

基于单视点视频的舰艇尾浪三维重建

郭亚飞, 姚怡超, 陆颖, 苏智勇

(南京理工大学自动化学院, 南京 210094)

摘要: 针对海战虚拟环境中对舰艇尾浪模拟的需要, 提出了一种基于单视点视频对舰艇尾浪进行三维重建的方法。对视频图像进行高斯滤波等预处理, 并从视频图像中提取尾浪区域, 基于 *Shape From Shading(SFS)* 方法计算舰艇尾浪的高度场, 生成尾浪网格模型, 基于 *Compute Unified Device Architecture(CUDA)* 技术计算网格顶点的法线, 重建出舰艇尾浪的三维模型。实验结果表明: 重建的尾浪具有较高的真实感, 提高了尾浪的重建速度, 在细节与实时性之间取得了较好的平衡, 能够满足实际应用需要。

关键词: 尾浪提取; SFS; 尾浪重建; CUDA

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2015) 10-2475-08

DOI:10.16182/j.cnki.joss.2015.10.037

Ship Stern Wave Reconstruction from Single Viewpoint Video

Guo Yafei, Yao Yichao, Lu Ying, Su Zhiyong

(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: To simulate the ship track in the marine virtual battle environment, a novel algorithm based on the single viewpoint video was proposed to reconstruct the ship stern wave. Gaussian filter was used to preprocess the video image and *extract the long tail region from the video image; SFS algorithm was employed to compute the height field of the ship stern wave and generate its grid mesh;* The normal of each vertex was computed based on the CUDA technology. The experimental results show that the reconstructed stern wave yields a good sense of reality. At the same time, it improves the stern wave reconstruction speed while achieving a balance between the realistic and real-time rendering performance to meet actual needs.

Keywords: long tail extract; SFS; wave reconstruction; CUDA

引言

舰艇尾浪是影响虚拟海战场景真实性的一个重要因素。随着计算机技术和仿真技术的不断发展, 虚拟现实技术在军事领域的应用变得越来越广泛深入。现如今虚拟现实技术已成为开发研制大型

海军电子信息系统、复杂舰载武器装备、海战作战环境下多武器平台和模拟海战战况的重要工具。而舰船的尾浪是模拟海战中的一个重要部分, 它不仅会影响虚拟海战环境的真实性, 而且还能刺激用户对虚拟场景的主观感受并且提高用户的沉浸感。所以, 对舰船尾浪的三维仿真具有很重要的意义。

近年来, 出现了很多针对舰艇尾浪的实时三维可视化研究。比较典型的是 Jerry^[1]运用 Iwave 方法建立可以交互干扰的静态水面, 算法比较复杂, 实时性较差。另一类主要是针对舰船航迹的不规则性和内在复杂性等特征, 通过分析简化航迹模型, 提



收稿日期: 2015-07-24 修回日期: 2015-08-07;
基金项目: 国家自然科学基金(61300160);
作者简介: 郭亚飞(1990-), 男, 山西长治人, 硕士生, 研究方向为系统仿真与虚拟现实; 陆颖(1990-), 女, 江苏南通人, 硕士生, 研究方向为图像分析; 姚怡超(1990-), 男, 江苏常州人, 硕士, 研究方向为三维仿真与图像分析。

http: www.china-simulation.com

出了基于粒子系统的实现方法。比较典型的是基于 Goss^[2-4]方法建立粒子的初始速度向量,对船行波简化近似,实现非常简单,但效果欠佳。赵欣^[5]等、胡一笑^[6]等根据 Kelvin 船行波模型建立粒子初始位置和随机发射速度,效果更真实。虞宏毅^[7]通过将多个粒子图样渲染成高度场纹理,作用于水面高度场实现尾浪效果。基于物理模型对尾浪进行三维仿真,虽然能够获得较好的精确性,但是也带来了建模复杂和计算开销大的问题。而基于粒子系统的尾浪三维模拟,虽然建模简单,但是其重建效果不够真实,而且不能灵活地对不同类型的尾浪实现快速的三维重建。

本文提出了一种基于单视点视频对舰艇尾浪进行三维重建的方法。在许多海战仿真领域,人们往往关注的是尾浪在视觉上的逼真性而对其物理上的精确性并没有要求。而基于视频三维重建技术的发展为本文提供了一种新的思路^[8-9]。本文提出的方法是从包含尾浪的视频图像中提取尾浪,然后重建其三维模型。这种方法不仅不需要考虑航迹复杂的物理方程,而且其重建的尾浪也具有较好的真实性。

1 算法流程

本文的算法流程主要由三部分构成:尾浪提取、尾浪的高度场计算和法线计算,具体流程图如图 1 所示。第一步先对包含尾浪的视频图像进行预处理,然后对视频图像进行分割,提取尾浪区域。第二步是基于 Shape From Shading(SFS)方法求得舰艇尾浪的高度场,从而生成尾浪的三维模型。第三步计算尾浪网格各顶点的法线。

2 基于单视点视频的尾浪三维重建

2.1 尾浪提取

通过对收集到的大量尾浪视频进行分析,发现视频中除了尾浪之外,通常还有其它物体,如天空,海岛等。因此,在重建之前,需要从这些视频图像源中提取所需要的尾浪信息,从而提高重建的精度

和准确度。

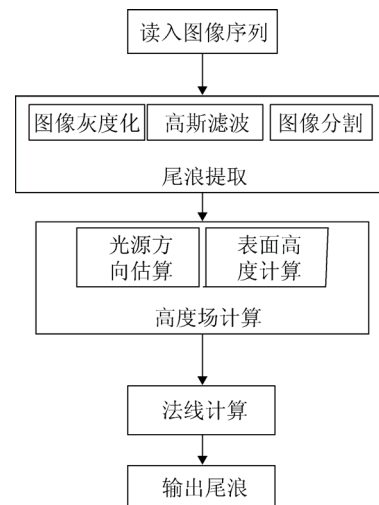


图 1 算法流程图

2.1.1 图像预处理

图像预处理的目的是消除图像中的无关信息恢复有用信息,从而改进图像分割的可靠性。本文中图像预处理主要包括图像灰度化和图像的高斯滤波。

将彩色图像转化为灰度图像的过程称为图像的灰度化。在彩色图像中每个像素的颜色有 R, G, B 三个分量,每个分量有 255 个值可取,这样一个像素点有 1 600 多万的颜色变化范围,而灰度化之后,像素点的变换范围为 255,这就大大减少了高斯滤波的计算量。

为了滤除高斯噪声,提高图像分割的准确性,本文设计了高斯滤波器来抑制高斯噪声。高斯滤波器是根据高斯函数的形状来选择权值的线性平滑滤波器。引入高斯函数为:

$$h(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

高斯滤波的思路是:对高斯函数进行离散化;以离散点上的高斯函数值为权值,对采集到的灰度矩阵的每个像素点做一定范围领域内的加权平均,即可有效消除高斯噪声。在图像处理中,高斯滤波的主要原理是采用高斯算子对图像进行卷积运算。在实现时,就是采用一个矩阵模板进行加权运算,

该模板我们称为高斯核。所以高斯核是高斯滤波的求解的关键。很显然,它是通过二维高斯函数计算得来的。

离散的高斯卷积核 $H: (2k+1) \times (2k+1)$ 维, 其元素计算方法为:

$$H_{i,j} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i-k-1)^2+(j-k-1)^2}{2\sigma^2}}$$

其中 σ 为方差, k 确定核矩阵的维数。

2.1.2 图像分割

经过预处理后,要对图像进行尾浪提取。通过对大量含有尾浪的视频进行统计分析发现,海洋的边界一般与天空相连接,具有明显的直线分界线。基于这一先验知识,本文采用概率霍夫变换来检测边界线^[10]。

在进行概率霍夫变换之前,要对图像进行轮廓检测。对图像的图像边缘检测可以通过对梯度大小进行简单的阈值化得到二值的边缘图像。然而,这有两大缺陷。第一点,检测到的边缘过粗,难以实现物体的精确定位。第二点,难以找到这样一个阈值,既能足够低以检测到所有重要边缘,同时又不至于包含过多次要的边缘。所以我们采用 Canny 算法来进行边缘检测。Canny 算子通常基于 Sobel 算子。核心思想是使用两个不同阈值以确定哪些点属于轮廓,一个低值和一个高值。其算法详解见文献[10]。

概率霍夫变化是对霍夫变化的改进。霍夫变化的思想为:在原始图像坐标系下的一个点对应了参数坐标系中的一条直线,同样参数坐标系的一条直线对应了原始坐标系下的一个点。然后,原始坐标系下呈现直线的所有点,它们的斜率和截距是相同的。所以它们在参数坐标系下对应于同一个点。这样在将原始坐标系下的各个点投影到参数坐标系下之后,看参数坐标系下有没有聚集点,这样的聚焦点就对应了原始坐标系下的直线。而概率霍夫变换的思想和霍夫变换的思想基本相同。

为了能够检测到带端点的直线,同时解决由于

霍夫变换查找边缘点的方式而可能带来的错误检测,研究人员提出了概率霍夫变换的方法。这种方法不仅提高了检测直线的准确性,也有利于边界线的选取。其与霍夫变换的不同之处在于:

- 1) 随机获取边缘图片上的前景点,映射到极坐标画曲线;
- 2) 当极坐标系里面有聚焦点达到最小投票数,将该点对应 $x-y$ 坐标系的直线找出来;
- 3) 搜索边缘图片上前景点,在直线上的点连成线段,然后这些点全部删除,并且记录该线段的参数,就是起始点和终止点。

用概率霍夫变换检测到所有的直线后,用直线的斜率 k 和直线的长度 L 为条件来选取边界线。

2.2 高度场计算

本文基于 SFS 方法求取尾浪的高度场。SFS 算法根据图像中物体表面的明暗变化,恢复物体表面各点的相对高度,从而重建物体的三维形状。SFS 算法主要由两个部分组成:光源方向估算和高度计算。

2.2.1 光源方向估算

物体的光源方向是 SFS 模型的一个重要参数,其准确性直接决定了三维重建的效果。本文采用 Penland^[11]的参数估计算法估算光源方向。

如图 2 所示, σ_s 为光源倾角, τ_s 为光源偏角。光源方向:

$$\vec{L} = (\sin \sigma_s \cos \tau_s, \sin \sigma_s \sin \tau_s, \cos \sigma_s) \quad (1)$$

本文采用文献[11]的方法计算光源的偏角和倾角。其中,偏角为:

$$\tau_s = \arctan \left\{ \frac{E_{xy} \left\{ \frac{\bar{y}_{Li}}{\sqrt{\bar{x}_{Li}^2 + \bar{y}_{Li}^2}} \right\}}{E_{xy} \left\{ \frac{\bar{x}_{Li}}{\sqrt{\bar{x}_{Li}^2 + \bar{y}_{Li}^2}} \right\}} \right\} \quad (2)$$

式中, \bar{x}_{Li} , \bar{y}_{Li} 分别为光照方向在点 (x_i, y_i) 沿 x, y 方向的局部估计值。即: $\begin{bmatrix} \bar{x}_{Li} \\ \bar{y}_{Li} \end{bmatrix} = (B^t B)^{-1} B^t d\vec{E}$, 而

$$B = \begin{bmatrix} 1 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 \\ 0 & 1 \\ -\sqrt{2}/2 & \sqrt{2}/2 \\ -1 & 0 \\ -\sqrt{2}/2 & -\sqrt{2}/2 \\ 0 & -1 \\ \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix}, \quad d\vec{E} = \begin{bmatrix} \delta E_1 \\ \delta E_2 \\ \vdots \\ \delta E_M \end{bmatrix}$$

其中 $\delta E_i (i=(0 \dots 7))$ 为该点与周围八个点的像素差值，从右边开始按逆时针方向求取。 $E_{xy} \{\square\}$ 为局部估计值的平均值。

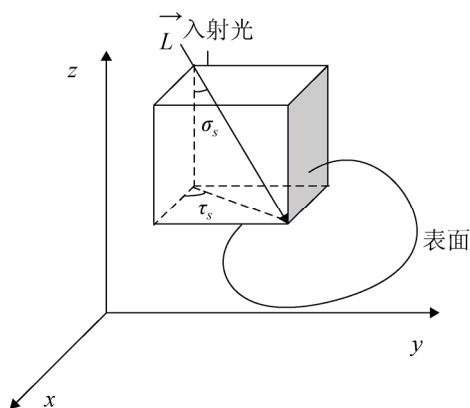


图 2 入射光源示意图

倾角由式(3)和式(4)可求得：

$$\begin{aligned} f_1(\sigma_s) &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} E(\alpha, \beta) d\alpha d\beta \\ f_2(\sigma_s) &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} E^2(\alpha, \beta) d\alpha d\beta \\ f_3 &= f_1(\sigma_s) / \sqrt{f_2(\sigma_s)} \end{aligned} \quad (3)$$

式中， α 为物体上任意一点 (x_i, y_i) 的法矢偏角， β 为点 (x_i, y_i) 的法矢倾角， $E(\alpha, \beta) = \max\{0, \cos(\alpha - \tau_s) \sin \beta \sin \sigma_s + \cos \sigma_s \cos \beta\}$ 而

$$\frac{E_{xy}(T)}{\sqrt{E_{xy}\{T^2\}}} = f_3(\sigma_s) \quad (4)$$

式中： T 表示图像像素值； $E_{xy} \{\square\}$ 是图像像素值均值。

当 $\frac{E_{xy}(E)}{\sqrt{E_{xy}(E)}} > f_3(0) = 0.96191$ 时，可令 $\sigma_s = 0$ 。

2.2.2 高度的计算

对尾浪表面的高度求取采用的是 Tsai 和 Shah^[12] 提出的有限差分(finite different)方法。尾浪表面满足朗伯体假设，其表面方程如式(5)所示：

$$E(x, y) = R(p, q) = \frac{pp_s + qq_s + 1}{\sqrt{p^2 + q^2 + 1} \sqrt{p_s^2 + q_s^2 + 1}} = \frac{\cos \sigma + p \cos \tau \sin \sigma + q \sin \tau \sin \sigma}{\sqrt{1 + p^2 + q^2}} \quad (5)$$

其中， $(p, q, 1)$ 以及 $(p_s, q_s, 1)$ 为矢量的空间梯度表示，分别为 (x, y) 处的表面方向以及光源方向， $z(x, y)$ 为该点的表面高度，而

$$p(x, y) = \frac{\partial z(x, y)}{\partial x}, \quad q(x, y) = \frac{\partial z(x, y)}{\partial y}。$$

所以方程(5)可以重新写为方程(6)形式：

$$\begin{aligned} 0 &= f(E(x, y), Z(x, y), Z(x-1, y), Z(x, y-1)) = \\ &E(x, y) - R(Z(x, y) - Z(x-1, y), \\ &Z(x, y) - Z(x, y-1)) \end{aligned} \quad (6)$$

对于一个给定的点 (x, y) 和图像像素值 $E(x, y)$ ，对方程(6)的 Z^{n-1} 线性近似为公式(7)。

对于一副 $N \times N$ 的图像，在线性化系统中总共有 N^2 这样的等式。他们可形成一个线性系统，用雅克比迭代方法可以解决这个系统。

$$\begin{aligned} 0 &= f(E(x, y), Z(x, y), Z(x-1, y), Z(x, y-1)) \approx \\ &f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), \\ &Z^{n-1}(x, y-1)) + (Z(x, y) - Z^{n-1}(x, y)) \times \\ &\frac{\partial}{\partial z(x, y)} f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), \\ &Z^{n-1}(x, y-1)) + (Z(x-1, y) - Z^{n-1}(x-1, y)) \times \\ &\frac{\partial}{\partial z(x-1, y)} f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x-1, y), \\ &Z^{n-1}(x, y-1)) + (Z(x, y-1) - Z^{n-1}(x, y-1)) \times \\ &\frac{\partial}{\partial z(x, y-1)} f(E(x, y), Z^{n-1}(x, y), \\ &Z^{n-1}(x-1, y), Z^{n-1}(x, y-1)) \end{aligned} \quad (7)$$

对式(7)进行简化可得到(8)式:

$$0 = f(Z(x, y)) \approx f(Z^{n-1}(x, y)) + (Z(x, y) - Z^{n-1}(x, y)) \frac{d}{dZ(x, y)} f(Z^{n-1}(x, y)) \quad (8)$$

经过 n 次迭代之后, $Z(x, y) = Z^n(x, y)$ 可以用公式(9)求解

$$Z^n(x, y) = Z^{n-1}(x, y) + \frac{-f(Z^{n-1}(x, y))}{\frac{d}{dZ(x, y)} f(Z^{n-1}(x, y))} \quad (9)$$

其中, $\frac{d}{dZ(x, y)} f(Z^{n-1}(x, y)(x, y)) = -1 \times$

$$\left(\frac{(p_s + q_s)}{\sqrt{p^2 + q^2 + 1} \sqrt{p_s^2 + q_s^2 + 1}} - \frac{(p + q)(pp_s + qq_s + 1)}{\sqrt{(p^2 + q^2 + 1)^3 \sqrt{p_s^2 + q_s^2 + 1}}} \right) \quad (10)$$

现在, 假设初始的高度为 $Z^0(x, y) = 0$, 那么物体的表面高度可以用(8)式求得。

2.3 法线计算

根据求得的尾浪高度场能够重构出尾浪的三维网格。对于网格中各顶点的法线计算, 本文用和该点相邻所有面的法线的平均值作为该点的法线, 计算公式如下式(11)所示, 图 3 所示为顶点与其相邻面法线的关系图, 图中 v_i 表示顶点, n_i 表示面的法线。为了提高计算速度, 本文在 CUDA^[13] 平台上进行法线的计算。

$$n_v = \sum_{i=1}^m n_i / m \quad (11)$$

式中: n_v 为待求顶点的法线; n_i 为邻接面的法线; m 为邻接面的个数。

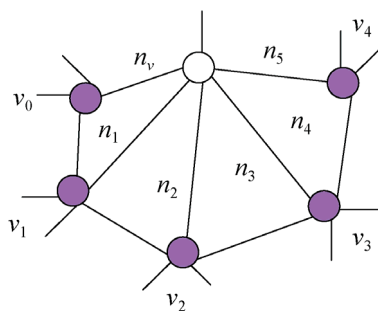


图 3 顶点与其相邻面法线的关系图

基于 CUDA 的法线计算中, 每个顶点的法线运算是相互独立的, 并且过程都很相似, 因此十分适合用于并行计算来进行加速。其算法步骤如下:

(1) 在 CPU 和 GPU 端开辟内存空间存放输入输出数据。在 CPU 端创建图像大小的三维数组, 以图像左上角为原点, 以行为顺序依次存放各个像素点的位置, 创建一个同样大小的数组用于存放从 GPU 中复制回来的计算结果。在 GPU 中分配两个和 CPU 中同样大小的三维数组, 一个在常量内存上, 一个在全局内存上。常量内存上存放从 CPU 中复制进来的位置数据, 全局内存上存放经过计算过的法线数据。

(2) 把计算得到的每个像素点的位置和高度数据拷贝到 GPU 中。

(3) 启动并行计算线程格。数据虽然是以三维数组的形式从 CPU 传递到 GPU 的, 但实际上是与二维的原理是一样的。而线程标识 (x, y) 是二维的, 所以只要把每个标识与每个像素点像对应即可。

(4) 进行并行计算, 每个线程处理对应索引的数据, 每个线程计算共用这个顶点的 6 个面的法线的平均值并存放 GPU 全局内存数组的对应的索引处。

(5) 并行计算结束后, 把 GPU 全局内存上法线结果数据拷贝回 CPU 用于存放法线的数组中。

3 实验结果及分析

本文以 VS 2008 为开发环境, 基于 OpenSceneGraph 3.0^[14] 和 OpenCV 2.4.3 平台对本文提出的基于单视点视频的尾浪重构算法进行了实验。实验硬件环境如下: CPU 型号为 Intel(R)Core(TM)i5-3470, 主频为 3.20GHz, GPU 型号为 GeForce GTX 650, 显存为 3723MB, 4G 内存。

为了证明分割算法的有效性, 本文取图 4 所示视频帧为例, 给出了经灰度化和高斯滤波后的直线检测结果图。如图 4 所示。图 4 中, (a) 为实验原

图, (b)为经过灰度化的图, (c)为未经过高斯滤波的 Canny 边缘检测图, (d)为未经过高斯滤波用概率霍夫检测到的直线图。图 4 的表格中, 第一行是不同方差 σ 值时, Canny 算法检测到的边缘轮廓图, 第二行是不同方差 σ 值时, 概率霍夫检测检测直线的结果图。

观察图 4 可以看出, 高斯滤波器对图 4(b)的去噪效果直接影响了尾浪区域提取的准确性。经过高斯滤波之后, 检测到的直线明显减少, 说明视频图像中的噪声被有效地抑制。从图中可以看出, 随着方差 σ 值的增大, Canny 检测到的边缘轮廓越来越少, 而概率霍夫变换检测到的直线先增多后减少。高斯核中方差 σ 越高, 视频图像就越平滑, 图像的梯度差越小, 故 Canny 检测到的边缘轮廓越来越少。检测到的轮廓线会直接影响概率霍夫变换检测直线的结果。而概率霍夫检测到的直线结果会直接影响尾浪区域提取的准确性。由于尾浪区域提取的准确性越高, 高度场计算的结果越精确, 综合分析

在不同方差下概率霍夫变换检测到的直线情况, 本文实验中选取高斯核中方差 $\sigma = 3$ 。

图 5 中图(a)是经过预处理后取的尾浪区域图, 选取的分界线直线的斜率 $k = 0$, 并且斜率为零的直线中长度最长的直线为分界线, 图(b)是只有高度场没有法线和颜色信息的三维重构结果。图(c)是三维重建的网格图, 图(d)是最终重建的三维结果图, 颜色信息可直接从视频图像中读取。从三维网格图中可以看出, 尾浪的高度趋势是非常明显的, 而且重建的结果和原图像是非常相近的。图 6 和图 7 是一部视频的不同帧的效果显示, 其分辨率大小为 496×160 。图 8 和图 9 是另一部视频的结果, 其分辨率大小为 432×230 。图 10 是视频二在 CUDA 下和在 CPU 下三维重建的用时对比图。

根据实验结果可以看出, 本文能够在 CUDA 平台下, 实现尾浪的实时三维重建, 其结果也非常接近于其原视频。但是也可看出, 本文算法对尾浪的水花飞溅的重建的效果并不是很理想。

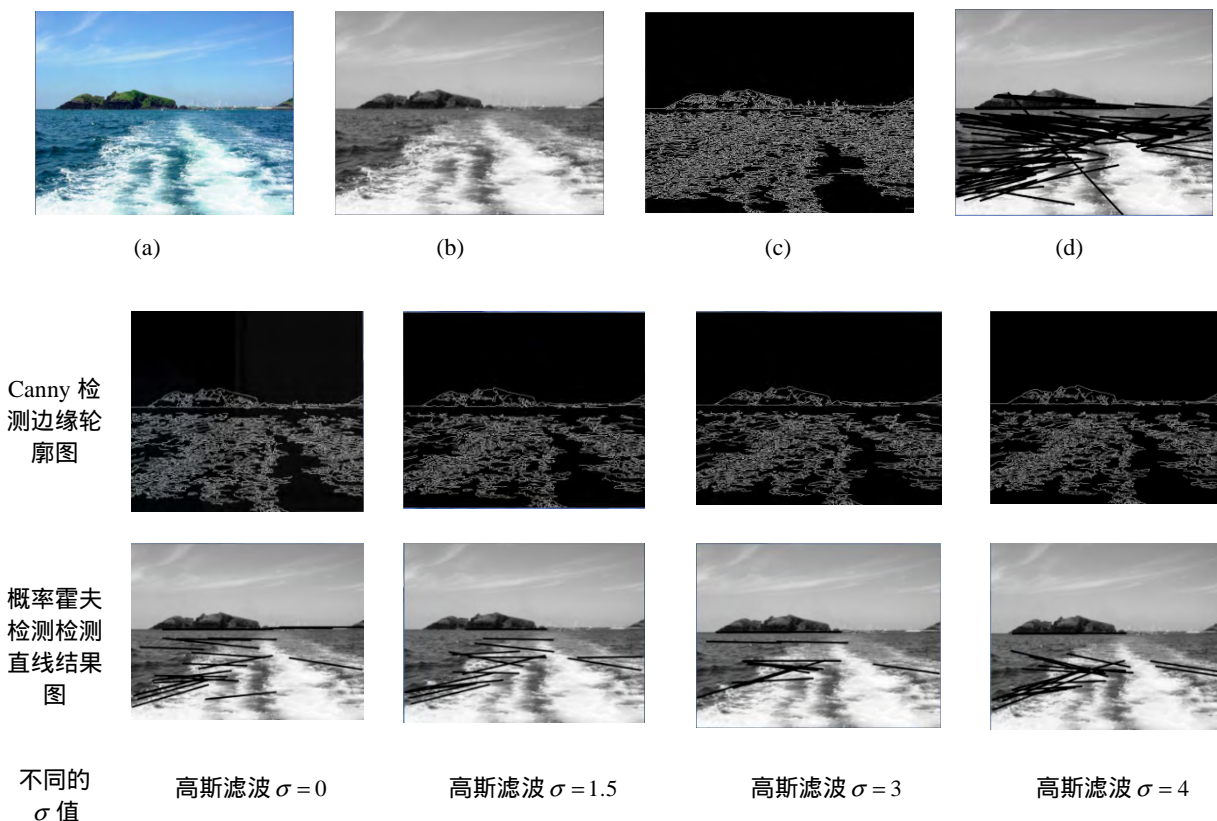


图 4 预处理结果

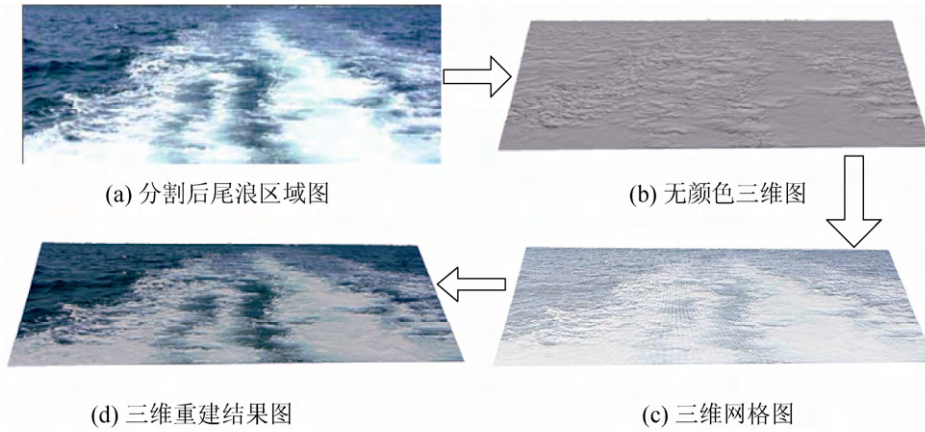


图 5 三维重建结果图

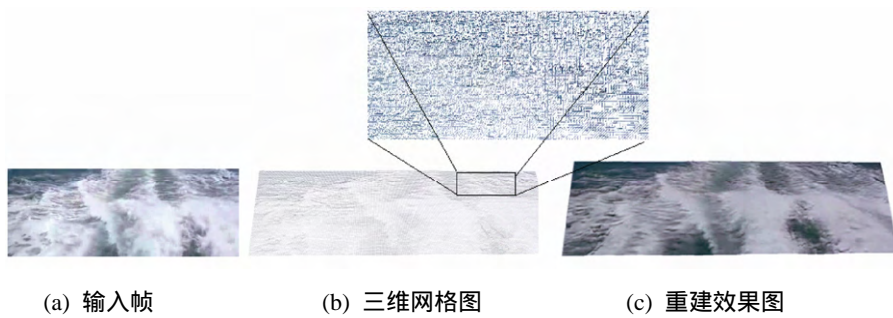


图 6 视频一第 20 帧三维网格

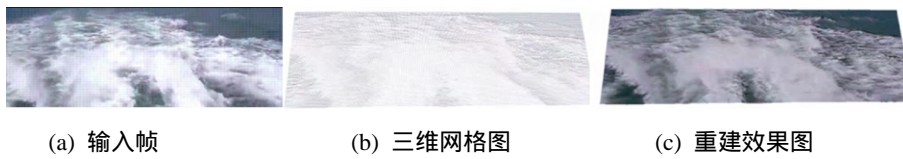


图 7 视频一第 110 帧三维网格图

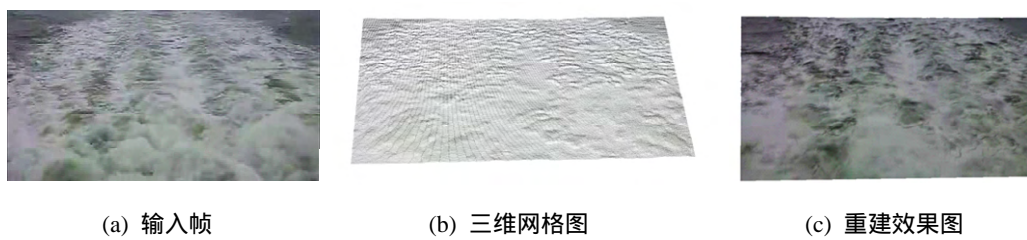


图 8 视频二第 2 帧三维重建效果图

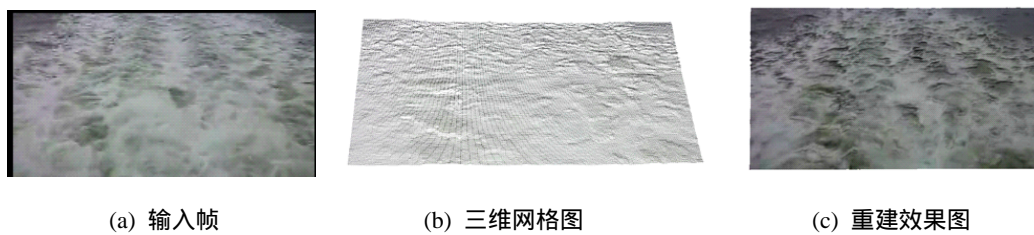


图 9 视频二第 100 帧三维重建效果图

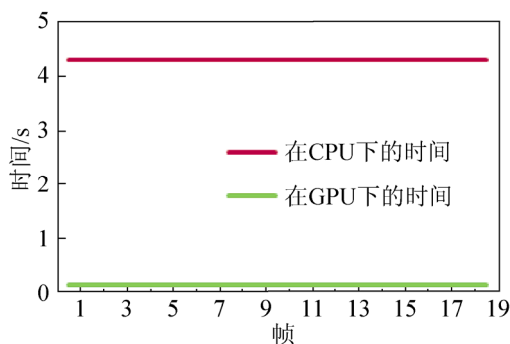


图 10 在 CUDA 和 GPU 下的时间对比图

4 结论

本文提出了一种基于单视点视频对舰艇尾浪重建的方法，并利用 CUDA 对其重建速度进行了优化。实验结果表明，本文方法避免了复杂的物理方程计算，并且其灵活性大大的增加。基于这种方法可以对不同海况，不同型号的舰船的尾浪进行三维重建。但是，本文也有其不足之处：第一，算法要求含有尾浪的视频图像中尾浪和天空必须相接，有明显的分界线；第二，无法模拟出尾浪中水花四处飞溅的效果。这也是本文下一步的工作重点。

参考文献：

[1] Tessendorf J. Interactive water surfaces [C]// Game Programming Gems 4. Hingham, MA, USA: Charles River Media, 2004.
[2] Goss M E. A real time particle system for display of ship wakes [J]. IEEE Computer Graphics & Applications (S0272-1716), 1990, 10(3): 30-35.
[3] 尹勇, 任鸿翔, 张秀凤, 等. 航海仿真虚拟环境的海浪

视景生成技术 [J]. 系统仿真学报, 2002, 14(3): 313-315.
[4] 马天, 黄建国. 应用水波弥散特性的舰船航迹三维可视化新方法 [J]. 西安交通大学学报, 2011, 45(8): 49-53.
[5] 赵欣, 李凤霞, 战守义. 基于粒子系统的舰船航迹仿真 [J]. 计算机工程, 2008, 34(15): 22-24.
[6] 胡一笑, 姜昱明. 船行波的三维可视化仿真 [J]. 计算机应用, 2008, 28(z1): 247-249.
[7] 虞宏毅. 大规模水场景的实时绘制 [D]. 杭州: 浙江大学, 2007.
[8] 王承博, 朱登明. 数据驱动的大规模水面动画合成方法 [J]. 图学学报, 2014, 35(4): 491-497.
[9] Marshall D, Cosker D, Saunders T, et al. Water Surface Modeling from a Single Viewpoint Video [J]. IEEE Transactions on Visualization and Computer Graphics (S1077-2626), 2013, 19(7): 1242-1251.
[10] Robert Laganirer, 张静. OpenCV2 计算机视觉编程手册 [M]. 北京: 科学出版社, 2013.
[11] Zheng Q, Chellappa R. Estimation of illuminant direction, albedo, and shape from shading [C]// Computer Vision and Pattern Recognition, 1991, Proceedings CVPR '91., IEEE Computer Society Conference on IEEE. USA: IEEE, 1991: 540-545.
[12] Ping-Sing T, Shah M. Shape from shading using linear approximation [J]. Image & Vision Computing (S0262-8856), 1994, 12(94): 487-49.
[13] Sanders, kandrot E, 聂学军. GPU 高性能编程 CUDA 实战 [M]. 北京: 机械工业出版社, 2011.
[14] 肖鹏, 刘更代, 徐明亮. OpenSceneGraph 三维渲染引擎编程指南 [M]. 北京: 清华大学出版社, 2010.
[15] 胡一笑, 姜昱明. 船行波的三维可视化仿真 [J]. 计算机应用, 2008, 28(z1): 247-249.