Secured Cloud-based 3D Medical Data Visualization

Manoranjan Mohanty[1] Pradeep Atrey[2] Wei Tsang Ooi[1]

[1] Department of Computer Science, National University of Singapore, Singapore

[2] Department of Applied Computer Science, University of Winnipeg, Canada.

MOTIVATION:

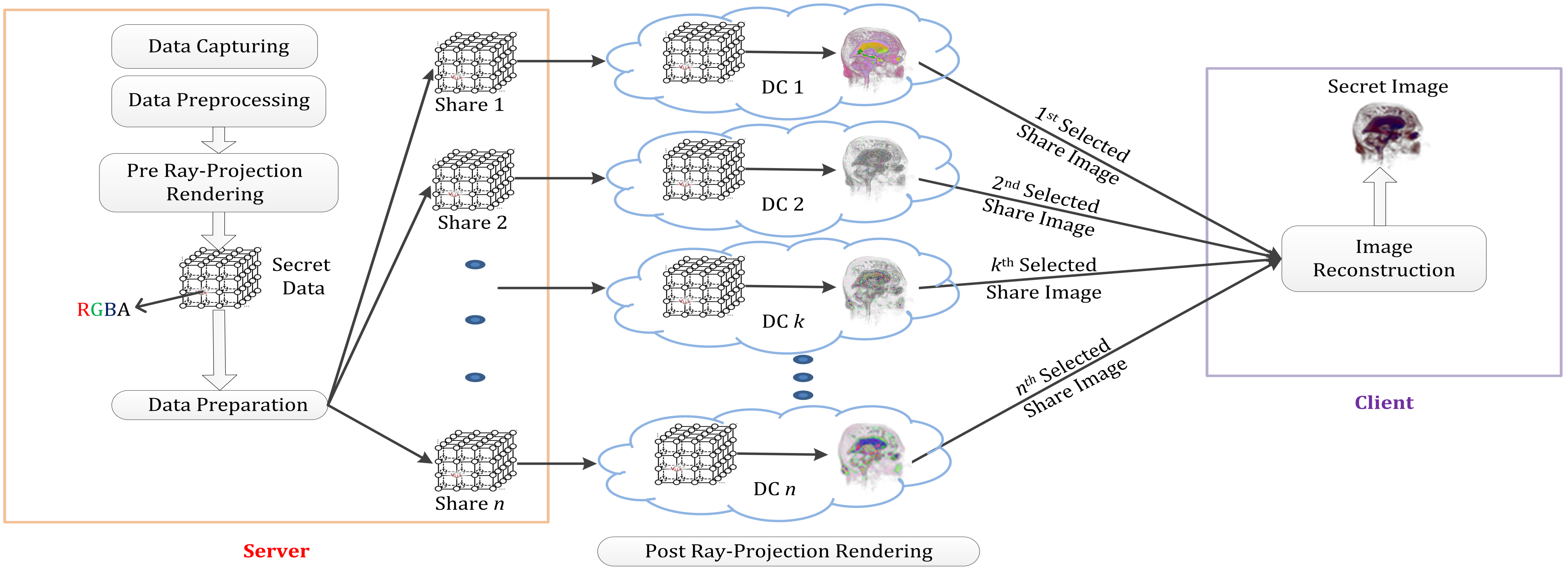
- Rendering complex medical data on cloud servers frees the hospitals from management of its own infrastructure.
- Allowing a third party cloud data center access to medical data leads to security issues as private data can be leaked (data confidentiality issue), and medical data can be tampered (data integrity issue).
- We focus on 3D volumetric medical data rendering using pre-classification volume ray-casting and hiding of color-coded health information within these volumes.

OBJECTIVE:

- To integrate a cryptographic technique with a direct volume rendering algorithm such that
  - Cloud data centers render 3D medical volume as images that hide the color-coded health information.
  - An authorized user (e.g., a doctor) can recover the color-coded information from the rendered images.
  - Tampering with the 3D medical data or the rendered image can be detected by the authorized user.

METHOD

- We integrate a variant of Shamir's (*k, n*) secret sharing method that does not use modular prime operation with the pre-classification volume ray-casting. The important rendering steps are explained in details below.



Data Capturing

Data Preprocessing

Pre Ray-Projection Rendering

Secret Data

RGBA

Data Preparation

Share 1

Share 2

Share *n*

Server

DC 1

DC 2

DC *k*

DC *n*

Post Ray-Projection Rendering

1st Selected Share Image

2nd Selected Share Image

*k*th Selected Share Image

*n*th Selected Share Image

Secret Image

Image Reconstruction

Client

Pre Ray-Projection Rendering

The pre ray-project rendering step includes gradient estimation, classification, and shading. It maps scalar value of each data voxel to color and opacity. This step can be pre-computed.

Data Preparation

- The data preparation step creates *n* number of *share volume data* by
  - Sharing the color of each data voxel by a variant of Shamir's (*k,n*) secret sharing method. The *p*th share of the volume data contains the *p*th share of the data voxels.
  - Copying the opacity value of each data voxel into each share.
- Each share volume data is sent to a cloud data center for rendering/storage.

Post Ray-Projection Rendering

- Each cloud data center performs interpolation and composition on its share volume data.
- Along a ray, composited opacity in the *p*th share volume data is equal to the rendered opacity in secret volume data.
- Along a ray, composited color in the *p*th share volume data is a share of the rendered color in secret volume data, i.e.,

$$C_p = C + \alpha_p, \text{where } \alpha_p = \sum_{i=0}^{k-1} a_i p_i, 0 \le a_i \le C$$

color in pth share volume data

color in secret volume data

random number

- Each of the *n* cloud data center renders the share volume into a *share image* with hidden color information.

Image Reconstruction

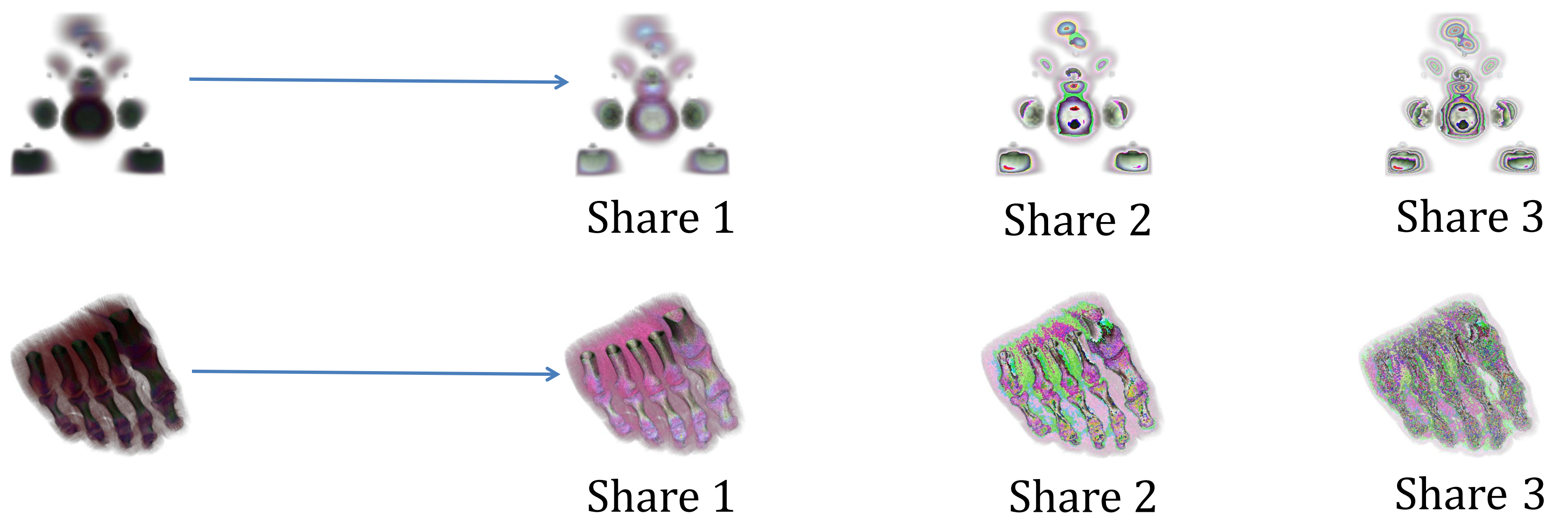
- The user recovers the *secret image*, with the color information from the share images of at least *k* cloud data centers.
- The opacity of a pixel in a share image become the opacity of the pixel in the secret image.
- The colors of a pixel of *k* share images are Lagrange interpolated to find the color of the pixel in the secret image.

PROPERTIES

- Color coded health information hidden to an adversary having access to at most (*k-1*) data centers.
- Tampering with 3D data or rendered image can be detected if total number of data centers is more than the minimum number of data centers required to reconstruct the secret image (i.e., if *n* > *k*).
- The variant of Shamir's secret sharing is not as secured as Shamir's secret sharing as it is not a perfect secret sharing technique.
- Shape of the image is available to cloud data centers.

IMPLEMENTATION AND RESULTS

- We simulate the server, cloud data centers, and the client in a laptop with Intel 2.0 GHz dual core processor and 4GB RAM.
- We modify the volume ray-casting module in open source visualization software VTK by integrating (3, 5) secret sharing.



Images of different volume data

Images from different viewpoints