



Построение модели головы человека на основе нескольких фотографий

Акимов Евгений Владимирович

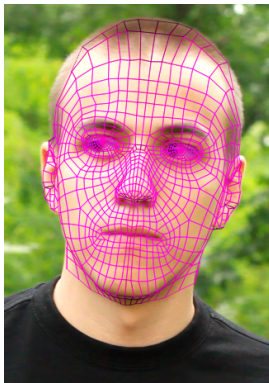
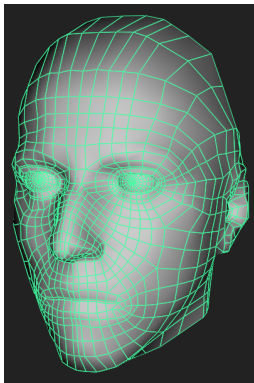
Научный руководитель: Белов Роман Вячеславович

САНКТ-ПЕТЕРБУРГСКИЙ АКАДЕМИЧЕСКИЙ УНИВЕРСИТЕТ

13.06.2017



- Трекинг - определение трехмерной позиции движущегося объекта
- Для трекинга необходима модель объекта, которую получать трудозатратно





- Детализация
 - Достаточность для трекинга, полная аналогия не нужна
 - Хорошая топология и uv-развертка полигональной сетки
- Входные данные (условия съемки, оборудование, калибровка)
 - Без необходимости съемки с фиксированных ракурсов
 - Без необходимости постоянного источника света
 - Без необходимости специальных камер
 - Без долгой “живой” калибровки перед использованием
- Возможность влиять на результат
 - Возможность вручную исправить решение если оно немного ошиблось



Цель

- Реализовать систему построения трехмерной модели головы человека по набору фотографий для ее последующего использования в трекинге

Задачи

- Выбрать способ представления модели и реализовать его
- Составить подходящую модель головы
- Реализовать эффективную оптимизацию параметров модели для получения модели конкретной головы
- Интегрировать систему в плагин для Foundry Nuke

Представление модели головы

Способ получения модели конкретной головы



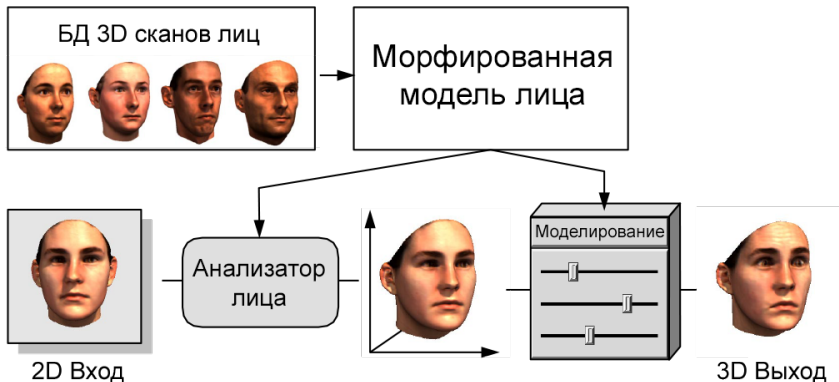
- Ручная установка “пинов” - соответствия 3D точки на поверхности модели с 2D точкой на фотографии
- Решение сводится к подбору трансформации геометрии и значений модификаторов модели

Представление модели головы



Моделирование с использованием базы трехмерных моделей лиц

- Модели с ручными модификаторам отдельных частей лица
- Статистические модели



Представление модели головы

Пример применения модификатора





Упрощенно (для одного пина и одной вершины):

$$\begin{array}{c}
 \begin{array}{c} \text{параметры} \\ \text{внутренней} \\ \text{калибровки} \\ \text{камеры} \end{array} \\
 \downarrow \\
 \begin{array}{c} \text{параметры} \\ \text{внешней} \\ \text{калибровки} \\ \text{камеры} \end{array} \\
 \downarrow \\
 \begin{array}{c} \text{преобразование} \\ \text{координат} \\ \text{модели} \end{array} \\
 \downarrow \\
 \begin{array}{c} \text{модификаторы} \\ \text{вершины} \\ \text{модели} \end{array} \\
 \downarrow
 \end{array}
 \begin{array}{c}
 s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 & X_1 & \dots & X_n \\ Y_0 & Y_1 & \dots & Y_n \\ Z_0 & Z_1 & \dots & Z_n \\ 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} 1 \\ m_1 \\ \vdots \\ m_n \end{bmatrix}
 \end{array}$$

В реальности:

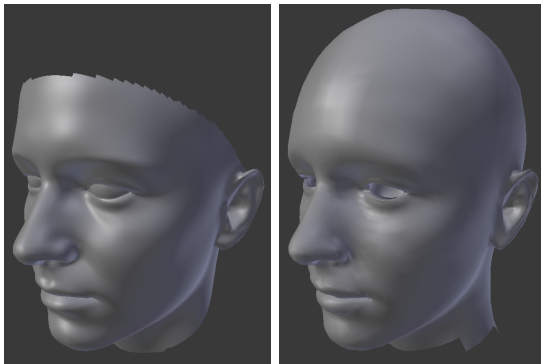
- несколько пинов
- барицентрические координаты точки на поверхности
- вектор параметров вращения
- регуляризация



- Целевая функция алгоритма оптимизации
 - ~ 220 аргументов функции
 - Вычисляет значение функции и матрицу Якоби
 - Одновременно для деформации и трансформации модели
- Алгоритм оптимизации
 - Размерность матрицы Якоби $\sim 250 \times 220$
 - ~ 20 итераций для сходимости алгоритма
 - ~ 10 пользовательских действий в секунду
- Время выполнения алгоритма
 - В исходной матричной форме: ~ 900 мс
 - После оптимизации: ~ 50 мс

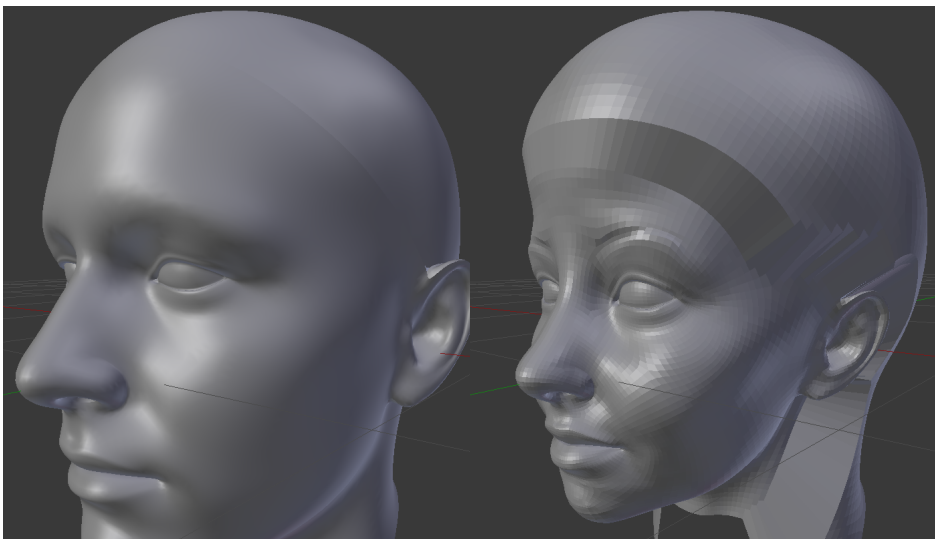


- Модификаторы лица лучше у example-based моделей
- Модификаторы головы можно получить из ручных моделей
- Модификаторы лица должны влиять на остальную часть головы



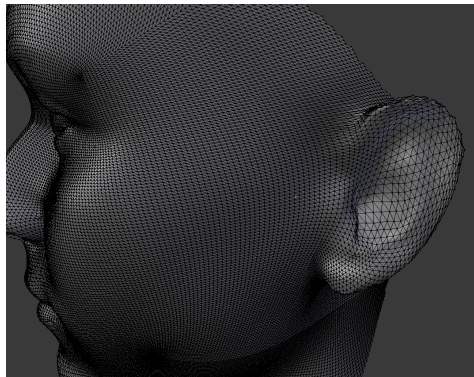
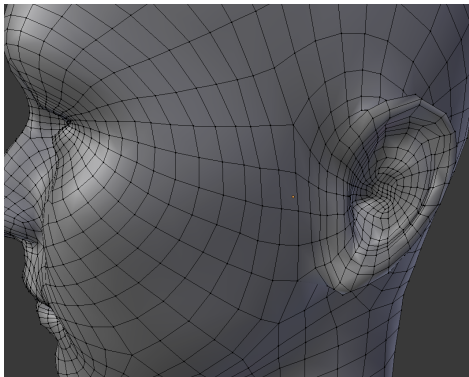
Создание модели головы

Модификаторы лица



Создание модели головы

Топология разных полигональных сеток





Требования

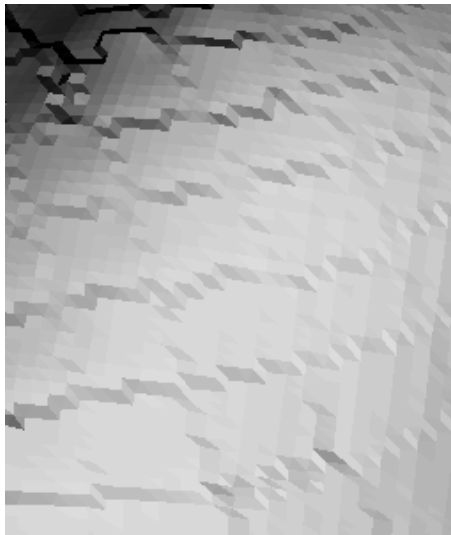
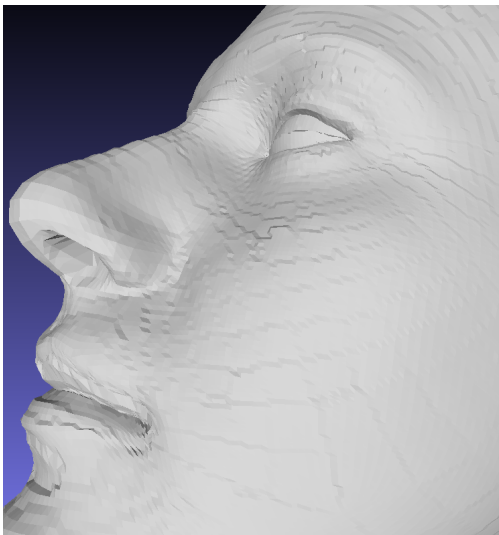
- Объединение модификаторов
- Плавность границы склеивания
- Хорошая топология полигональной сетки

Создание модели

- Подбор нелицевой части для совпадения по границе с лицом
- Поиск соответствий и интерполяция для модификаторов модели головы
- Трехмерная оптимизация для каждого из 200 модификаторов лица
 - Установка в максимальное значение
 - Совпадение вершин на границе
 - Жесткость относительно среднего положения
- Линейный переход для вершин граничной области лица и головы

Создание модели головы

Изменение топологии





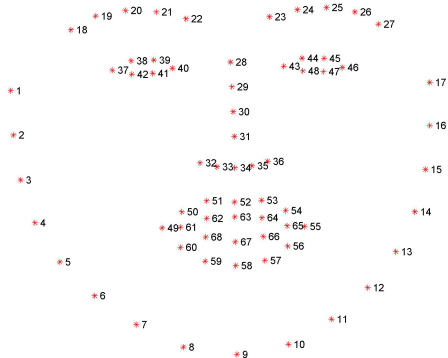
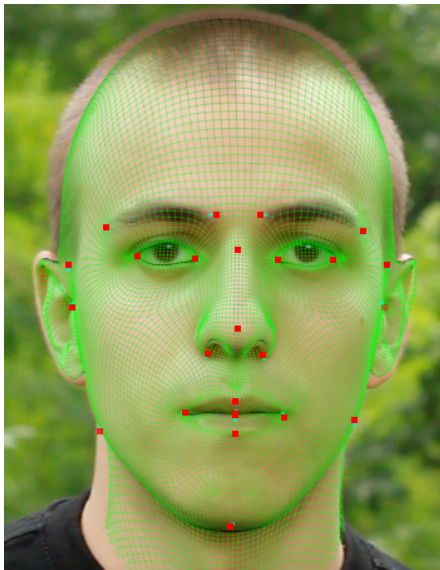
Интерполяция вершин поверхности по статье
Gregory M. Nielson. Scattered Data Modeling. IEEE Computer
Graphics and Applications, 1993

- Построение и минимизация интерполирующей функции
- Аргументы: расстояния до некоторых фиксированных точек поверхности

Интеграция



Автоматическое определение положения головы





Реализована система для построения модели головы человека по набору фотографий

- Составлена подходящая модель головы
- Алгоритм оптимизации параметров модели позволяет взаимодействовать на моделью в реальном времени
- Система интегрирована в плагин для Foundry Nuke



▶ video

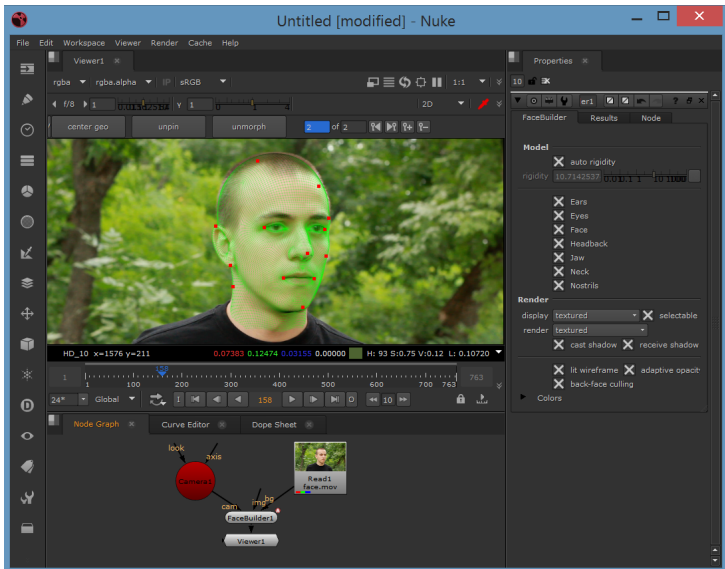


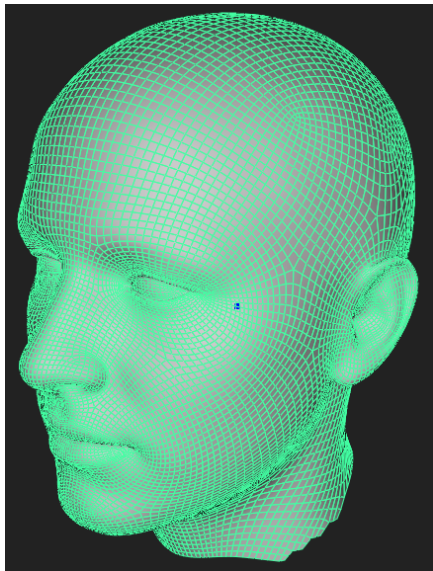
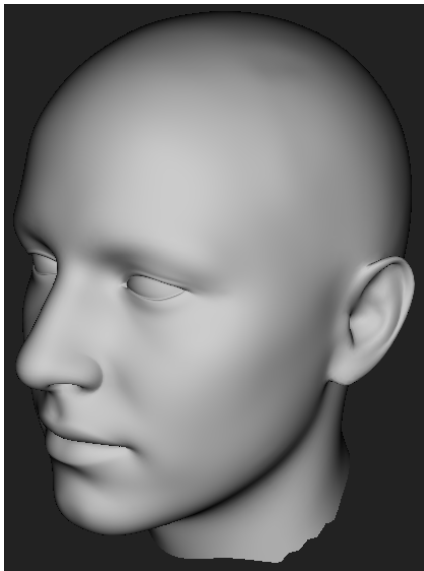
Вопросы?



Приложение

Интерфейс плагина



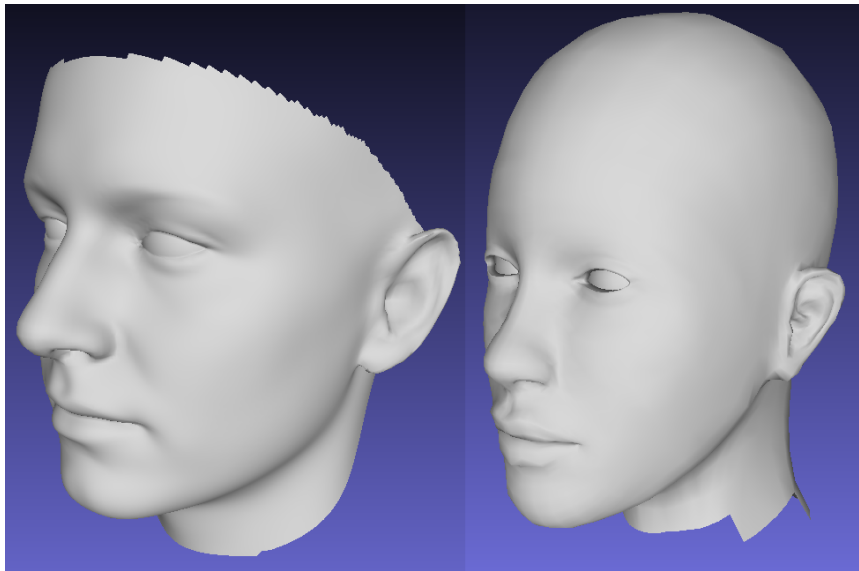


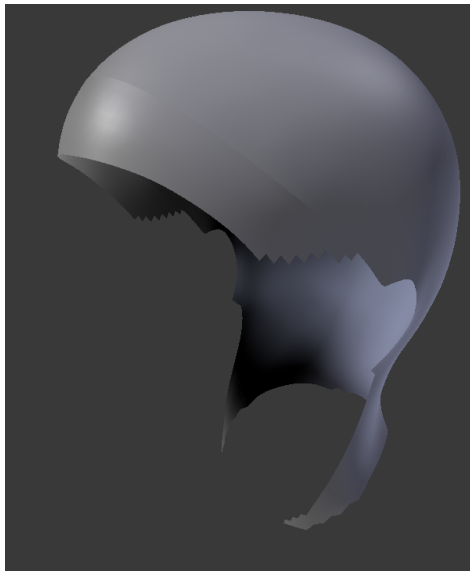


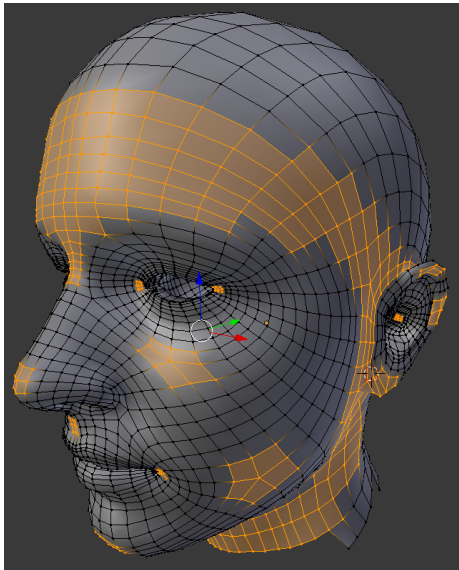
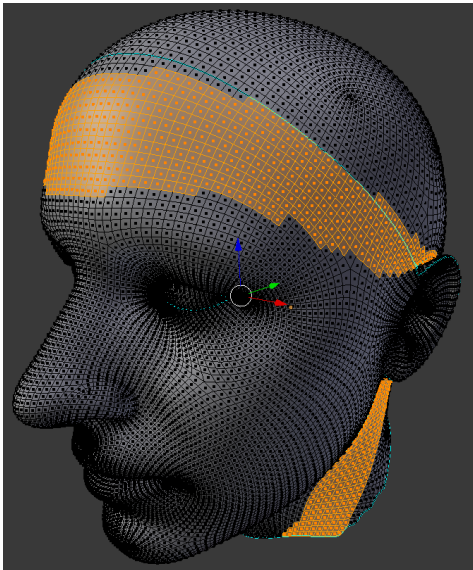
$$J = \begin{bmatrix} \frac{\partial E_{uv}}{\partial \vec{r}} & \frac{\partial E_{uv}}{\partial \vec{t}} & \frac{\partial E_{uv}}{\partial \vec{a}} \\ 0 & 0 & \frac{\partial E_a}{\partial \vec{a}} \end{bmatrix} \quad (1)$$

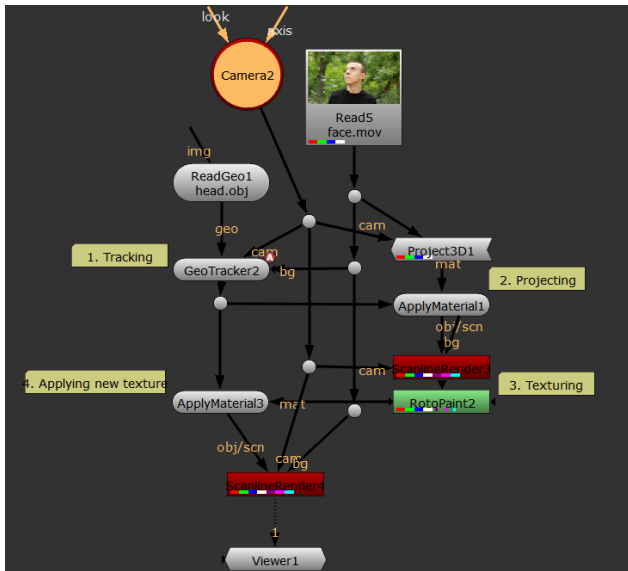
$$[J^T(\vec{x}_k)J(\vec{x}_k) + \lambda_k I] \vec{p}_k = -J^T(\vec{x}_k) \vec{f}(\vec{x}_k) \quad (2)$$

$$\begin{aligned} J^T J &= \begin{bmatrix} \frac{\partial E_{uv}}{\partial \vec{r}} & 0 \\ \frac{\partial E_{uv}}{\partial \vec{t}} & 0 \\ \frac{\partial E_{uv}}{\partial \vec{a}} & \frac{\partial E_a}{\partial \vec{a}} \end{bmatrix} \begin{bmatrix} \frac{\partial E_{uv}}{\partial \vec{r}} & \frac{\partial E_{uv}}{\partial \vec{t}} & \frac{\partial E_{uv}}{\partial \vec{a}} \\ 0 & 0 & \frac{\partial E_a}{\partial \vec{a}} \end{bmatrix} = \\ &= \begin{bmatrix} \frac{\partial E_{uv}}{\partial \vec{r}} \\ \frac{\partial E_{uv}}{\partial \vec{t}} \\ \frac{\partial E_{uv}}{\partial \vec{a}} \end{bmatrix} \begin{bmatrix} \frac{\partial E_{uv}}{\partial \vec{r}} & \frac{\partial E_{uv}}{\partial \vec{t}} & \frac{\partial E_{uv}}{\partial \vec{a}} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (\frac{\partial E_a}{\partial \vec{a}})^2 \end{bmatrix} \end{aligned} \quad (3)$$









Приложение Камера в Foundry Nuke



The screenshot displays the Foundry Nuke interface for a 3D camera setup. The main viewport shows a 3D scene with a white sphere and a camera rig. The camera rig consists of several interconnected nodes: a red 'Camera' node, a 'FaceBuilder' node, a 'TransformGeo' node, and a 'Camera' node. The 'TransformGeo' node is selected, and its properties are visible in the right-hand pane. The 'TransformGeo' node has the following settings:

- Look: Node
- display: unchanged
- render: unchanged
- transform normals: [checked]
- Transform order: BRT
- rotation order: ZXY
- translate: x = 2.0299999, y = 0, z = 2.7300000
- rotate: x = 0, y = 2.00129766, z = 0
- scale: x = 1, y = 1, z = 1
- uniform scale: 1
- skew: x = 0, y = 0, z = 0
- pivot: x = 0, y = 0, z = 0
- Local matrix: [empty]

The 'FaceBuilder' node is also visible in the right-hand pane, with the following settings:

- Facebuilder: Node
- Results: model type: [set]
- file: C:/diploma2/model_new/compass/mh/
- model regularization: 10

The bottom of the interface shows a Node Graph with a red 'Camera' node connected to a 'FaceBuilder' node, which is connected to a 'TransformGeo' node, which is connected to a 'Camera' node. The 'FaceBuilder' node is also connected to a 'Camera' node. The 'Camera' node is connected to a 'Camera' node. The 'Camera' node is connected to a 'Camera' node.