# Введение в методы переменного уровня детализации полигональных сеток

*Алексей Игнатенко*

*ignatenko@graphics.cs.msu.su*

# Основные темы


- *Что такое уровень детализации*
- *Статические  и динамические методы*
- *Видозависимость*
- *Реализация*

*По материалам учебного курса при конфереренции*
*ACM SIGGRAPH 2001*

# Мотивация


***Очень важен интерактивный рендеринг геометрических данных большого объема***

- Научная и медицинская визуализация
- CAD
- Симуляторы
- Игры

# Мотивация: большие модели


## *Проблема*

- Полигональные модели часто слишком сложны для интерактивной визуализации
- Более того, чем быстрее становится «железо» тем больше становятся модели.

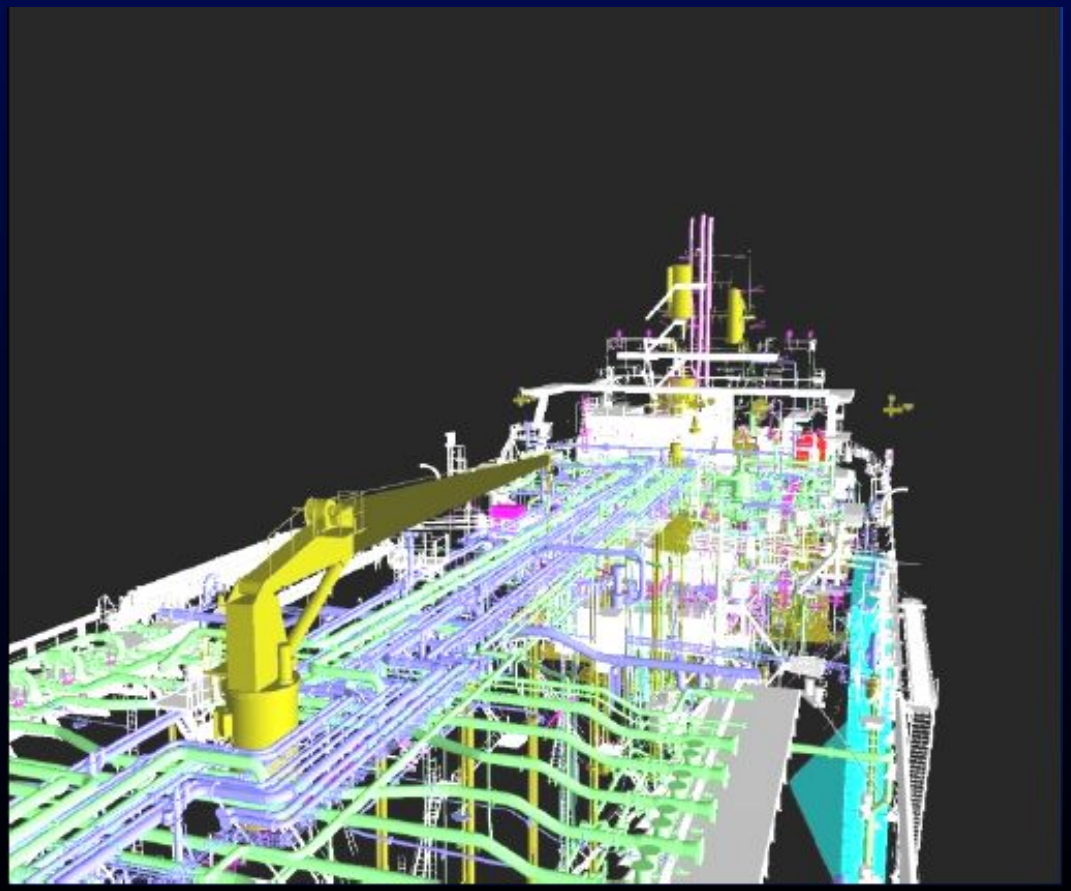
# Пример модели

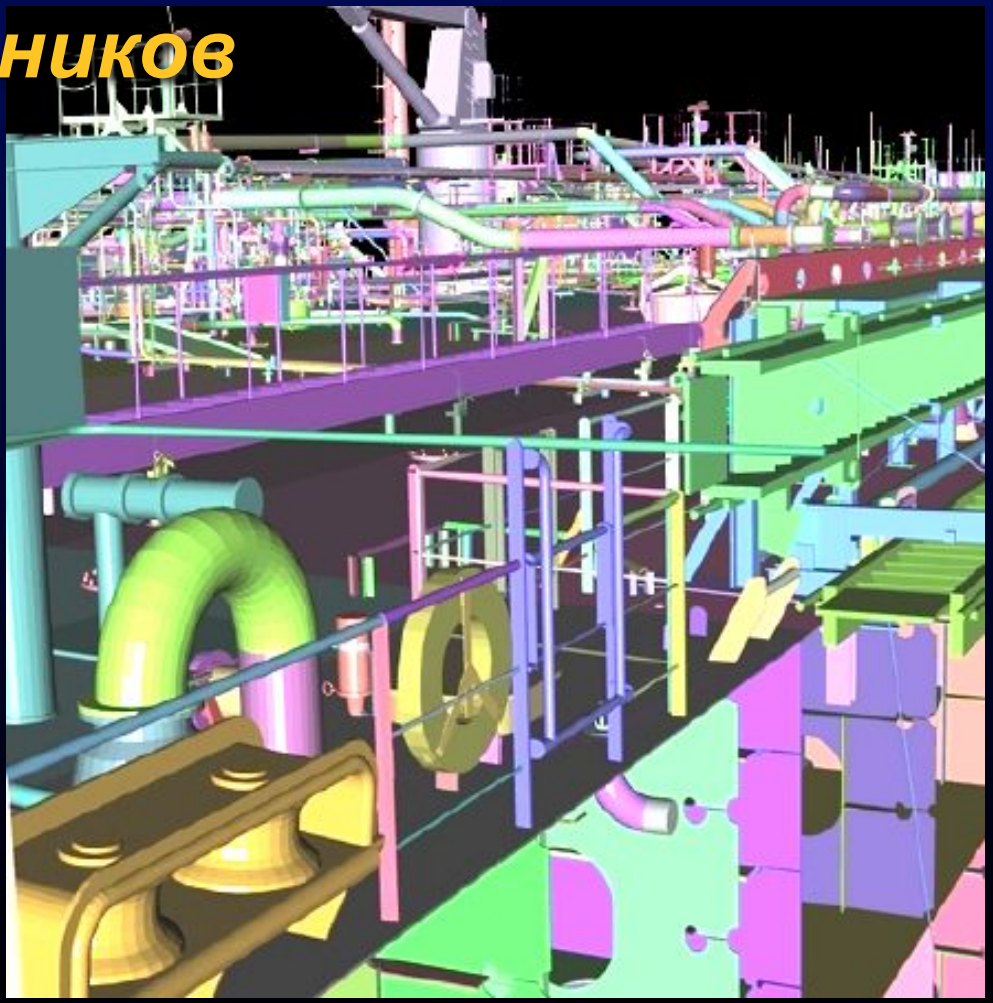
*10 миллионов треугольников*

# Пример модели

GRAPHICS & MEDIA LAB

*80 миллионов треугольников*



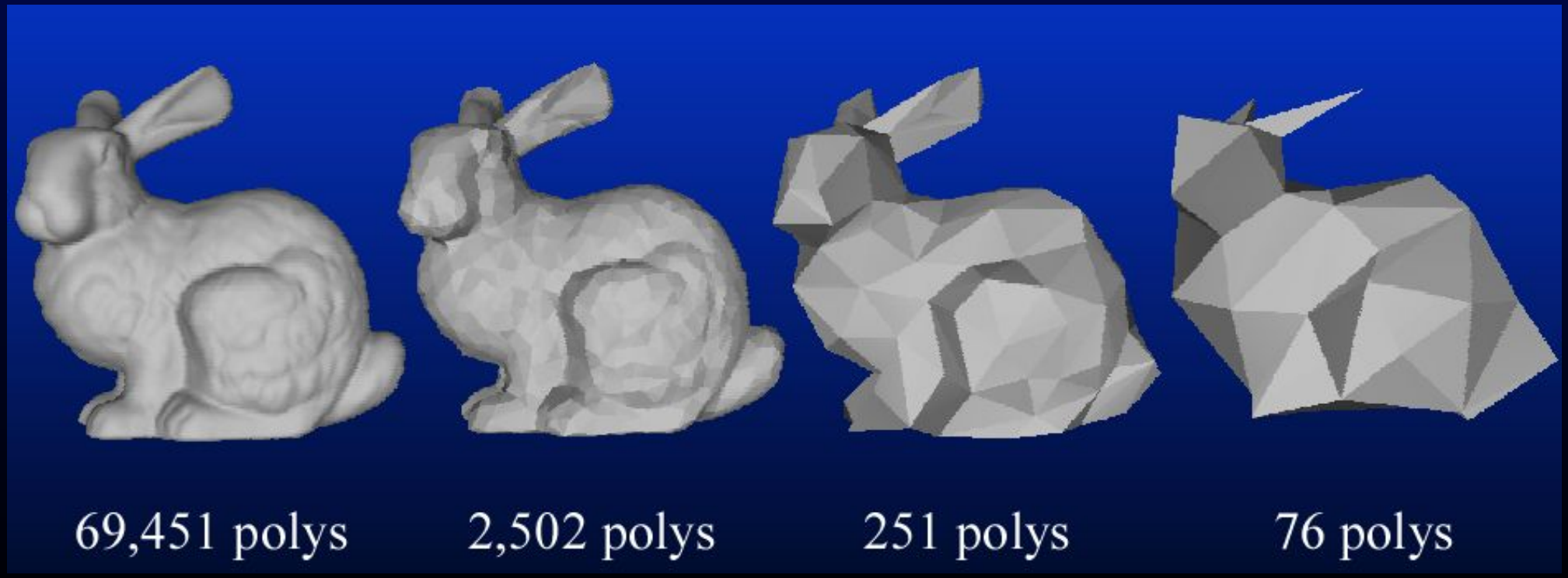


# Уровень детализации: Основная идея


- ***Возможное решение:***
  - Упрощение геометрии малых или удаленных объектов
  - Известно как Level of Detail (LOD)
    - Так же известно как polygonal simplification, geometric simplification, mesh reduction, multiresolution modeling,…

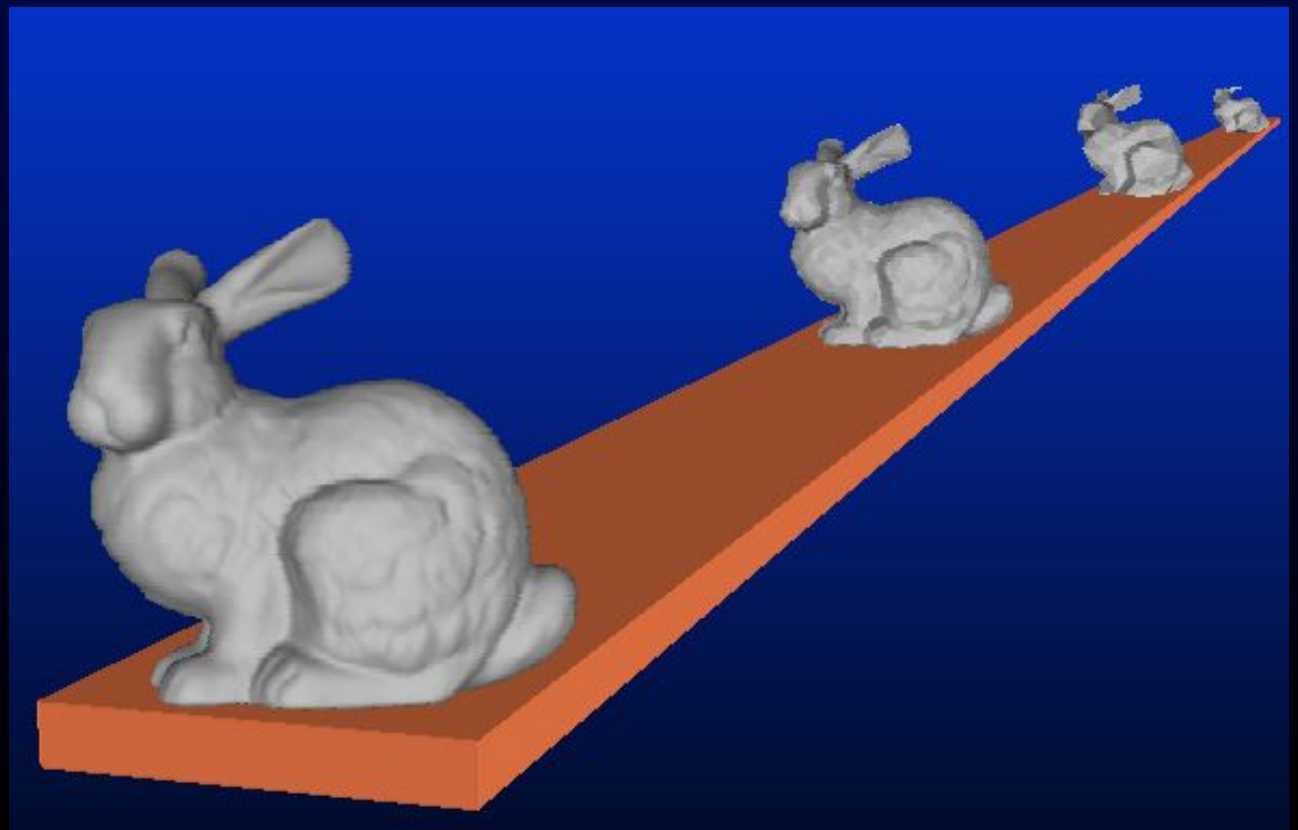
# Уровень детализации: традиционный подход


***Создание уровней детализации объектов (LODs)***



# Уровень детализации: традиционный подход


***Более удаленные объекты используют более грубые уровни детализации***



# Традиционные подход: статический LOD


***Традиционный подход:***

- Во время препроцессинга создается отдельный LOD для каждого объекта
- Во время визуализации для каждого объекта выбирается один из уровней детализации в зависимости от расстояния до объекта.

***Уровни создаются на этапе препроцессинга и имеют фиксированное разрешение. Такой подход называется статический LOD***

# Преимущества статического LOD


***Легко программируется***

***Создание и визуализация разделены***

- Создание LODов не учитывает ограничений интерактивной визуализации
- От визуализации требуется только выбрать нужный LOD

# Преимущества статического LOD


***Позволяет на полную использовать аппаратное ускорение***

- Статические LODы можно легко упаковать в дисплейные списки, вершинные массивы, страйпы и т.д.
- Это позволяет намного ускорить визуализацию, т.к. такие структуры рисуются на современной аппаратуре в 3-5 раз быстрее чем неорганизованные полигоны

# Недостатки статического LOD


***Вопрос: Почему же возникает необходимость использовать что-то кроме статического LOD?***

***Ответ: Потому что иногда статический LOD не подходит для радикального упрощения геометрии***

***Например:***

- Ландшафты
- Трехмерные изоповерхности
- Данные с трехмерных дистанционных сканеров
- Сложные CAD-модели

# Радикальное упрощение


***Для радикального упрощения геометрии:***

- сложные объекты должны быть разбиты на простые
- простые – скомбинированы

***Это очень сложно или невозможно при статическом подходе***

***Так что же можно сделать?***

# Динамический уровень детализации


***Отличие от традиционного подхода:***

- Статический LOD: создание уровней заранее перед визуализацией
- Динамический LOD: создание структуры данных, из которой геометрия с нужной детализацией может быть извлечена в реальном времени во время визуализации.

# Динамический уровень детализации: преимущества


***Данный уровень детализации определяется точно, а не выбирается из заранее полученного набора LODов.***

- Следовательно, объекты используют не больше полигонов чем необходимо
- Более эффективное использование ресурсов

# Динамический уровень детализации: преимущества


## *Более плавная визуализация*

- Переключение между статическими LODами может привести к эффекту «перепрыгивания»
- Динамические LOD позволяют плавно изменять уровень разбиения, устраняя этот эффект

# Динамический уровень детализации: преимущества

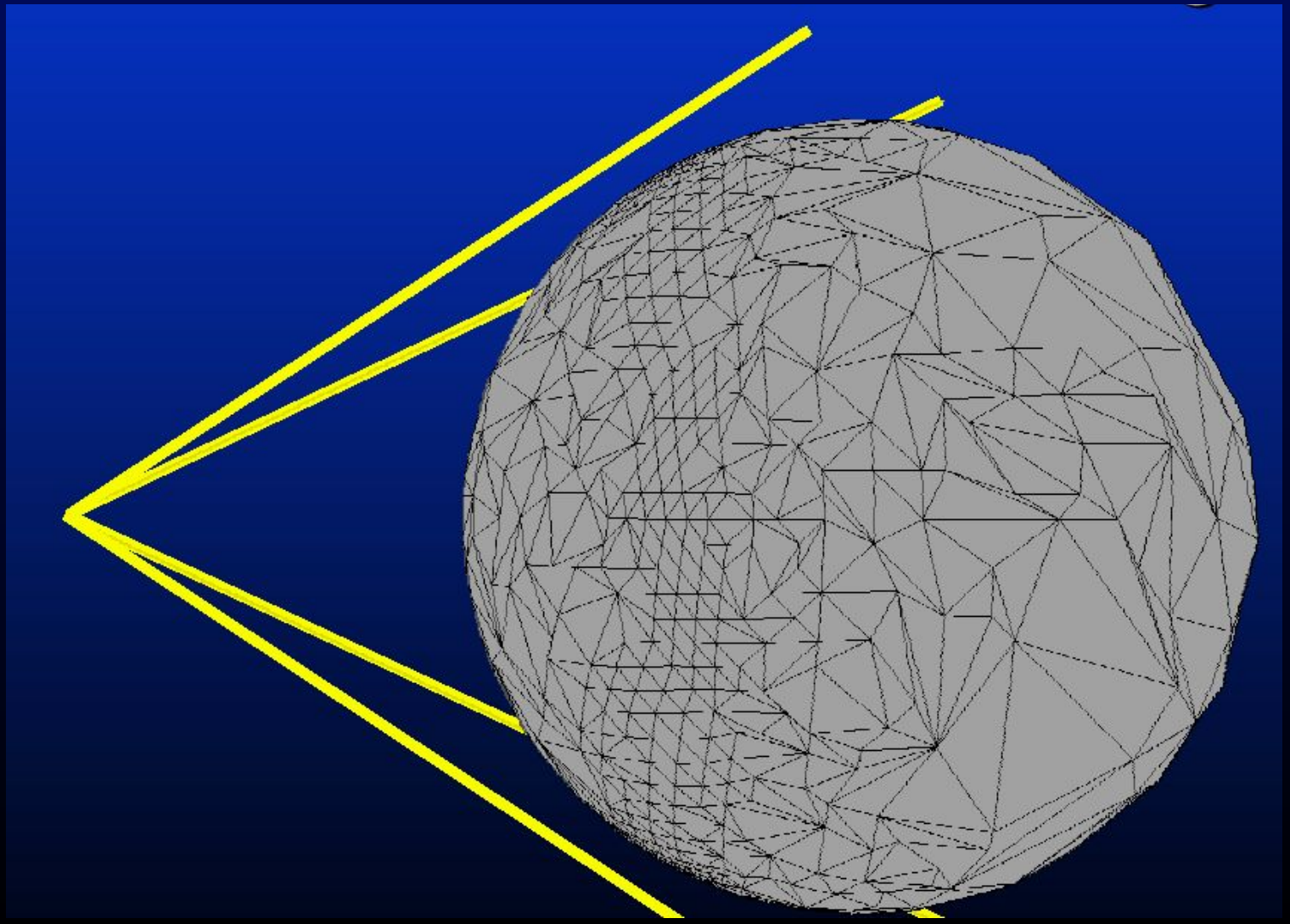

- Поддерживает прогрессивную передачу
- Поддерживает видозависимые (view-dependent) LOD
  - Можно использовать текущие параметры виртуальной камеры для определения лучшего представления для данного вида.

# Видозависимый LOD: варианты применения


***Видозависимость используется для того, чтобы***

- Показывать близкие часть объекта с большей точностью, чем удаленные
- Показывать границы объекта или наиболее интересные части с большей точностью
- Показывать больше деталей там, куда смотрит пользователь и меньше в области периферического зрения

# Динамический уровень детализации: пример



# Иерархический LOD


***Видозависимые LOD могут решить проблему с большими объектами***

***Иерархические LOD решают проблему маленьких объектов***

- Разделение объектов на составляющие
- На достаточном расстоянии упрощение составляющих, а не объектов

***Иерархический LOD подразумевает алгоритмы , модифицирующие топологию объекта***

# Иерархический LOD


***Очень хорошо взаимодействует с видозависимым LOD.***

- Можно рассматривать всю сцену как один объект и упрощать его видозависимым способом.

***Иерархический LOD также может использоваться в паре с традиционным***

# Алгоритмы видозависимых LOD


*Разработано достаточно много хороших алгоритмов видозависимых LOD*

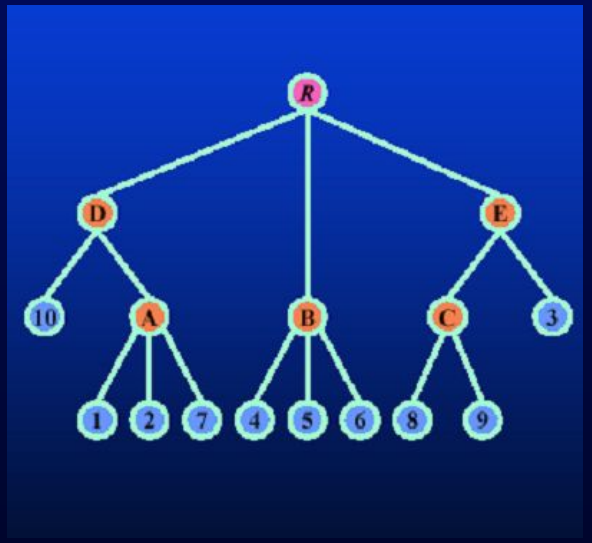
*Далее для примера будет рассмотрен алгоритм VDS (View-Dependent simplification)*

# Обзор метода VDS


## *Обзор метода VDS:*

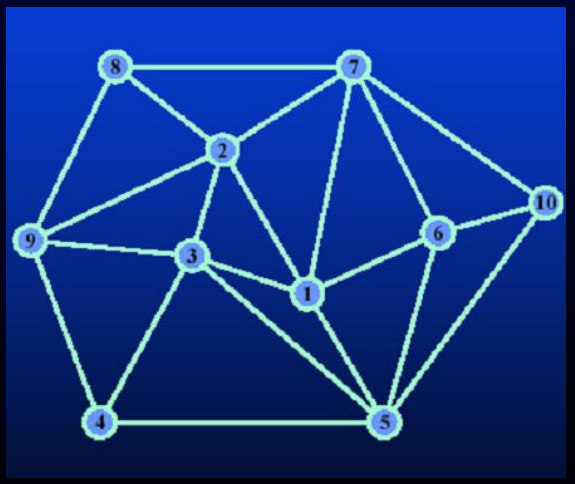
- На эпапе препроцессинга строится дерево вершин - иерархическая кластеризация вершин данной полигональной сетки
- Во время рисования кластеры разворачиваются и сворачиваются в зависимости от положения виртуальной камеры
  - Кластеры, которые становятся слишком маленькими, сворачиваются, уменьшая общее число треугольников

# Структуры данных




## *Дерево вершин*

- Содержит всю модель
- Представляет иерархию всех вершин
- Используется на каждом кадре для обновления



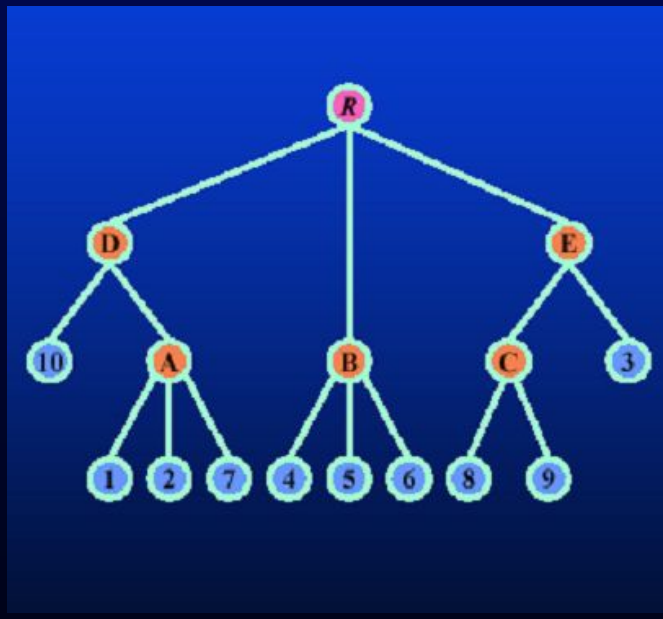

## *Список активных треугольников*

- Представляет текущее упрощение
- Содержит список треугольников для рисования
- Треугольники добавляются или удаляется в результате операций с деревом вершин.

# Дерево вершин

***С каждым узлом дерева ассоциировано подмножество вершин модели***

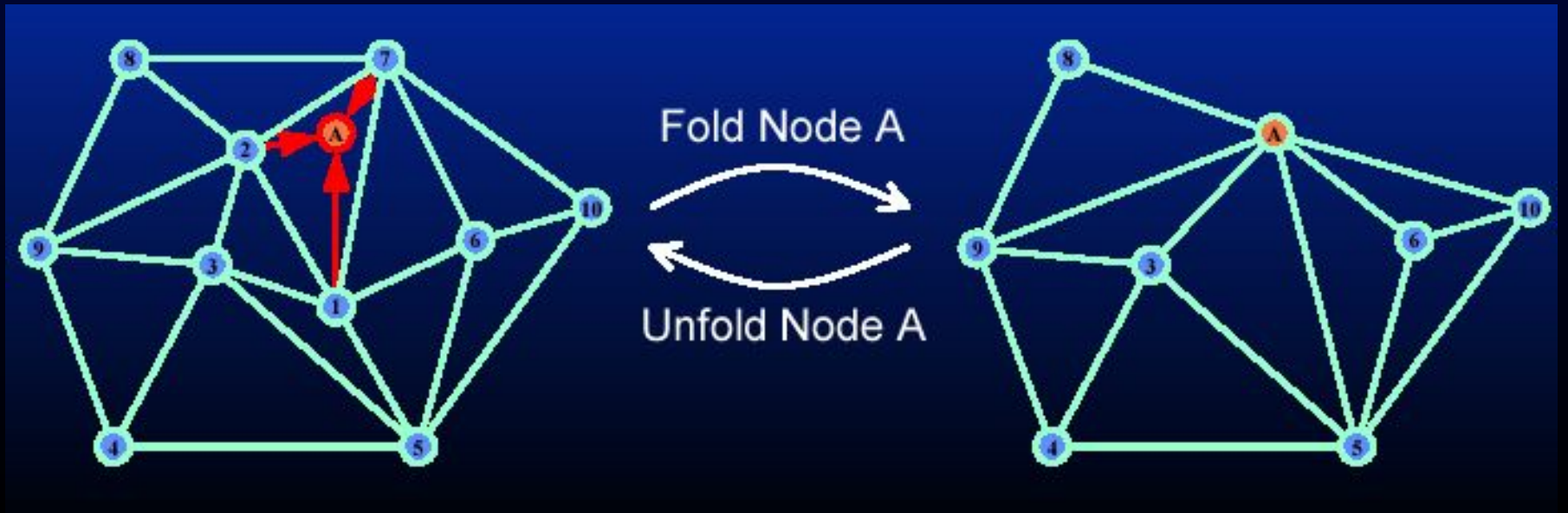
- С каждым листовые узлом ассоциирована вершина из оригинальной модели
- С каждым промежуточным узлом ассоциировано множество вершин, ассоциированных с каждым из его потомков.



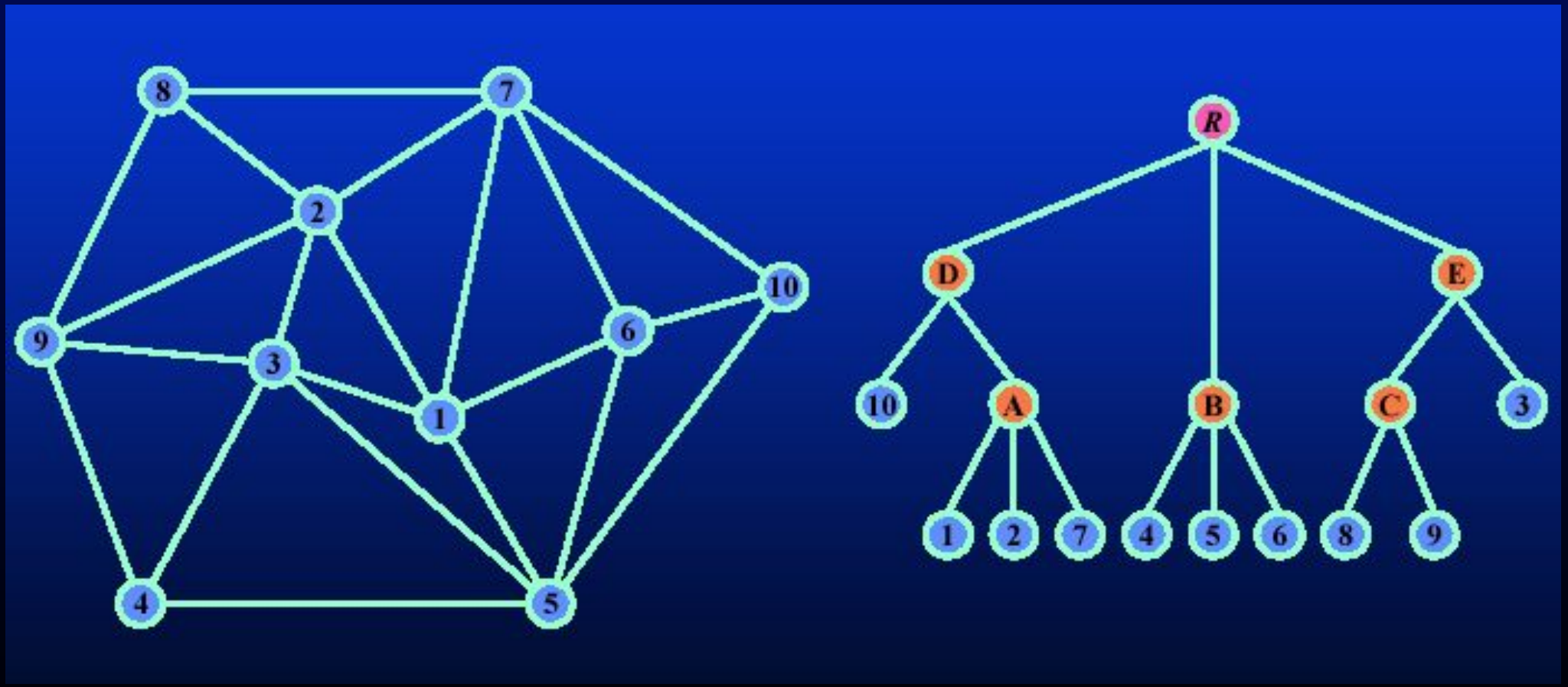

***Каждому узлу также присваивается представляющая его вершина, или заместитель (proxy)***

# Дерево вершин: сворачивание и разворачивание

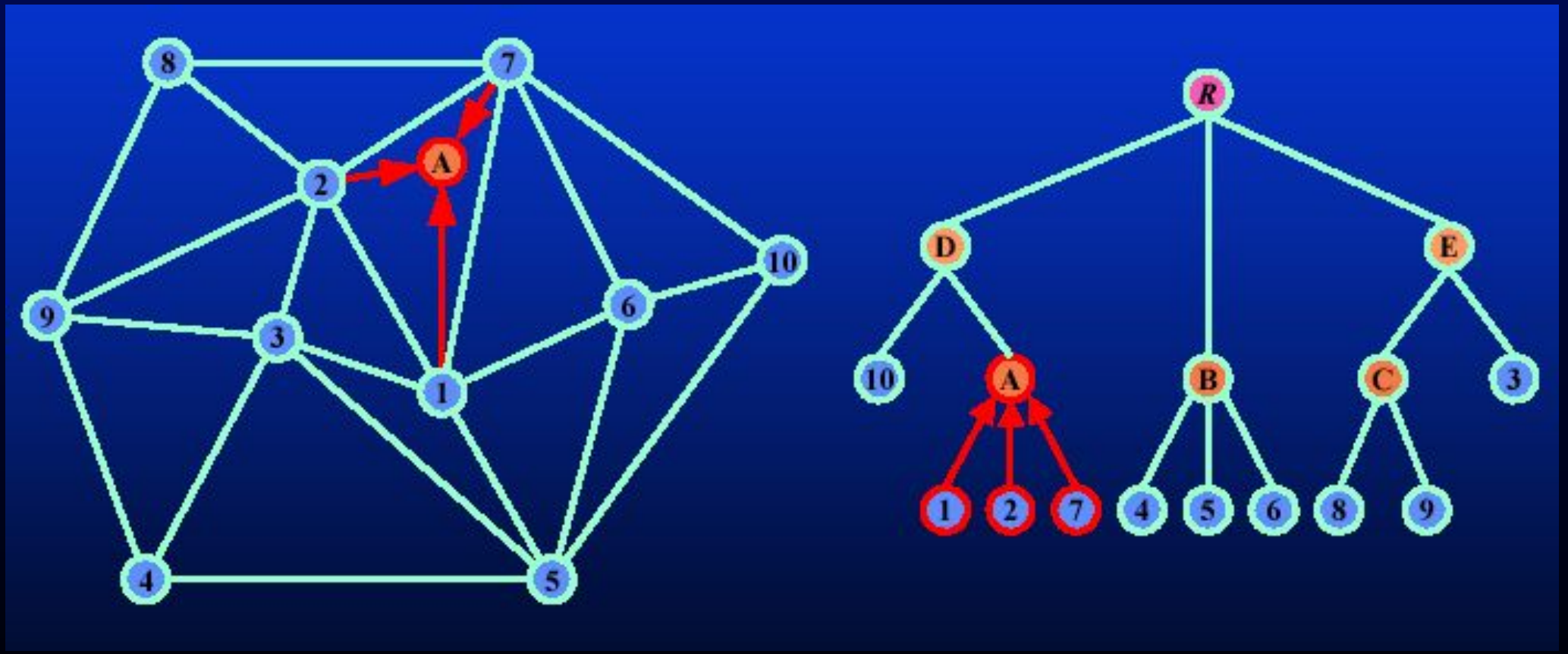

- Сворачивание узла дерева заменяет ассоциированные с ним вершины на вершину-заместитель
- Разворачивание узла заменяет заместитель на ассоциированные вершины



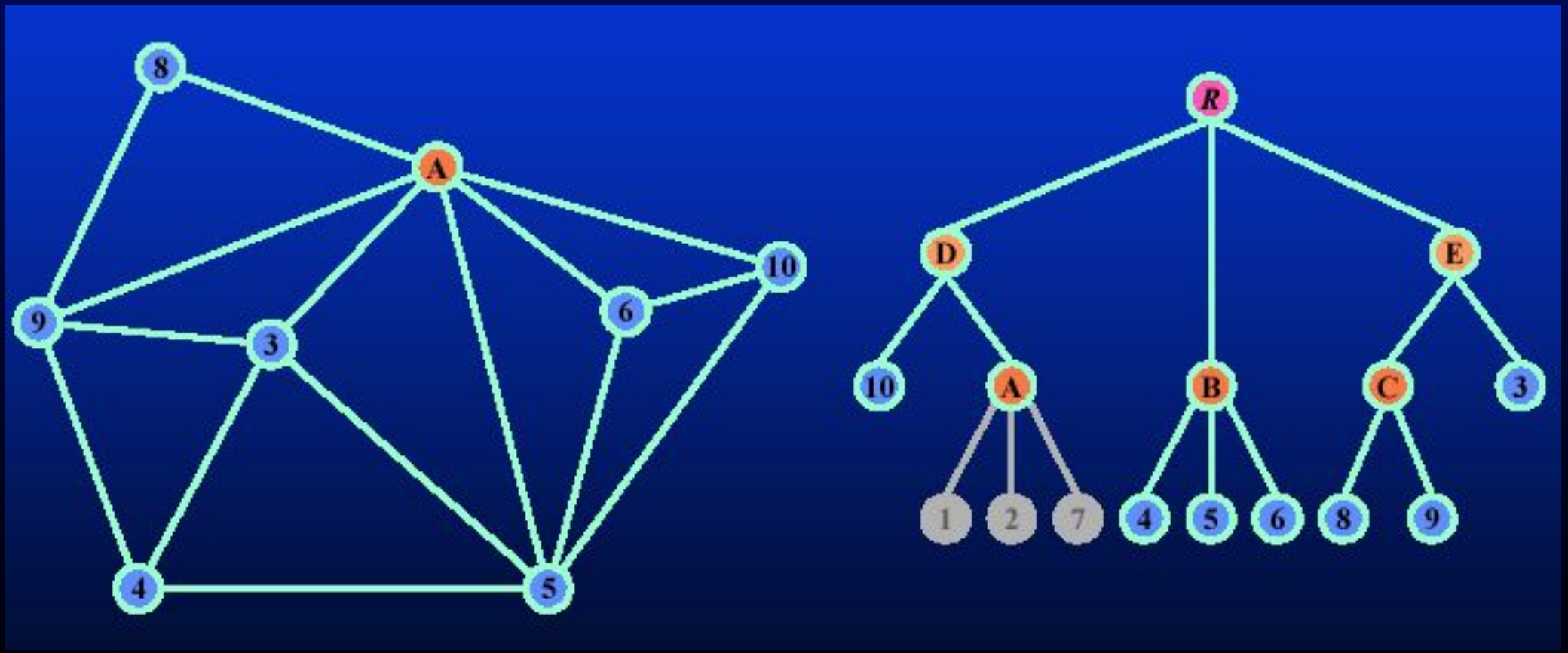
# Дерево вершин: пример



# Дерево вершин: пример



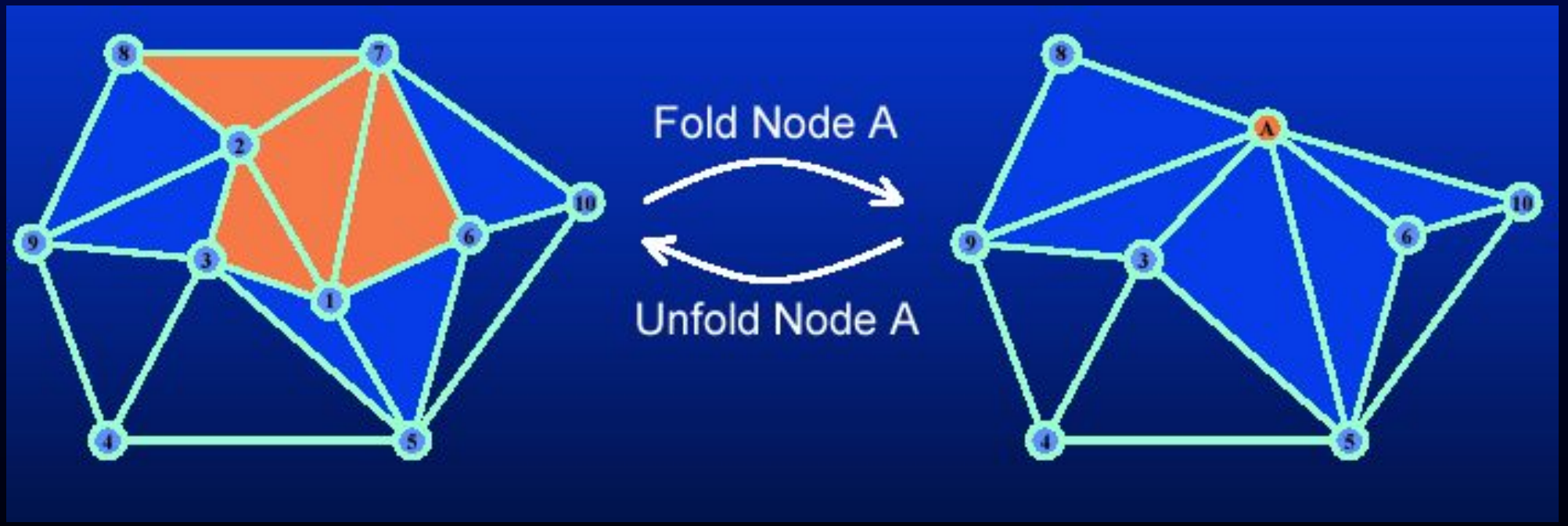
# Дерево вершин: пример



# Дерево вершин: livetris и subtris

GRAPHICS & MEDIA LAB

*2 категории треугольников*



# Дерево вершин: livetris и subtris


***Ключевое наблюдение:***

- Для каждого узла дерева subtris может быть вычислено заранее
- Для каждого узла дерева livetris можно поддерживать в реальном времени.

# Видозависимое упрощение


***Может быть использован любой интерактивный критерий сворачивания узла.***

***Примеры видозависимых критериев упрощения***

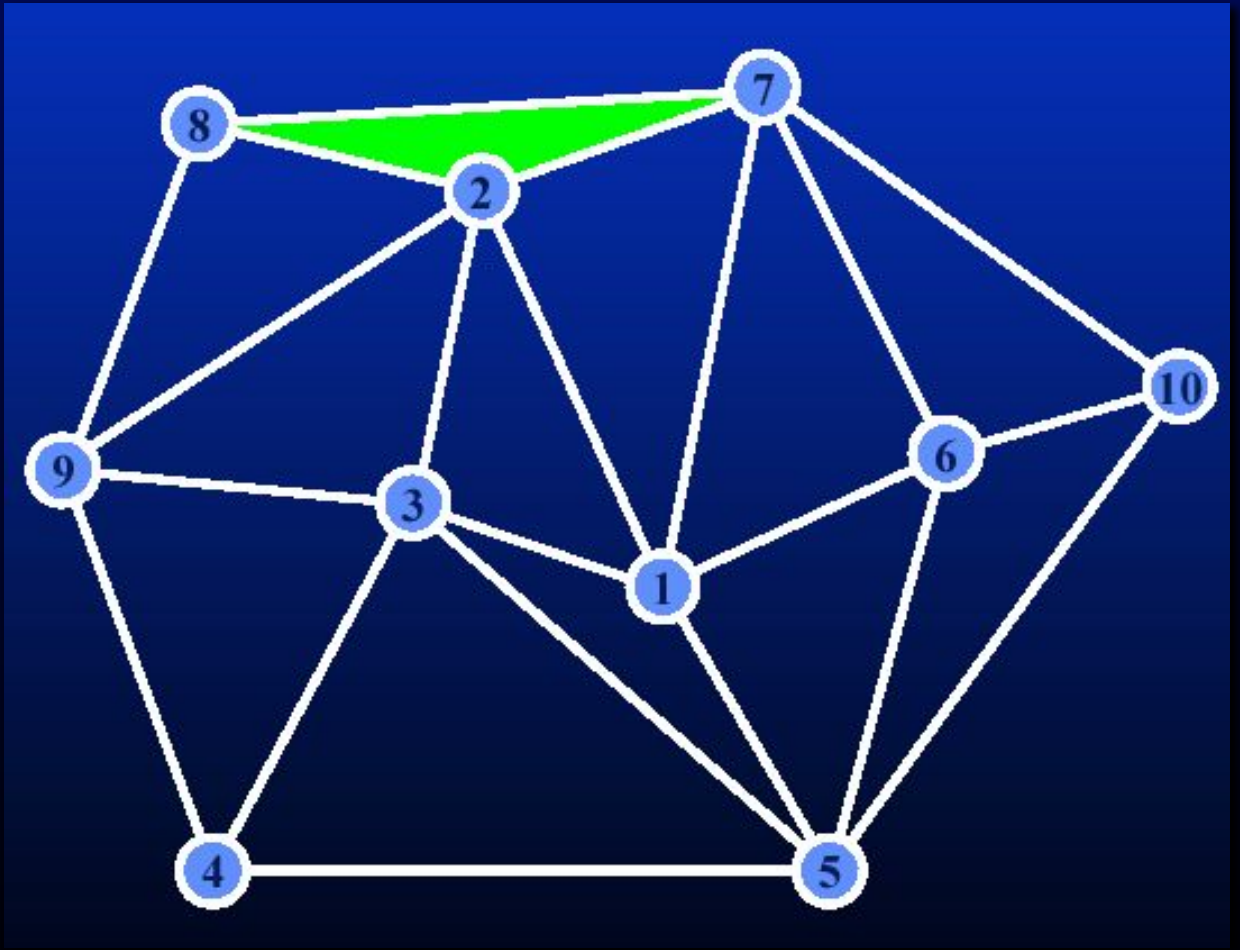
- Размер в экранных координатах
- Сохранение силуэта
- Поддержание фиксированного суммарного числа треугольников (бюджет треугольников)
- Критерии, основанные на человеческом восприятии изображения
- Временные критерии

# Оптимизация

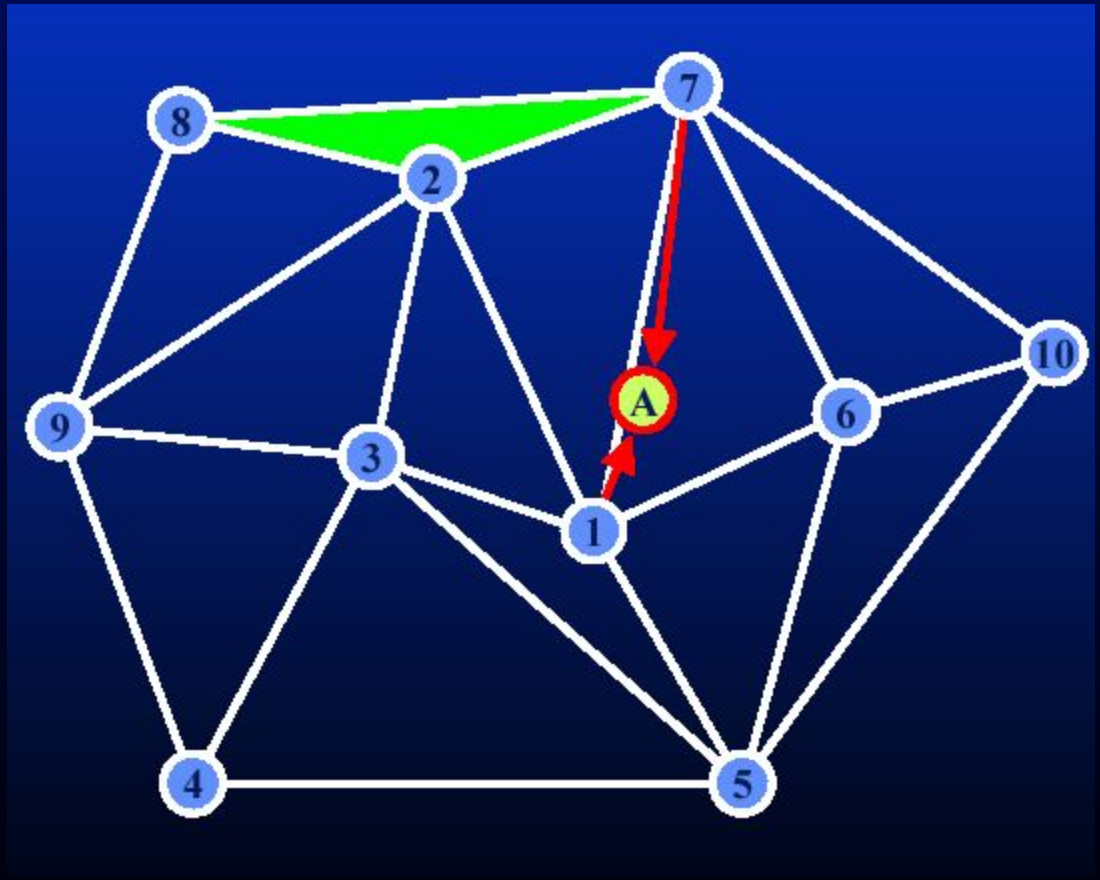

## *Методы оптимизации*

- Асинхронное упрощение сетки
  - Распараллеливание  алгоритма
- Использование пространственной связности
  - За малый промежуток времени сцена меняется мало
- Поддержание связанной геометрии в памяти
  - Оптимизация для аппаратных ускорителей

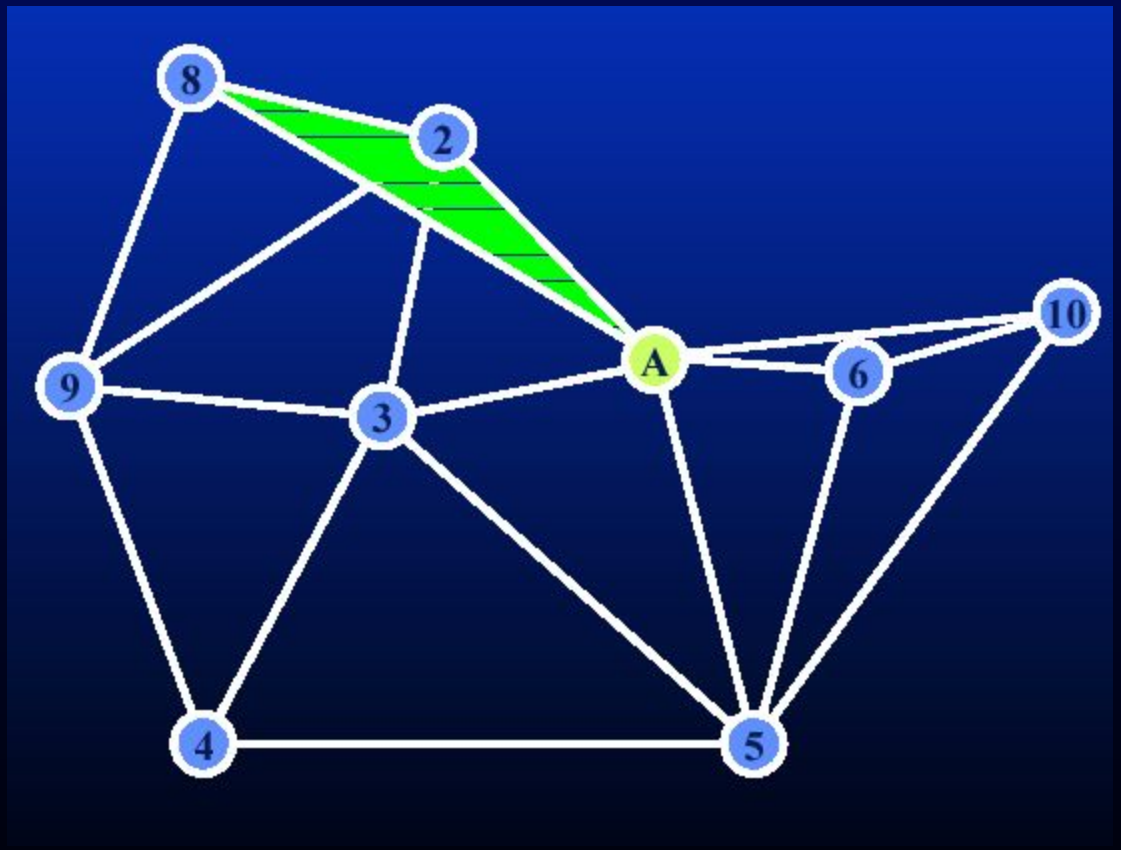
# Проблема: перехлестывание сетки



# Проблема: искажение сетки (mesh foldovers)



# Проблема: искажение сетки (mesh foldovers)



# Видозависимый LOD vs. Статического LOD


***Видозависимый LOD предпочтительнее, когда***

- Модели содержат большое число небольших объектов (напр. ландшафт)
- Упрощение должно выполняться автоматически (напр. CAD)
- Необходима гибкая система видозависимых критериев упрощения сетки

# Видозависимый LOD vs. Статического LOD


***Статический LOD является лучшим выбором, так как обладает свойствами***

- Простая для реализации модель
- Малая загрузка CPU
- Проще использовать аппаратуру

# Видозависимый LOD vs. Статического LOD


***Приложения, которые используют***

- Статический LOD
  - Видеоигры
  - Симуляторы
- Видозависимый LOD
  - Инструменты CAD
  - Научные и медицинские приложения
  - Виртуальные музеи

# Вопросы?