|Méso|Star>

« Les outils de l'industrie du cinéma appliqués au transfert radiatif.»

benjamin.piaud@meso-star.com

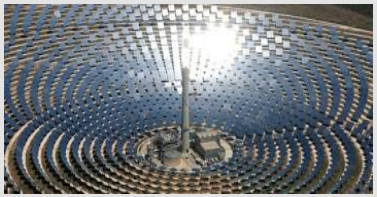
En collaboration avec R. Fournier, S. Blanco, M. El Hafi, C. Caliot et d'autres ...

Energétique et complexité.

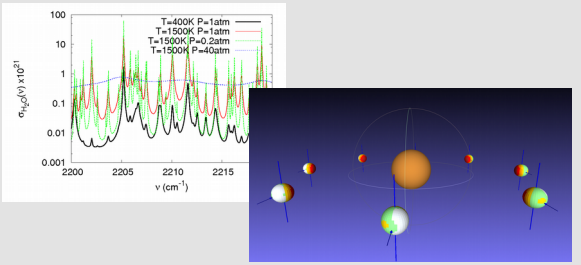
Résolution de problèmes thermiques, énergétiques complexes

- échelle spatiale « infinie »,
- échelle temporelle « infinie »,
- couplage de phénomènes,
- ...

Procédés solaires

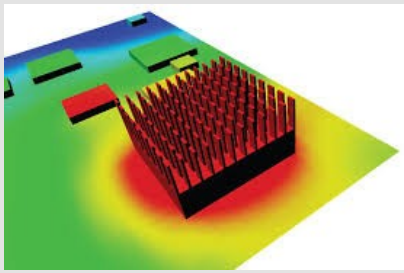


Physique atmosphérique / planétaire



Thermique habitat/quartier

Thermique électronique / systèmes embarqués



1- Qu'a réalisé l'industrie du cinéma ?

2- Star-Engine et quelques projets.

3- Un exemple complet : le code 4V/S.

4- Conclusion.

Une rupture dans l'industrie du cinéma :
une ingénierie statistique autour de la maquette numérique.

Le *Faucon Millenium* :
~ **80 Go** de données géométriques
+ **120 Go** pour les textures

isotropix

Star Wars : The Force Awakens (2015)

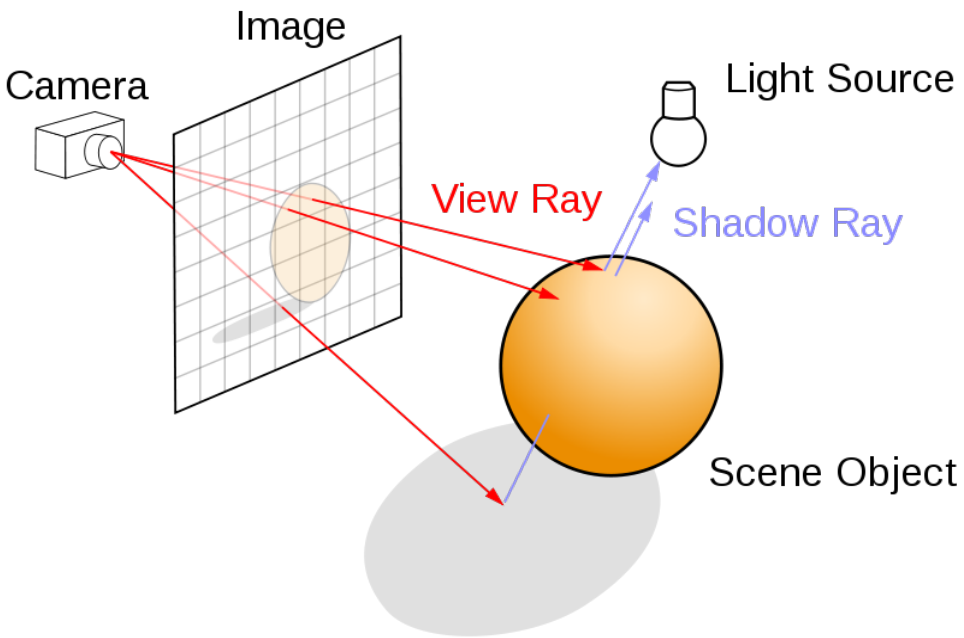
Big Hero 6 (2014)

Métropole de *San Fransokyo :*
83000 bâtiments

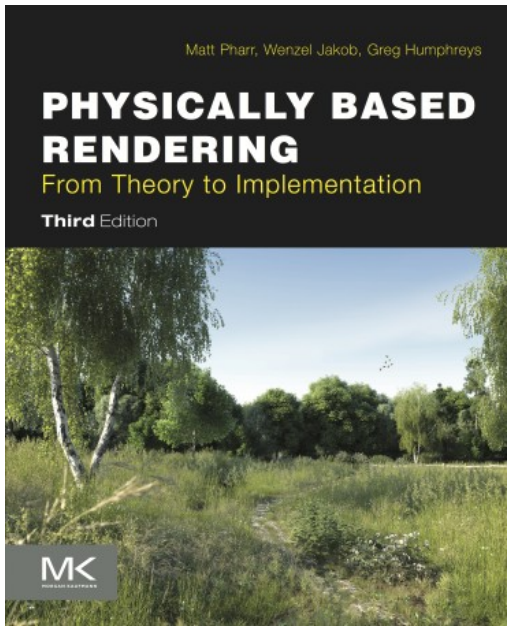
Walt Disney Animation Studios

Une rupture dans l'industrie du cinéma :
une ingénierie statistique autour de la maquette numérique.

- **Modèle** de propagation de la lumière.
- Résolution du modèle par une exploration **statistique** des **chemins optiques**.
- Techniques d'**accélérations d'accès à la données**, peu sensible aux détails et à la complexité.



Une communauté « open-source » active.



intel

LuxRender
GPL PHYSICALLY BASED RENDERER

ALEMBIC
open computer graphics interchange framework

Mitsuba
Physically Based Renderer

Manta
interactiveraytracer

Y · A · F · A · R · A · Y

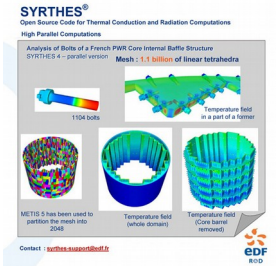
1- Qu'a réalisé l'industrie du cinéma ?

2- Star-Engine et quelques projets.

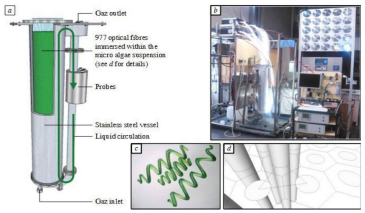
3- Un exemple complet : le code 4V/S.

4- Conclusion.

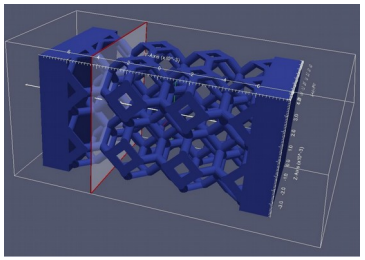
Star-Engine.



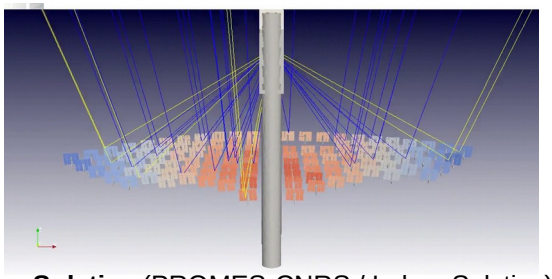
Star-gebhartfactor (EDF R&D)



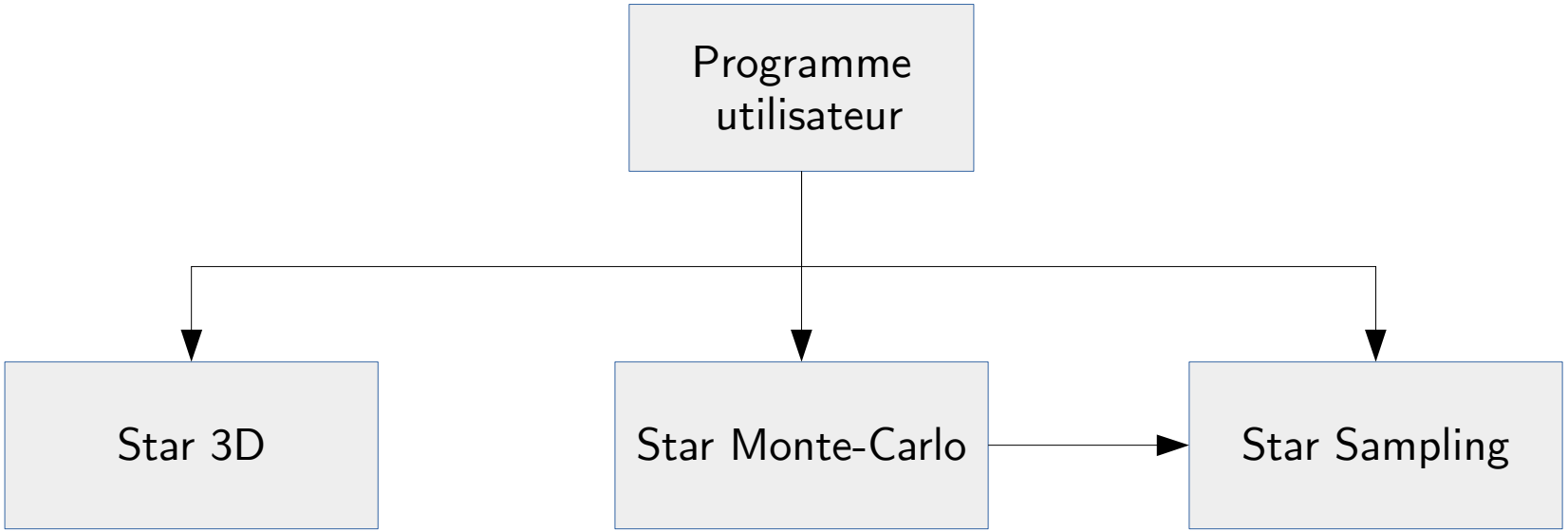
Schiff (IBP, RAPSODEE)



Star-therm (PROMES-CNRS)



Solstice (PROMES-CNRS / Labex Solstice)



Collections de bibliothèques open-source (CeCILL) pour programmeurs.

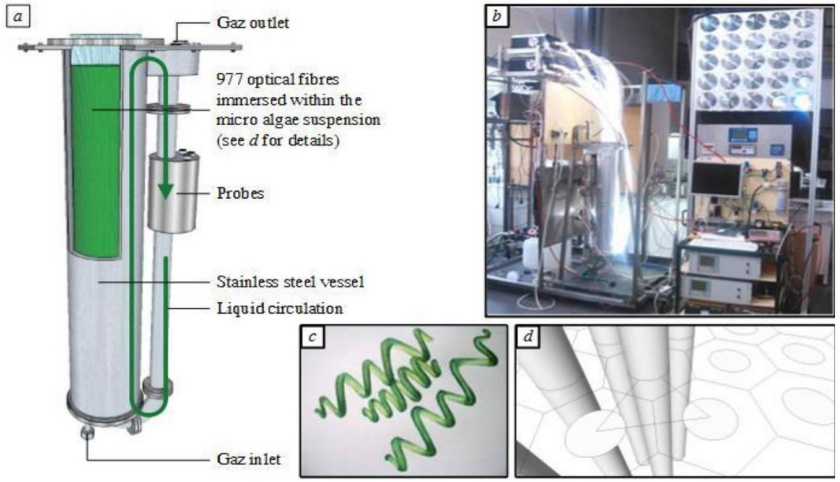
Contre Labo. Entreprise
Contrat d'Appui

https://gitlab.com/meso-star/star-engine
https://www.meso-star.com/projects/index.html

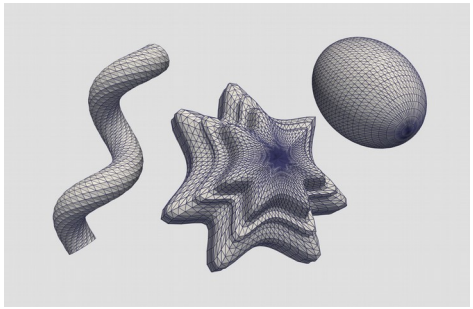
Le code Schiff.

(Institut Pascal, Rapsodee, Laplace, CNRS)

Contexte : Optimisation de procédés Photo-bio-reactifs.



Objet du code : Evaluation des propriétés radiatives (ka, ks, phi) de « soft particles ».



Schiff - Radiative properties of soft particles

The Schiff program computes the **radiative properties** of soft particles. It uses the **Monte-Carlo** method to solve Maxwell's equations within the L. Schiff's approximation as presented in Charon et al. 2015.The main advantages of using Monte-Carlo are: the possibility to address **any shape** of particle, and the results are provided with a **numerical accuracy**.

Download Schiff 0.3.1

- GNU/Linux: tarball / pgp
- Sources: tarball / pgp

For a mixture of soft particles, Schiff estimates the total **cross-sections** (absorption, scattering and extinction cross-sections) in addition of the **phase function**, its cumulative and its inverse cumulative.

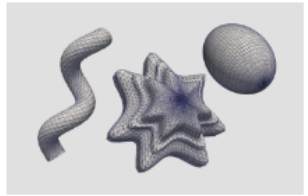
The set of particles to simulate is defined by its **refractive index** - provided at various wavelengths - and a **geometry distribution** that controls how the particles look like into the mixture. More precisely, this distribution describes the main shapes of the particles (sphere, cylinder, helical pipe, *etc.*) and their statistical variation according to the distribution of their parameters (gaussian, lognormal, *etc.*).

Schiff is available on GNU/Linux 64-bits. It is a free software release under the GPLv3+ license. You are welcome to redistribute it under certain conditions; refer to the license for details.

A straight interface

The Schiff program is a **command-line tool** that processes input data, performs computations, write results and that's all. It makes no assumptions on how the input data are created excepted that it has to follow the expected file formats. The simulation results are also provided as is, in a raw ASCII format.

This thin interface is particularly well suited to be **extended** and **integrated into any toolchain**. According to the user needs, the optical properties of the particles can be entered manually or generated by an external program. In the same way, the output data can either be directly interpreted or post-processed by any script with respect to the targeted toolchain.



Examples of particle shapes handled by Schiff.

Quick start

Download the desired archive of Schiff and verify its integrity against its PGP signature. Then extract it. Finally source the provided `schiff.profile` file to register the Schiff installation for the current shell priorly to the invocation of the `schiff` program.

```
$ source ~/Schiff-0.3.1-GNU-Linux64/etc/schiff.profile
$ schiff -h
```

The Solstice **reference documentation** is provided trough man pages. Use the `man` command-line to consult it.

```
$ man schiff
$ man schiff-geometry
$ man schiff-output
```

https://www.meso-star.com/projects/schiff.html

Le code Solstice.

(labex Solstice, PROMES, CNRS)

Contexte : Optimisation de champ optique pour procédés solaires.

Objet du code : Evaluer le flux solaire concentré dans une géométrie complexe.

```
- sun: {dni: 1000, pillbox: {half_angle: 0.1}}
- geometry: &small-circle
  - material: {matte: {reflectivity: 1}}
    plane: {clip: [{operation: AND, circle: {radius: 0.5}}]}
- geometry: &big-sphere
  - material: {virtual}
    sphere: {radius: 2, slices: 128}
- entity: {name: reflector, primary: 1, geometry: *small-circle}
- entity: {name: receiver,  primary: 0, geometry: *big-sphere}
```

Solstice - The solar plant simulation tool

Solstice computes the **total power** collected by a concentrated solar plant, and evaluates various **efficiencies** for each primary reflector: it computes losses due to cosine effect, to shadowing and masking, to orientation and surface irregularities, to reflectivity and to atmospheric transmission. These data provide insightful information when looking for the optimal design of a concentrated solar plant. Solstice is powered by a **Monte-Carlo solver**, which means that each result is provided with its **numerical accuracy**.

Solstice 0.6.1 is available

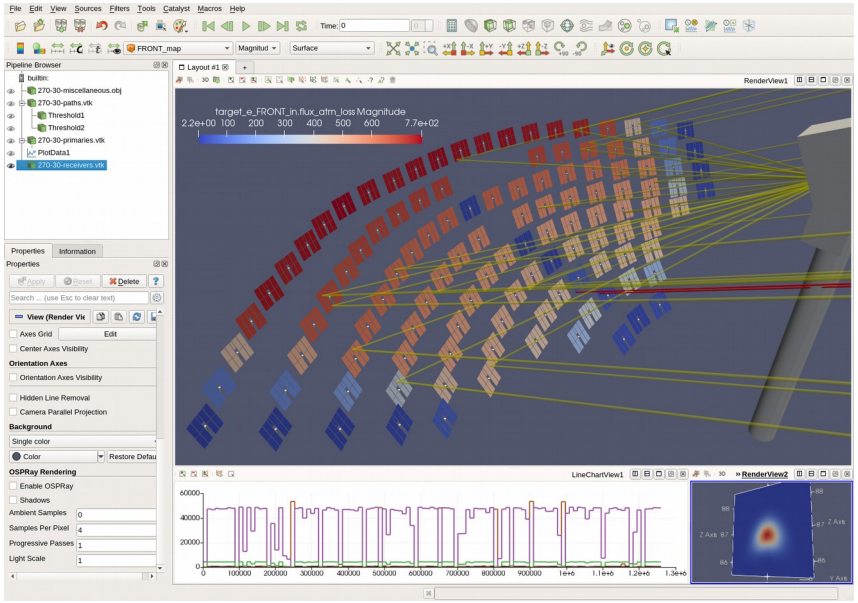
- GNU/Linux: tarball / pgp
- Windows: zip / pgp
- Sources: zip / pgp

Solstice is specifically designed to handle **complex solar facilities**. A solar plant can be composed of any number of geometries of different types like hyperbolas, parabolas, cylindro-parabolas, planar polygons, cylinders, spheres, hemispheres and cuboids. Behind analytic shapes, one can also use any **external mesh** stored in a **ST**ereo **L**ithography file format.

The orientation of the reflectors can be either defined manually or **automatically computed** by Solstice according to the sun direction and the animation constraints of the reflectors.

Mirror, matte and dielectric materials are supported. **Spectral effects** are also taken into account as long as the relevant physical properties are provided; it is possible to define the spectral distribution of any physical property, including the input solar spectrum and the absorption of the atmosphere, at any spectral resolution.

Solstice is available on GNU/Linux and Microsoft Windows 7 or later. It is a free software released under the GPLv3+ license. You are welcome to redistribute it under certain conditions; refer to the license for details.



https://www.meso-star.com/projects/solstice.html

1- Qu'a réalisé l'industrie du cinéma ?

2- Star-Engine et quelques projets.

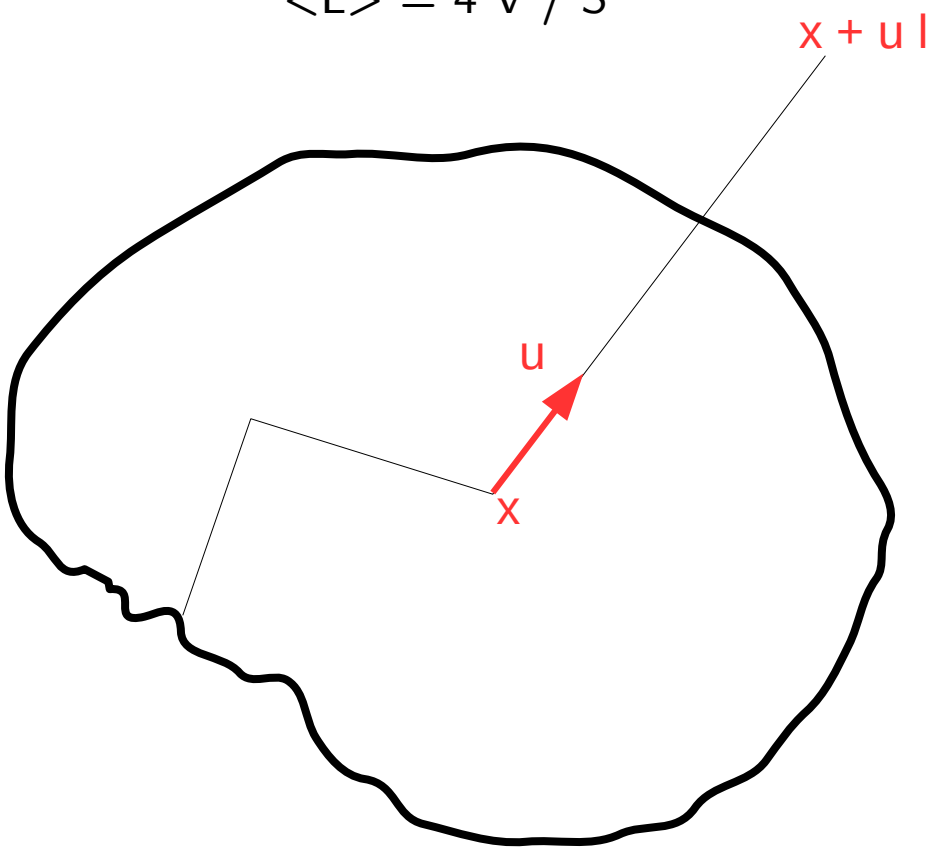
3- Un exemple complet : le code 4V/S.

4- Conclusion.

L'exemple 4V/S.

Propriété d'invariance d'une marche aléatoire diffuse dans un domaine de volume V et de surface S.

<L> = 4 V / S




```
sum = 0
for i=1 to N
        poids = 0
        tirer un point x uniformément sur la surface
        tirer une direction u lambertienne
        while ( )
                tirer une longueur l de diffusion en k exp(-k l)
                if (x + u l > hit.distance)
                        poids = poids + hit.distance
                        break
                else
                        poids = poids + l
                        x = x + u l
                        tirer une direction u de diffusion
        sum = sum + poids

L_moy =  sum / N
```

S. Blanco and R. Fournier, *An invariant property of diffuse random walks*, **Europhysics Letter** 2003

https://gitlab.com/meso-star/star-4v_s

L'exemple 4V/S.

```c
#include <star/s3d.h>
#include <star/smc.h>

int
main(int argc, char* argv[])
{
  struct s3d_scene* scene = NULL;
  struct smc_device* smc = NULL;
  struct smc_integrator integrator;
  struct smc_estimator* estimator = NULL;

/* ----------------------------
 ...
Récupération des données d'entrée
…
 ----------------------------*/

 /* Setup Star-MC */
  smc_device_create(NULL, NULL, SMC_NTHREADS_DEFAULT, NULL, &smc);
  integrator.integrand = &s4vs_realization; /* Realization function */
  integrator.type = &smc_double; /* Type of the Monte Carlo weight */
  integrator.max_steps = max_steps; /* Realization count */
  integrator.max_failures = max_steps / 1000; /* cancel if 0.1% of the realization fail */

  /* Solve */
  smc_solve(smc, &integrator, &ctx, &estimator);

  return 0 ;
}
```

L'exemple 4V/S.

```c
res_T
s4vs_realization(void* out_length, struct ssp_rng* rng, void* context)
{
  struct s4vs_context* ctx = (struct s4vs_context*)context;
  struct s3d_attrib attrib;
  struct s3d_primitive prim;
  double sample[4];
  double normal[3];
  float u[3], x[3], st[2];
  const float range[2] = {0.f, FLT_MAX};
  struct s3d_hit hit;
  double w = 0;
  double sigma = 0;
  int keep_running = 0;
  float r0, r1, r2;

  /* Sample a surface location, i.e. primitive ID and parametric coordinates */
  r0 = ssp_rng_canonical_float(rng);
  r1 = ssp_rng_canonical_float(rng);
  r2 = ssp_rng_canonical_float(rng);
  s3d_scene_view_sample(ctx->view, r0, r1, r2, &prim, st);

  /* retrieve the sampled geometric normal and position */
  s3d_primitive_get_attrib(&prim, S3D_GEOMETRY_NORMAL, st, &attrib);
  d3_normalize(normal, d3_set_f3(normal, attrib.value));
  s3d_primitive_get_attrib(&prim, S3D_POSITION, st, &attrib);
  f3_set(x, attrib.value);

  /* Cosine weighted sampling of the hemisphere around the sampled normal */
  ssp_ran_hemisphere_cos(rng, normal, sample);
  f3_set_d3(u, sample);

  /* Find the 1st hit from the sampled location along the sampled direction */
  s3d_scene_view_trace_ray(ctx->view, x, u, range, &prim, &hit);

  /* No intersection <=> numerical imprecision or geometry leakage */
  if(S3D_HIT_NONE(&hit)) return RES_UNKNOWN_ERR;

  keep_running = 1;
  while(keep_running) { /* Here we go for the diffuse random walk */
    sigma = ssp_ran_exp(rng, ctx->ks); /* Sample a length according to ks */

    if(sigma < hit.distance) {
      int i;
      FOR_EACH(i, 0, 3) x[i] = x[i] + (float)sigma*u[i];
      d3_normalize(sample, d3_set_f3(sample, u));
      f3_set_d3(u, ssp_ran_sphere_hg(rng, sample, ctx->g, sample));

      /* sample a new direction */
      s3d_scene_view_trace_ray(ctx->view, x, u, range, NULL, &hit);

      w = w + sigma;

      /* No intersection <=> numerical imprecision or geometry leakage */
      if(S3D_HIT_NONE(&hit)) return RES_UNKNOWN_ERR;

    } else { /* Stop the random walk */
      w = w + hit.distance;
      keep_running = 0;
    }
  }

  SMC_DOUBLE(out_length) = w;
  return RES_OK;
}
```

1- Qu'a réalisé l'industrie du cinéma ?

2- Star-Engine et quelques projets.

3- Un exemple complet : le code 4V/S.

4- Conclusions, Perspectives.

Conclusions, Perspectives.

- L'utilisation du « **lancer de rayons** » n'est pas nouvelle mais ...

- Méso-Star porte l'ambition de **transposer la rupture** qui s'est produite dans l'industrie du cinéma à **l'ingénierie de systèmes physiques.**

- Projet de développement d'un « **bac à sable** » pour faciliter l'entrée des données.

- Projet thermique Monte-Carlo.
Couplage conducto-convecto-radiatiaf. Evaluation du propagateur (Green).
Monte-Carlo « symbolique »*. MC non-linéaire ...

→ **Outils d'analyse, Modèle rapide, outils pour l'inversion et l'optimisation ...**

*M Galtier et al, *A symbolic approach for the identification of radiative properties*, **JQSRT** 2017