DISTRIBUTED RAY TRACING – some implementation notes
multiple distributed sampling
jitter - to break up patterns

DRT - theory v. practice
brute force - generate multiple rays at every sampling opportunity
alternative: for each subsample, randomize at each opportunity

DRT COMPONENTS
anti-aliasing and motion blur
  supersampling - in time and space: anti-aliasing
  jitter sample in time and space: motion blur
depth of field - sample lens: blurs
shadows – sample light source: soft shadows
reflection – sample reflection direction: rough surface
transparency – sample transmission direction: translucent surface

REPLACE CAMERA MODEL
shift from pinhole camera model to lens camera model
picture plane at -w, not +w
camera position becomes lens center
picture plane is behind 'pinhole'
negate u, v, w, trace ray from pixel to camera

ANTI-ALIASING: ORGANIZING subpixel samples
Options
1. Do each subsample in raster order
2. do each pixel in raster order, do each subsample in raster order
3. do each pixel in raster order, do all subsamples in temporal order
4. keep framebuffer, do all subsamples in temporal order

SPATIAL JITTERING
for each pixel 200x200, i,j
  for each subpixel sample 4x4 s,t
  JITTERED SAMPLE
  jitter s,t

MOTION BLUR – TEMPORAL JITTERING
for subsample
get delta time from table
jitter delta +/- 1/2 time division
move objects to that instant in time

DEPTH OF FIELD
generate ray from subsample through lens center to focal plane
generate random sample on lens disk - random in 2D u,v
generate ray from this point to focal plane point

VISIBILITY - as usual
intersect ray with environment
find first intersection at point p on object o with normal n

SHADOWS
generate random vector on surface of light - random on sphere

REFLECTIONS
computer reflection vector
generate random sample in sphere at end of R

TRANSPARENCY
compute transmission vector
generate random sample in sphere at end of T

SIDE NOTE
randomize n instead of ramdomize R and T