|  |  |
| --- | --- |
|  | **Российский государственный социальный университет** |

**Лабораторная работа № 1.**

**по дисциплине «Интеллектуальные информационные системы»**

|  |  |
| --- | --- |
| **ФИО студента** | Салов Артём Владимирович |
| **Направление подготовки** | Программная инженерия |
| **Группа** | ПИН-Б-О-Д-2021-1 |

# **Москва 2023**

# DLSS

DLSS — это технология NVIDIA, которая призвана повысить производительность игр без потери качества картинки. Впервые мы увидели, как это работает в Battlefield V и Metro: Exodus . Говоря простым языком, DLSS вначале снижает разрешение изображения, детализирует его с помощью нейросети и подтягивает по качеству, а затем выдает красивую картинку. Все равно сложно? Тогда давайте по порядку.

Если переводить аббревиатуру DLSS (Deep Learning Super Sampling) на русский язык, то будет выглядеть это примерно так “сглаживание с использованием глубокого обучения”. Вряд ли это вносит какую-то решающую ясность. Становится понятно лишь то, что это технология сглаживания, и то, что используется обучение. Обучение обычно свойственно нейросетям. Именно это и относится к технологии DLSS. Ее появление напрямую связано с технологией трассировки лучей (Ray Tracing). Это новшество, как ожидалось, слишком сильно скажется на производительности, поэтому специалисты из NVIDIA должны были придумать что-то для оптимизации. И придумали. DLSS — технология сглаживания, которая без потери качества картинки улучшает быстродействие. Звучит не реалистично, но факт остается фактом.

Первая версия — DLSS 1.0 подразумевала следующее: создателям нужно было “загнать” весь графический контент в специальный сверхмощный компьютер NVIDIA, который, с помощью искусственного интеллекта, обрабатывал изображение. Получается, что под каждую отдельную игру писался свой алгоритм DLSS. Это было очень затратно. Если одна нейросеть работала бы на все игры, то трудно представить объем данных, который бы она хранила. Да и картинку DLSS 1.0 выдавала "мыльную". Первое время ходили даже мемы про это. Что-то наподобие такого:



# Описание технологий

Cглаживание по типу «суперсэмплинг». Если упрощенно представить себе принцип действия, он выглядит так: картинка отрисовывается в более высоком, чем нужно, разрешении, а затем ужимается до нужного. В итоге цвет каждого пикселя реального разрешения вычисляется на основе нескольких субпикселей виртуального, и качество картинки заметно улучшается.

«Суперсэмплинг» работает в паре с технологиями глубокого обучения. Алгоритмы используют три типа данных:

* изображения текущего кадра в низком разрешении, сгенерированные игровым движком;
* предыдущий кадр в высоком разрешении;
* векторы движения для тех же изображений, также сгенерированные движком.

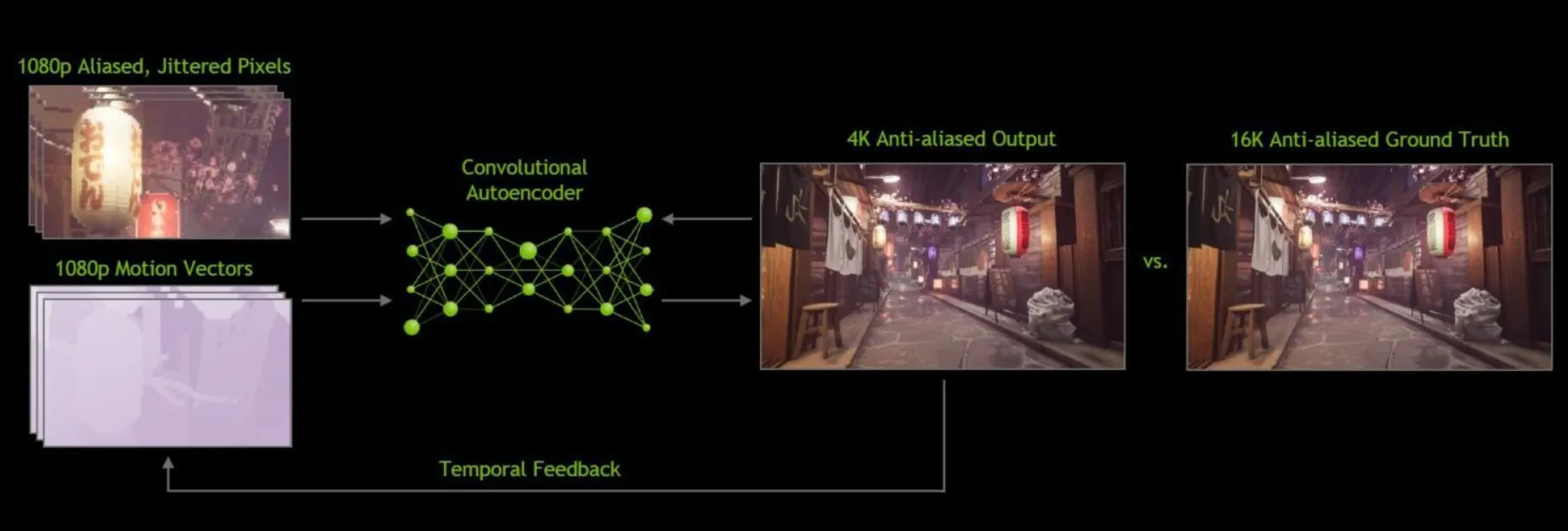
Используя векторы движения, ИИ понимает, как трансформируется предыдущий кадр в высоком разрешении. И создает текущий тоже в высоком разрешении.

Теперь вкратце алгоритм работы DLSS:

1. *Технология изучает исходное изображение в высоком разрешении (разработчик отправляет весь графический контент в NVIDIA — под всеми ракурсами и углами в разрешении 16К).*
2. *Видеокарта выдает ту же самую картинку, но уже в более низком разрешении.*
3. *«Тензорные ядра» дорабатывают это изображение, чтобы оно внешне ничем не отличалось от аналога в высоком разрешении (на основе изучения исходного материала в 16К).*
4. *На монитор поступает оптимизированная картинка.*

Получается, что мы видим красивую графику, в основе которой лежит понижение разрешения, но без ущерба визуальной составляющей.

Апскейлинг – это далеко не идеальный метод, который вызывает многочисленные искажения отображаемого изображения, а конечный эффект очень чётко отличается от того, который даёт исходное разрешение. Здесь на помощь приходит NVIDIA с технологией DLSS.

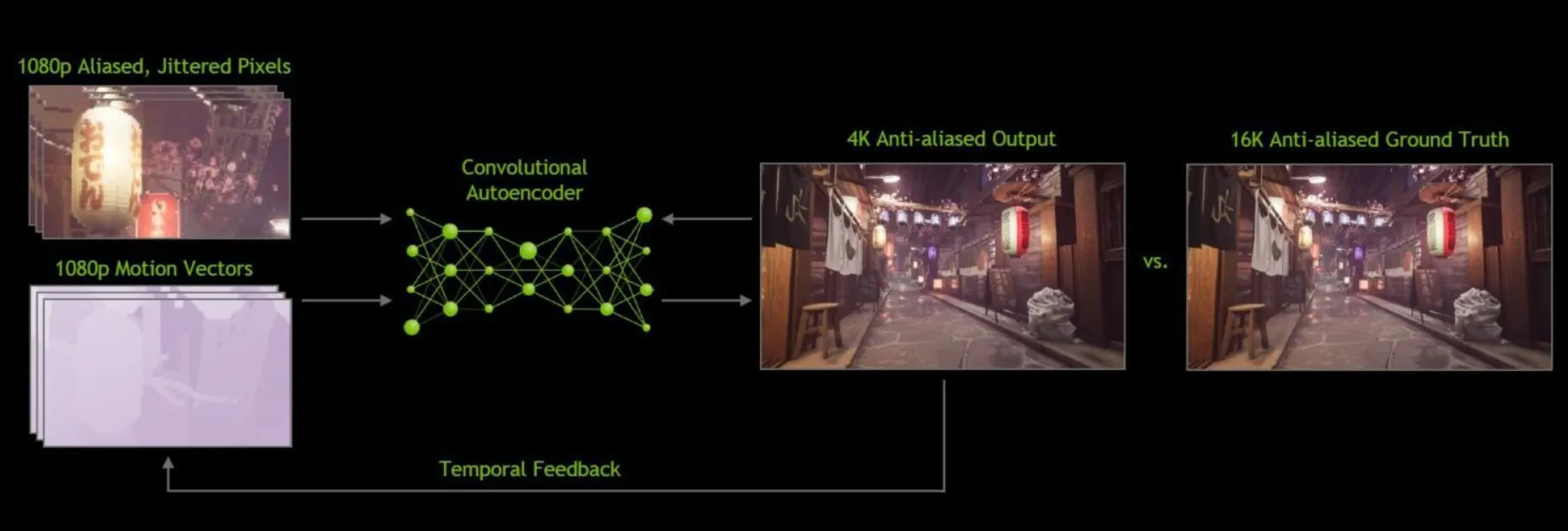


NVIDIA DLSS использует изображение с более низким разрешением, а затем масштабирует его до исходного разрешения, что похоже на обычный метод масштабирования. Однако, она делает гораздо больше. Самая большая разница заключается в том, что метод DLSS использует нейронную сеть, извлекая часть преимуществ глубокого обучения.

**Интеллектуальное масштабирование DLSS – что это значит**

NVIDIA использует специальную платформу под названием NGX (Neural Graphics Framework) – она занимается процессом обучения глубокой нейронной сети (DNN), которая основана на анализе тысяч «идеальных» изображений в очень высоком разрешении (16K).

Кроме того, NVIDIA показывает нейросети некачественные изображения без сглаживания. Затем они сравниваются в процессе обучения. Результат этого сравнения возвращается к сети SI, которая делает выводы и учится достигать целевой точки (шаблона), и конечный эффект производит лучшее впечатление (он приближается по качеству к изображению с более высокой детализацией).



На практике это означает, что, благодаря DLSS, игра работает с более высоким числом кадров в секунду, сохраняя – по крайней мере, теоретически – уровень графики, обеспечиваемый собственным разрешением. DLSS восстанавливает изображение, основываясь не только на информации, полученной из предыдущих кадров, но также на векторах движения, которые определяют направление, в котором объекты, представленные в сцене, перемещаются от кадра к кадру.

DLSS использует, так называемые, свёрточный автоэнкодер. Это особый тип нейронной сети, которая пиксель за пикселем определяет, какие края и формы нуждаются в коррекции, а какие нет. Во время этого процесса декодер сглаживает и обостряет те детали, которые были отмечены как важные. Это делается путём сравнения кадров низкого и высокого разрешения (текущего и последнего кадра, соответственно).

Примеры работы технологии:







# 3. Примеры архитектур нейронных сетей

Нейронные сети прямого распространения (feed forward neural networks, FF или FFNN) и перцептроны (perceptrons, P) очень прямолинейны, они передают информацию от входа к выходу. Нейронные сети часто описываются в виде слоёного торта, где каждый слой состоит из входных, скрытых или выходных клеток. Клетки одного слоя не связаны между собой, а соседние слои обычно полностью связаны. Самая простая нейронная сеть имеет две входных клетки и одну выходную, и может использоваться в качестве модели логических вентилей. FFNN обычно обучается по методу обратного распространения ошибки, в котором сеть получает множества входных и выходных данных. Этот процесс называется обучением с учителем, и он отличается от обучения без учителя тем, что во втором случае множество выходных данных сеть составляет самостоятельно. Вышеупомянутая ошибка является разницей между вводом и выводом. Если у сети есть достаточное количество скрытых нейронов, она теоретически способна смоделировать взаимодействие между входным и выходными данными. Практически такие сети используются редко, но их часто комбинируют с другими типами для получения новых.

Сети радиально-базисных функций (radial basis function, RBF) — это FFNN, которая использует радиальные базисные функции как функции активации. Больше она ничем не выделяется ?

Нейронная сеть Хопфилда (Hopfield network, HN) — это полносвязная нейронная сеть с симметричной матрицей связей. Во время получения входных данных каждый узел является входом, в процессе обучения он становится скрытым, а затем становится выходом. Сеть обучается так: значения нейронов устанавливаются в соответствии с желаемым шаблоном, после чего вычисляются веса, которые в дальнейшем не меняются. После того, как сеть обучилась на одном или нескольких шаблонах, она всегда будет сводиться к одному из них (но не всегда — к желаемому). Она стабилизируется в зависимости от общей “энергии” и “температуры” сети. У каждого нейрона есть свой порог активации, зависящий от температуры, при прохождении которого нейрон принимает одно из двух значений (обычно -1 или 1, иногда 0 или 1). Такая сеть часто называется сетью с ассоциативной памятью; как человек, видя половину таблицы, может представить вторую половину таблицы, так и эта сеть, получая таблицу, наполовину зашумленную, восстанавливает её до полной.

Цепи Маркова (Markov chains, MC или discrete time Markov Chains, DTMC) — это предшественники машин Больцмана (BM) и сетей Хопфилда (HN). Их смысл можно объяснить так: каковы мои шансы попасть в один из следующих узлов, если я нахожусь в данном? Каждое следующее состояние зависит только от предыдущего. Хотя на самом деле цепи Маркова не являются НС, они весьма похожи. Также цепи Маркова не обязательно полносвязны.

Машина Больцмана (Boltzmann machine, BM) очень похожа на сеть Хопфилда, но в ней некоторые нейроны помечены как входные, а некоторые — как скрытые. Входные нейроны в дальнейшем становятся выходными. Машина Больцмана — это стохастическая сеть. Обучение проходит по методу обратного распространения ошибки или по алгоритму сравнительной расходимости. В целом процесс обучения очень похож на таковой у сети Хопфилда.

Ограниченная машина Больцмана (restricted Boltzmann machine, RBM) удивительно похожа на машину Больцмана и, следовательно, на сеть Хопфилда. Единственной разницей является её ограниченность. В ней нейроны одного типа не связаны между собой. Ограниченную машину Больцмана можно обучать как FFNN, но с одним нюансом: вместо прямой передачи данных и обратного распространения ошибки нужно передавать данные сперва в прямом направлении, затем в обратном. После этого проходит обучение по методу прямого и обратного распространения ошибки.

Автокодировщик (autoencoder, AE) чем-то похож на FFNN, так как это скорее другой способ использования FFNN, нежели фундаментально другая архитектура. Основной идеей является автоматическое кодирование (в смысле сжатия, не шифрования) информации. Сама сеть по форме напоминает песочные часы, в ней скрытые слои меньше входного и выходного, причём она симметрична. Сеть можно обучить методом обратного распространения ошибки, подавая входные данные и задавая ошибку равной разнице между входом и выходом.

Разреженный автокодировщик (sparse autoencoder, SAE) — в каком-то смысле противоположность обычного. Вместо того, чтобы обучать сеть отображать информацию в меньшем “объёме” узлов, мы увеличиваем их количество. Вместо того, чтобы сужаться к центру, сеть там раздувается. Сети такого типа полезны для работы с большим количеством мелких свойств набора данных. Если обучать сеть как обычный автокодировщик, ничего полезного не выйдет. Поэтому кроме входных данных подаётся ещё и специальный фильтр разреженности, который пропускает только определённые ошибки.

Вариационные автокодировщики (variational autoencoder, VAE) обладают схожей с AE архитектурой, но обучают их иному: приближению вероятностного распределения входных образцов. В этом они берут начало от машин Больцмана. Тем не менее, они опираются на байесовскую математику, когда речь идёт о вероятностных выводах и независимости, которые интуитивно понятны, но сложны в реализации. Если обобщить, то можно сказать что эта сеть принимает в расчёт влияния нейронов. Если что-то одно происходит в одном месте, а что-то другое – в другом, то эти события не обязательно связаны, и это должно учитываться.

Шумоподавляющие автокодировщики (denoising autoencoder, DAE) — это AE, в которые входные данные подаются в зашумленном состоянии. Ошибку мы вычисляем так же, и выходные данные сравниваются с зашумленными. Благодаря этому сеть учится обращать внимание на более широкие свойства, поскольку маленькие могут изменяться вместе с шумом.

Сеть типа “deep belief” (deep belief networks, DBN) — это название, которое получил тип архитектуры, в которой сеть состоит из нескольких соединённых RBM или VAE. Такие сети обучаются поблочно, причём каждому блоку требуется лишь уметь закодировать предыдущий. Такая техника называется “жадным обучением”, которая заключается в выборе локальных оптимальных решений, не гарантирующих оптимальный конечный результат. Также сеть можно обучить (методом обратного распространения ошибки) отображать данные в виде вероятностной модели. Если использовать обучение без учителя, стабилизированную модель можно использовать для генерации новых данных.

Свёрточные нейронные сети (convolutional neural networks, CNN) и глубинные свёрточные нейронные сети (deep convolutional neural networks, DCNN) сильно отличаются от других видов сетей. Обычно они используются для обработки изображений, реже для аудио. Типичным способом применения CNN является классификация изображений: если на изображении есть кошка, сеть выдаст “кошка”, если есть собака — “собака”. Такие сети обычно используют “сканер”, не парсящий все данные за один раз. Например, если у вас есть изображение 200×200, вы не будете сразу обрабатывать все 40 тысяч пикселей. Вместо это сеть считает квадрат размера 20 x 20 (обычно из левого верхнего угла), затем сдвинется на 1 пиксель и считает новый квадрат, и т.д. Эти входные данные затем передаются через свёрточные слои, в которых не все узлы соединены между собой. Эти слои имеют свойство сжиматься с глубиной, причём часто используются степени двойки: 32, 16, 8, 4, 2, 1. На практике к концу CNN прикрепляют FFNN для дальнейшей обработки данных. Такие сети называются глубинными (DCNN).

Развёртывающие нейронные сети (deconvolutional networks, DN), также называемые обратными графическими сетями, являются обратным к свёрточным нейронным сетям. Представьте, что вы передаёте сети слово “кошка”, а она генерирует картинки с кошками, похожие на реальные изображения котов. DNN тоже можно объединять с FFNN. Стоит заметить, что в большинстве случаев сети передаётся не строка, а какой бинарный вектор: например, <0, 1> — это кошка, <1, 0> — собака, а <1, 1> — и кошка, и собака.