Mutated Kd-tree Importance Sampling

Perttu Hämäläinen[*], Timo Aila[*], Tapio Takala[*]
[*]Helsinki University of Technology
Telecomm. Software and Multimedia Laboratory
{pjhamala,timo,tta}@tml.tkk.fi

Jarmo Alander[†]
[†]University of Vaasa
Dept. of Electrical Engineering and Automation
Jarmo.Alander@uwasa.fi

Abstract

This paper describes a novel importance sampling method with applications in multimodal optimization. Based on initial results, the method seems suitable for real-time computer vision, and enables an efficient frame-by-frame global search with no initialization step. The method is based on importance sampling with adaptive subdivision, developed by Kajiya, Painter, and Sloan in the context of stochastic image rendering. The novelty of our method is that the importance of each kd-tree node is computed without the knowledge of its neighbours, which saves computing time when there's a large number of samples and optimized variables. Our method can be considered a hybrid of importance sampling, genetic algorithms and evolution strategies.

1 Introduction

Model-fitting in real-time computer vision is often a difficult multimodal, multivariable optimization problem. It is common to only search a portion of the parameter space based on prior knowledge, such as a Kalman filter that predicts the next solution based on the previous solutions (Sonka *et al.*, 1999). In cases where the tracked shape moves in an unpredictable and rapid manner, e.g., when tracking the user in gesture-controlled software, the prediction may not be accurate. In this case, the tracker may need to be re-initialized.

From the point of view of usability, initialization and too strong priors should be avoided so that the user controls the software instead having to adapt to the limits of technology. For example, Pfinder initializes the tracking only if the user assumes a pose in which the body parts can be reliably labelled based on the user's silhouette (Wren *et al.*, 1997).

To avoid re-initialization, more efficient global optimization strategies need to be developed. This paper describes a novel optimization method that we call *mutated kd-tree importance sampling*. The method is a hybrid of importance sampling, genetic algorithms (GA) and evolution strategies (ES). It is based on the following design principles:

Principle 1: Optimization as sampling
To inspect all peaks of the objective function (fitness function) carefully without neglecting any part of the parameter space, the density of samples at a given location should be proportional to the fitness of the location. In other words, we inspect optimization as importance sampling, treating the fitness function as a probability distribution function.

Principle 2: Make every sample count
Evaluating the fitness of generated samples is often time-consuming. For maximum efficiency, the decision of where to place the next sample should be supported by all past samples. Each new sample increases the resolution of the perceived fitness function shape.

Principle 3: Simple user interface
An optimization method should have as few parameters as possible for it to be easy to use. However, parameters are necessary to incorporate problem-specific prior knowledge to the optimization. The No Free Lunch (NFL) theorems for optimization state that on average, all optimization methods are equal when applied to all possible cost functions without problem-specific adjustments (Wolpert and MacReady, 1997).

Based on the design principles, we have developed a novel variant of kd-tree based importance sampling introduced by Kajiya (1986) and developed further by Painter and Sloan (1989).

2 Description of the method

Our method uses a kd-tree to adaptively subdivide the search space. With *N* optimized variables, each node of the tree represents an *N*-dimensional hypercube in the search space. Each node has two children. The children are created by bisecting the parent along a coordinate axis. There's one sample inside each leaf node.

Samples are generated so that a leaf node is sampled from a discrete probability distribution where the relative probability of leaf node k equals a single point estimate of the integral of the fitness inside the node, $p_k=f_kv_k$, where f_k is the fitness of the sample inside the node and v_k is the volume of the node.

Contrary to previous kd-tree importance sampling methods, we do not generate a sample uniformly inside the selected hypercube, which would correspond to using the kd-tree as a piecewise constant approximation of the fitness function. Instead, we mutate the existing sample of the selected hypercube using a multivariate normal distribution centered at the sample or the hypercube center. The covariance matrix is diagonal and the standard deviations are proportional to the dimensions of the hypercube. This way, the kd-tree approximates the fitness function as an additive mixture of Gaussians.

A new Gaussian is created for each mutated sample by finding the leaf node (hypercube) inside which the sample is located, and splitting the node into two new hypercubes, one containing the old sample and the other containing the mutated one. In optimized kd-trees for databases, the splitting dimension is often chosen to be that whose distribution exhibits the most spread (Yianilos, 1993). In our case this corresponds to the dimension of maximum distance between the old and new sample.

The Gaussians could also be stored in some other data structure than a kd-tree. The benefit of using a kd-tree is that the node selection time grows logarithmically as a function of the number of samples. After a new sample is stored in a leaf node, the probabilities are updated recursively by traversing from the leaf to the root and setting $p_{parent}= p_{child1} + p_{child2}$. When selecting a leaf, the tree is traversed from the root to a leaf so that at each node, a uniformly distributed pseudorandom number r is generated in the range $0\ldots p_{child1}+p_{child2.}$ Child 1 is visited if $r< p_{child1}$. If $r= p_{child1}$, the child is selected randomly.

The sampling is not prone to getting stuck inside a local optimum. A sample with high fitness attracts more samples, and if a sample belongs to a peak much smaller than the cube containing the sample, the fitness function approximation is inaccurate and the peak attracts disproportionately many samples. However, as the cubes get split to smaller and smaller ones, the approximation gets more accurate.

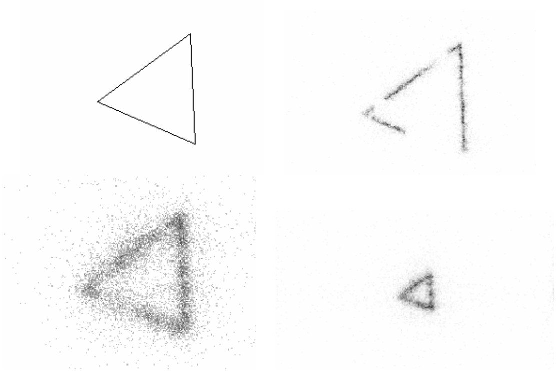

Figure 1. Top-left: 2d fitness function. Top-right: the distribution of samples with low mutation variance. Bottom-left: the distribution of samples with high mutation variance. Bottom-right: decreasing the high-fitness area reduces blurring.

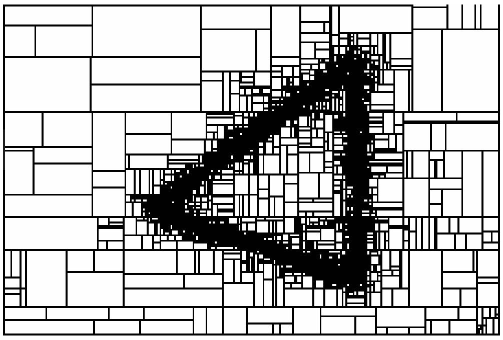

Figure 2. An example of the adaptive subdivision of space.

The normally distributed mutations provide a crucial improvement compared to a piecewise constant approximation of the fitness function: The spreading of samples beyond node boundaries ensures that a node can get split even if the fitness of its sample is initially zero ($p_k=f_kv_k=0$). The normally distributed sampling propagates the selection probabilities so that if the fitness of a cube is high, it will attract more samples both inside it and its neighbours.

Figure 1 illustrates the effect of the mutation variance. The 2d fitness function is a 256x256 pixel bitmap, sampled using 10000 samples (0.16 samples per pixel). The fitness function is zero except along the edges of the triangle, so that there's no gradient information to guide the search. With low variance, samples are highly concentrated on the triangle edges, but regions of the space remain unexplored due to zero samples. Increasing the mutation variance adjusts the compromise between greediness and thoroughness of the search. With high variance, the whole space gets searched, but the samples also fall on the zero areas due to a Gaussian blurring effect.

Thanks to the mutation variance proportional to the dimensions of the selected hypercube, the blurring is adaptive considering the number of samples and the size of the high-fitness area. The bottom-right image in Figure 1 shows how there's less blurring when the high-fitness area is smaller. The mutation variances at the bottom-left and bottom-right images are equal.

Figure 2 illustrates the adaptive subdivision resulting from the sampling.

3 Related work

3.1 Genetic algorithms

In multimodal, multivariable optimization, stochastic methods are generally a safer choice than hill-climbing methods that use the gradient of the objective function to direct the search. The hill-climbing (or descent) methods are prone to ending up in a local optimum instead of the global one.

Genetic algorithms (GA) are a vast family of stochastic optimization methods. GA methods generate samples (individuals) and evaluate their fitness. After an initial population has been generated by random sampling or based on prior knowledge, the population is evolved, which consists of selection, cross-over and mutation operations (e.g., Goldberg, 1989). In model-fitting, fitness can be formulated as a function of fitting error so that smaller error yields greater fitness. The optimized variables are formulated into a genome, e.g., a binary string or a vector of real values.

Our sampling method is related to GA in that, the corners of a kd-tree node can be thought as parents that produce offspring samples by cross-over. The structured subdivision provides the benefit of knowing that there's only one previous sample in the same space as the offspring. Even if the parents are fit, they should not be selected for cross-over if there already are several samples in the same space as the offspring. In this case, the offspring does not provide much additional information about the fitness function shape.

3.2 Evolution strategies

Evolution strategies (ES) are another big family of stochastic optimization methods. In contrast to GA, ES view the optimized variables not as single genes but as the features that are function of several genes, e.g., musical or mathematical talent (e.g., Beyer, 2001). Such features are often normally distributed in real populations. ES exploit this, e.g., by generating samples by sampling from a normal distribution centered at a selected individual.

Our method is related to ES in that we also select a sample and mutate it using a normal distribution. Our main difference to ES is the structured subdivision of space. The variance of the normal distribution is relative to the size of the selected hypercube, which focuses the search based on how much resolution there already is in fitness function approximation.

3.3 Importance sampling

Population based optimization can be sometimes replaced by importance sampling, and vice versa, although importance sampling is perhaps more typically used in Monte Carlo integration, i.e., estimating integrals of difficult integrands based on random samples. There are various importance sampling methods, of which overviews can be found in textbooks (e.g., Dutré, 2003). Borrowing ideas from computational chemistry, Sminchisescu (2002) has investigated hyperdynamic importance sampling for computer vision model-fitting. In computer graphics, both genetic algorithms and importance sampling have been used to minimize the number of light paths that need to be evaluated in image rendering (Szirmay-Kalos, 1999).

A simple way to estimate a definite integral is to compute the mean of random samples of the integrand within the integrating bounds. However, the estimate has a high variance, and much of the literature is devoted to variance reduction techniques.

Variance can be reduced by importance sampling, which means that more samples are produced in areas that contribute more to the integral. The relation to optimization is clear when considering a fitness function as the integrand: Areas of high fitness contribute more to the integral, and concentrating samples there makes it more probable to generate a sample close to the optimum.

Importance sampling is also related to stochastic optimization in that simulated annealing is an adaptation of Metropolis Monte Carlo (MMC) importance sampling. In MMC, a new sample is generated by displacing the previous one randomly. The change of the energy of the system, ΔE, is then computed and the new location is accepted if $\Delta E<0$ or if $\xi<\exp(-\Delta E/kT)$, where ξ is a random number between 0 and 1, and T is the temperature of the system. In the original Metropolis case, the sample is formed as a vector of the locations of the particles of a substance (Metropolis *et al.*, 1953).

In simulated annealing, energy is replaced by an arbitrary cost function, and T is gradually decreased so that the system freezes in a (near) optimal configuration (Kirkpatrick *et al.*, 1983). Although simulated annealing has been found powerful in many problems, it violates our principle 2 in that new samples are not based on all previous samples. The method is prone to get trapped inside a local optimum, especially if the temperature is decreased too rapidly. Variants of simulated annealing have been developed to improve transitions between optima, e.g., using gradient and curvature information to find saddle points representing ‘mountain passes’ connecting adjacent cost basins (Sminchisescu and Triggs, 2005). To our knowledge, none of the variants have exploited the adaptive subdivision of space that is characteristic to our method.

3.4 Sampling and adaptive subdivision using kd-trees

In Monte Carlo integration, variance can be reduced through stratification. This means that the sampling space is divided into equal size strata, and samples are generated inside each stratum. This ensures that the samples don’t get cluttered up in some portion of the space. Unfortunately, this is impractical in high-dimensional problems, because the number of strata grows exponentially as a function of dimensionality.

Crucial to this paper is the unification of importance sampling and stratification via adaptive subdivision using a kd-tree. This was first proposed by Kajiya (1986) in context of 3d image rendering. He writes that *“So far, our experiments in finding adaptive criteria have not been terribly successful”*, adaptive criteria denoting the means to decide which node to split and generate a new sample in. Although he outlined the use of the kd-tree to divide the space, he didn’t use the adaptive subdivision in generating the images in the paper.

Painter and Sloan (1989) were successful in developing the approach further. They were able to considerably reduce the number of light paths that need to be evaluated to provide an anti-aliased image.

Comparing Painter and Sloan to Kajiya, a major improvement is that the selection priority of a node equals the product of *external variance* and the volume of the node. External variance is computed from the means of the samples of the node and its neighbour nodes. This ensures that even if the priority of a node is initially zero, the priority grows along with the priorities of its neighbours and no part of the space is left unexplored. The effect is similar to the mutation in our method, but it is impractical in high-dimensional problems. The number of neighbours of a node grows exponentially as the number of dimensions increase. In high-dimensional spaces, updating the priorities may require thousands of operations for each new sample. This may be why Painter and Sloan apply their method in only two dimensions.

Compared to the randomized selection in our method (sampling from the approximated fitness distribution), Painter and Sloan always select the node with the highest priority. We also tried this, but it seems to increase the exploring of local optima and the time needed to find the global optimum. Randomized selection enhances the parallel exploration of several optima.

In our method, the concept of external variance can be implemented so that instead of fitness times volume, the relative probability of node k is computed as $p_k=|f_k-f_s|v_k$, where f_s is the fitness of the sibling of the node. This improves sampling of, e.g., images with areas of constant colour, but so far we have not observed a clear difference in optimization.

There’s been quite some time since Painter and Sloan’s paper, but the ideas have recently had revived interest. SUAVE, a method similar to Painter and Sloan’s has been implemented as a part of the CUBA library for multidimensional numerical integration (Hahn, 2005 and 2006). SUAVE improves previous methods by global estimation of Monte Carlo integration error and stops sampling automatically when desired precision has been reached.

4 Test results

Figure 1 demonstrates the ability of our method to concentrate samples in areas of high fitness, even when the gradient of the fitness function is either zero or infinite. In the following, we present the results from optimization in the case of two multidimensional test functions and a real-world computer vision problem with three optimized variables.

4.1 Multidimensional test functions

Figures 3 and 4 illustrate the average convergence of 100 test runs of our method with 1 to 10 dimensional test functions. In the figures, the Euclidian distance between the true optimum and the best sample so far is plotted as a function of samples generated. The distance curves are normalized so that they start at 1, independent of the number of dimensions. In Figure 3, the fitness is defined as

$$f(\mathbf{x}) = \prod_i \left|\mathrm{sinc}(x_i)\right|, \qquad (1)$$

where x_i denotes optimized parameter i. The parameter range is $-4\pi \le x_i \le 4\pi$. In Figure 4, the fitness is an exponential peak,

$$f(\mathbf{x}) = e^{-10\sqrt{|\mathbf{x}|}}. \qquad (2)$$

Because the samples are distributed according to the fitness, convergence is rapid if high fitness is concentrated close to the optimum, as in the case of the exponential peak. In general, exponentially peaked fitness functions work better than, e.g., quadratic peaks, since the convergence slows down if the gradient of the fitness approaches zero near the optimum.

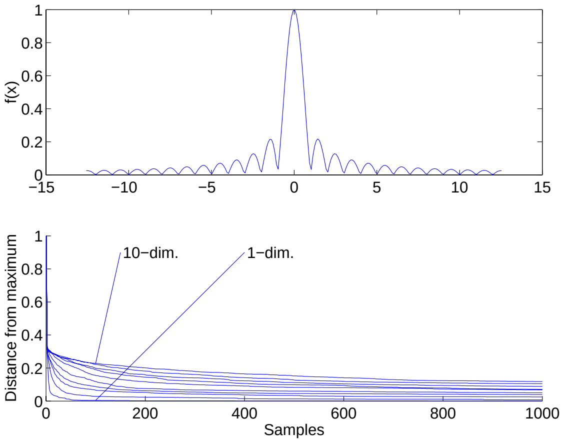

Figure 3. Convergence when finding the maximum of a multimodal test function in 1 to 10 dimensions.

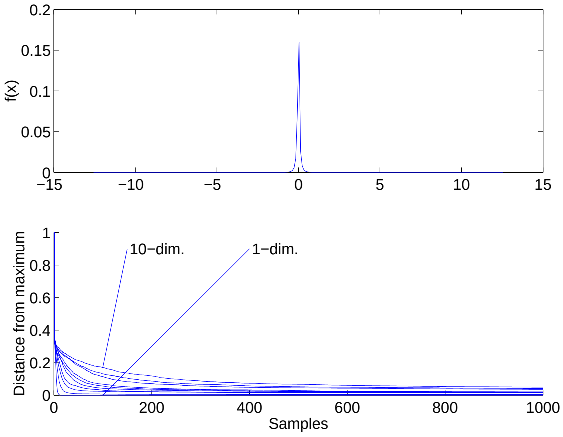

Figure 4. Convergence when finding the maximum of an exponential peak in 1 to 10 dimensions.

4.2 Exploiting spatiotemporal coherence in computer vision

Considering practical applications, we have tested our method in a recent version of a computer vision based user interface, originally presented by Mäki-Patola *et al.* (2006). Figure 5 shows the view of a camera placed inside a djembe drum. The shadows of the player's hands control musical synthesis. The hands don't need to be fully tracked, because moment-based image features provide enough information. However, the moments should only be computed at the drum membrane, for which a circular mask is needed. The mask is obtained by fitting a circle to the drum membrane, shown in Figure 5. This is an optimization problem with the circle coordinates and radius as the variables. The drum membrane stays mostly still in the camera view, but it may move when the drum is hit hard.

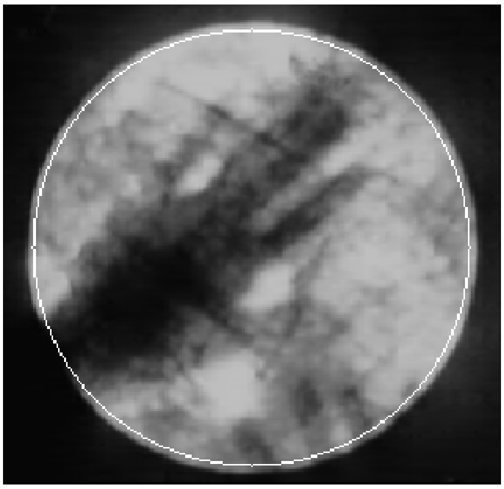

Figure 5. The view of a camera placed inside a djembe drum, showing the shadows of the player's hands on the drum membrane, and the circle fitted to the outline of the membrane.

We formulate the fitness function as

$$f = e^{-\sum_i I_{1i}} e^{-\sum_i (255 - I_{2i})} r, \qquad (3)$$

where I_{1i} denotes the intensity of pixel i of the circle, and I_{2i} denotes the intensity of pixel i of a circle with radius 1.05r. The intensity values are in the range 0...255. The circle is sampled uniformly with 14 points. In a good solution, the circle is on the drum membrane, but a slightly larger circle falls outside the drum. Multiplying by r favours large circles.

According to our tests, 1000 samples are enough when optimizing the circle parameters at each new video frame with 30 frames per second. However, the performance can be boosted considerably by exploiting both spatial and temporal coherence.

The basic version of our sampling method exploits spatial coherence, that is, samples are likely to be generated in the vicinity of regions that are already known to yield high fitness. Temporal coherence can be utilized so that instead of initializing the optimization at each video frame, the kd-tree generated at the previous frame is pruned to only contain a number of best samples. The fitness of the samples is then re-evaluated in the context of the new video frame, after which optimization is continued normally. When preserving 10 best samples, we only need about 100 new samples for each frame. The computational and memory costs are considerably less than, e.g., those of Hough transform.

In effect, we initialize the optimization with guesses based on the previous optimization results, which is a common technique in computer vision. What makes our case special is that thanks to the adaptive subdivision and the sampling based on all previous samples, erroneous initial guesses do not prevent the optimization from converging. Sudden movements of the djembe only cause a few frames of confusion, after which the optimization converges again without any additional initialization.

5 Conclusion and future work

We have described a novel importance sampling method that improves previous methods by combining kd-tree adaptive subdivision with stochastic blurring of the sample distribution. The blurring is implemented via mutations that make it unnecessary to find and evaluate neighbours in the kd-tree. This can save computation time, especially in multi-parameter optimization.

The method satisfies all the three design criteria defined in the introduction: optimization as sampling from the fitness distribution, utilization of every past sample in the generation of a new sample, and a simple user interface. Besides the formulation of the fitness function, which typically requires skill and insight, the only user-adjustable parameters are the number of samples and the mutation variance that adjusts greediness. The sampling can also be initialized with a number of initial guesses, and it recovers from erroneous guesses.

We find the method promising especially in the following aspects that we are investigating for future publications:

- Light transport in image rendering. This is natural, as the roots of our method are in importance sampling of light paths.
- Improving the utilization of temporal coherence by incorporating ideas from particle filters, such as Condensation (Blake and Isard, 1998). It seems that using the kd-tree may ensure global sampling so that the particles don't get trapped inside a single mode of the estimated distribution.

Acknowledgements

We wish to thank all our colleagues for the fruitful discussions, especially Samuli Laine, Janne Kontkanen, Jaakko Lehtinen, and Teemu Mäki-Patola. This work has been partly funded by Academy of Finland.

References

Hans-Georg Beyer. *The Theory of Evolution Strategies*, Springer-Verlag, 2001

A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998

P. Dutré, P. Bekaert, and K. Bala. *Advanced Global Illumination*. A K Peters Ltd., 2003

D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989

T. Hahn. CUBA - a library for multidimensional numerical integration. *Computer Physics Communications*, 168:78–95, 2005

T. Hahn. The CUBA library. *Nuclear Instruments and Methods in Physics Research A*, 559:273–277, 2006

J.T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and interactive Techniques (SIGGRAPH '86)*, 143-150, 1986

S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by Simulated Annealing, *Science*, 220(4598): 671-680, 1983

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines, *The Journal of Chemical Physics*, 21(6): 1087-1092, 1953

T. Mäki-Patola, P. Hämäläinen, and A. Kanerva. The Augmented Djembe Drum – Sculpting Rhythms. To appear in *Proceedings of NIME'2006*, Ircam, France, June 5-8, 2006

J. Painter and K. Sloan. Antialiased ray tracing by adaptive progressive refinement. In *Proceedings of the 16th Annual Conference on Computer Graphics and interactive Techniques (SIGGRAPH '89)*, 281-288, 1989

C. Sminchisescu. *ESTIMATION ALGORITHMS FOR AMBIGUOUS VISUAL MODELS Three Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*, doctoral thesis, 2002, http://www.cs.toronto.edu/~crismin/thesis.html, link visited 29th April 2006

C. Sminchisescu and B. Triggs. Hyperdynamic Sampling. *Journal of Image & Vision Computing*, 2005, available online at http://lear.inrialpes.fr/pubs/2005/ST05a, link visited 29th April 2006

M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*, 2nd edition, Brooks/Cole Publishing Company, 1999

L. Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination*. Institute of Computer Graphics, Vienna University of Technology, Vienna, 1999. http://citeseer.ist.psu.edu/szirmay-kalos00montecarlo.html, link visited 29th April 2006

D.H. Wolpert and W.G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): 67-82, 1997

C.R. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfinder: Real-Time Tracking of the Human Body, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7): 780-785, 1997

P.N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 311-321, 1993