

学号	姓名	论文规范性 (10)	问题分析与调研 (30)	方案创新性 (20)	实验结果分析与讨论 (40)	结课论文总成绩 (100)
21301160	刘荧	8	28	18	38	92

北京交通大学

基于 MSAA 和 TAA 的抗锯齿算法研究与实现

学号：21301160

姓名：刘荧

导师：吴雨婷

北京交通大学

2024 年 6 月

摘要

走样（aliasing）问题是信号处理和图像处理中的一个常见现象，会导致信号或图像的失真和质量下降。抗锯齿（Anti-Aliasing）技术在现代 3D 计算机图形学渲染中扮演着至关重要的角色，它能够有效地改善图像的真实感和美观度。

虽然 SSAA（Super Sample Anti-Aliasing，超级采样抗锯齿）通过增加空间采样率来抗锯齿，虽然能获得较好的效果，但其高昂的计算时间和硬件成本使其在实时渲染中的应用并不理想。MSAA（MultiSampling Anti-Aliasing，多重采样抗锯齿），为节省计算时间和硬件成本提供了方法，MSAA 是寻找出物体边缘部分的像素，然后对它们进行缩放处理。由于只是物体的外层像素进行缩放处理，忽略掉了不会产生锯齿的内部像素，所以不会像处理 SSAA（超级采样抗锯齿）那样需要庞大的计算量，比起 SSAA 来更有效。为了应对实时渲染中的抗锯齿需求，TAA（Temporal Anti-Aliasing，时间抗锯齿）技术提供了一种解决方案。它通过在时间维度上分摊采样，提高了采样率而不显著增加计算成本。TAA 通过对连续帧之间的像素进行插值计算，有效减少了锯齿、抖动和失真，同时保持了细节的准确性和图像的真实感。

本文学习了 MSAA 和 TAA 的原理，并通过实验运用了 MSAA 和 TAA，改进锯齿边缘和纹理的处理效果，可消除空间和时间走样现象，提高图像质量。

关键词：抗锯齿技术；MSAA；TAA

目录

摘要	II
1 引言	1
2 相关工作介绍.....	2
3 实验方法.....	3
3.1 MSAA	3
3.2 TAA	4
4 实验设置.....	5
4.1 实验环境	5
4.2 实验设计	5
5 实验结果与分析.....	7
5.1 实验结果	7
5.2 实验结果分析	10
6 结论	12
参考文献	13

1 引言

在计算机图形学中，图像通过将图形光栅化为像素来进行显示。然而，由于像素是离散的，图形边缘会出现锯齿状，这影响了图像的质量。为了解决这一问题，人们开始研究抗锯齿技术。早期的抗锯齿技术依赖于几何算法，例如线段平滑和多边形平滑算法，这些算法在图形生成过程中可以避免锯齿边缘的出现，但由于效率较低，不适合实时渲染。

随着计算机硬件的发展，研究重点转向了基于硬件的抗锯齿技术。最早的硬件抗锯齿技术是超级采样抗锯齿（SSAA），通过对图像进行超采样以减少锯齿边缘的出现。然而，SSAA 需要大量计算资源，因此不适用于实时渲染。为了解决这一问题，研究人员开发了基于硬件的多重采样抗锯齿（MSAA）技术。MSAA 通过对每个像素进行多次采样，减少了锯齿边缘的出现，同时保持较高的渲染速度，因此广泛应用于游戏和其他实时渲染应用中。

然而在实时渲染中，MSAA 并不能有很好的效果。一方面，TAA 通过在时间维度上分摊计算负荷，大幅度降低了每帧的计算需求，从而提高了性能效率，使其更适合实时渲染。另一方面，TAA 利用前后多帧的信息，通过插值计算实现平滑过渡，能更有效地减少闪烁和抖动，更好地捕捉和保留图像中的细节。TAA 被提出是为了提供一种在性能和图像质量之间取得良好平衡的抗锯齿解决方案，特别适合实时渲染中的复杂和动态场景。

本文基于 MSAA 和 TAA 的抗锯齿算法进行实验，分别研究和运行这两种算法，并对比和分析了不同的采样方法对这两种抗锯齿算法的效果的影响。这两种算法都可以在不同的程度上消除图像中的走样问题，同时保持图像的真实感和细节，改进图形渲染效果，减少锯齿状边缘的出现，从而提高画面的质量和流畅度。

2 相关工作介绍

二十一世纪初，人们提出超采样方法，每个像素使用多个样本来综合评判颜色。Beets^[1]等人提出的 SSAA（Super-Sample Anti-Aliasing）技术，采用高于屏幕分辨率的方式渲染图像，并将其缓存到帧缓冲区中，然后下采样到屏幕分辨率进行显示。虽然通过更高的分辨率渲染图像可以减少锯齿边缘的产生，但这也增加了需要绘制的像素数量，从而导致渲染性能显著下降。

基于 SSAA 技术，MSAA^[2]（Multi-sample Anti-aliasing，多重采样抗锯齿）技术对采样计算进行了优化。与 SSAA 不同，MSAA 在每个像素中采样若干位置不同的子样本，并通过平均这些子样本的颜色值来得到最终的像素颜色。由于计算的像素数量更少，MSAA 可以显著减少性能开销。然而，在延迟渲染方法中，MSAA 会丢失几何信息，导致无法正常工作。为了解决这一问题，TAA 被提出，TAA 通过利用前后帧的信息来进行抗锯齿，减少了对每帧需要进行的大量计算，从而提升了整体性能。TAA 还能够更好地处理这些次像素细节，通过在时间上对像素进行平均，能够消除更多的锯齿和闪烁问题。其次 TAA 主要依赖于软件算法，可以在更多种类的设备上实现，具有更广泛的兼容性。

此外，研究人员也开始探索将深度学习与帧间抗锯齿结合，以实现更优异的抗锯齿效果。NVIDIA 公司最近推出了 DLSS（Deep Learning Super Sampling，深度学习超级采样）技术，这种方法能够显著提升图像质量。DLSS 成功地将帧间抗锯齿与深度学习技术融合，带来了更加支持这一技术。总体而言，抗锯齿技术的发展速度很快，不仅提升了图像质量，还能够在低端设备上实现实时性需求，从而大幅改善用户体验。从最早的超采样技术到如今结合深度学习的抗锯齿技术，抗锯齿技术的不断进步为用户提供了更加自然的视觉感受。

3 实验方法

3.1 MSAA

超采样技术（SSAA）高了采样率，以一个更大的分辨率来渲染场景，然后再把相邻像素值做一个过滤，比如计算相邻几个子像素的平均值，来得到最终的图像。如图 1 所示，首先经对每个像素取 n 个子采样点，然后针对每个子像素点进行着色计算。然后将所有子像素的平均值作为该像素的值，最后合成最终的图像。SSAA 中每个子采样点都要进行单独的着色，在像素着色器比较复杂的情况下很费时。

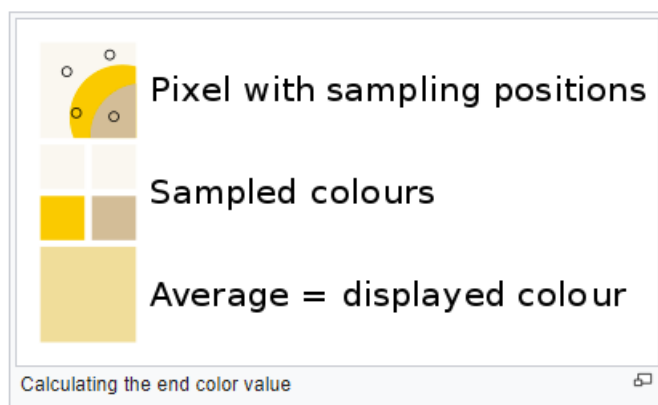


图 1 SSAA 像素值计算示例

MSAA 对于那些子采样点只计算一个覆盖信息和遮挡信息来把像素的颜色信息写到每个子采样点里面，最终根据子采样点里面的颜色值来通过重建过滤器来降采样生成目标图像。

覆盖是通过判断一个图形是否跟一个指定的像素重叠来决定的。覆盖是通过测试一个采样点是否在像素的中心来决定的。如图 2 所示的是一个三角形的覆盖信息。蓝色的点代表采样点，每一个都在像素的中心位置。红色的点代表三角形覆盖的采样点。遮挡表示被一个图形覆盖的像素是否被其它的像素覆盖了，此时采用 z -buffer 的深度测试。深度和遮挡测试是对每个子样本独立进行的。这意味着即使一个像素内部的不同子样本被不同的多边形覆盖，这些子样本的深度和遮挡信息仍然会被单独存储和处理。MSAA 对当前像素覆盖掩码不为 0 的子采样点进行着色计算。具体而言，只对像素的中心点的顶点属性进行插值，然后使用这个插值值在片段程序中进行一次着色计算。将这次着色计算的结果应用到所有覆盖掩码不为 0 的子采样点上。最终对这些子采样点的颜色结果进行平均，得到最终像素的颜色值。由于每个像素只进行一次或少量的着色计算，而不是对每个子

采样点都进行完整的着色计算，大大减少了计算量。

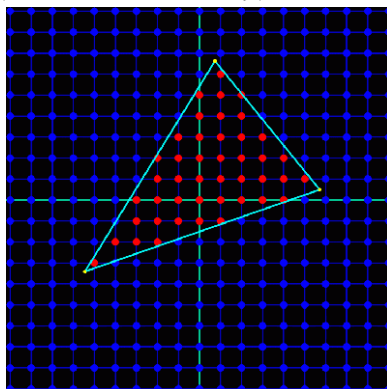


图 2 MSAA 覆盖

3.2 TAA

TAA 的核心思想是利用帧之间的时间信息，对每一帧进行多次采样并结合之前帧的结果，从而得到更加平滑的图像。通过在时间维度上进行平均，可以减少动态场景中的锯齿和闪烁。帧与帧之间有很多信息都可以复用，根据摄像机位置，照明条件和可见表面区域等相关信息，我们可以提取出两帧之间的相机、灯光、可见表面点等许多可复用的信息。TAA 在每一帧中对像素进行采样，并且采样点会在每帧之间稍微移动。这种移动通常是通过一个抖动（jittering）矩阵实现的，该矩阵在不同的帧之间应用不同的偏移量，使得每帧的采样点位置不同。为了将当前帧与之前的帧对齐，TAA 使用了运动矢量（motion vectors）。这些运动矢量表示了物体在屏幕上从一帧到下一帧的移动方向和距离。通过这些矢量，可以将之前帧的像素位置映射到当前帧的像素位置。将当前帧的像素颜色与之前帧的映射像素颜色进行融合。常见的方法是对当前帧和之前帧的颜色进行加权平均。权重通常取决于运动速度和相似性。如图 3 所示。TAA 还会保留一个历史缓存，用于存储前几帧的颜色和深度信息，以便在未来的帧中使用。

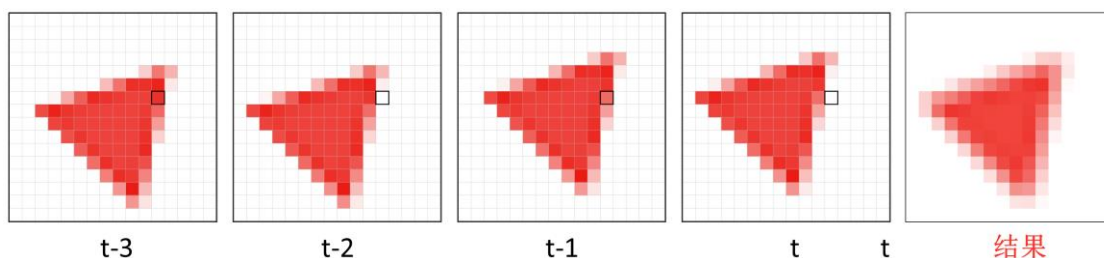


图 3 TAA

4 实验设置

4.1 实验环境

1) 系统配置

		版本
System Info	操作系统	Windows 10 家庭中文版 64-bit
	渲染器	Intel(R) UHD Graphics Family
	处理器	intel(R) Core(TM) 7-10870H CPU @2.20GHz
	屏幕显示	1920*1080*32bpp (144Hz)
OpenGL		4.3
Visual Studio		Microsoft Visual Studio Community 2019 版本 16.11.26

图 4 系统配置

2) 其他配置

运用 DirectX 11 框架，主要包括 C 标准库；C++标准库；DirectX 数学库，用于向量和矩阵运算；DirectXTex 库用于纹理处理；gdiplus 库，用于图形编程的 API；Assimp 库，用于导入和导出各种 3D 模型文件格式。HosekSky 库：用于生成物理上逼真的天空光照和颜色，适用于需要自然光照和天空渲染的应用程序。

4.2 实验设计

结合运用两个算法，分别对两个算法的不同参数配置进行实验，获取实验结果.算法参数设计如表 1。

算法	MSAA	TAA
参数	None	None
参数	8	None
参数	4	None
参数	2	None

参数	None	0.9
参数	None	0.7
参数	None	0.5
参数	8	0.9

表 1 实验设计表

5 实验结果与分析

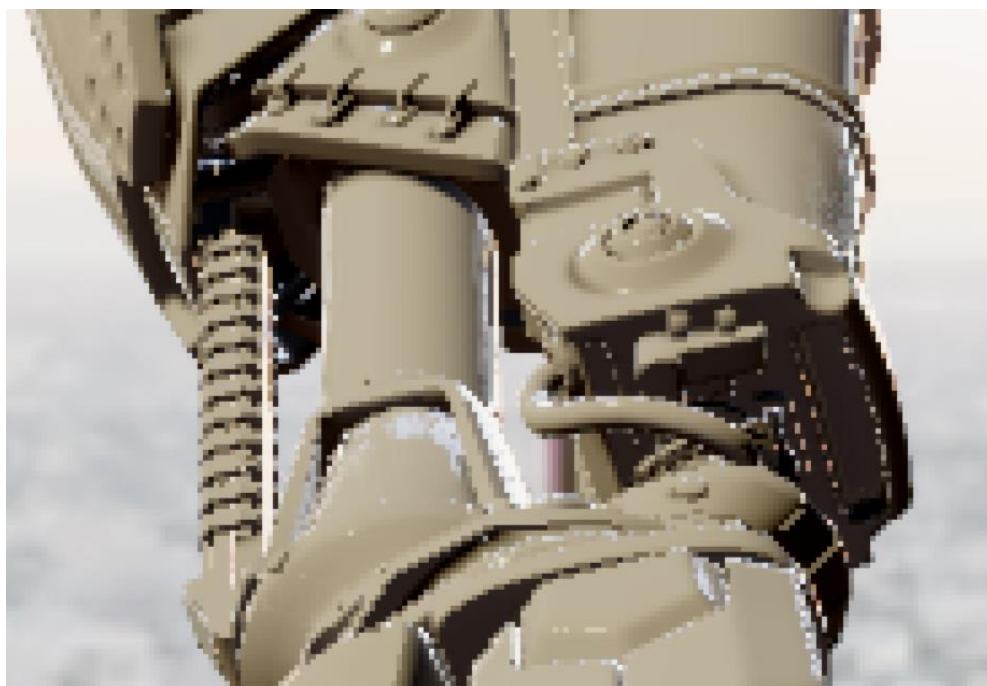
5.1 实验结果

1) 没有开启 MSAA 和 TAA 的原始图像，如图 5 所示。

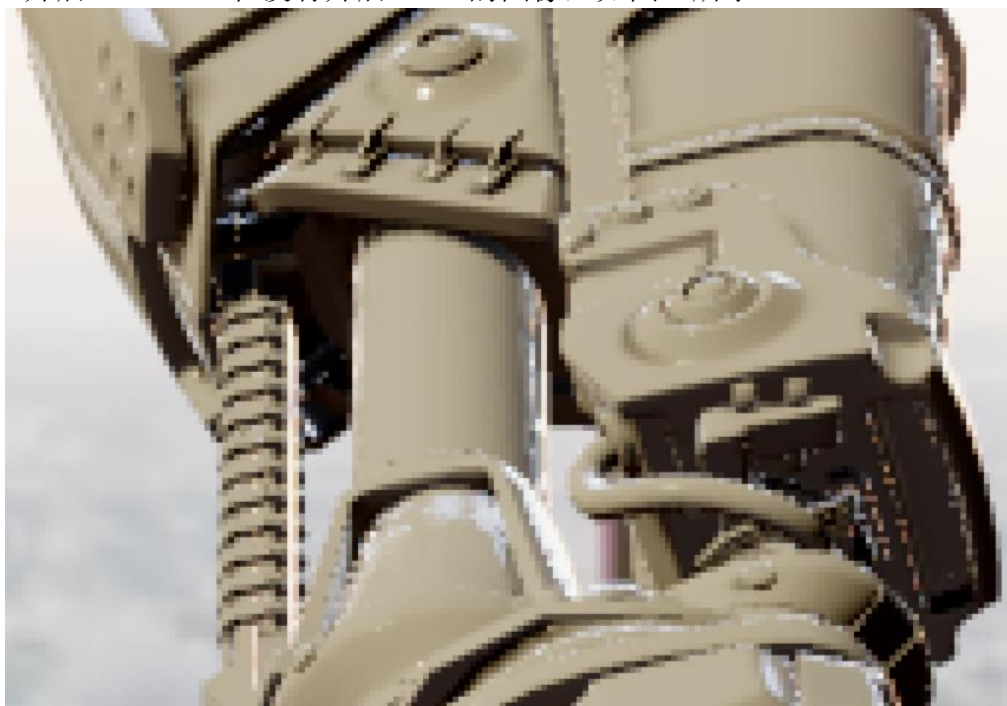


图 5 没有开启 MSAA 和 TAA 的原始图像

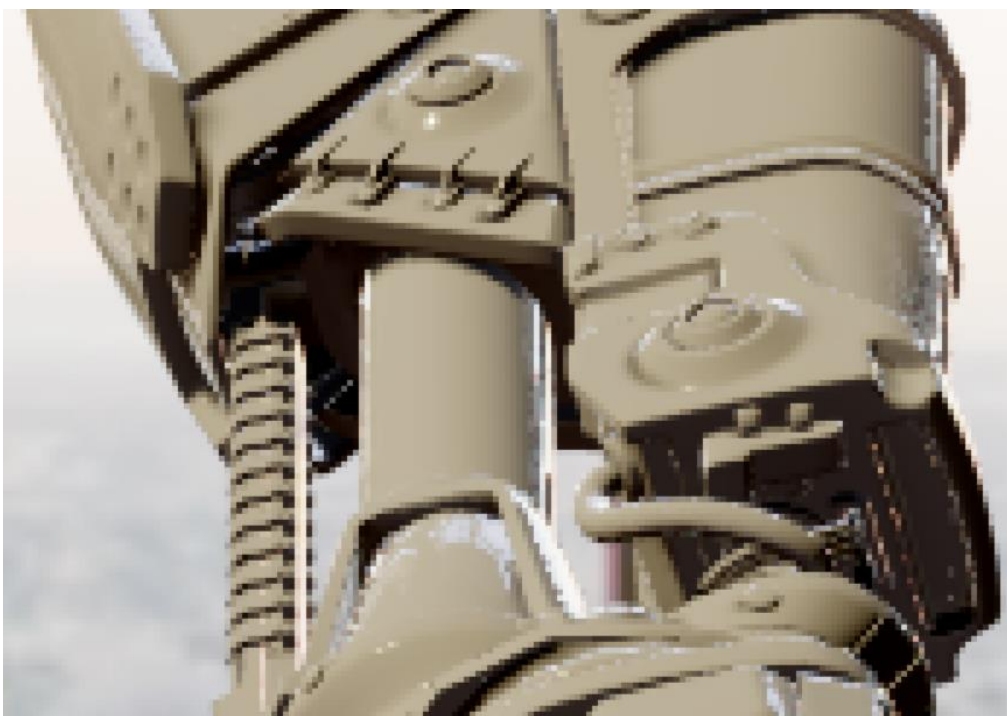
2) 开启 2xMSAA 和没有开启 TAA 的图像，如图 6 所示。



-
- 图 6 开启 2xMSAA 和没有开启 TAA 的图像
- 3) 开启 4xMSAA 和没有开启 TAA 的图像，如图 7 所示。



- 图 7 开启 4xMSAA 和没有开启 TAA 的图像
- 4) 开启 8xMSAA 和没有开启 TAA 的图像，如图 8 所示。



- 图 8 开启 8xMSAA 和没有开启 TAA 的图像
- 5) 没有开启 MSAA 和开启 0.5xTAA 的图像，如图 9 所示。

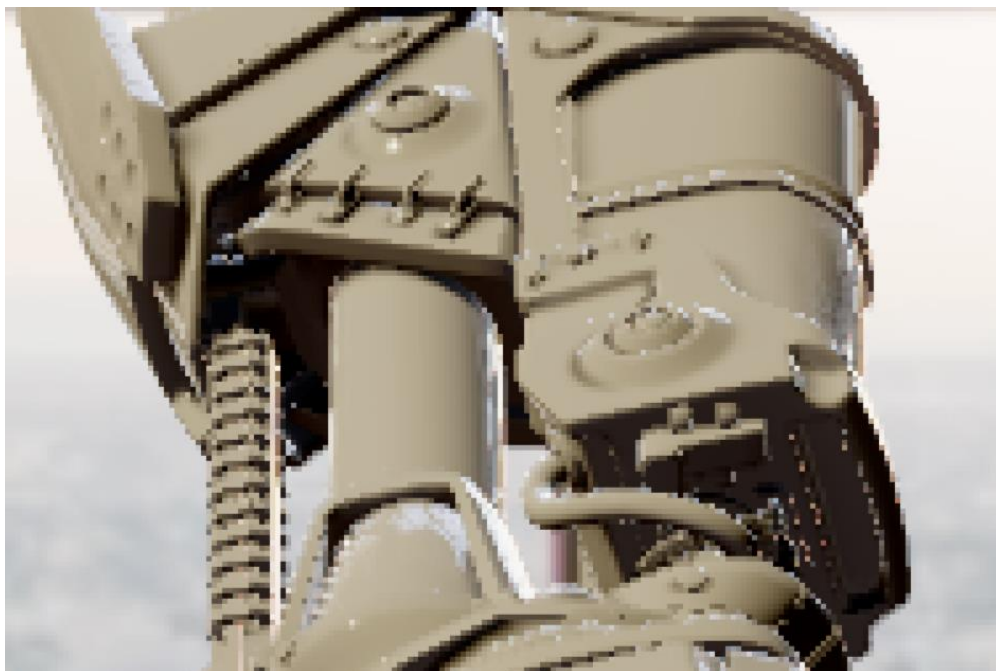


图 9 没有开启 MSAA 和开启 0.5xTAA 的图像

6) 没有开启 MSAA 和开启 0.7xTAA 的图像, 如图 10 所示。



图 10 没有开启 MSAA 和开启 0.7xTAA 的图像

7) 没有开启 MSAA 和开启 0.9xTAA 的图像, 如图 11 所示。

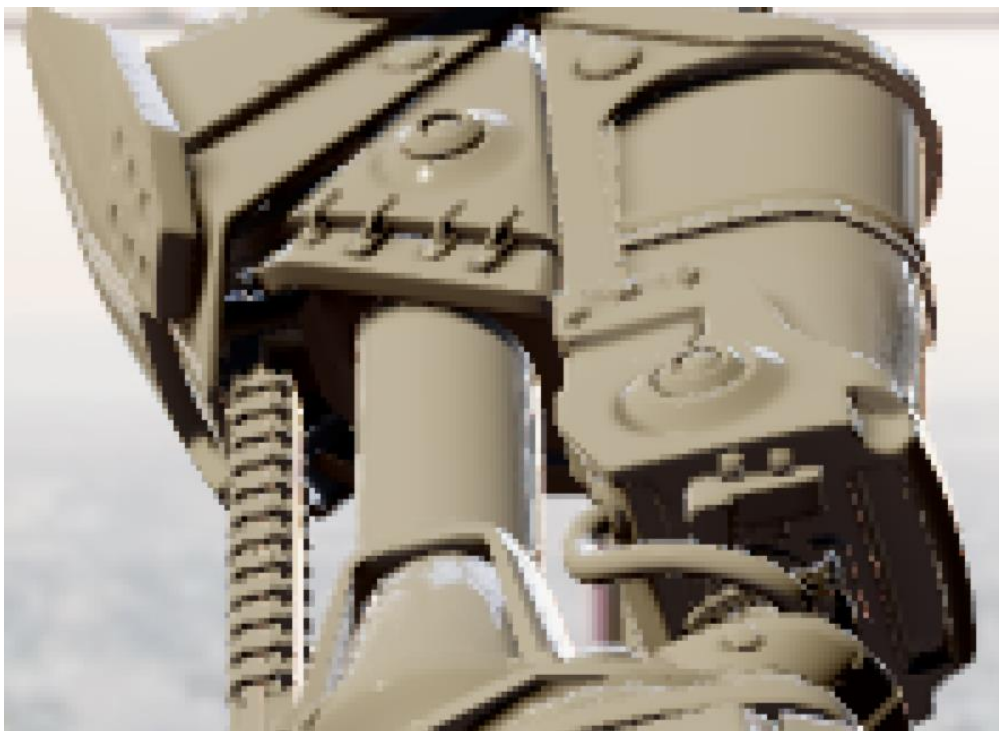


图 11 没有开启 MSAA 和开启 0.9xTAA 的图像

8) 开启 8xMSAA 和开启 0.9xTAA 的图像，如图 12 所示。



图 12 开启 8xMSAA 和开启 0.9xTAA 的图像

5.2 实验结果分析

MSAA 在抗锯齿方面有很好的效果，减少了锯齿和失真，都增加了细节的准确性和图像的真实感。但是不同参数配置的 MSAA 有着不同的效果。2x MSAA:

相比没有抗锯齿的图像，2x MSAA 能显著减少锯齿现象，但效果相对较弱。适合对图像质量要求不高或硬件性能有限的情况。而且性能消耗相对较低，仅需处理两倍的采样点数，因此对帧率的影响较小；4x MSAA：提供了更好的抗锯齿效果，相比 2x MSAA，图像边缘更加平滑，锯齿现象进一步减少。适合对图像质量有中等要求的场景。性能消耗比 2x MSAA 高，需要处理四倍的采样点数，对硬件性能要求较高；8x MSAA：提供了最好的抗锯齿效果，图像边缘非常平滑，几乎没有锯齿现象。适合对图像质量要求很高的情况。性能消耗最高，需要处理八倍的采样点数，对显卡性能要求极高。对比如图 13 所示，从左往右分别是 8xMSAA、4xMSAA、2xMSAA、原始图像。



图 13 MSAA 对比图

TAA 在抗锯齿方面有很好的效果，减少了锯齿和失真，都增加了细节的准确性和图像的真实感。但是不同缩放因子配置的 TAA 有着不同的效果。0.5x TAA：将渲染分辨率降低到原始分辨率的一半，然后再通过 TAA 处理来减少锯齿。尽管 TAA 能有效地减少锯齿，但由于初始分辨率较低，最终图像会显得模糊，细节损失较大，特别是在高对比度的边缘。性能消耗最低，因为渲染的像素数量减少到原始的一半，非常适合低端硬件或需要高帧率的情况；0.7x TAA：将渲染分辨率降低到原始的 70%。相比 0.5x TAA，图像质量有明显提升，细节保留更多，但仍然会比原始分辨率有所模糊。TAA 处理可以有效减少锯齿，但细节锐利度稍有欠缺。性能消耗中等，适合在需要一定图像质量和性能平衡的情况下使用；0.9x TAA：将渲染分辨率降低到原始的 90%。图像质量接近原始分辨率，细节基本得以保留，同时 TAA 能有效减少锯齿，使得图像边缘非常平滑。画面较为清晰，细节损失最小。性能消耗最高，但仍比原始分辨率的全渲染要低一些，适合在对图像质量要求较高的场景中使用。对比如图 14 所示，从左往右分别是 0.9x TAA、0.7x TAA、0.5x TAA、原始图像。

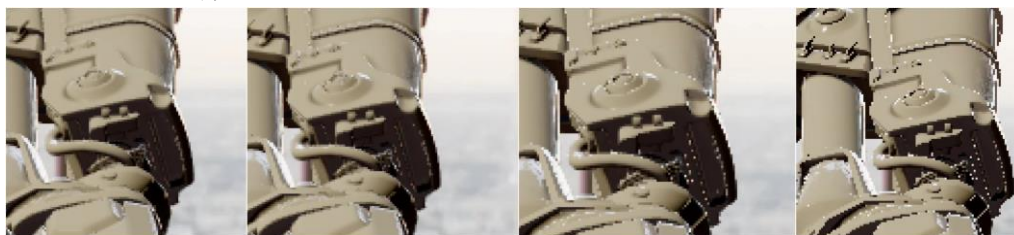


图 14 TAA 对比图

6 结论

抗锯齿技术在图形渲染中起到了重要的作用，可以显著提升图像的质量。这里主要讨论了 MSAA 和 TAA 两种常见的抗锯齿技术，它们各有优缺点，适用于不同的硬件配置和场景需求。MSAA 适合对图像静态质量要求较高的场景，在图像边缘附近进行多重采样，从而在边缘处生成更平滑的过渡效果，减少锯齿的出现。TAA 则更适合动态场景，利用时间维度上的信息，通过在连续帧之间进行像素位置的模糊化处理，以减少锯齿和运动模糊，从而改善图像质量，使图像看起来更平滑和真实。两者均能有效地减少锯齿现象，提升图像的真实感和细节表现。但是 MSAA 通常在增加采样点数时会增加硬件的负荷，会对显卡性能有一定的要求。相对于 MSAA，TAA 的性能消耗通常较低，因为它不需要增加过多的采样点数，而是依赖于可以并行处理的后处理的模糊化技术。选择合适的抗锯齿技术和参数配置需要考虑具体的使用场景和硬件性能，从而在图像质量和性能之间找到最佳平衡。

通过本次作业，我探索了抗锯齿的 MSAA 和 TAA 算法，学习了 MSAA 和 TAA 的基本实现，初步实现了这两个算法，并进行了不同参数的设置，进行实验结果的对比。总体而言，本次实验不仅强化了我在抗锯齿方面的实践技能，还加深了我对这门技术的理论理，为今后的学习和科研打下了坚实的基础。这次宝贵的经历，将激励我们在计算机图形学领域继续深入探索和创新。

参考文献

- [1] Beets, Kristof, and Dave Barron. "Super-sampling anti-aliasing analyzed." (2000).
- [2] Fridvalszky A, Tóth B. Multisample Anti-aliasing in Deferred Rendering[C]//Eurographics (Short Papers). 2020: 21-24.