pattern design or biological distribution), color stippling (e.g. pointillism effects in paintings), or sensor layout (e.g. RGB color sensors in cameras). The consideration of anisotropy [Li et al. 2010] enables us to apply blue noise distribution to anisotropic scenarios, such as the placement of anisotropic glyphs for vector field visualization. The methods in [Bowers et al. 2010] not only allows generation of samples over manifold surfaces but also the standard Fourier spectrum analysis method that previously has been restricted to regular planar domains.

Future work Many well known procedural distribution methods are based on heuristics. Even though their properties have been verified by methods such as Fourier spectrum analysis, there is a lack of fundamental theoretical understanding in general. I am current pursuing this direction, starting with the mathematical connection between spatial and spectral properties of various sampling methods [Wei and Wang 2011]. I believe such a more fundamental understanding can lead to better analysis tools as well as better synthesis methods with a wider range of applications.

1.2 Data Driven Approach

Even though procedural methods have been applied for computing several important classes of distributions, they are often not suitable for general scenarios with diverse application characteristics. Thus, I have spent significant efforts focusing on *data driven* methods, for which a single algorithm can handle a variety of situations depending on the specific user inputs. For example, to design patterns for a building interior, the user should only need to supply a few small swaths of exemplars and specify which regions each exemplar is for (e.g. walls, floors, and ceilings) and the algorithm should automatically produce outputs that covers the entire building interior satisfying the user specifications. My goal is to design algorithms that are general, effective, efficient, and easy to use. To achieve this, I have proceeded to first design fundamental algorithms that capture commonality among natural repetitions. I then build upon these basic algorithms more specialized and advanced features for different application domains.

Fundamentals I first focus on a specific kind of natural repetition, 2D textures, which are a common visual phenomenon and have been an important subject of study in computer graphics, computer vision, and image processing. The first task is to figure out how to effectively and efficiently synthesize an output texture with arbitrary size and shape from a small user exemplar. For this I have developed a very simple algorithm [Wei 1999; Wei and Levoy 2000] that uses local and fixed neighborhoods to characterize textures. Due to natural repetitions, this approach works very well. The next step is to consider the situation where the output can reside in a different domain from the input, e.g. manifold surfaces versus planar grids [Wei and Levoy 2001]. Both of these early works have spawned hundreds of subsequent methods. I have also developed an inverse texture synthesis algorithm [Wei et al. 2008] that runs in the opposite direction of traditional forward synthesis, for computing a compact summarization of potentially large textures acquired through scanning. Given the proliferation of large textures in both 2D images and 3D scenes, this serves a useful technique for data reduction, comparison, and summarization.

Extensions The fundamental algorithms that I have developed facilitate further extensions for different problem domains and applications. This includes a numerous list of work that has been partially surveyed in [Wei et al. 2009a]. Some of my efforts include parallelization [Wei 2002], synthesis from multiple sources to achieve effects like texture mixtures and cross dimensional synthesis [Wei 2003], globally varying synthesis for natural effects depending on environment factors such as rusting on a metal sculpture [Lu et al. 2007], real-time rendering on graphics hardware [Wei 2004b], interactive image editing to fix missing details caused by high dynamic range [Wang et al. 2007b], generating potentially artistic and non-physics-based fluid motion [Ma et al. 2009], and modeling 3D object distributions [Ma et al. 2011].

Future work I believe enough fundamental infrastructures have already existed to enable data driven synthesis to be applied to a wide variety of applications. One direction that I plan to take on is to combine physical simulation and data driven detail synthesis. Specifically, even though physical simulation can produce realistic effects, they are often computationally expensive and hard to control. I have taken several initiatives in fluid animation [Ma et al. 2009] and 3D modeling [Ma et al. 2011] while others have pursued other possibilities such as cloth wrinkles [Wang et al. 2010]. I anticipate exciting opportunities in this direction.

1.3 Future Directions

Hybrid approach There have been many methods proposed for both procedural and data-driven repetition synthesis and each camp has its relative pros and cons. One general future direction that I would like to pursue is a hybrid approach, combining the strengths of both camps. Recent advances have shown promise for such a hybrid approach and I believe my interest and knowledge put me in a good position to be productive in this pursuit.

Analysis & reduction The flip side of synthesis would be analysis. Even though I have been mainly focused on the former, I have not only worked on the latter (including texture compression [Wang et al. 2007a], texture reduction [Wei et al. 2008], and interaction summarization [Chen et al. 2011]) but also considered it an important future direction. In a nutshell, we live in an age of data deluge, with the amount of information, either captured or computed, increasing in an ever rapid speed. Thus, the ability to manage, analyze, search, and reduce large amount of information would be an important future direction. Since natural repetition is a ubiquitous phenomenon, I believe it could be harnessed for such analysis & reduction tasks with potential applications including search, compression, and data mining.

2 Parallelism

Parallelism is another interest that I derived from my background in computing natural repetitions as well as designing + programming parallel architectures. Due to the sheer complexity of many natural phenomena, computing detailed repetitions can be time and/or memory consuming. Parallel computing comes not only as a potent rescue but also a natural solution due to the repetitive nature of computations.

Traditional parallel computing focuses mainly on algorithms with *sequential* consistency, where results produced by a parallel algorithm should be equivalent to those produced by the original sequential algorithm. However, for certain applications such sequential consistency might be too strict, as the final results might only need to be statistically or perceptually correct. Some examples include pseudo random number generation as well as image synthesis in graphics. For these applications, I would like to propose a looser model, termed *semantic* consistency [Wei 2009], that can facilitate faster and more flexible parallelization not possible under traditional consistency models.

I have conducted several works based on the notion of semantic consistency. One is a parallel random number generator via cryptographic hash [Tzeng and Wei 2008] that I have described before. Since most applications only need the output numbers to be of high randomness instead of obeying specific rules, semantic consistency is a better and more flexible model than sequential consistency. Another example is Poisson disk sampling [Cook 1986], a particular form of blue noise distribution where samples are randomly located but remain at least a minimum distance away from each other. The standard method for generating such a distribution is dart throwing, where samples are generated one by one so that each subsequent sample is sufficiently far away from all existing ones. Dart throwing is inherently sequential. To overcome this issue, I have figured out a parallel algorithm for dart throwing [Wei 2008; Bowers et al. 2010]. My basic idea is to partition the sample domain into cells and draw samples concurrently among cells that are sufficiently far apart so that their containing samples are guaranteed to be sufficiently far away. This idea was inspired by a random event that someone (whom I previous did not know) sent me a copy of his Ph.D. dissertation related to Poisson disk distribution. I read about it, and realized that the process can be parallelized similar to how I parallelize texture synthesis [Wei 2002].

Future work A main future direction I plan to investigate is to formalize the notion of semantic consistency and apply it to classical algorithms that could benefit from a more flexible parallelization, followed by proper categorization [Asanovic et al. 2006]. I would also like to work on the related issue of real-time computing [Wei 2004b; Hou et al. 2006; Wu et al. 2006; Wang et al. 2007a; Wei et al. 2009b] and hardware architecture [Wei 2004a; Chen and Wei 2009].

3 Future Work

In addition to what mentioned above, I would like to embark on several major future directions that could benefit from my past experience as well as have potential impacts on both academic research and industrial practice.

3.1 Imaginative computing

One of the most powerful features of a virtual world is that we do not have to obey any real-world laws. What we can do is only limited only by our imaginations. This feature has been pursued in various science fictions, movies, arts, and games. For computer graphics, a significant portion of efforts has been devoted for reproducing realism. There are more recent efforts devoted to *non-photorealistic animation and rendering* [NPAR 2011], but these are still most about reproducing physical art forms, which are still part of the real world.

I believe a potentially very interesting direction is *imaginative computing*, in which we utilize computational methods to produce imaginative effects that are not possible or not yet thought of via real-world mediums. For example, in a virtual world, it is cool to produce realistic fluid motions and there has been a large body of work based on fluid dynamics [Bridson and Müller-Fischer 2007]. However, why limiting ourselves to physical fluids? We can produce physically impossible/infeasible motions with artistic effects such as heart shaped swirls as in [Ma et al. 2009]. Or, it is cool to produce realistic geometrical structures in a virtual world through physical or procedural simulations. But as shown in [Cho et al. 2007], sometimes it is desirable to have physically unstable configurations. I have shown how to achieve both physically plausible and non-plausible geometrical element distributions in [Ma et al. 2011]. A common fundamental principle for both animation in [Ma et al. 2009] and modeling in [Ma et al. 2011] is a hybrid solver that combines both physical simulation (fluid simulation in [Ma et al. 2009] and dynamic simulation in [Ma et al. 2011]) and data-driven computation (motion field texture synthesis in [Ma et al. 2009] and discrete element texture synthesis in [Ma et al. 2011]) to achieve both realistic and artistic effects. I believe such hybrid solver approach can be applicable to produce many other imaginative effects in modeling, animation, rendering, and imaging.

3.2 Assistive computing

Content creation and interaction is one of the central human activities. Ideally, the ability to create content should be limited only by our imagination instead of technologies, and once the content is created, it should be easy to edit, share, and use. While technologies have enabled textual content relative easy to create and interact with, this is not yet so for other kinds of contents, such as images, videos, models, and games. For example, even though a talented individual can single-handedly author an entire novel series, both natural and technological factors still present significant obstacles for turning these stories into movies or games.

I believe a fruitful future direction to design computational methods to facilitate such content creation and interaction, with a particular focus on simplicity, generality, and effectiveness. For this, it is vital to distinguish between unique tasks that require human creativity and discretion, and repetitive tasks that are well suited for computer automation. I have devoted significant research effort in such *assistive computing*, such as recovering details in ill-exposed photo regions from nearby regions with similar textures [Wang et al. 2007b], reducing repetitive sampling artifacts in digital painting via proper sample map design [Chu et al. 2010], and a non-linear revision control system for images that can encode and replay editing history [Chen et al. 2011].

Of course, all such assistive computational tasks also closely involve human computer interaction. Thus, in addition to algorithms, an equally important component for all such work [Wang et al. 2007b; Chu et al. 2010; Chen et al. 2011] is a proper user interface design. Despite the impressive advances in processor speeds and memory capacity, the computer interfaces have been seriously lacking. The main stream interfaces have been mostly about mouse and keyboard for a long time, and even touch screens, which can provide more intuitive interactions, still have a long way to go for a sufficiently natural user interface. As a step towards this goal, I have started an interdisciplinary project with Jaron Lanier on "somatic computing" [Lanier and Wei 2010], with the goal to map human body movements into interactive or computational gestures for a variety of applications in gaming, education, and science.

Some people like to say computer graphics is a relatively mature field and there is not much left to be done. I believe it is never really so until one day we can provide authoring tools that can allow average people to create graphics contents such as images, models, and animations as easily as they would for text documents. Content creation should be limited only by our imaginations, not tools.

Acknowledgement

I would like to thank all my collaborators for making me wiser, happier, tougher, healthier, lovelier, and in general, better.

This document is a perpetual work in progress (last updated on April 21, 2011).

References

ASANOVIC, K., BODIK, R., CATANZARO, B. C., GEBIS, J. J., HUSBANDS, P., KEUTZER, K., PATTERSON, D. A., PLISHKER, W. L., SHALF, J., WILLIAMS, S. W., AND YELICK, K. A. 2006. The landscape of parallel computing research: A view from berkeley. Tech. Rep. UCB/EECS-2006-183, EECS Department, University of California, Berkeley.

BOWERS, J., WANG, R., WEI, L.-Y., AND MALETZ, D. 2010. Parallel Poisson disk sampling with spectrum analysis on surfaces. In *SIGGRAPH Asia '10 Papers*, 166:1–10.

BRIDSON, R., AND MÜLLER-FISCHER, M. 2007. Fluid simulation: Siggraph 2007 course notes. In *SIGGRAPH 2007 courses*, 1–81.

CHEN, W.-C., AND WEI, L.-Y., 2009. A method and system for processing texture samples with programmable filter weights. U.S. Patent 7623136 granted November 24, 2009.

CHEN, H.-T., WEI, L.-Y., AND CHANG, C.-F. 2011. Nonlinear revision control for images. In *SIGGRAPH '11 Papers*.

CHO, J. H., XENAKIS, A., GRONSKY, S., AND SHAH, A. 2007. Course 6: Anyone can cook: inside ratatouille's kitchen. In *SIGGRAPH 2007 Courses*.

CHU, N., BAXTER, W., WEI, L.-Y., AND GOVINDARAJU, N. 2010. Detail-preserving paint modeling for 3d brushes. In *NPAR '10: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*.

COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1, 51–72.

HOU, X., WEI, L.-Y., SHUM, H.-Y., AND GUO, B. 2006. Real-time multi-perspective rendering on graphics hardware. In *Eurographics Symposium on Rendering '06*, 93–102.

HUANG, J., SHI, X., LIU, X., ZHOU, K., WEI, L.-Y., TENG, S.-H., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Subspace gradient domain mesh deformation. In *SIGGRAPH '06 Papers*, 1126–1134.

KNUTH, D. 1998. *The Art of Computer Programming*. Addison-Wesley Publishing Company.

LANIER, J., AND WEI, L.-Y., 2010. Somatic computing. `http://research.microsoft.com/en-us/projects/somatic/`.

LI, H., WEI, L.-Y., SANDER, P. V., AND FU, C.-W. 2010. Anisotropic blue noise sampling. In *SIGGRAPH Asia '10 Papers*, 167:1–11.

LLOYD, S. 1983. An optimization approach to relaxation labeling algorithms. *Image and Vision Computing 1*, 2.

LU, J., GEORGHIADES, A. S., GLASER, A., WU, H., WEI, L.-Y., GUO, B., DORSEY, J., AND RUSHMEIER, H. 2007. Context-aware textures. *ACM Trans. Graph. 26*, 1, 3.

MA, C., WEI, L.-Y., GUO, B., AND ZHOU, K. 2009. Motion field texture synthesis. In *SIGGRAPH Asia '09 Papers*, 110:1–8.

MA, C., WEI, L.-Y., AND TONG, X. 2011. Discrete element textures. In *SIGGRAPH '11 Papers*.

NPAR, 2011. Npar: Non-photorealistic animation and rendering. `http://www.npar.org/`.

TZENG, S., AND WEI, L.-Y. 2008. Parallel white noise generation on a GPU via cryptographic hash. In *I3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*, 79–87.

WANG, L., WANG, X., SLOAN, P.-P., WEI, L.-Y., TONG, X., AND GUO, B. 2007. Rendering from compressed high dynamic range textures on programmable graphics hardware. In *SI3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, 17–24.

WANG, L., WEI, L.-Y., ZHOU, K., GUO, B., AND SHUM, H.-Y. 2007. High dynamic range image hallucination. In *Eurographics Symposium on Rendering '07*, 321–326.

WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. 2010. Example-based wrinkle synthesis for clothing animation. In *SIGGRAPH '10 Papers*, 107:1–8.

WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00 Papers*, 479–488.

WEI, L.-Y., AND LEVOY, M. 2001. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH '01 Papers*, 355–360.

WEI, L.-Y., AND WANG, R. 2011. Differential domain analysis for non-uniform sampling. In *SIGGRAPH '11 Papers*.

WEI, L.-Y., HAN, J., ZHOU, K., BAO, H., GUO, B., AND SHUM, H.-Y. 2008. Inverse texture synthesis. In *SIGGRAPH '08 Papers*, 52:1–9.

WEI, L.-Y., LEFEBVRE, S., KWATRA, V., AND TURK, G. 2009. State of the art in example-based texture synthesis. In *Eurographics '09 State of the Art Report*, 93–117.

WEI, L.-Y., LIU, B., YANG, X., MA, C., XU, Y.-Q., AND GUO, B. 2009. Nonlinear beam tracing on a GPU. Tech. Rep. MSR-TR-2009-183, Microsoft Research.

WEI, L.-Y. 1999. Deterministic texture analysis and synthesis using tree structure vector quantization. In *SIBGRAPI '99 Papers*, 207–214.

WEI, L.-Y. 2002. *Texture Synthesis by Fixed Neighborhood Searching*. PhD thesis, Stanford University. Adviser-Marc Levoy.

WEI, L.-Y. 2003. Texture synthesis from multiple sources. In *SIGGRAPH '03 Sketches & Applications*, 1–1.

WEI, L.-Y., 2004. A system and method of generating white noise for use in graphics and image processing. U.S. Patent Pending No. 10/956,954, filed September 30, 2004.

WEI, L.-Y. 2004. Tile-based texture mapping on graphics hardware. In *HWWS '04: Proceedings of the SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, 55–63.

WEI, L.-Y. 2008. Parallel Poisson disk sampling. In *SIGGRAPH '08 Papers*, 20:1–9.

WEI, L.-Y. 2009. Semantics consistent parallelism. In *ISC 09 GPU panel: making your desktop a supercomputer*. `http://research.microsoft.com/apps/pubs/default.aspx?id=81096`.

WEI, L.-Y. 2010. Multi-class blue noise sampling. In *SIGGRAPH '10 Papers*, 79:1–8.

WU, H., WEI, L.-Y., WANG, X., AND GUO, B. 2006. Silhouette texture. In *Eurographics Symposium on Rendering '06*, 285–296.