We regard an image as a vector with a component corresponding to each pixel. So the ith pixel has a corresponding basis vector B_i. There is also a function, $B(x, y)$, which maps points on the image plane to the basis vector of the pixel which contains them. Then the image, I, can be written as:

$$I = \int_{x \in X} \int_{y \in Y} B(x, y) R(x, y)\, dy\, dx$$

where X is the range of x values in the image, Y is the range of y values in the image, and $R(x, y)$ represents point (x, y)'s contribution to its pixel's response. So $R(x, y)$ is itself an integral over solid angle, time, and wavelength. The point here is that it is possible to write the entire image as a vector-valued integral and to change variables so that the image can be expressed as an integral over the image plane and lens of an integrand, call it $f(x, y, u, v)$. The fact that f is itself an integral is not important.

It is also possible to parameterize f in terms of the lens and the set of visible points in the scene. This integral can be extended to all points in the scene, which we will assume are parameterized by one continuous parameterization in s over some range S and t over some range T, by introducing a visibility term, $V(s, t, u, v)$, which is 1 if the point (s, t) is visible from (u, v) and is 0 otherwise. We will assume that the Jacobian for the transformation from x, y, u, and v to s, t, u, and v does not depend on u and v (the position on the lens). So we will write $dv\, du\, dy\, dx = J(s, t)\, dv\, du\, dt\, ds$. This allows us to do the following transformations on the integral:

$$\begin{aligned}
I &= \int_X \int_Y \int_U \int_V f(x, y, u, v)\, dv\, du\, dy\, dx \\
&\approx \int_S \int_T \int_U \int_V V(s, t, u, v) f(x_{s,t,u,v}, y_{s,t,u,v}, u, v) J(s, t)\, dv\, du\, dt\, ds \\
&= \int_S \int_T \frac{\int_U \int_V V(s, t, u, v)\, dv\, du}{\int_U \int_V V(s, t, u'', v'')\, dv''\, du''} \int_U \int_V V(s, t, u', v') f(x_{s,t,u',v'}, y_{s,t,u',v'}, u', v') J(s, t)\, dv'\, du'\, dt\, ds \\
&= \int_S \int_T \int_U \int_V \frac{V(s, t, u, v) \int_U \int_V V(s, t, u', v') f(x_{s,t,u',v'}, y_{s,t,u',v'}, u', v')\, dv'\, du'}{\int_U \int_V V(s, t, u'', v'')\, dv''\, du''} J(s, t)\, dv\, du\, dt\, ds \\
&\approx \int_X \int_Y \int_U \int_V \frac{\int_U \int_V V(s_{x,y,u,v}, t_{x,y,u,v}, u', v') f(x_{s_{x,y,u,v}, t_{x,y,u,v}, u', v'}, y_{s_{x,y,u,v}, t_{x,y,u,v}, u', v'}, u', v')\, dv'\, du'}{\int_U \int_V V(s_{x,y,u,v}, t_{x,y,u,v}, u'', v'')\, dv''\, du''}\, dv\, du\, dy\, dx
\end{aligned}$$

As an algorithmic reference point, the regular rendering algorithm uses an estimator to the first form of the integral:

$$\frac{C}{n} \sum_{i=1}^{n} f(X_i, Y_i, U_i, V_i)$$

where X_i, Y_i, U_i, and V_i are selected uniformly at random from their domains and C is the reciprocal of the probability of all four random variables hitting arbitrary target values.

The corresponding estimator for our above resulting form of the integral is:

$$\frac{C}{n}\sum_{i=1}^{n}\frac{\int_U\int_V V(s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},u',v')f(x_{s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},u',v'},y_{s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},u',v'},u',v')\,dv'\,du'}{\int_U\int_V V(s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},u'',v'')\,dv''\,du''}$$

So for each choice of X_i, Y_i, U_i, and V_i, we have two more integrals to evaluate. We can use the same techniques to get an unbiased estimate for each of these integrals, but note that an unbiased estimated of the denominator cannot simply be inverted to get an unbiased estimate of the reciprocal. We will use this approach anyway, which will introduce bias but still guarantee convergence on the correct answer.

For each choice of the above random variables, we will need a collection of additional random variables $U'_{i,j}$, $V'_{i,j}$, $U''_{i,k}$, and $V''_{i,k}$. We can choose the amount to sample each sub-integral independently and arbitrarily for each choice of X_i, Y_i, U_i, and V_i. On the other hand, we could correlate them as much as we like and re-use the first two as the latter two. This introduces dependence but has no effect on bias. Fortunately, the pdfs for these variables are all 1 everywhere, so there are no constants to worry about (although they would cancel between the numerator and the denominator anyway). The final (biased) estimator is:

$$\frac{C}{n}\sum_{i=1}^{n}\frac{\frac{1}{m_i}\sum_{j=1}^{m_i}V(s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},U'_{i,j},V'_{i,j})f(x_{s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},U'_{i,j},V'_{i,j}},y_{s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},U'_{i,j},V'_{i,j}})}{\frac{1}{l_i}\sum_{j=1}^{l_i}V(s_{X_i,Y_i,U_i,V_i},t_{X_i,Y_i,U_i,V_i},U''_{i,k},V''_{i,k})}$$

The denominator considers the point hit by the ray from (X_i, Y_i) to the point (U_i, V_i) and estimates the fraction of the lens that can see this point by checking at randomly selected points on the lens.

The numerator considers the same point. It randomly samples points on the lens. If the scene point is not visible from the lens point, that lens point has no contribution. When the point is visible, the situation is similar to one in the original algorithm. Namely, the situation where the ith sample sends a ray through the current point on the lens and it hits the current point in the scene. Note the similarity of the values in the estimators for these two situations. The visibility term is 1 and can be ignored, so this estimator has the same contribution as with the usual algorithm except divided by m_i (the number of lens samples) and the denominator (an estimate of the fraction of the lens seeing the scene point).