外 文 翻 译

**毕业设计题目：基于UE4的交互渲染及其在服饰设计中的应用**

**Interactive rendering based on UE4 and its application in apparel design**

**原文 1：Modeling and Rendering of Volumetric Clouds in Real-Time with Unreal Engine 4**

**译文1：使用虚幻四引擎实时建模和渲染体积云**

**原文 2：Virtual Try-On through Image-Based Renderin**

**译文 2：基于图像渲染的虚拟试穿**

原文1

**Modeling and Rendering of Volumetric Clouds**

**in Real-Time with Unreal Engine 4**

*Lukasz Nowak, Artur Bak, Tomasz Czajkowski, Konrad Wojciechowski*

*Polish-Japanese Academy of Information Technology, Warsaw, Poland*

*Orka Postproduction Studio, Warsaw, Poland*

**3.4 Rendering**

We use a ray marching method implemented according to for rendering clouds in our system. The Ray marching algorithm is implemented in a separate material. Such material is then applied to the mesh, which was chosen to contain clouds. Method presented in and used by us is suited for cube-shaped meshes. The general idea of the ray marching method consists of “shooting” rays from every pixel of the mesh (with ray marching material applied) and moving through them with constant step defined by the user. At each step, operations defining this pixel colour are executed, e.g. the amount of absorbed light is added up. Number of steps is finite and defined by user. Usually, the end condition for a given pixel is the execution of all number of steps. Graphical presentation of ray marching technique is presented in Fig. 2.

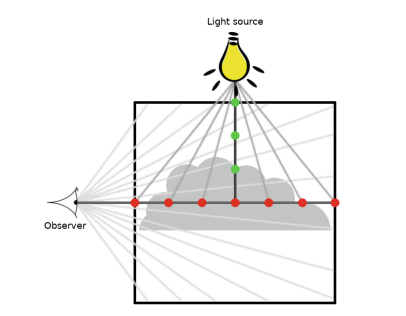


Fig. 2. Brief overview of ray marching technique. At every step (marked by red circles) of ray coming from observer or more precisely mesh’ pixel, additional steps (green circles) to light source are performed for volume shading calculation. (Color figure online)

**3.5 Lighting**

Implementation of a lighting model based strictly on the real world can be difficult and impossible in real-time. This is due to how the light scatters in the cloud. The most difficult task is the approximation of in-scattering existing in the cloud (multiple scattering of light inside the cloud). We currently use the lighting model proposed in, which implements only out-scattering. It gives a decent visual result but we consider implementation of more realistic looking solution. Similar approach was presented in and extension of that method in.

**3.6 Shadows**

For obtaining a shadow of the cloud we created a simple material with Blend mode (material parameter) set to Masked, which takes the pseudo volume texture as a parameter. The material calculates the shape of the cloud based on the pseudo volume texture by summing the density values from its every slice in a given position. Such material is then applied on the regular UE4 plane mesh. The plane mesh is set to not be visible, but casts its shadow. Thanks to the usage of Masked Blend Mode, the shadow shape matches the cloud shape. The shadow shape calculated in this material is in fact a projection of the cloud to the XY plane. One drawback of this method results from this fact, the created cloud shadow will always present the cloud in the XY plane projection independently of the light direction. But for the scenario in which the clouds are high in the atmosphere, this problem would not even be noticeable by the observer in most cases.

The cloud should not only cast shadows, but also allow other objects to cast their shadows on it. Two solutions of this problem were considered by us, using a distance field shadow as in and using a custom per object shadow map presented in. First method gives good visual results but has one drawback, the distance field generation in Unreal Engine 4 does not work with Skeletal meshes, i.e. meshes containing the skeleton for animations. We have found a workaround for this limitation by using sockets. Sockets allow to connect any mesh to specific Skeletal mesh bone. We connected regular spheres with changed scale and/or rotation to some bones in a way that shape of mesh was covered by these spheres. Spheres were set to be not visible but cast shadow. As added spheres are not Skeletal meshes, distance fields can be generated for them. This way we have obtained an approximated shadow of Skeletal mesh on volumetric cloud. This method allows the shadow to move as the added spheres are moving along with the bones to which they are connected while the Skeletal mesh is playing animation. One inconvenience of this workaround is that additional spheres must be added and connected to bones manually in Unreal Engine 4 editor. The second method of achieving the shadows of other objects on clouds, i.e. using custom per object shadow map, was implemented in UE4 according to. Although this method does not have a problem with various types of meshes (e.g. Skeletal meshes) it has other significant drawbacks. The first drawback is a poor performance compared to the previous method. The second drawback is the necessity of making the shadows softer to look more realistic on cloud that may impact the performance even more. Another problem are graphical artifacts like distortion or disappearance of shadow when object casting it is moved away from the cloud (as it could not be captured by Scene Capture component). Because of these drawbacks we decided to use the first of described methods, i.e. distance fields shadows.

译文1

作者：Lukasz Nowak, Artur Bak, Tomasz Czajkowski, Konrad Wojciechowski

国籍：波兰

出处：Lecture Notes in Computer Science Vol: 11114 68-78

使用虚幻四引擎实时建模和渲染体积云

**3.4 渲染**

我们使用光线行进的方法来渲染我们系统中的云。光线行进算法作用于单独的材质中，然后这种材质附到云的网格模型上，我们提供和使用的方法适用于立方体网格。光线行进方法的主要思想包括从网格的每个像素“发射”射线（应用于光线行进的材质）并持续不断的按照用户定义的步骤来行进。在每一步的操作中，定义了执行之后的像素颜色，吸收的光量相加。用户定义的执行的步骤是有限的，通常，结束的条件是给定的像素执行了所有的步骤。光线行进的光线图如图2所示。

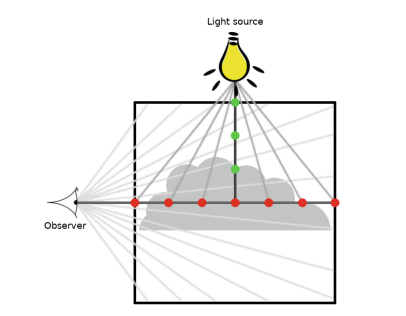


图2 光线行进图

简要概述一下光线行进技术，在光线来自观察者或更精确的网格像素的每个步骤（红圆圈标记的地方），执行光源的附加步骤（绿色圆圈）以进行体积着色的计算（线上的颜色图）

**3.5 光照**

严格遵循现实世界的光照模型实施起来是非常困难，几乎不可能的。这是由于光线在云中是会内散射的，最困难的任务是模拟内散射（光线在云内会多次散射）。我们目前使用的光照模型，仅能实现外散射，它给了我们一个表面的视觉效果，但我们会考虑实现更加逼真的外观，在该方法中提出了类似的方法并对其进行了扩展。

**3.6 阴影**

为了获取云的阴影，我们是用Blend创建了一个简单的材质mode（材质参数）设置为Masked，使用伪体积纹理作为参数，材质会基于将每个给定位置的切片密度值相加得到的伪体积纹理计算云的形状，然后将这种材质应用于常规的UE4平面网格，平面网格设置为不可见，但会投射其阴影。由于使用了Masked Blend Mode的关系，使得阴影的形状和云的形状一致。在该材质中计算的阴影形状实际上是云对XY平面的投影，这也导致了该方法的一个缺点，所创建的云阴影将始终独立于光方向在XY平面呈现云的投影，但是对于大气层中云层较高的情况，大多数情况下观察者都不会注意到这个问题。

云不仅应该投射阴影，还应该允许其他对象投射他们的影子，我们考虑了两个解决方案，使用距离场阴影和使用每个对象的自定义阴影贴图。第一种方法提供了良好的视觉效果，但有一个缺点，虚幻四引擎中生成的距离场并不适用于骨骼网格物体，即包含动画骨骼的网格物体。我们通过使用插槽（sockets）找到了解决此类限制的方法，插槽允许将任何网格连接到特定的骨骼网格的骨骼位置，我们通过这些球体覆盖网格形状的方式将具有变化的缩放和旋转的常规球体连接到一些骨骼，这些球设置为不可见但投射阴影，由于添加的球体不是骨骼网格物体，因此可以为它们生成距离场。通过这种方式，我们在体积云上获得了骨骼网格的近似阴影，此方法允许当骨骼网格播放动画时，阴影随着添加的球体与它们所连接的骨骼一起移动。此解决方法的一个不便之处是必须在虚幻四引擎编辑器中手动添加其他球体并将其连接到骨骼。第二种实现方法是对每个对象自定义阴影贴图，尽管该方法对各种类型的网格（例如骨骼网格物体）没有问题，但是它具有其他显著的缺点，第一个缺点是与之前的方法相比性能较差，第二个缺点是需要使阴影更柔和，在云上看起来更逼真，这可能会对性能产生更大的影响。另一个问题是当对象投射远离云时（因为它无法被Scene Capture组件捕获），会造成图形失真或阴影消失等图形伪像。由于这些缺点，我们决定使用所描述的第一种方法，即距离场阴影。

原文2

**Virtual Try-On through Image-Based Rendering**

*Stefan Hauswiesner, Student Member, IEEE, Matthias Straka,*

*And Gerhard Reitmayr, Member, IEEE*

**2 RELATED WORK**

Interactive virtual try-on applications need to track the user’s position and pose to augment him or her with garments. The garments themselves can be reconstructed from camera images. Virtual- and mixed-reality dressing applications that combine these aspects have been suggested.

**2.1 Motion Capture**

Motion capture, or human pose tracking, is the task of determining the user’s body pose. It usually involves a pose and shape model that is fitted to sensor data and, therefore, comprises a model-based tracking problem. We only consider optical, marker-less pose tracking, because we do not want users to wear markers.

Pose tracking from multiple video streams was used for animating and rendering people. Recent GPU based implementations adapt pose and shape in real time. A system for markerless human motion transfer was suggested to transfer motions from one person to another. Another system can modify the user’s body shape in a virtual mirror. Recent developments in sensor technologies have enabled the acquisition of depth images in real time, which opened up new possibilities for pose tracking with a single camera. Ganapathi et al. [9] have shown how to track full body motions using a time-of-flight camera. The more recent Microsoft Kinect device allows for real-time recording of color and depth images at a very low cost, as well as high-quality real-time human pose estimation.

**2.2 Clothes Reconstruction**

Many virtual dressing applications draw a textured clothes mesh over a camera image. Obtaining that mesh is a key aspect of such systems. Some approaches use CAD models, which are labor intensive to create.

Clothes can also be reconstructed from images. Reconstructing the garment from a video sequence is a hard task, especially because of occlusions and the nonrigid shape of cloth. Many approaches use markers on the cloth for capturing, which makes them less suitable for our method. More recent approaches do not require markers. They usually use a shape-from-stereo approach and apply complex processing to the data to account for occlusions. However, all approaches that rely on point correspondences that are computed from the image data assume a certain texturedness of the garment. By using the light dome of or a laser scanner this limitation can be removed, but such hardware is expensive and processing cannot be performed in real time.

Once the shape of a garment is digitized it needs to be fitted to the user’s body model. This is a complex problem that is usually not handled in real time.

**2.3 Virtual Try-On**

Previous methods work by finding the best matching data set in a previously recorded database that contains all possible poses of the user. These systems first learn and then search a database of poses by using a pose similarity metric. The best match is used to deform a texture to fit the user. However, like many other retexturing approaches they operate in 2D and, therefore, do not allow the user to view him- or herself from arbitrary viewpoints.

The Virtual Try-On project offers a set of applications for various tailoring, modeling, and simulation tools. Three-dimensional scans of real garments are acquired by color-coded cloth. Cloth surface properties are measured from real samples. MIRACloth is a clothes modeling application that can create garments, fit them to avatars, and simulate them. However, both Virtual TryOn and MIRACloth do not include a mixed reality component that allows users to see realistic clothing on themselves immediately.

Kinect-based body scanning enables virtual try-on applications at low costs but systems with a single sensor require multiple views. A system that uses manually modeled garment meshes was introduced. Similar to our system, it performs a nonrigid registration to align garments and user.We use a nonlinear formulation instead of Laplacian surface editing to avoid finding correspondences.

Transferring garment meshes from one human model to another is an important step for many virtual try-on applications. It requires a shape and pose adaption process. Volumetric Laplacian deformation can achieve this.

Motion capture, reconstruction, and retexturing are used to render dressed people. Other virtual mirrors are restricted to specific tasks, like augmenting logos or shoes and even faces.

**2.4 Prerequisites**

The virtual dressing room that is used for this work consists of a 2x3 meter footprint cabin with green walls. Ten cameras are mounted on the walls: two in the back, two on the sides, and six in the front. The cameras are synchronized and focused at the center of the cabin, where the user is allowed to move freely inside a certain volume (see Fig. 2 left). All cameras are calibrated intrinsically and extrinsically and connected to a single PC. Most processing is performed on the graphics processing unit (GPU). The output device is a 4200 TV that is mounted to the front wall in a portrait orientation. In such a setup, silhouettes can be extracted from the camera images quickly and robustly by background subtraction. Silhouettes make novel view synthesis very efficient. The image-based visual hull algorithm creates a depth map for arbitrary viewpoints. Such a depth map can be textured with the camera images for realistic image-based rendering.

译文2

作者：Stefan Hauswiesner, Matthias Straka, Gerhard Reitmayr

国籍：奥地利

出处：Transactions on Visualization and Computer Graphics, v 19, n 9, p 1552-1565, 2013

基于图像渲染的虚拟试穿

**2 相关工作**

交互式虚拟试穿应用程序需要跟踪用户的位置和姿势，以添加他或她的服装，服装本身可以从相机图像重建，这种虚拟现实和混合现实相结合的服饰试穿应用已经被提出。

**2.1 动作捕捉**

动作捕捉，或者说是人体动作跟踪，它是确定用户身体姿势的方法，通常需要用传感器捕捉模型的姿势或形状，因此，这就需要我们解决基于模型的跟踪问题。我们只考虑光学无标记姿势跟踪，我们不希望用户需要佩戴标记。

渲染动画和人体模型需要来自多个视频流的动作捕捉，近年来人们使用基于GPU的方法来实时调整姿势和形状，推荐使用无标记人体运动转移的系统来将运动从一个人转到另一个人，另一个系统可以在虚拟环境中修改用户的身体形状。传感器技术的最新发展使得我们能够实时获取到深度图像，这为使用单个相机进行姿势跟踪创造了新的可能性。例如Ganapathi，已经能够使用一个飞行相机来跟踪全身运动，最近的Microsoft Kinect设备也允许用户以非常低的成本实时记录彩色和深度图像，以及高质量的实时人体姿势捕捉。

**2.2 服装重建**

大部分虚拟试穿软件通过绘制衣服的纹理图片并添加到网格模型上来实现，获取该网格是这种系统的关键方面，一些人使用CAD模型，这样的工作量是非常庞大的。

服装也可以通过图像来重建，然而从一些视频序列中重建服装是非常困难的，尤其在一些被遮挡或者布料是非刚体的情况下，许多方法使用布料上的标记来进行捕获，但这并不太适合我们的方法。最近人们使用的方法并不需要标记，他们通常使用立体声形状方法，并对数据应用复杂的处理来解决遮挡问题。然而，所有依赖于图像数据计算的方法都假定了服装的某种纹理，使用灯罩或激光扫描仪可以消除这种限制，但硬件过于昂贵并且不能进行实时处理。

一旦服装的形状被数字化，它还需要适合用户的身体模型，这是一个复杂的问题，通常不能实时处理。

**2.3 虚拟试穿**

先前的方法通过在之前记录的数据库中找到包含用户的所有可能姿势的最佳匹配数据集来工作，这些系统首先通过使用姿势相似性度量来学习，然后搜索姿势数据库，最佳的匹配结果用于创建纹理来适应用户。然而与许多其他重新构造的方法一样，这需要在2D中操作，因此，用户无法从任意视角查看自己。

Virtual Try-On项目为各种裁剪、建模和仿真工具提供了一组应用程序，通过彩色编码布料获得真实服装的三维扫描，从真实样品测量布料的表面特性。MIRACloth是一款服装造型应用程序，可以创建服装，使其适合用户，并模拟他们。但是，Virtual Try-On和MIRACloth都不包含混合现实组件，用户无法立即看到真实的衣服。

基于Kinect的人体扫描能够用低成本实现虚拟试穿效果，但具有单个传感器的系统需要多个视图，引入了使用手动建模的服装网格的系统，与我们的系统相类似，它能够以非刚体的形式来对齐到服装和用户，我们使用非线性公式而不是拉普拉斯表面编辑来避免找到对应关系。

将服装网格从一个人体模型转移到另一个人体模型是许多虚拟试穿应用程序的重要步骤，它需要一个形状和姿势适应的过程，体积拉普拉斯变形可以实现这一点。

渲染一个穿着衣服的人需要用到动作捕捉、重建和重新构造，其他虚拟试穿仅限于特定的任务，例如给脸部或鞋子增加徽章标志。