Monte Carlo Anti-Aliasing

TR88-018
April 1988

John H. Halton

The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175
MONTE CARLO ANTI-ALIASING

by

John H. Halton

The University of North Carolina at Chapel Hill
Department of Computer Science
Sitterson Hall, CB#3175
Chapel Hill, NC 27599 U.S.A.

ABSTRACT

Several anti-aliasing strategies are proposed, which generate Monte Carlo discretized estimates of color and intensity at each pixel of a raster display.
1. We are given a function \( f : \mathbb{R}^2 \to \mathbb{R}^1 \), specifying color and intensity at any point of a screen area \( S \subseteq \mathbb{R}^2 \). The screen \( S \) is subdivided into \( n \) pixels \( P_h \) (\( h = 1, 2, \ldots, n \)), all disjoint and of equal area and shape.

2. It is intended to approximate the function \( f \) on \( S \) by a function \( \mathbb{R}^2 \to \mathbb{R}^1 \) which takes the value \( \gamma_h \) on the pixel \( P_h \), for \( h = 1, 2, \ldots, n \).

3. One approach is to define, for the pixel \( P_h \) centered at \( c_h \), a weight function \( \omega_h(r - c_h) = \omega_h(r) \) and let

\[
\phi_h = \int_{\mathbb{R}^2} f(r) \omega_h(r),
\]

where \( \mathbb{R}^2 \) denotes \( \mathbb{R}^2 \) and \( \int_{\mathbb{R}^2} dr \) denotes \( \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \), with \( r = (x, y) \).

4. A very general Monte Carlo scheme for estimating \( \gamma_h \) would select an integer \( n_h \) and a set of estimator-probability pairs \( (g_{h,i}(r), p_{h,i}(r)) \), for \( i = 1, 2, \ldots, n_h \); so that one samples points \( \xi \in \mathbb{R}^2 \) with probability density \( p_{h,i}(\xi) \), independently of each-other, and uses the estimator

\[
\hat{\gamma}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} g_{h,i}(\xi)
\]

for \( \gamma_h \). For example, "crude Monte Carlo" could define \( p_{h,i}(r) = N/A \), where \( A \) is the area of \( S \) (so that \( A/N \) is the area of the pixel \( P_h \)), and use the estimator \( g_{h,i}(r) = cf(r) \) in \( P_h \); but this would not work, since we would want that the estimator be unbiased, i.e., that

\[
\mathbb{E}_{i=1}^{n_h} E[g_{h,i}] = \phi_h,
\]

and this reduces, by (1), to \( c = A\phi_h / Nn_h \phi_h \), where
and we would need to know both $\hat{z}_h$ and $\hat{z}_h'$ to get $\hat{z}_h'$. Another approach is to use $\hat{u}_{h_c}(r) = \hat{u}_{h}(r) = \hat{u}(r - c_h)$ in the whole of $\mathcal{Q}$ (though, of course, most of the probability will be in or near $\mathcal{P}_h$), and use the estimator $\hat{z}_{h'c}(r) = c(r)$; whence the condition (3) reduces to $\tau = 1/n_h$, provided that the weight function $\hat{u}_{h}$ satisfies (as is usual) the normalizing condition

$$\int_{\mathcal{Q}} \hat{u}_{h}(r) = \int_{\mathcal{Q}} \hat{u}(r - c_h) = \int_{\mathcal{Q}} \hat{u}(r) = 1. \quad (5)$$

Of course, this condition is not at all unreasonable. Note that we may, yet again, choose, over the whole of $\mathcal{Q}$, $\hat{u}_{h'}(r) = \hat{u}_h(r)$, a different normalized weight function from $\hat{u}_{h}$ (for instance, the normal distribution centered at $c_h$ and with standard deviation of the order of the diameter of a pixel), and then the estimator would be $\hat{u}_{h'}(r) = \hat{u}_h(r)f(r)/\hat{u}_h(r)n_h$, as is readily verified, and this is again feasible; so we note the pair:

$$(\hat{g}_{h'}, \hat{\rho}_{h'}) = \left[ \frac{u_h(r)f(r)}{u_h(r)n_h}, \ u_h'(r) \right]. \quad (6)$$

5. An alternative approach would be to use a form of stratified sampling. Note that, in the technique developed above, all $n_h$ estimators are identical and identically distributed. Suppose, instead, that the pixel $P_h$ is dissected into $m$ identical sub-pixels $R_{h,j}$, and that $\delta_j$ identical estimators $\hat{u}_{h,j}(r)$ are sampled with density $\rho_{h,j}(r)$ in $Q$, where $u_{h,j}(r) = \delta(r - b_{h,j})$ and $b_{h,j}$ is the center of $R_{h,j}$. We then require, by (3), that
As an example, we could choose the function \( \phi \), and then put

\[
\var{\tilde{g}_{h \tilde{c} j}^j}(r) = \frac{f(r) \omega(r - c_h)}{m \sigma \frac{r}{b_{h j}}},
\]

where we also must have that

\[
\sum_{j=1}^{m} \var{\tilde{g}_{h \tilde{c} j}^j} = \lambda_h.
\]

6. What we must do to make the method efficient is to minimize (or at least diminish) the variance of our estimate. Thus, we note that, for the first technique, given by (6), we have

\[
\var{\tilde{g}_{h \tilde{c} j}^j} = \var{\tilde{g}_{h \tilde{c} j}^j} = \lambda_h \left\{ \int_{Q} \left[ \frac{\omega''(r)f(r)}{\omega'(r) \omega_h(r)} \right]^2 \omega_h(r) \right\} - \left\{ \int_{Q} \omega_h(r) f(r) \right\} = \frac{1}{\lambda_h} (\lambda_h - \phi_h^2),
\]

where

\[
\lambda_h = \int_{Q} \frac{[\omega_h(r)]^2 [f(r)]^2}{\omega_h'(r)}.
\]

For the second technique, given by (8), we similarly get that

\[
\var{\tilde{g}_{h \tilde{c} j}^j} = \frac{1}{m} \sum_{j=1}^{m} \var{\tilde{g}_{h \tilde{c} j}^j} = \lambda_h \left\{ \int_{Q} \left[ \frac{f(r) \omega(r - c_h)}{m \sigma \frac{r}{b_{h j}}} \right]^2 \sigma(r - b_{h j}) \right\} - \left\{ \int_{Q} \frac{f(r) \omega(r - c_h)}{m \sigma \frac{r}{b_{h j}}} \sigma(r - b_{h j}) \right\} = \frac{1}{m} (\lambda_h - \phi_h^2),
\]

where

\[
\lambda_h = \int_{Q} \frac{[\omega_h(r)]^2 [f(r)]^2}{\omega_h'(r)}.
\]
where
\[ \omega_n = \int_{a}^{b} \omega(r) \cdot \frac{2 \omega'(r - c)}{\omega(r - b_n)} dr. \]  

1. If we consider the case of (6), (10), and (11), and first assume that \( f, \omega, \omega' \), and so \( \lambda_n \) and \( \beta_n \) are all given \textit{a priori}; then we may ask how to choose the numbers of function-evaluations \( n_n \) by pixels, so as to make all variances the same, given the sum \( n = \sum_{k=1}^{N} n_k \). The answer is evidently
\[ n^* = n \left( \lambda_n - \frac{\phi_n^2}{n} \right) / \sum_{k=1}^{N} \left( \lambda_k - \frac{\phi_k^2}{n} \right), \]  
and the common value of the variance at every pixel is then
\[ \text{var} \left( \sum_{j=1}^{m} \omega_{n_j} \right) = \sum_{k=1}^{N} \left( \bar{\lambda}_k - \frac{\phi_k^2}{n} \right). \]  

In the case of (8), (12), and (13), with \( f, \omega, \mu, \) and so \( \phi_n \) and \( \nu_n \) given, we similarly see that we can first optimize over the strata in a single pixel; Lagrangian theory shows that
\[ s_j^* = n_{n_j} \left( \omega_{n_j} - \frac{\phi_n^2}{n} \right) / \sum_{k=1}^{m} \left( \nu_{n_k} - \frac{\phi_n^2}{n} \right), \]  
minimizes the variance at \( \bar{x}_n \) to the value
\[ \min \text{var} \left( \sum_{j=1}^{m} \omega_{n_j} \right) = \frac{1}{m^2 n_H} \left[ \sum_{j=1}^{m} \left( \nu_{n_j} - \frac{\phi_n^2}{n} \right) \right]^2. \]  

Note that the Cauchy-Schwartz-Bunyakovsky inequality shows that indeed
\[ \frac{1}{m^2 n_H} \left[ \sum_{j=1}^{m} \left( \nu_{n_j} - \frac{\phi_n^2}{n} \right) \right]^2 \leq \frac{1}{m^2 n_H} \left[ \sum_{j=1}^{m} \left( \nu_{n_j} - \frac{\phi_n^2}{n} \right) \right] \left( \sum_{k=1}^{m} s_k \right). \]
and the right-hand side of the inequality is the general variance (12), by (9); so that (16) does indeed minimize (not maximize or point-of-inflexion) the variance. Now we proceed, as before, to make all the variances (17) the same; yielding that

\[ v_k = \frac{1}{n^2} \sum_{j=1}^{m} (w_k - \bar{w}_k)^2 \]

This makes the common value of the variance

\[ \min \text{var} \left[ \sum_{j=1}^{m} z_{k,j} \right] = \frac{1}{m^2 n} \sum_{k=1}^{N} \left( \sum_{j=1}^{m} (w_{k,j} - \bar{w}_k)^2 \right)^2. \]  

8. As a specific example, we may suppose that \( S \) is a rectangle

\[ S = (0 < x < L_1, \ 0 < y < L_2); \]  

and that the index \( h \) is \((h_1, h_2)\), with \( N = N_1 N_2 \) and \( 0 < h_1 < N_1, \ 0 < h_2 < N_2 \) \((t = 1, 2)\), so that \( P_h \) is the rectangle

\[ P_h = P_{h_1 h_2} = \left( \frac{L_1}{N_1} h_1 \leq x \leq \frac{L_1}{N_1} (h_1 + 1), \ \frac{L_2}{N_2} h_2 \leq y \leq \frac{L_2}{N_2} (h_2 + 1) \right), \]  

centered at \( c_h = (c_{h_1}, c_{h_2}) \) with \( c_{h_t} = \frac{L_t}{N_t} (h_t + \frac{1}{2}) \) \((t = 1, 2)\).  

Similarly, we take \( j = (j_1, j_2) \), \( m = m_1 m_2 \), and \( 0 < j_z \leq m_t \) \((t = 1, 2)\), so that \( P_{h,j} \) is the \((l_1 / N_1 m_1, l_2 / N_2 m_2)\) rectangle centered at

\[ b_{h,j} = (b_{h_1,1}, b_{h_1,2}) \]  

with \( b_{h,t} = \frac{L_t}{N_t m_t} (m_t h_t + j_t - \frac{1}{2}) \) \((t = 1, 2)\).  

We may further postulate that both \( \omega_h \) and \( \rho_{h,j} \) take the form of the normal distribution, with
where
\[ \gamma = (\frac{\mu_1 - \mu_2}{\sigma_1^2 + \sigma_2^2}) \sigma = (\frac{A}{\gamma}) \sigma, \]
and
\[ \varepsilon = (\frac{\mu_1 - \mu_2}{\sigma_1^2 + \sigma_2^2}) \sigma = (\frac{A}{\gamma}) \sigma. \]

Here, \( \varepsilon \) is a constant for the system, related to the weight function \( \omega \) but not to \( f \) or to \( S \) and its subdivisions.

Then we have that
\[
\psi = \frac{1}{2\pi} \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \ [f(x, y)]^2 [\omega(x - \sigma_1, y - \sigma_2)]^2
\times \exp(\frac{\gamma}{A} (x - \sigma_1)^2 + (y - \sigma_2)^2)/2\sigma. \tag{29}
\]
and
\[
\psi_\mu = \frac{1}{2\pi m} \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \ [f(x, y)]^2 [\omega(x - \sigma_1, y - \sigma_2)]^2
\times \exp(\frac{\gamma m}{\sigma} (x - \sigma_1)^2 + (y - \sigma_2)^2)/2\sigma. \tag{30}
\]

9. The strategies investigated here so far are adaptive only insofar as the optimizing numbers of samples (14) and (16) are to be estimated from Monte Carlo estimates of the \( \lambda_i \) and \( \omega_i \), which can be obtained simultaneously with the estimates of \( \varphi \) generated by the estimators (6) and (8), respectively.

Since only small samples are to be taken, because \( f \) is so laborious to get, the relative sample-sizes (14) and (16) will not be very accurately optimal.
Another approach would attempt to perform importance sampling by sequentially approximating $f(x, y)w_n(x, y)$ with $w_n'$. Since $w_n$ is given and $f$ is experimentally determined (so, also given), we may write $f(x, y)$ for the product. As we accumulate values of $C$ by sampling (initially with an arbitrary distribution), we can form an increasingly accurate picture of the functional dependence of $C$ on $(x, y)$ and model $w_n'$ on this.

Alternatively, we may do a sequential correlated sampling calculation, in which we fix the sampling density arbitrarily, and then use an estimator of the form $(C(x, y) - \psi(x, y))/w_n'(x, y) - \int q dx \psi(x, y)$, where $\psi$ is the best approximation to $C$ for which the integral on the right is easily computable.

Yet another approach which should be empirically investigated is to use an ordering of the sampled values of $C$ to indicate where stratification should occur. First, we sample $C$ at a small number of points in each pixel and tabulate $C, x,$ and $y$, in order of increasing $C$. If there is a strong correlation of $C$ with $x$ or with $y$, split the pixel accordingly and sample a few more points. Repeat, if necessary.

Note that the stratification and sampling are done in the whole of $x$, not within the pixel or sub-pixel only. This is to conform with the global form of $u$. Note also that $u$ may be given the full theoretical form, and need not be approximated by a normal distribution itself.