

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Высшая школа теоретической механики

Работа допущена к защите

Директор в.ш.т.м, д.ф.-м.н., чл.-корр. РАН

_____ А. М. Кривцов

«___» _____ 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

«ТЕХНОЛОГИЯ СОЗДАНИЯ ИНДИВИДУАЛЬНЫХ ОРТОПЕДИЧЕСКИХ СТЕЛЕК С ИСПОЛЬЗОВАНИЕМ 3D ПЕЧАТИ»

по направлению 01.04.03 Механика и цифровое математическое моделирование
профиль 01.04.03_03 Механика и цифровое производство

Выполнила
студентка гр. 3640103/80301



А.П. Пеева

Руководитель
к.ф.-м.н., доцент



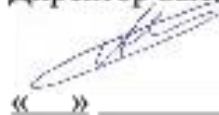
М.Б. Бабенков

Санкт-Петербург

2020

УТВЕРЖДАЮ

Директор высшей школы



А.М. Кривцов

« » 2020 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Пеевой Алисе Павловне 3640103/80301

1. Тема работы: «Технология создания ортопедических индивидуальных ортопедических стелек с использованием 3D печати».
2. Срок сдачи студентом законченной работы: 8 июня 2020 г.
3. Исходные данные по работе: справочная литература отечественных и зарубежных ученых в исследуемой и смежных областях, конструкторско-технологическая документация, актуальные публикации по теме исследования.
4. Содержание работы (перечень подлежащих разработке вопросов): разработать полный цикл создания и реализации конечного продукта. Повысить качество и конкурентоспособность индивидуальных ортопедических стелек на основе совершенствования процессов их проектирования.
5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрены.
6. Консультанты по работе: не предусмотрены.
7. Дата выдачи задания: 22 января 2020 г.

Руководитель ВКР



М.Б. Бабенков

Задание принял к исполнению 22 января 2020 г.



А.П. Пеева

Студент

РЕФЕРАТ

На 45 с., 28 рисунков, 5 таблиц, 1 приложение.

**ОРТОПЕДИЧЕСКИЕ СТЕЛЬКИ, ДЕФОРМАЦИЯ ОРТЕЗОВ,
ФОТОГРАММЕТРИЯ, 3D-МОДЕЛЬ СТЕЛЕК, СЕБЕСТОИМОСТЬ
ИЗГОТОВЛЕНИЯ ИНДИВИДУАЛЬНЫХ СТЕЛЕК, 3D ПЕЧАТЬ, ТПУ**

В работе представлены результаты исследований, нацеленных на повышение качества индивидуальной ортопедической стельки путем разработки и внедрения в процесс проектирования передовых методов компьютерного моделирования, фотограмметрии, анализа материалов и учета их механического поведения при эксплуатации. Проведен анализ существующих аналогов. Предложена технология изготовления индивидуальных ортопедических стелек, выбрана рациональная конструкция и материал стелек. Проведены расчеты деформации стелек при нагрузке. Имея 3D-модель стельки, можно не только напечатать на 3D – принтере, но изготовить ее на станке с ЧПУ. Важная часть работы – это обоснование себестоимости изготовления индивидуальных ортопедических стелек.

THE ABSTRACT

45 pages, 28 pictures, 5 tables, 1 application.

**CORRECTIVE INSOLES, ORTHOSES DEFORMATION, PHOTOGRAMMETRY, IN-
SOLE 3D MODEL, COST OF INDIVIDUAL INSOLES PRODUCTION, 3D PRINT-
ING, TPU**

The paper presents the results of studies aimed at improving the quality of an individual corrective insole by developing and introducing modern methods of computer modeling, analysis of materials and consideration of their mechanical behavior during operation into the design process. The analysis of existing analogues. The technology of manufacturing individual corrective insoles is proposed, the rational design and material of the insoles is selected. The insoles deformation under load was calculated. Having a 3D model of the insole, you can not only print on a 3D printer, but also make it on a CNC machine. An important part of the work is the substantiation of the cost of manufacturing individual corrective insoles.

Содержание

ВВЕДЕНИЕ	5
Глава 1. Обзор аналогов и существующих решений	6
1.1. Термоформовочные каркасные стельки	6
1.2. Термоформовочные бескаркасные стельки	6
1.3. Стельки по гипсовому позитиву	6
1.4. Индивидуальные стельки на эва-матрице	7
1.5. Индивидуализируемые стельки с ламельной опорой под своды	7
Глава 2. Сканирование стопы	8
2.1. Обзор существующих 3D сканеров	8
2.2. Создание фотостола	9
2.3. Материал для слепка	11
2.4. Фотограмметрия	12
Глава 3. Создание стельки	15
3.1. Моделирование скана ступней	15
3.2. Проектирование стельки	18
3.2.1. Статический анализ ступней до использования стелек	18
3.2.2. Gensole	19
3.3. Материал для 3D печати	22
3.3.1. Термопластичный полиуретан	22
3.3.2. Деформация и давление на модель стельки	24
Глава 4. Изготовление индивидуальных ортопедических стелек с использованием 3D печати	27
4.1. Обзор оборудования	28
4.2. Процесс изготовления	29
Глава 5. Расчет себестоимости изготовления индивидуальных стелек	31
Глава 6. Выводы	33
Заключение	34
Список использованных источников	34
Приложение 1. Код использования Meshroom	36

ВВЕДЕНИЕ

Предприятия обувной промышленности ежегодно поставляют миллионы товаров людям по всему миру. Главная цель отрасли – удовлетворить потребности покупателя. Существует множество конструкций повседневной обуви для каждого сегмента населения: ортопедическая, спортивная, демисезонная, производственная и др.

Сложно себе представить удобную обувь без стелек – функциональный «вкладыш» между стопой и внутренней части подошвы, обеспечивающий комфорт при ношении.

Полезная составляющая стелек заключается в коррекции деформации ступней, а также в снятии нагрузок с плантарных участков стоп при артрите и прочих заболеваниях опорно-двигательного аппарата [4]. Такие стельки принято называть ортопедическими, в большинстве случаев «индивидуальные» корректирующие стельки содержат вставку – супинатор для продольного свода стопы, который «снимает нагрузку».

Ежегодное увеличение численности населения неизбежно приводит к росту спроса на обувь, что в свою очередь стимулирует обувную индустрию в целом. Изготовление стелек должно отвечать обширным требованиям потребителей, таким как: комфорт, эргономичность, соответствие цена/качество.

В соответствии со стратегией утвержденной главой Российской Федерации (7 февраля 2008 года № Пр-212) необходимо «повысить качество жизни граждан, обеспечить конкурентоспособность России, путем внедрения новых технологий и инноваций на период до 2025 года» [16].

На первом этапе разработки инновационной модели стелек имеется ряд «узких мест», формализация которых с применением передовых технологий посредством фотограмметрии, смогла бы существенно снизить субъективность решений производителя. Однако, множество предприятий завышают цену на изготовление индивидуальных ортопедических стелек и не учитывают специфику стопы при изготовлении. Использование предложенной методики направленно на сокращение и оптимизацию временных ресурсов, направленных на производство. Следовательно, разработка технологии изготовления индивидуальных ортопедических стелек с использованием средств 3D печати является актуальной.

Результаты исследований направлены на повышение качества индивидуальных ортопедических стелек путем разработки и внедрения в процесс проектирования современных методов компьютерного моделирования, фотограмметрии, анализа материалов и учета их механического поведения при эксплуатации.

Глава 1. Обзор аналогов и существующих решений

1.1. Термоформовочные каркасные стельки

Основная функция таких стелек - поддержка продольного свода при ходьбе [15]. В состав поддерживающего каркаса входит 2-3 мм термопластика. Используется в обуви с узкой подошвой. При изготовлении стелька нагревается до мягкого состояния и формируется по ноге, либо по заранее зафиксированному следу, полученному на вакуумной подушке. После застывания форма стелек фиксируется, и они готовы к использованию.

Время изготовления: 20-25 минут

Средняя цена: 2500 – 7000 рублей

Минусы: поскольку изгибы отформованного продольного свода остаются незаполненными, пластик в этом месте теряет свою форму за 2-3 месяца из-за нагрузок. При занятиях спортом – еще быстрее, вплоть до растрескивания несущей пластины. В большинстве салонов своды перед формовкой не дорабатывают, и клиент получает стельку, которая никоим образом не корректирует его стопу.

Плюсы: тонкие стельки, легко адаптируются под любую обувь, изготавливаются прямо при пациенте.

1.2. Термоформовочные бескаркасные стельки

Этот тип стелек выполнен из термоформуемого вспененного материала. Подобные стельки используются в спортивной обуви. Технология изготовления практически та же, что и в п.1.1. Стельки нагреваются до мягкого состояния, затем формуются прямо в обуви под весом пациента при ходьбе.

Время изготовления: 20-25 минут

Средняя цена: 6500 – 7500 рублей

Минусы: поскольку первичное формование происходит уже в обуви, для нанесения корректоров нужен повторный визит и доработка стелек. Коррекция затруднена отсутствием полной опоры пяточной части стельки на обувь.

Плюсы: в отличие от предыдущего типа 1.1 эти конструкции полнотелые, и не склонны к столь быстрому проседанию, как каркасные стельки. Максимальный срок применения – 3 года.

1.3. Стельки по гипсовому позитиву

Сложнонаборные стельки с пластиковой, наборной кожаной или карбоновой несущей. Могут быть как профилактическими, так и корректирующими. При изготовлении очень важна квалификация ортезиста.

Время изготовления: 20-25 минут
Средняя цена: 2500 – 7000 рублей

Минусы: время изготовления достаточно большое, высокая стоимость. При обработке гипсового позитива часто возникают ошибки, и, как результат, малоэффективная конструкция. Сложно поддаются коррекции. Не подходят для каблучной (более 5 см) обуви.

Плюсы: срок службы – примерно 2 года

1.4. Индивидуальные стельки на эва-матрице

Сложные стельки, основа – матрица из ЭВА жесткостью 45-50 по Шор. Корректирующие элементы наносятся на матрицу по результатам обследования. Могут быть как профилактическими, так и корректирующими.

Время изготовления: 1,5 часа – 3 дня
Средняя цена: 3000 – 7000 рублей

Минусы: нельзя нагревать более 60 градусов. Не подходят для каблучной (более 5 см) обуви.

Плюсы: легко поддаются коррекции. Срок службы матрицы 3-5 лет.

1.5. Индивидуализируемые стельки с ламельной опорой под своды

Этот вид стелек истинно индивидуальными назвать очень сложно, поскольку изготовление подразумевает всего лишь регулировку высоты ламелей под сводами. Технология индивидуализации выглядит следующим образом: стелечная заготовка размещается на специальной установке в решетке, через которую пропущены ламели. Под стелькой вставляется специальный клин для предотвращения избыточного проседания пяточной зоны. Сзади на установке крепится пяточный позиционер. Пациент приставляет пятку к позиционеру, потом делает шаг по заготовке, ламели проходят через решетку, тем самым формируя своды. После этого лишние части ламелей срезаются.

Время изготовления: 1,5 часа – 3 дня
Средняя цена: 3000 – 7000 рублей

Минусы: первично изготовленную стельку откорректировать практически невозможно. Высоту сводов можно только уменьшить. Конструкция не подходит для сложных деформаций стопы.

Плюсы: силиконовые стельки прекрасно компенсируют нагрузки на стопу, конструкции могут быть адаптированы под каблучную обувь, легко моются.

Изучив множество аналогов и сравнив их между собой, приходим к выводу, что варианты 1.1 и 1.5 подходят для краткосрочного ношения в узкой модельной и каблучной обуви, а также для профилактики развития продольного плоскостопия. Варианты 1.2, 1.3 и 1.4 – коррекционные, все варианты исполнения стелек принесут хороший результат, и при правильном подходе к коррекции плоскостопия индивидуализация и контроль проводится раз в 3-6 месяцев, что на варианте 3 несколько проблематично.

Глава 2. Сканирование стопы

За последнее пятилетие, множество производителей стелек внедряют в свой рабочий процесс 3D сканеры для ног для создания трехмерных моделей, которые, в отличие от двухмерных изображений, дают полное представление о проблемах и потребностях клиента и предоставляют более широкий инструментарий для создания индивидуальных ортопедических стелек. Технология 3D сканирования помогает разрабатывать и создавать спортивную и терапевтическую обувь, идеально подходящую для каждого заказчика. Помимо терапевтического эффекта, обувь, разработанная по технологии 3D сканирования, помогает уменьшить усталость при длительных физических нагрузках, снизить риск развития плоскостопия, снизить нагрузку на позвоночник.

2.1. Обзор существующих 3D сканеров

Одним из примеров, является компания ScanPod3D, которая специализируется на производстве аддитивного оборудования для ортопедии [12]. В линейке сканеров для ног на данный момент представлено множество моделей: от небольшого сканера для одной ноги весом менее 5 кг до более продвинутых с разрешением снимков FULL HD.

Сканер USOL (рис. 2.1) разработан для 3D сканирования одной ноги с возможностью проводить различные измерения, масштабировать и поворачивать изображения. Это самый компактный сканер из представленных, его вес составляет всего 4.7 кг. Отличается хорошей скоростью, портативными размерами и бюджетной ценой. За один раз сканируется одна стопа, но также есть модель USOL-DUO, которая может сканировать 2 стопы одновременно. На передней панели расположена вертикальная планка с лазером, которая помогает выровнять стопу и лодыжку. Сзади идет небольшая выдвижная планка с камерой для выравнивания пятки. Сканер можно использовать и вертикально с помощью специального штатива. Скорость сканирования – 1,9 – 3 секунды. Платформа выдерживает вес до 180 кг. Высота сканирования 80 мм. Сканер USOL – это отличное решение для изготовления и подбора индивидуальных ортопедических стелек и обуви.

Все сканеры подобного типа принадлежат к разным ценовым категориям и заточены под различные задачи. Все аналоговые модели создают точное цветное или монохромное 3D изображение стопы, с которым далее можно работать в фирменном ПО. Все они сканируют ноги с разной степенью нагрузки: с полной, частичной и без нагрузки, что позволяет ортопеду поставить более точный диагноз пациенту и дать соответствующие рекомендации. Стоимость варьируется от 400 000 рублей до 1 000 000 рублей, одной из задач дипломной работы является создание индивидуальных ортопедических стелек доступных каждому. Стельки, которые производятся, используя представленные в этой главе сканеры стоят от 6 000 рублей за штуку, что не является приемлемым для выполнения задачи работы.

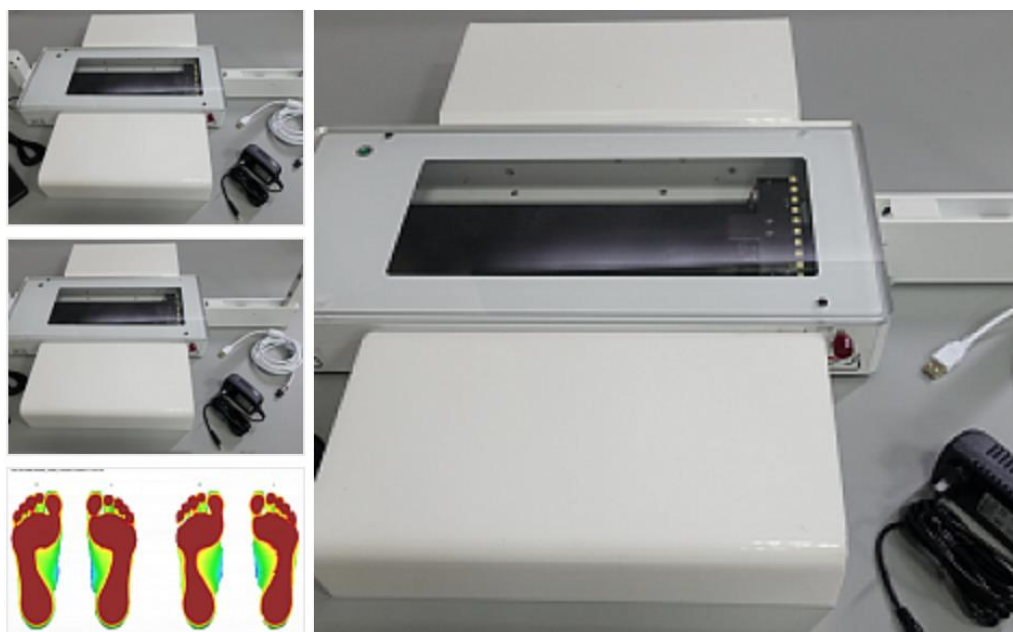


Рисунок 2.1. 3D - сканер USOL

Уже в ближайшие годы обувные 3D-технологии получат максимально широкое применение, стоимость подобных услуг снизится, географическая доступность вырастет и воспользоваться ими сможет любой. Но на данный момент удовлетворяющих запросы потребителей существующих аналогов нет, поэтому мной была разработана другая технология создания индивидуальных ортопедических стелек с использованием 3D печати [17].

2.2. Создание фотостола

Одно из приспособлений, значительно облегчающее предметную фотосъемку и повышающее качество изображений, – это поворотный фотостол. Используя программу Solidworks для моделирования фотостола, ЧПУ для изготовления деталей, создала фотостол (рис. 2.2.1), состоящий из вращающегося диска и неподвижной основы (на которой он закреплен), светодиодной ленты (рис.2.2.4), пластиковых

рассеивателей света и крепления для камеры [9]. Использование этого устройства позволяет получать объемные изображения предметов.

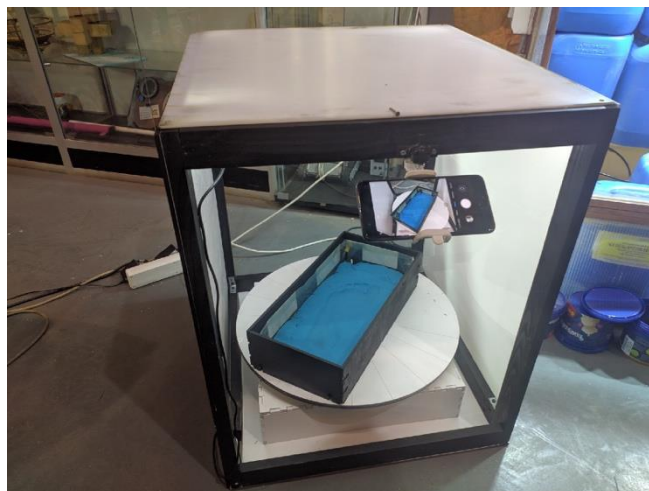


Рис. 2.2.2 Фотостол с боксом

Диск фотостола разделен на определенное количество одинаковых секторов.



Рис. 2.2.3 Диск фотостола

Сама фотосъемка происходит следующим образом: на поверхность стола помещается бокс со слепком, камера устанавливается в крепеж, после чего делается первый снимок.



Рис. 2.2.4 Светодиодная лента



Рис. 2.2.5 Фотостол

Затем диск проворачивается на один сектор, и цикл повторяется. На выходе мы получаем количество снимков, равное количеству секторов поворотного стола.

2.3. Материал для слепка

Благодаря своим свойствам, выбор материала для слепка пал на «кинетический песок», в состав которого входят всего лишь 2 компонента: диоксид кремния (98%) и полидиметилсилоксан (2%), который выступает в качестве связующего агента [7]. Смесь не рассыпается на мелкие частицы, благодаря образованию между ними полимерных волокон, которые их удерживают. А также не прилипает и не липнет к тканям. Кинетический песок не требует никаких специальных условий хранения и не теряет физических свойств с течением времени.

Диоксид кремния — это не что иное, как чистейший кварцевый песок. Полидиметилсилоксан также известен нам как «Энтеросгель» или «Пищевая добавка E900» — то есть, он не только абсолютно безвреден, но и съедобен. Материал не вызывает аллергии, не токсичен и не поддерживает размножение болезнетворных микроорганизмов. Заполнив боксы пятью килограммами песка, клиент встает в боксы с нагрузкой своего тела, и мы получаем слепок стопы для дальнейшего сканирования.



Рис. 2.3. Слепок стопы

2.4. Фотограмметрия

Используемая программа для фотограмметрии - Meshroom с открытым исходным кодом, написанная на Python, выполняющая вызовы команд. Обработка изображений выполняется отдельными программами на C++, работающими из командной строки, где каждую операцию мы можем отслеживать в терминале. Meshroom и AliceVision запустятся не на каждой платформе [10]. Используемый скрипт (приложение 1). Программа Meshlab применяется для обработки изображений, используемые файлы:

- build_files: файлы, которые собрали
- dataset_monstree-master: исходные изображения
- Meshroom-2018.1.0: двоичные файлы Meshroom/AliceVision
- Всё остальное: скрипты для их запуска

Файл Python получает 5 аргументов:

- baseDir: папка, в которую помещаются временные файлы.
- imgDir: папка, содержащая исходные изображения.

- binDir: папка, содержащая исполняемые файлы AliceVision, например, aliceVision_cameraInit.exe.
- numImages: количество изображений в imgDir.
- runStep: выполняемая операция.

Получив набор изображений – равному количеству делений фотостола (20 под углом 30 градусов), выглядящих следующим образом (рис. 2.4.1):



Рис. 2.4.1 Изображение слепка ступни

С помощью python-скрипта run_alicevision.py создаем следующую структуру папок. В папке 11_Texturing находится готовая модель, открываемая в Meshlab:

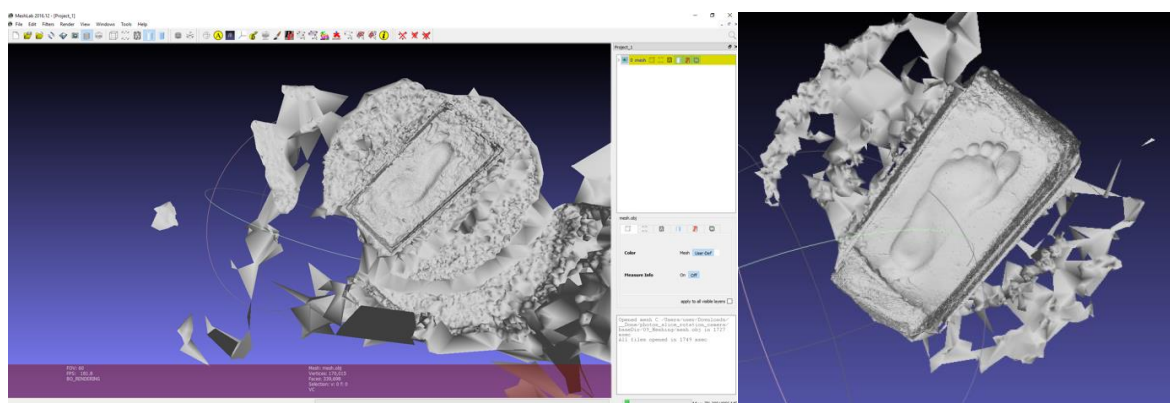


Рис. 2.4.2 Модель в Meshlab

Процесс обработки включает в себя 11 папок-этапов, которые можно запускать по очереди с помощью файлов `run_monstree_runXX.bat`, или воспользоваться `run_monstree_all.bat`, чтобы собрать их все сразу.

Первый этап сгенерирует файл SFM. Файлы SFM (в формате JSON), содержащие в себе информацию о камерах (размеры, сенсор, точки наблюдения и др.)

Второй итерация отделяет особенности силуэта изображений, в том числе дескрипторы. Этап постобработки: `02_ImageMatching`, определяющий, какие из изображений разумно сравнивать друг с другом. В случае если бы применялся набор из большего количества изображений, понадобилось бы большое количество времени, этот этап обработки не включает найденные пары.

`03_FeatureMatching` находит соответствия между изображениями с помощью дескрипторов характерных черт.

Этап `04_StructureFromMotion` представляет собой этап визуализации фотограмметрического диапазона для оценки трехмерных структур из последовательностей двумерного изображения, которые позволяют правильно позиционировать камеры.

- `bundle.sfm`: файл SFM со всеми наблюдениями.
- `cameras.fm`: файл SFM с данными только привязанных камер.
- `cloud_and_poses.ply`: найденные точки и камеры.

Следующий этап `05_PrepareDenseScene` — устраняет искажения в изображениях. После этого этапа появляются чёрные области, но последующие этапы `AliceVision` не используют настоящую матрицу камеры. Мы лишь делаем вид, что теперь имеем камеру без искажений, а `05_PrepareDenseScene` искажает исходные данные под эту искусственно созданную матрицу.

`06_CameraConnection` — этот этап нарушает принцип рабочего процесса. Все этапы были разработаны так, дабы любая папка становилась совершенно уникальным отдельным этапом.

`07_DepthMap` - это самый временно затратный этап программы, где происходит генерирование карт глубин. Создается карта глубин для каждого изображения как файл `EXR`.

`08_DepthMapFilter` — этап, изолирующий области, перекрытые другими картами глубин.

`09_Meshing` — этап генерации меша. Дабы избежать проблемы с генерацией, `10_MeshFiltering` — этап получает меш `09_Meshing` и усовершенствует его. Он

выполняет следующие операции (при необходимости все эти параметры можно настроить):

- сглаживает меш
- устраняет большие треугольники
- сохраняет наибольший меш, но удаляет все остальные.

11_Texturing – заключительный этап, в котором происходит UV преобразование (соответствие между координатами на поверхности трёхмерного объекта и координатами на текстуре) и проецирование текстур.

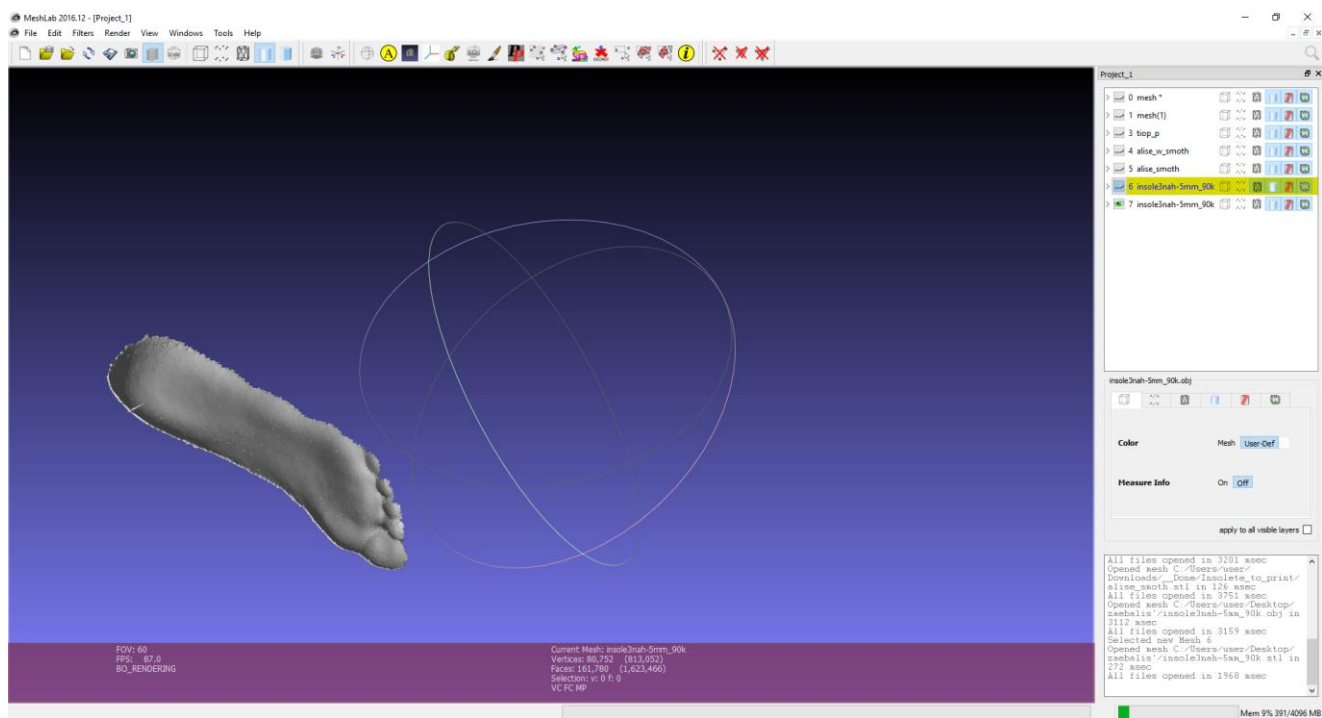


Рис. 2.4.2 Извлеченный скан ступни в Meshlab

Глава 3. Создание стельки

3.1. Моделирование скана ступней

Изучив огромный спектр программ 3D моделирования: Zbrush, 3DsMax, Agisoft photoscan, FreeCAD, K-3D, Paint 3D, Fusion 360, SOLIDWORKS, Autocad, Компас 3D, ни одна из них не подошла для работы с облаком точек [3].

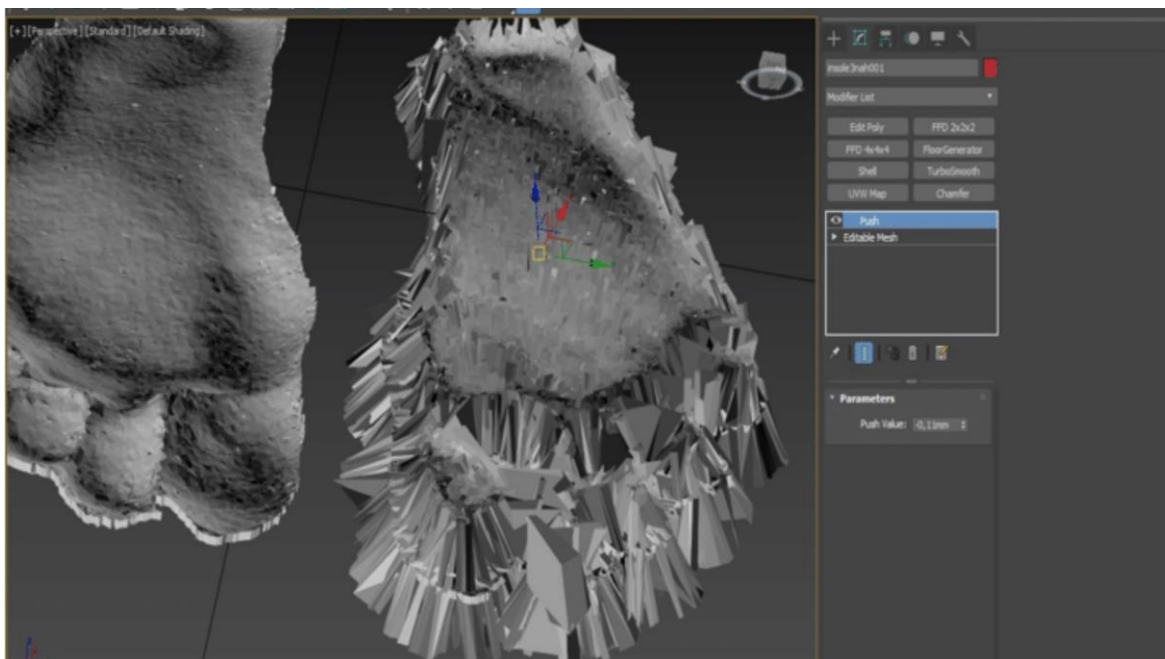


Рис. 3.1.1 Интерфейс программы 3DsMax

Используя софт Materialise Magics 22.03 - программа Magics от разработчика Materialise, помогла подготовить данные. Это мощный инструмент для эффективного управления всем процессом аддитивного производства, начиная от простого импорта данных и заканчивая постобработкой. С помощью данной программой полученное облако удалось создать STL модель. Осуществив всестороннее ручное редактирование сетки, включая анализ толщины, обработку пустот, сглаживание плохих краев и наиболее сложных ошибок треугольника.

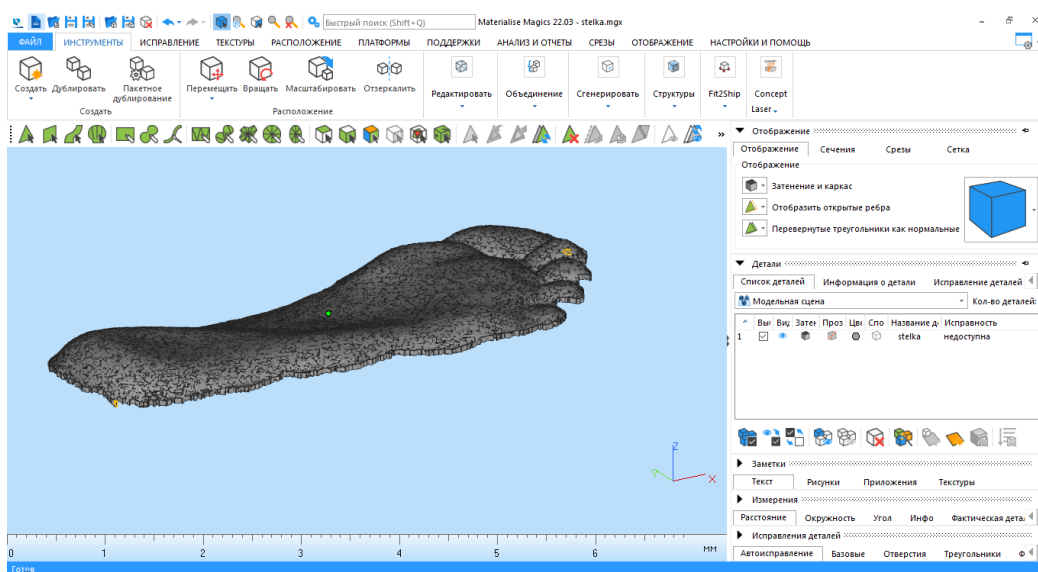


Рис. 3.1.2 Интерфейс программы Materialise Magics 22.03

Исправление ошибок выполнено с помощью мастера исправлений Fix Wizard. Нажав Go to Advised Step (Перейти к рекомендованному шагу), чтобы открыть диагностическую таблицу для проверки исправления различных ошибок. Для больших сеток рекомендуется снять флажки Overlapping (Перекрывающиеся треугольники и Пересекающиеся треугольники) и сначала исправить основные ошибки. После выбора Go to Advised Step (Обновить) снова нажав на Advised Step (Перейти к рекомендованному шагу), а затем Automatic Fixing (Автоматическое исправление) — это исправит большинство ошибок. В случае, если мастер исправления не может исправить поврежденные края и перекрывающиеся треугольники, то Stitch функция под Stitching в Fix Wizard меню часто оказывается полезной с достаточно высоким значением допуска.

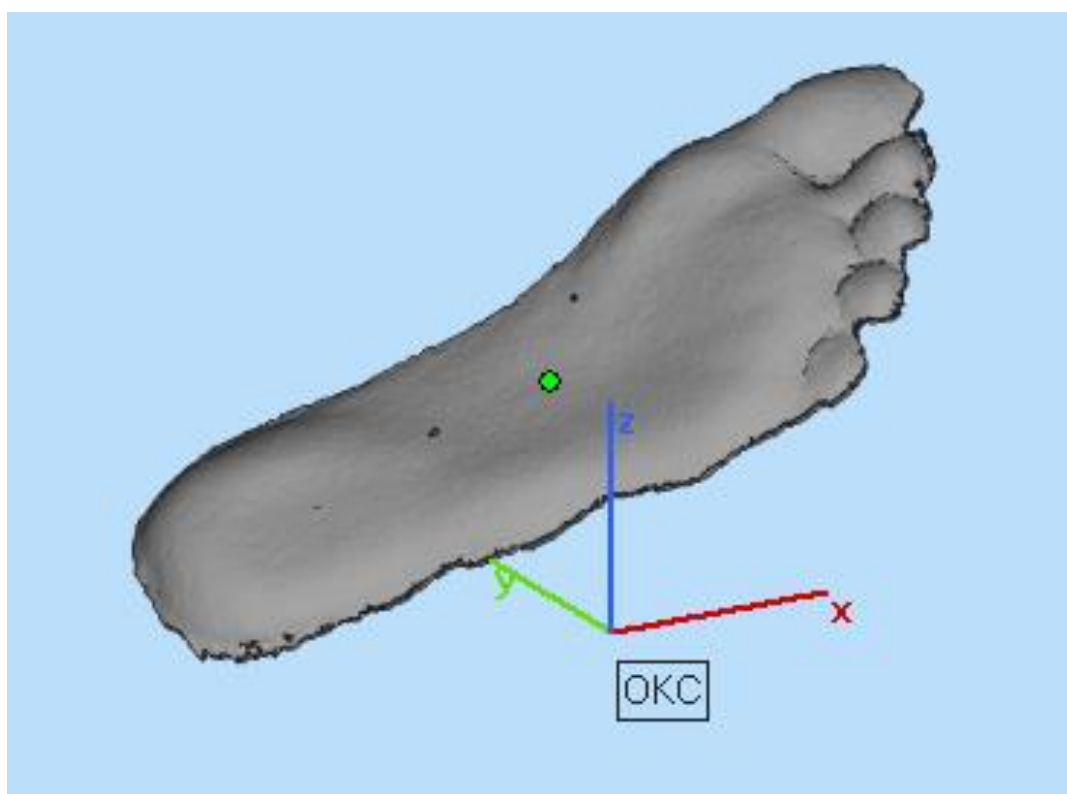


Рис. 3.1.3 Процесс удаления пересекающихся треугольников

Для оставшихся перекрывающихся треугольников либо снова использовать мастер исправлений, либо выбрав Detect Overlapping (Обнаружение перекрытий) в разделе Overlaps (Перекрытия) в меню Fix Wizard (Мастер исправлений). При этом выбираются все перекрытия, которые затем будут удалены нажатием кнопки Delete Marked (Удалить помеченные). Точно так же это можно сделать для пересекающихся треугольников в разделе Triangles → Detect Intersecting (Треугольники → Определить пересекающиеся). Если сшивание оставшихся зазоров не завершает восстановление, кнопка Create позволяет заполнить оставшиеся пробелы вручную. Разделы сетки

также можно пометить вручную на вкладке Marking (Маркировка) в главном меню.

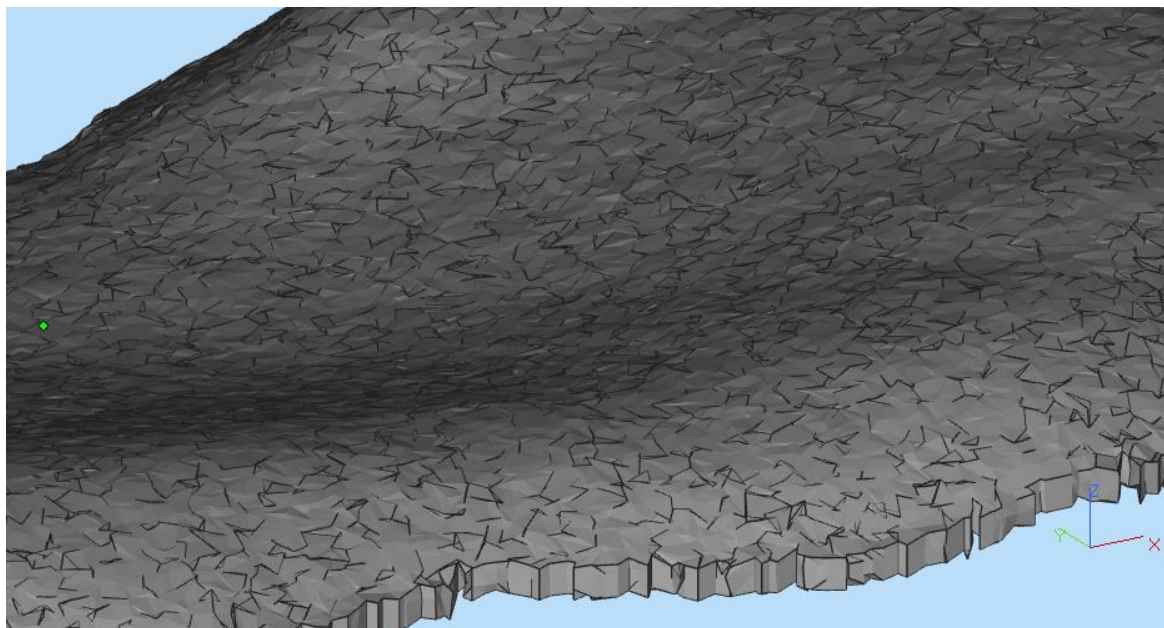


Рис. 3.1.4 Процесс сглаживания треугольников

Плавающие секции могут быть удалены под секцией Noise Shells. Для больших неплоских отверстий их заполнение вручную с помощью параметра Freeform (Свободная форма) в разделе Holes (Отверстия) в меню Fix Wizard (Мастер исправлений) приведет к получению самого гладкого участка заполнения. Опция Ruled позволяет указать направление, по которому должна следовать дырка и в этом случае она используется для преодоления цилиндрического зазора в одной из стержней после создания вручную нескольких соединительных треугольников. Иногда Fix Wizard не объединяет различные оболочки в сетке. Чтобы это исправить, Part Pages → Part List (Страницы деталей → Список деталей) на главном экране и выберите Shells to Parts (Оболочки для деталей). Это создает отдельные сетки, которые при проверке теперь могут быть объединены с помощью функции Tools → Boolean.

3.2. Проектирование стельки

3.2.1. Статический анализ ступней до использования стелек

Для определения эффективности реализованной индивидуальной ортопедической стельки я воспользовалась услугами клиники для проведения статической анализа ступней [5]. Использование плантографии в программно-аппаратных комплексах, с расчетом плантографических индексов стопы, позволяет объективно оценить деформацию стопы, эффективность лечения и отдаленные

результаты ортопедического обеспечения. Подоскан – устройство, которое позволяет получить изображение «отпечатков» ступней на плоской поверхности в режиме статической нагрузки. Анализ позволяет оценить позиционную установку стоп с выявлением отклонений от нормы и опорно-двигательного аппарата в целом [11].

Все системы можно разделить по принципу получения подошвенного давления: подоскопы, изменение интенсивности света, гидроячейки, тензометрические, резистивные и емкостные датчики. В системах для измерения подошвенных давлений используются емкостные датчики, объединённые в измерительную матрицу. Принцип работы емкостных датчиков основан на изменении ёмкости при воздействии внешних механических сил.

Визуально видно (см. рис. 3.2.1.1), что на статическом анализе «не хватает пальцев», что говорит о неправильном распределении веса на каждую ногу: 25 кг на левую ногу и 30 кг на правую [18].

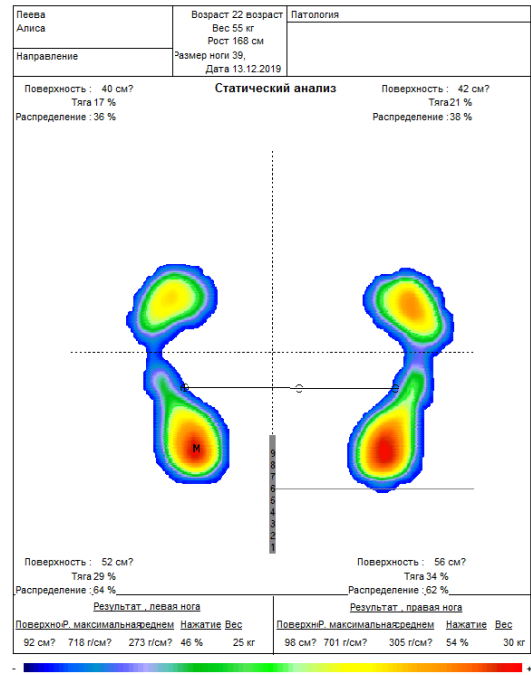


Рис. 3.2.1.1 Статический анализ ступней до использования стелек (декабрь)

3.2.2. Gensole

Используя удобный и бесплатный инструмент Gensole от Gyrobot, созданный Стивом Вудом - британский инженер-механик с опытом работы в автомобильной, аэрокосмической и атомной промышленности, а также эксперт в области гибкой FDM печати, удалось спроектировать стельки прямо в своем браузере. Сам процесс включает в себя 4 этапа:

- 1) создание общей формы, используя данные сканирования;
- 2) определение плотности для конкретного участка в зависимости от давления;
- 3) выбор открытой или закрытой конструкции, оптимизирующей поток воздуха;
- 4) профилирование под конкретную обувь.

Сгруппированные вместе параметры значительно облегчают производство (простое изменение цифровой модели позволяет ввести корректирующие свойства), помогая клиентам получить стельки, которые идеально подходят для их ног.

Заполняем имеющиеся параметры клиента (рис. 3.2.2.1): вес – 55 кг, имя клиента, левая нога.

The screenshot shows a software window titled 'Patient' with a green header. At the top, there are four numbered tabs: 1 (selected), 2, 3, and 4. The main area contains the following fields and controls:

- 'Load an insole project (zip):' with a button labeled 'Выберите файл' and a text label 'Файл...ран'.
- 'Patient name (or identifier)' with a text input field containing 'Alice'.
- 'Bodyweight (kg)' with a text input field containing '55'.
- 'Which foot?' with two buttons: 'Left' and 'Right'.

Рис. 3.2.2.1 Интерфейс программы Gensole

Создаем общую форму стелек (рис. 3.2.2.2) в программе по параметрам стопы, с учетом размера и формы обуви, в данном случае: закрытый тип, без каблука (с небольшой колодкой).

Outside Width - 48 мм; Toe Radius – 40 мм; Inside Width – 47 мм; Length – 248 мм; Inside Length – 47 мм; Outside Length – 155 мм; Instep Offset – 29 мм; Outstep Offset – 39 мм; Heel Radius – 30 мм; Inside Length – 192 мм; Variable Density Thickness – 30 мм; Chamfer Height – 10 мм; Heel Thickness – 10 мм; Draft Angle – 20 градусов.

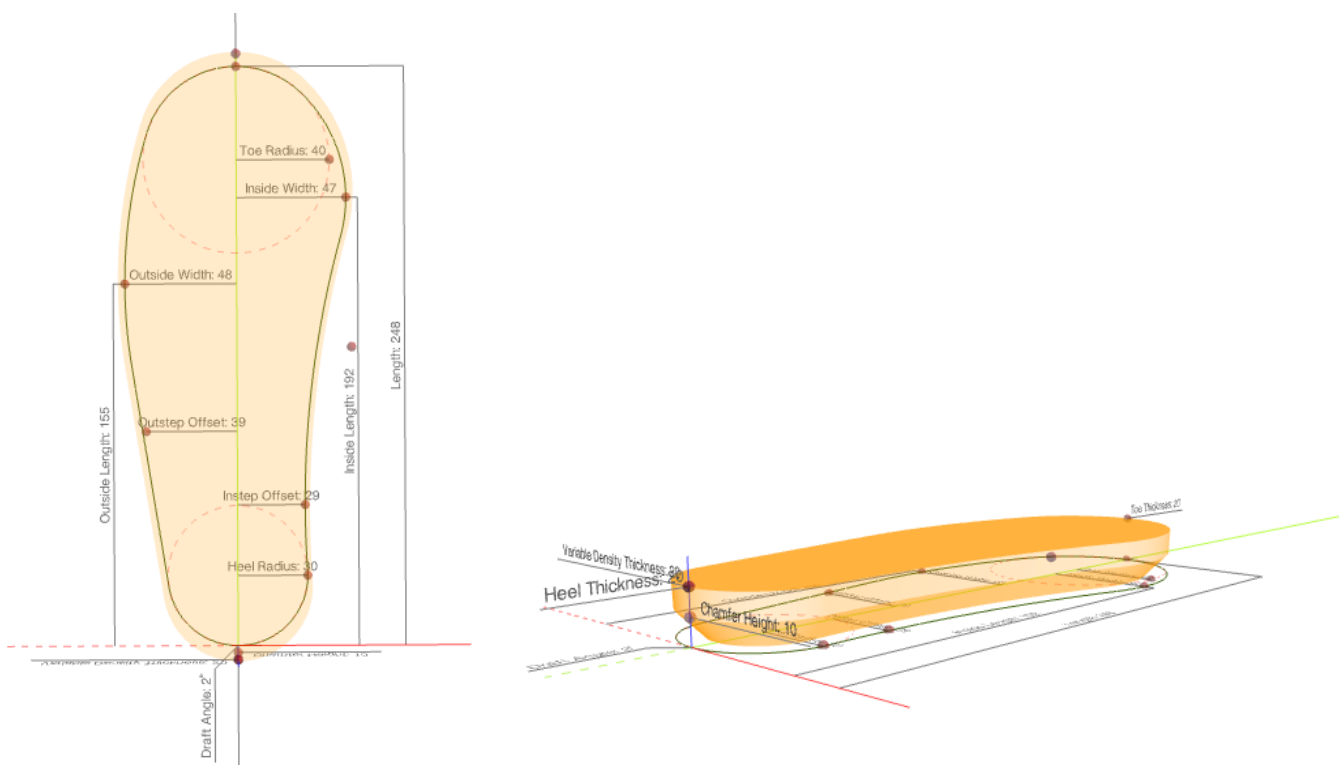


Рис. 3.2.2.2 Создание общей формы стельки

Поместив скан стопы (рис. 3.2.2.3) на модель стельки программа автоматически просчитывает плотность для конкретного участка в зависимости от давления, но мы можем изменить данные вручную, в зависимости от запроса:

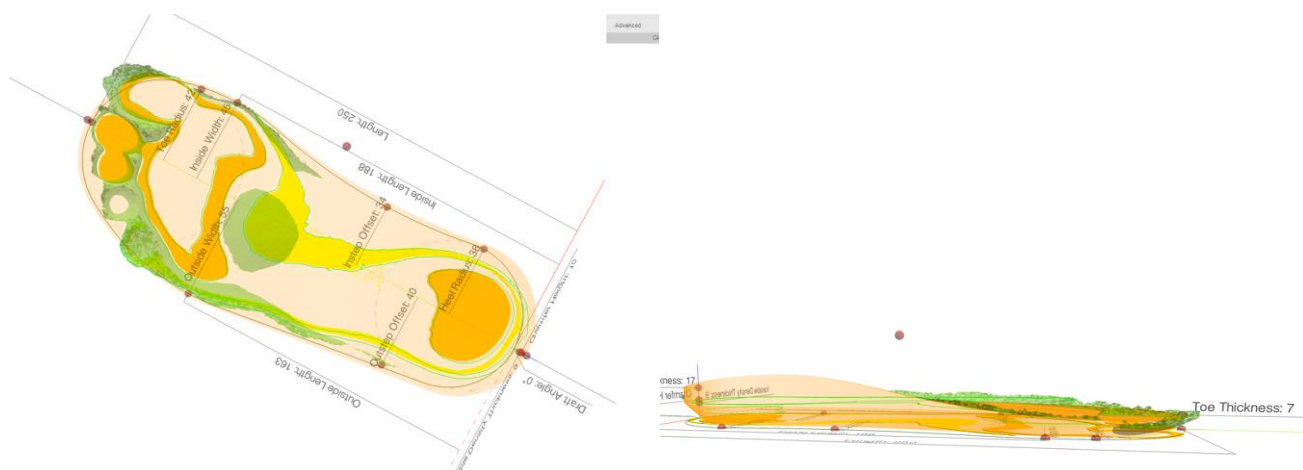


Рис. 3.2.2.3 Размещение скана ступни на модель стельки

Готовая модель индивидуальной ортопедической стельки (рис. 3.2.2.4):

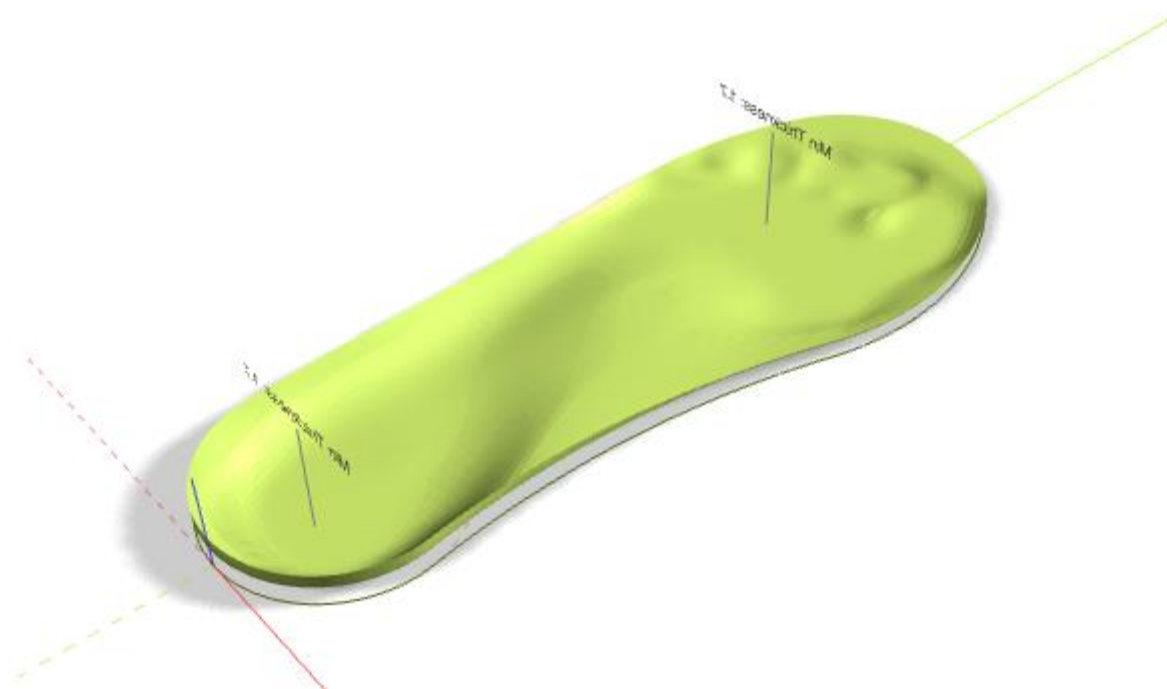


Рис. 3.2.2.4 Итоговая модель индивидуально ортопедической стельки в Gensole

3.3. Материал для 3D печати

Основными критериями в процессе подбора пластиков являются такие характеристики как: гибкость, прочность, стойкость, токсичность

Изучив данные от производителей различных пластиков [14]. Для изготовления стелек был выбран ТПУ пластик в виду его гибкости, прочности, износостойкости и не токсичности. Ближайшие конкуренты ТПУ - это термопластичные полиэфирные эластомеры, ТПЭ (ТРЕ). Из плюсов ТПЭ можно отметить более низкую стоимость, очень низкую гигроскопичность. Но ТПЭ часто расплываются при печати мелких деталей (нужен увеличивать время печати слоя), как правило у ТПЭ хуже адгезия к столу, и несравнимо гораздо меньшая эластичность - способность восстанавливать форму при деформациях. При одинаковой мягкости, ТПУ будет печататься проще, чем ТРЕ - небольшие изгибы ТПУ в экструдере будут стремиться выправиться, в то время как ТРЕ может необратимо замяться и печать будет прервана [13].

3.3.1. Термопластичный полиуретан

TPU (термопластичный полиуретан) - это гибкий, стойкий к истиранию термопластик. Является весьма распространенной формой упругого полимера, или термопластичного эластомера (TPE). Представляет собой смесь твердого пластика и резины. Материал, который очень близок к настоящей резине, повторяя все её свойства.

В традиционной промышленности это довольно распространенный материал, который применяется для изготовления широкого спектра различных изделий. В 3D печать ТПУ пластик пришел относительно недавно, и еще не успел обрести достаточную популярность. Тем не менее, по своим физическим и механическим свойствам этот полимер очень интересен. Он довольно прочен при высокой гибкости, а при охлаждении практически не дает усадки.

Несмотря на то, что ТПУ приспособлен к работе практически со всеми настольными 3D принтерами, перед его использованием рекомендую ознакомиться с особенностями печати гибкими материалами. Работа с такими материалами может вызывать сложности у новичков, потому во избежание напрасного расхода материала желательно знать обо всех тонкостях заранее. Из недостатков полимера можно отметить сложность в постобработке (он практически не поддается шлифовке, окраске и т.д.).

Табл. 3.3.1.1 Общие характеристики ТПУ материала

Общие характеристики	
Плотность	0,95 г/см ³
Температура эксплуатации	от -35°C до +85°C
Температура размягчения	н/д
Страна производитель	Россия

Табл. 3.3.1.2 Механические характеристики ТПУ материала

Механические характеристики	
Ударная вязкость по Изоду	25 кДж/м ²
Прочность при растяжении вдоль слоев	19,7 Мпа
Модуль упругости при растяжении вдоль слоев	н/д
Прочность на изгиб	3,4 Мпа
Модуль упругости на изгиб	н/д
Максимальная нагрузка на изгиб	5 Н

Продолжение таблицы 3.3.1.2

Прочность при растяжении поперек слоев	н/д
Модуль упругости при растяжении поперек слоев	н/д
Максимальная нагрузка на растяжение	365 Н
Прочность на сжатие	2,3 МПа
Модуль упругости на сжатие	62,3 МПа
Максимальная нагрузка на сжатие	287 Н
Коэффициент удлинения	500%
Биоразлагаемость	н/д
Диэлектрическая проницаемость	$1,65 \times 10^{15}$ Ом/см
Предел текучести при растяжении при температуре 23°C	н/д
Прочность при изгибе 2,8 мм/мин. 23°C	н/д
Твердость по Шору (шкала А)	95
Масло - бензостойкость (максимальное изменение формы за 24 часа)	4,5%
Кислородный индекс, %O ₂ по ГОСТ 21793-76	17.5-17.8%
Массовая доля золы по ГОСТ 15973	менее 0,4%

3.3.2. Деформация и давление на модель стельки

Материал ТПУ обладает нелинейными свойствами и большим числом параметров, влияющих на напряженно-деформированное состояние. В отличие от физико-механических свойств металлов, которые имеют 2 параметра, позволяющие описать НДС по линейному закону, что даст нам оценку износостойкости. Ортопедические стельки из ТПУ носятся в сложных динамических условиях. Стопа совершает разнообразные движения при стоянии, ходьбе, беге с различной амплитудой и частотой. Процесс эксплуатации ортеза сопровождается сложным комплексом повторных механических действий между стопой, стелькой и обувью.

Для исследования деформации стельки воспользовались программой ANSYS , воспроизведя поведение свойств материала [1]. Для анализа был выбран материал

– ТПУ, данные о материале предоставил производитель. Высота образца 20 ± 2 мм. На верхнюю поверхность была приложена половина нагрузки – 150 Н, и в свойства модели были заложены графики поведения материалов, полученные из результатов эксперимента на сжатие, запрошенные у производителя (рис.3.3.2). Видно, что при нагрузке до 30 кг ТПУ ведет себя по закону линейной упругости.

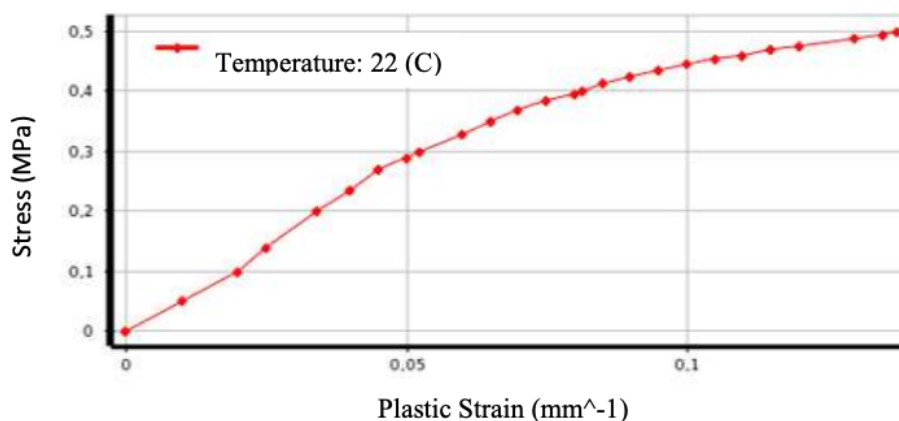


Рис. 3.3.2 Результаты эксперимента на сжатие

Основным свойством ортопедической стельки является перераспределение давления на стопу за счет 2 факторов: эластичных свойств материала (амортизация) и геометрических параметров стельки (выкладка свода и др. конструктивов). Известно [20], что конструкционный материал ортеза при ходьбе испытывает нагрузки, как на сжатие, так и на удар, что приводит к деформации стельки и изменению геометрии, особенно элемента, поддерживающего продольный свод.

Важным параметром исследования деформации стельки является определение максимальной силы нагружения стопы. Этот параметр был исследован Кевином Кёрби [19]. Он проводил на Он предложил проводить на пациентах клинический тест «сопротивление супинации». Что позволяет получить количественную оценку силы, которая должна воздействовать на свод для сохранения стопы в нейтральном (корректированном) положении. В среднем сила может варьироваться от 60 до 300 Н, поэтому была принята нагрузка 300 Н. По нижней поверхности модели стельки задаем опору. Ограничиваем перемещение модели по осям Z и Y.

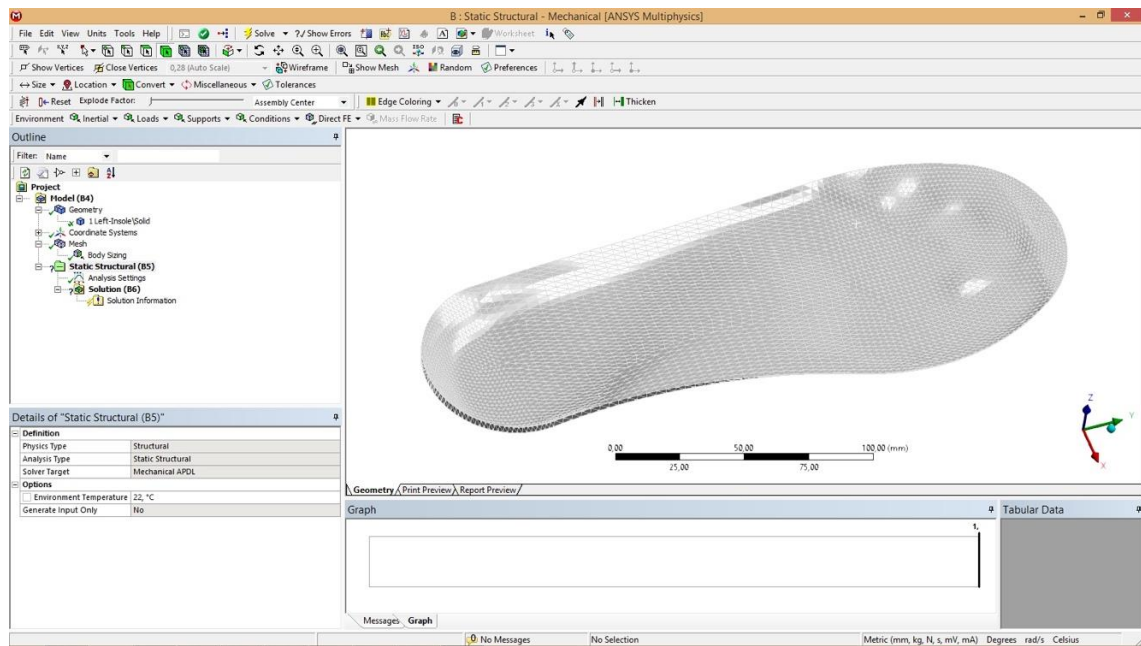


Рис. 3.3.3 Интерфейс программы ANSYS

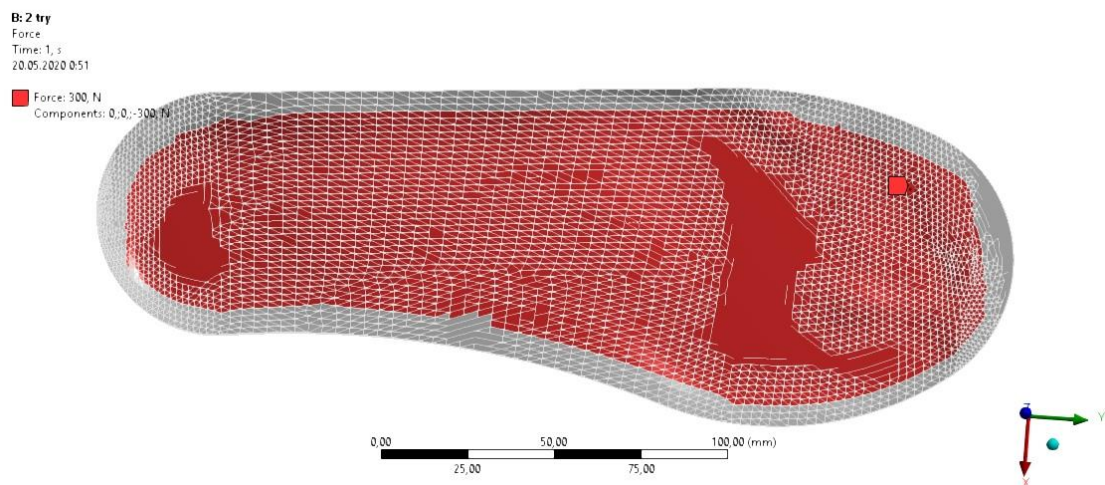


Рис. 3.3.4 Прикладываемая сила

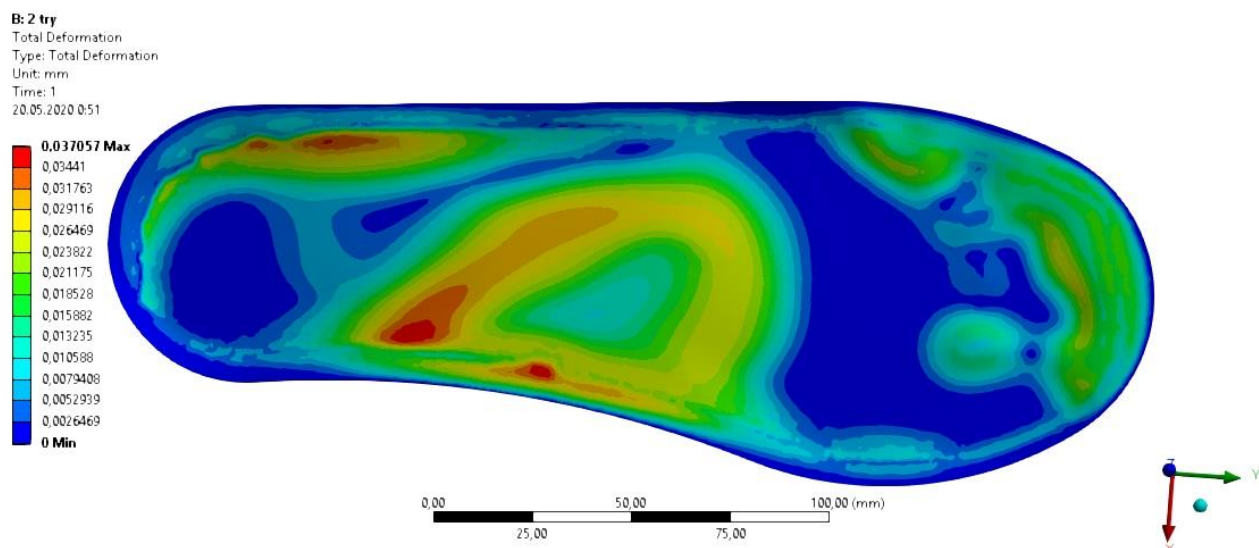


Рис. 3.3.5 Деформация

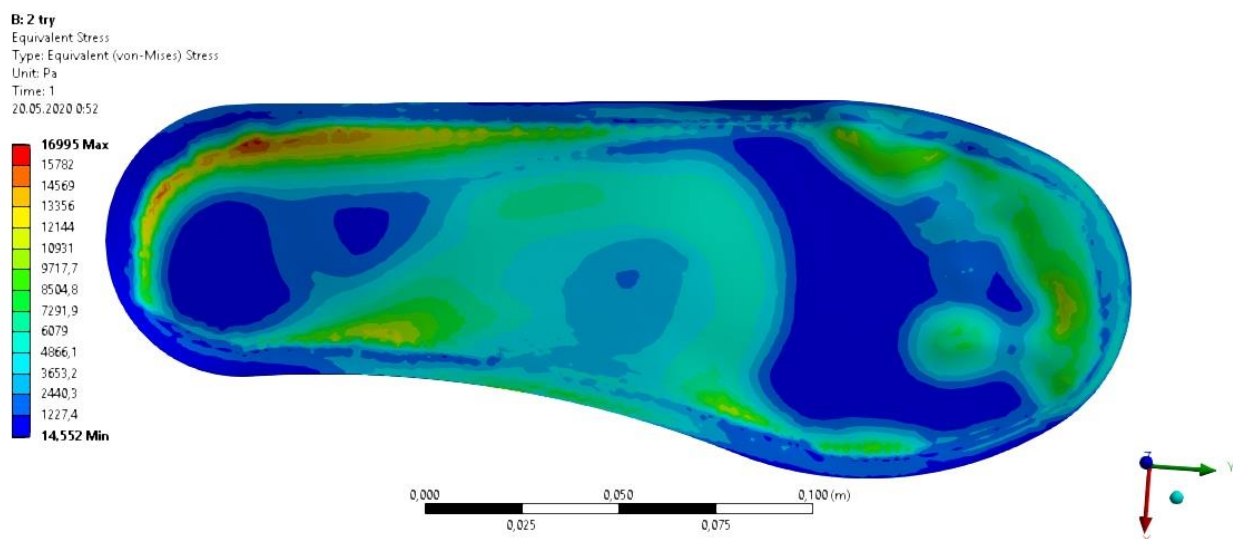


Рис. 3.3.6 Давление

Из рисунков 3.3.5 и 3.3.6 видно, что максимальная деформация равна 0,037 мм, а максимальное давление – 16 995 Па, соответственно. Выбранный материал подходит для изготовления стелек и прослужит долгое время [2].

Глава 4. Изготовление индивидуальных ортопедических стелек с использованием 3D печати

Изготовление опытного образца планировалось выполнить на базе лаборатории цифрового производства Фаблаб Политех, к сожалению, в условиях карантина мне пришлось обратиться к индивидуальному предпринимателю, покупка материалов и

аренда оборудования осуществлялась за счёт моих денежных средств. Но для дальнейшей реализации, планируется использовать иные способы печати, поэтому в работе производится обзор оборудования и материалов, с помощью которых можно производить данный продукт.

4.1. Обзор оборудования

Рассмотрим стандартный FDM принтер на примере 3D-принтера DF-Print (рис. 4.1):

- Технология печати: FDM
- Рабочая область: 300x300x300 мм
- Диапазон рабочих температур экструдера: 120-300 С°
- Максимальная температура нагревательной платформы: 120 С°
- Материалы печати: ABS, PLA, SBS, PETG, Нейлон и т.д.

Технология FDM печати («Fused Deposition Modeling») - заключается в последовательном наплавлении тонких пластиковых нитей друг на друга [8]. Технология FDM подразумевает создание трёхмерных объектов за счёт нанесения последовательных слоёв материала, повторяющих контуры цифровой модели. Как правило, в качестве материалов для печати выступают термопластики, поставляемые в виде катушек нитей или прутков. 3D принтер DF Print - FDM 3D принтер, выполненный по кинематической схеме H-bot. Станок оснащен подогреваемой платформой мощностью 600 Вт, обеспечивающей возможность печати ABS пластиком, нейлоном. Печатающая головка выполнена по Direct системе с механизмом подачи, расположенным непосредственно на портале станка. В принтере используется хотенд E3D v6, доказавший свою надежность и позволяющий печатать при температурах до 300°С. Печатающая головка оснащена турбо-кулером для обдува зоны печати, что необходимо для достижения высокой точности изготавливаемых деталей.



Рис. 4.1 3D принтер DF-Print

4.2. Процесс изготовления

Для формирования управляющих программ использует программное обеспечение Cura (рис. 4.2.1). Предварительный расчет времени печати модели массой 230 грамм показал 15 часов (без учета времени на разгон и торможение печатающей головки).

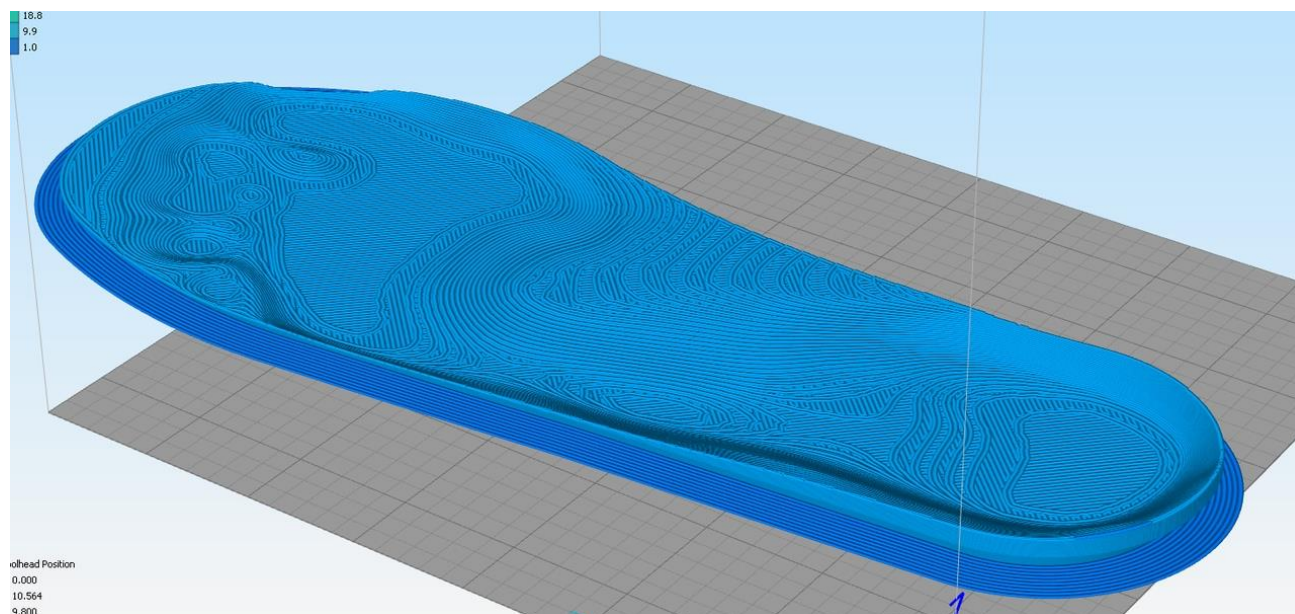


Рисунок 4.2.1. Интерфейс программы Cura с настройками печати

Табл. 3.3.1.1 Общие настройки для печати ТПУ материала

Настройки для печати	
Температура сопла	225-245°C
Температура стола	90-110°C
Обдув	не рекомендуется
Рекомендуемый адгезив	Клей The3D
Мин. диаметр сопла	0.4 мм



Рис. 4.2.2 Процесс изготовления стелек на 3D принтере

Результат печати приведен на рис. 4.2.3:



Рис. 4.2.3 Изготовленная пара стелек на 3D принтер

Глава 5. Расчет себестоимости изготовления индивидуальных стелек

Используя программу Microsoft Excel, были выполнены расчетные калькуляции себестоимости услуг по проектированию и изготовлению одной пары индивидуальных стелек по методике [6]. Все расчетные данные занесены в таблицы 5.1-5.2.

Таблица 5.1 Затраты на проектирование индивидуальных стелек

№	Наименование статей расходов	Расходы, руб/мес.
1	Оплата труда	30 000,00
2	Начисление на оплату труда 30,2 %	9 060,00
Итого		39 060,00
3	Амортизация	
	Ноутбук Acer Aspire – 40 000,00 руб./36 мес. (3 года)=1 111,11	1 111,11
4	Услуги связи	
	Интернет	1 000,00
5	Коммунальные услуги	815,53
	1. Водоснабжение – 145,24 руб. на 1 чел. (водопотребление х/в – 20,47 руб/куб. м; водоотведение – 8,71 руб/куб. м; гор. водоснабжение – 116,06 руб/куб. м)	
	На 1 человека – 1,25 куб. м/мес.*1*145,24 руб/куб. м=181,55	181,55
	2. Отопление 1 кв. м – 35,08 руб.	
	12 кв. м*35,08 руб.=420,96 руб.	420,96
	3. Электроэнергия	
	Лампы: 0,04 кВт/ч*8 шт.*8 ч*21 день=53,76 кВт	
	53,76 кВт*3,17 руб.=170,42 руб.	170,42
6	Расходы на содержание оборудования	21,30
	Ноутбук Acer Aspire – 0,04 кВт/ч*1*8 ч*21 день=6,72 кВт	
	6,72 кВт*3,17 руб.=21,30 руб.	21,30
7	Накладные расходы 38 %	15 963,02
8	Прибыль 10 %	4 200,79
9	Всего	62 171,75

Таблица 5.2 Затраты на изготовление индивидуальных стелек

№	Наименование статей расходов	Расходы, руб/мес.
1	Оплата труда (2 чел.)	60 000,00
2	Начисление на оплату труда 30,2 %	18 120,00
	Итого	78 120,00
3	Материалы	
	Пластик для 3D принтера	23 000,00
	Итого	23 000,00
4	Амортизация	
	3D принтер – 200 000,00 руб./84 мес. (7 лет)=2380,90	2 380,90
	Ноутбук Acer Aspire – 40 000,00 руб./36 мес. (3 года)=1 111,11	1 111,11
	Итого	3 492,01
5	Услуги связи	
	Интернет	1 000,00
	Итого	1 000,00
6	Коммунальные услуги	
	1. Водоснабжение – 145,24 руб. на 1 чел. (водопотребление х/в – 20,47 руб/куб. м; водоотведение – 8,71 руб/куб. м; гор. водоснабжение – 116,06 руб/куб. м)	
	На 2 человека – 1,25 куб. м/мес.*2*145,24 руб/куб. м=363,10 руб.	363,10
	2. Отопление 1 кв. м – 35,08 руб.	2 315,28
	54 кв. м*35,08 руб.=1 894,32 руб.	1 894,32
	12 кв. м*35,08 руб.=420,96 руб.	420,96
	3. Электроэнергия	
	Лампы: 0,04 кВт/ч*48 шт.*8 ч*21 день=322,56 кВт	
	322,56 кВт*3,17 руб.=1 022,52 руб.	1 022,52
	Итого	3 700,90
7	Расходы на содержание оборудования	
	Ноутбук Acer Aspire – 0,04 кВт/ч*1*8 ч*21 день=6,72 кВт	
	6,72 кВт*3,17 руб.=21,30 руб.	21,30
	ПК Intel Core i7-4770 – 0,15 кВт/ч*1*8 ч*21 день=25,20 кВт	
	25,20кВт*3,17=79,88 руб.	79,88
	3D принтер – 3 кВт/ч*1*24 ч*21 день=1512 кВт	
	1512 кВт*3,17=1 597,68 руб.	4 793,04
	Итого	4 894,22
8	Накладные расходы 38 %	43 398,71
9	Прибыль 10 %	11 420,71
	Итого	54 819,42
10	Всего	164 132,33

Расчет времени работы по проектированию: при 8-часовом рабочем дне - 2 000 раб. ч в год. Себестоимость 1 ч – 373,03 руб. Экспериментальное время проектирования одной пары индивидуальных стелек составило 2 ч. Расчет времени работы по изготовлению: себестоимость 1 ч – 984,79 руб. Экспериментальное время изготовления 1 пары индивидуальных стелек составило 1 ч.

Суммарная цена на одну пару индивидуальных стелек составляет 2 000 руб. Прибыль составляет 20 % от цены. Проведя маркетинговое исследование (средняя цена производства стелек для взрослых 7 000 рублей) и изучив все доступные предложения по изготовлению стелек на рынке, можно с уверенностью заявить, что разработанная технология создания индивидуальных ортопедических стелек является конкурентоспособной и позволит клиентам, которых в г. Санкт-Петербурге около 5 млн., обоснованно выбрать предприятие-изготовителя по более низкой цене.

Глава 6. Выводы

Эксплуатация стелек длилась 5 месяцев с декабря по апрель. Посетив клинику для создания статического анализа ступней, было замечено следующее: распределение веса стало равномерным по 29, 28 килограмм на левую и правую ногу соответственно; визуально появились «пальцы», чего не было ранее. А также уменьшились боли в спине, так как удалось снизить нагрузку на позвоночник.

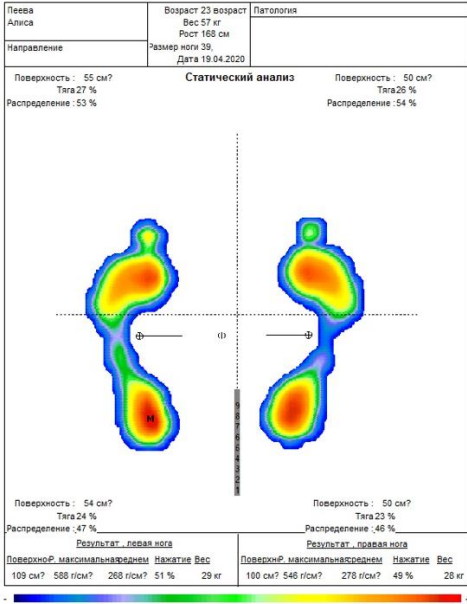


Рис. 6.1 Статический анализ ступней после использования стелек

Данный анализ показывает нам эффективность предложенной технологии для создания индивидуальных ортопедических стелек с использованием 3D печати.

Заключение

В настоящей работе был разработан полный цикл создания и реализации конечного продукта: проведен анализ существующих технологий изготовления индивидуальных ортопедических стелек, материалов и наиболее распространенных конструкций ортезов; внедрены в процессы проектирования современные методы компьютерного моделирования и фотограмметрии; выполнен анализ возможности использования материала на основе ТПУ с необходимостью подбора твердости в соответствии с весовой категорией пациента; проведено оценочное моделирование напряженно-деформированное состояния стельки из ТПУ при силе давления 300 Н; произведены расчетные калькуляции себестоимости услуг по проектированию и изготовлению одной пары индивидуальных стелек; доказана эффективность предложенного метода технологии создания индивидуальных ортопедических стелек с использованием 3d печати, путём статического анализа ступней.

Список использованных источников

- [1]. Басов К.А. ANSYS в примерах и задачах. М.: КомпьютерПресс, 2002. 224 с. (дата обращения: 04.06.2020)
- [2]. Борисов В.В. Клинико-экспериментальное исследование материала для изготовления защитных зубных шин на основе этиленвинилацетата методом термоформирования: автореф. дис канд. мед. наук. М., 2017. 24 с. (дата обращения: 04.06.2020)
- [3]. Виды 3D–моделирования. [Электронный ресурс] // URL: <https://koloro.ua/blog/3d-tekhnologii/vidy-3d-modelirovaniya-poligonalnoesplajnovoe-i-nurbs-modelirovanie.html> (дата обращения: 03.06.2020)
- [4]. Виды ортопедических стелек [Электронный ресурс] // URL: <https://ctelki.ru/info/articles/vidy-ortopedicheskikh-stelek/> (дата обращения 01.06.2020)
- [5]. Корнилов Н.В. Травматология и ортопедия: Руководство для врачей / В 4 томах. СПб.: Гиппократ, 2004 – Т. 3: Травмы и заболевания нижней конечности / Под ред. Н.В. Корнилова и Э.Г. Грязнухина.– СПб.: Гиппократ, 2006.– Гл.8.– С.566–582. (дата обращения 03.06.2020)
- [6]. Мельник Е.А. Экономика предприятия. М.: Юрайт, 2013. 303 с. (дата обращения: 05.06.2020)

- [7]. Новинки медтехники. Подоскоп // ЛФК и массаж. – 20- 03. - № 4(7). – С. 44-53. (дата обращения: 03.06.2020)
- [8]. Описание 3D принтера DF-Print [Электронный ресурс] // URL: <https://3dtool.ru/product/3d-printer-df-print/> (дата обращения: 05.06.2020)
- [9]. Практическое руководство по освоению программы Solidworks [Электронный ресурс] // URL: <https://autocad-lessons.ru/solidworks/> (дата обращения 02.06.2020)
- [10]. Программа MeshLab [Электронный ресурс] // URL: <http://meshlab.sourceforge.net/> (дата обращения 03.06.2020)
- [11]. Свойства диоксида кремния [Электронный ресурс] // URL: http://wiki.web.ru/wiki/Диоксид_кремния (дата обращения 02.06.2020)
- [12]. Скан Usol equipment information [Электронный ресурс] // URL: <http://scanpod3d.com/> (дата обращения 02.06.2020)
- [13]. Сравнение ABS, PLA, SBS, PETG. Различие пластиков: свойства, хранение, применение. [Электронный ресурс] // URL: <http://rusabs.ru/blogs/blog/razlichie-mezhdu-abs-i-pla-dlya-3d-pechati> (дата обращения: 04.06.2020)
- [14]. Сравнение всех материалов REC (для профессионалов). [Электронный ресурс] // URL: <https://rec3d.ru/wiki/materialy/sravnenie-vsekhmaterialov-rec-dlya-professionalov> (дата обращения: 04.06.2020)
- [15]. Сравнение различных технологий изготовления индивидуальных стелек [Электронный ресурс] // URL: <https://www.zdravohod.ru/articles/sravnenie-individualnyh-stelek> (дата обращения 01.06.2020)
- [16]. Стратегия развития информационного общества в Российской Федерации от 7 февраля 2008 г. N Пр-212. (дата обращения 01.06.2020)
- [17]. Услуги и продажа оборудования в сфере 3D печати и сканирования [Электронный ресурс] // URL: <http://3dboom.su/> (дата обращения 02.06.2020)
- [18]. Яременко Д.А. Методика исследования, диагностика и ортопедическое снабжение при статических деформациях стоп. – Харьков, 1984. – 45с. (дата обращения: 03.06.2020)
- [19]. Anggoro P.W., Saputra E., Tauviqirrahman M., Jamari J., Bayuseno A.P. A 3-dimensional finite element analysis of the insole shoe orthotic for foot deformities // International Journal of Applied Engineering Research. 2017. Vol. 12. № 15. P. 5254–5260. (дата обращения: 05.06.2020)

[20]. Taha Z., Norman M.S., Omar S.F.S., Suwarganda E. A finite element analysis of a human foot model to simulate neutral standing on ground // Procedia Engineering. 2016. Vol. 147. P. 240–245. (дата обращения: 04.06.2020)

Приложение 1. Код использования Meshroom

```
import sys, os
import shutil
```

```
def SilentMkdir(theDir):
```

```
    try:
        os.mkdir(theDir)
    except:
        pass
    return 0
```

```
def Run_00_CameraInit(baseDir, binDir, srcImageDir):
```

```
    SilentMkdir(baseDir + "/00_CameraInit")
```

```
    binName = binDir + "\\aliceVision_cameraInit.exe"
```

```
    dstDir = baseDir + "/00_CameraInit/"
```

```
    cmdLine = binName
```

```
    cmdLine = cmdLine + " --defaultFieldOfView 45.0 --verboseLevel info --sensor-  
Database \"\" --allowSingleView 1"
```

```
    cmdLine = cmdLine + " --imageFolder \"\" + srcImageDir + "\""
```

```
    cmdLine = cmdLine + " --output \"\" + dstDir + "cameraInit.sfm\""
```

```
    print(cmdLine)
```

```
    os.system(cmdLine)
```

```
    return 0
```

```
def Run_01_FeatureExtraction(baseDir, binDir, numImages):
```

```
    SilentMkdir(baseDir + "/01_FeatureExtraction")
```

```
    srcSfm = baseDir + "/00_CameraInit/cameraInit.sfm"
```

```
    binName = binDir + "\\aliceVision_featureExtraction.exe"
```

```
    dstDir = baseDir + "/01_FeatureExtraction/"
```

```
    cmdLine = binName
```

```
    cmdLine = cmdLine + " --describerTypes sift --forceCpuExtraction True --verbose-  
Level info --describerPreset normal"
```

```

cmdLine = cmdLine + " --rangeStart 0 --rangeSize " + str(numImages)
cmdLine = cmdLine + " --input \"" + srcSfm + "\""
cmdLine = cmdLine + " --output \"" + dstDir + "\""
print(cmdLine)
os.system(cmdLine)

```

```

return 0

```

```

def Run_02_ImageMatching(baseDir,binDir):

```

```

    SilentMkdir(baseDir + "/02_ImageMatching")

```

```

    srcSfm = baseDir + "/00_CameraInit/cameraInit.sfm"

```

```

    srcFeatures = baseDir + "/01_FeatureExtraction/"

```

```

    dstMatches = baseDir + "/02_ImageMatching/imageMatches.txt"

```

```

    binName = binDir + "\\aliceVision_imageMatching.exe"

```

```

    cmdLine = binName

```

```

    cmdLine = cmdLine + " --minNbImages 200 --tree "" --maxDescriptors 500 --ver-
boseLevel info --weights "" --nbMatches 50"

```

```

    cmdLine = cmdLine + " --input \"" + srcSfm + "\""

```

```

    cmdLine = cmdLine + " --featuresFolder \"" + srcFeatures + "\""

```

```

    cmdLine = cmdLine + " --output \"" + dstMatches + "\""

```

```

    print(cmdLine)

```

```

    os.system(cmdLine)

```

```

return 0

```

```

def Run_03_FeatureMatching(baseDir,binDir):

```

```

    SilentMkdir(baseDir + "/03_FeatureMatching")

```

```

    srcSfm = baseDir + "/00_CameraInit/cameraInit.sfm"

```

```

    srcFeatures = baseDir + "/01_FeatureExtraction/"

```

```

    srcImageMatches = baseDir + "/02_ImageMatching/imageMatches.txt"

```

```

    dstMatches = baseDir + "/03_FeatureMatching"

```

```

    binName = binDir + "\\aliceVision_featureMatching.exe"

```

```

    cmdLine = binName

```

```

    cmdLine = cmdLine + " --verboseLevel info --describerTypes sift --maxMatches 0 -
-exportDebugFiles False --savePutativeMatches False --guidedMatching False"

```

```

    cmdLine = cmdLine + " --geometricEstimator acransac --geometricFilterType fun-
damental_matrix --maxIteration 2048 --distanceRatio 0.8"

```

```

    cmdLine = cmdLine + " --photometricMatchingMethod ANN_L2"

```

```

cmdLine = cmdLine + " --imagePairsList \"" + srcImageMatches + "\""
cmdLine = cmdLine + " --input \"" + srcSfm + "\""
cmdLine = cmdLine + " --featuresFolders \"" + srcFeatures + "\""
cmdLine = cmdLine + " --output \"" + dstMatches + "\""

```

```

print(cmdLine)
os.system(cmdLine)
return 0

```

```

def Run_04_StructureFromMotion(baseDir,binDir):

```

```

    SilentMkdir(baseDir + "/04_StructureFromMotion")

```

```

    srcSfm = baseDir + "/00_CameraInit/cameraInit.sfm"
    srcFeatures = baseDir + "/01_FeatureExtraction/"
    srcImageMatches = baseDir + "/02_ImageMatching/imageMatches.txt"
    srcMatches = baseDir + "/03_FeatureMatching"
    dstDir = baseDir + "/04_StructureFromMotion"

```

```

    binName = binDir + "\\aliceVision_incrementalSfm.exe"

```

```

    cmdLine = binName
    cmdLine = cmdLine + " --minAngleForLandmark 2.0 --minNumberOfObservationsForTriangulation 2 --maxAngleInitialPair 40.0 --maxNumberOfMatches 0 --localizerEstimator acransac --describerTypes sift --lockScenePreviouslyReconstructed False --localBAGraphDistance 1"
    cmdLine = cmdLine + " --initialPairA "" --initialPairB "" --interFileExtension .ply --useLocalBA True"
    cmdLine = cmdLine + " --minInputTrackLength 2 --useOnlyMatchesFromInputFolder False --verboseLevel info --minAngleForTriangulation 3.0 --maxReprojectionError 4.0 --minAngleInitialPair 5.0"

```

```

    cmdLine = cmdLine + " --input \"" + srcSfm + "\""
    cmdLine = cmdLine + " --featuresFolders \"" + srcFeatures + "\""
    cmdLine = cmdLine + " --matchesFolders \"" + srcMatches + "\""
    cmdLine = cmdLine + " --outputViewsAndPoses \"" + dstDir + "/cameras.sfm\""
    cmdLine = cmdLine + " --extraInfoFolder \"" + dstDir + "\""
    cmdLine = cmdLine + " --output \"" + dstDir + "/bundle.sfm\""

```

```

print(cmdLine)
os.system(cmdLine)
return 0

```

```

def Run_05_PrepareDenseScene(baseDir,binDir):

```

```

    SilentMkdir(baseDir + "/05_PrepareDenseScene")

```

```
#srcSfm = baseDir + "/04_StructureFromMotion/cameras.sfm"
srcSfm = baseDir + "/04_StructureFromMotion/bundle.sfm"
dstDir = baseDir + "/05_PrepareDenseScene"
```

```
binName = binDir + "\\aliceVision_prepareDenseScene.exe"
```

```
cmdLine = binName
cmdLine = cmdLine + " --verboseLevel info"
cmdLine = cmdLine + " --input \"" + srcSfm + "\""
cmdLine = cmdLine + " --output \"" + dstDir + "\""
```

```
print(cmdLine)
os.system(cmdLine)
return 0
```

```
def Run_06_CameraConnection(baseDir,binDir):
    SilentMkdir(baseDir + "/06_CameraConnection")
```

```
srcIni = baseDir + "/05_PrepareDenseScene/mvs.ini"
```

```
# This step kindof breaks the directory structure. Tt creates
# a camsPairsMatrixFromSeeds.bin file in in the same file as mvs.ini
binName = binDir + "\\aliceVision_cameraConnection.exe"
```

```
cmdLine = binName
cmdLine = cmdLine + " --verboseLevel info"
cmdLine = cmdLine + " --ini \"" + srcIni + "\""
```

```
print(cmdLine)
os.system(cmdLine)
return 0
```

```
def Run_07_DepthMap(baseDir,binDir,numImages,groupSize):
    SilentMkdir(baseDir + "/07_DepthMap")
```

```
numGroups = (numImages + (groupSize-1))/groupSize
```

```
srcIni = baseDir + "/05_PrepareDenseScene/mvs.ini"
binName = binDir + "\\aliceVision_depthMapEstimation.exe"
dstDir = baseDir + "/07_DepthMap"
```

```
cmdLine = binName
cmdLine = cmdLine + " --sgmGammaC 5.5 --sgmWSH 4 --refineGammaP 8.0 --
refineSigma 15 --refineNSamplesHalf 150 --sgmMaxTCams 10 --refineWSH 3 --"
```

```
downscale 2 --refineMaxTCams 6 --verboseLevel info --refineGammaC 15.5 --sgmGammaP 8.0"
```

```
cmdLine = cmdLine + " --refineNriters 100 --refineNDepthsToRefine 31 --refineUseTcOrRcPixSize False"
```

```
cmdLine = cmdLine + " --ini \"" + srcIni + "\""
```

```
cmdLine = cmdLine + " --output \"" + dstDir + "\""
```

```
for groupIter in range(numGroups):
    groupStart = groupSize * groupIter
    groupSize = min(groupSize, numImages - groupStart)
    print("DepthMap Group %d/%d: %d, %d" % (groupIter, numGroups, groupStart, groupSize))

    cmd = cmdLine + (" --rangeStart %d --rangeSize %d" % (groupStart, groupSize))
    print(cmd)
    os.system(cmd)
```

```
#cmd = "aliceVision_depthMapEstimation --sgmGammaC 5.5 --sgmWSH 4 --refineGammaP 8.0 --refineSigma 15 --refineNSamplesHalf 150 --sgmMaxTCams 10 --refineWSH 3 --downscale 2 --refineMaxTCams 6 --verboseLevel info --refineGammaC 15.5 --sgmGammaP 8.0 --ini \"c:/users/geforce/appdata/local/temp/MeshroomCache/PrepareDenseScene/4f0d6d9f9d072ed05337fd7c670811b1daa00e62/mvs.ini\" --refineNriters 100 --refineNDepthsToRefine 31 --refineUseTcOrRcPixSize False --output \"c:/users/geforce/appdata/local/temp/MeshroomCache/DepthMap/18f3bd0a90931bd749b5eda20c8bf9f6dab63af9\" --rangeStart 0 --rangeSize 3"
```

```
#cmd = binName + " --sgmGammaC 5.5 --sgmWSH 4 --refineGammaP 8.0 --refineSigma 15 --refineNSamplesHalf 150 --sgmMaxTCams 10 --refineWSH 3 --downscale 2 --refineMaxTCams 6 --verboseLevel info --refineGammaC 15.5 --sgmGammaP 8.0 --ini \"c:/users/geforce/appdata/local/temp/MeshroomCache/PrepareDenseScene/4f0d6d9f9d072ed05337fd7c670811b1daa00e62/mvs.ini\" --refineNriters 100 --refineNDepthsToRefine 31 --refineUseTcOrRcPixSize False --output \"build_files/07_DepthMap/\" --rangeStart 0 --rangeSize 3"
```

```
#cmd = binName + " --sgmGammaC 5.5 --sgmWSH 4 --refineGammaP 8.0 --refineSigma 15 --refineNSamplesHalf 150 --sgmMaxTCams 10 --refineWSH 3 --downscale 2 --refineMaxTCams 6 --verboseLevel info --refineGammaC 15.5 --sgmGammaP 8.0 --ini \"" + srcIni + "\" --refineNriters 100 --refineNDepthsToRefine 31 --refineUseTcOrRcPixSize False --output \"build_files/07_DepthMap/\" --rangeStart 0 --rangeSize 3"
```

```
#print(cmd)
```

```
#os.system(cmd)
```



```
return 0
```

```
def Run_08_DepthMapFilter(baseDir,binDir):
```

```
    SilentMkdir(baseDir + "/08_DepthMapFilter")
```

```
    binName = binDir + "\\aliceVision_depthMapFiltering.exe"
```

```
    dstDir = baseDir + "/08_DepthMapFilter"
```

```
    srcIni = baseDir + "/05_PrepareDenseScene/mvs.ini"
```

```
    srcDepthDir = baseDir + "/07_DepthMap"
```

```
    cmdLine = binName
```

```
    cmdLine = cmdLine + " --minNumOfConsistensCamsWithLowSimilarity 4"
```

```
    cmdLine = cmdLine + " --minNumOfConsistensCams 3 --verboseLevel info --
```

```
    pixSizeBall 0"
```

```
    cmdLine = cmdLine + " --pixSizeBallWithLowSimilarity 0 --nNearestCams 10"
```

```
    cmdLine = cmdLine + " --ini \"" + srcIni + "\""
```

```
    cmdLine = cmdLine + " --output \"" + dstDir + "\""
```

```
    cmdLine = cmdLine + " --depthMapFolder \"" + srcDepthDir + "\""
```

```
    print(cmdLine)
```

```
    os.system(cmdLine)
```

```
    return 0
```

```
def Run_09_Meshing(baseDir,binDir):
```

```
    SilentMkdir(baseDir + "/09_Meshing")
```

```
    binName = binDir + "\\aliceVision_meshing.exe"
```

```
    srcIni = baseDir + "/05_PrepareDenseScene/mvs.ini"
```

```
    srcDepthFilterDir = baseDir + "/08_DepthMapFilter"
```

```
    srcDepthMapDir = baseDir + "/07_DepthMap"
```

```
    dstDir = baseDir + "/09_Meshing"
```

```
    cmdLine = binName
```

```
    cmdLine = cmdLine + " --simGaussianSizeInit 10.0 --maxInputPoints 50000000 --  
    repartition multiResolution"
```

```
    cmdLine = cmdLine + " --simGaussianSize 10.0 --simFactor 15.0 --voteMarginFac-  
    tor 4.0 --contributeMarginFactor 2.0 --minStep 2 --pixSizeMarginFinalCoef 4.0 --max-  
    Points 5000000 --maxPointsPerVoxel 1000000 --angleFactor 15.0 --partitioning single-  
    Block"
```

```
    cmdLine = cmdLine + " --minAngleThreshold 1.0 --pixSizeMarginInitCoef 2.0 --  
    refineFuse True --verboseLevel info"
```

```
    cmdLine = cmdLine + " --ini \"" + srcIni + "\""
```

```

cmdLine = cmdLine + " --depthMapFilterFolder \"" + srcDepthFilterDir + "\"
cmdLine = cmdLine + " --depthMapFolder \"" + srcDepthMapDir + "\"
cmdLine = cmdLine + " --output \"" + dstDir + "/mesh.obj\""

print(cmdLine)
os.system(cmdLine)
return 0

def Run_10_MeshFiltering(baseDir,binDir):
    SilentMkdir(baseDir + "/10_MeshFiltering")

    binName = binDir + "\\aliceVision_meshFiltering.exe"

    srcMesh = baseDir + "/09_Meshing/mesh.obj"
    dstMesh = baseDir + "/10_MeshFiltering/mesh.obj"

    cmdLine = binName
    cmdLine = cmdLine + " --verboseLevel info --removeLargeTrianglesFactor 60.0 --
iterations 5 --keepLargestMeshOnly True"
    cmdLine = cmdLine + " --lambda 1.0"

    cmdLine = cmdLine + " --input \"" + srcMesh + "\"
    cmdLine = cmdLine + " --output \"" + dstMesh + "\"

    print(cmdLine)
    os.system(cmdLine)

    return 0

def Run_11_Texturing(baseDir,binDir):
    SilentMkdir(baseDir + "/11_Texturing")

    binName = binDir + "\\aliceVision_texturing.exe"

    srcMesh = baseDir + "/10_MeshFiltering/mesh.obj"
    srcRecon = baseDir + "/09_Meshing/denseReconstruction.bin"
    srcIni = baseDir + "/05_PrepareDenseScene/mvs.ini"
    dstDir = baseDir + "/11_Texturing"

    cmdLine = binName
    cmdLine = cmdLine + " --textureSide 8192"
    cmdLine = cmdLine + " --downscale 2 --verboseLevel info --padding 15"
    cmdLine = cmdLine + " --unwrapMethod Basic --outputTextureFileType png --flip-
Normals False --fillHoles False"

```

```

cmdLine = cmdLine + " --inputDenseReconstruction \"" + srcRecon + "\"
cmdLine = cmdLine + " --inputMesh \"" + srcMesh + "\"
cmdLine = cmdLine + " --ini \"" + srcIni + "\"
cmdLine = cmdLine + " --output \"" + dstDir + "\"

print(cmdLine)
os.system(cmdLine)

return 0

def main():
    print("Prepping Scan, v2.")

    print(sys.argv)

    print(len(sys.argv))
    if (len(sys.argv) != 6):
        print("usage: python run_alicevision.py <baseDir> <imgDir> <binDir> <numImages> <runStep>")
        print("Must pass 6 arguments.")
        sys.exit(0)
    baseDir = sys.argv[1]
    srcImageDir = sys.argv[2]
    binDir = sys.argv[3]
    numImages = int(sys.argv[4])
    runStep = sys.argv[5]

    print("Base dir : %s" % baseDir)
    print("Image dir : %s" % srcImageDir)
    print("Bin dir : %s" % binDir)
    print("Num images: %d" % numImages)
    print("Step : %s" % runStep)

    SilentMkdir(baseDir)

    if runStep == "runall":
        Run_00_CameraInit(baseDir,binDir,srcImageDir)
        Run_01_FeatureExtraction(baseDir,binDir,numImages)
        Run_02_ImageMatching(baseDir,binDir)
        Run_03_FeatureMatching(baseDir,binDir)
        Run_04_StructureFromMotion(baseDir,binDir)
        Run_05_PrepareDenseScene(baseDir,binDir)

        Run_06_CameraConnection(baseDir,binDir)

```

```

        Run_07_DepthMap(baseDir,binDir,numImages,3)
        Run_08_DepthMapFilter(baseDir,binDir)
        Run_09_Meshing(baseDir,binDir)
        Run_10_MeshFiltering(baseDir,binDir)
        Run_11_Texturing(baseDir,binDir)
elif runStep == "run00":
    Run_00_CameraInit(baseDir,binDir,srcImageDir)

elif runStep == "run01":
    Run_01_FeatureExtraction(baseDir,binDir,numImages)
elif runStep == "run02":
    Run_02_ImageMatching(baseDir,binDir)
elif runStep == "run03":
    Run_03_FeatureMatching(baseDir,binDir)
elif runStep == "run04":
    Run_04_StructureFromMotion(baseDir,binDir)
elif runStep == "run05":
    Run_05_PrepareDenseScene(baseDir,binDir)
elif runStep == "run06":
    Run_06_CameraConnection(baseDir,binDir)

elif runStep == "run07":
    Run_07_DepthMap(baseDir,binDir,numImages,3)
elif runStep == "run08":
    Run_08_DepthMapFilter(baseDir,binDir)
elif runStep == "run09":
    Run_09_Meshing(baseDir,binDir)
elif runStep == "run10":
    Run_10_MeshFiltering(baseDir,binDir)

elif runStep == "run11":
    Run_11_Texturing(baseDir,binDir)

else:
    print("Invalid Step: %s" % runStep)

#print("running")
#Run_00_CameraInit(baseDir,binDir,srcImageDir)
#Run_01_FeatureExtraction(baseDir,binDir,numImages)
#Run_02_ImageMatching(baseDir,binDir)
#Run_03_FeatureMatching(baseDir,binDir)
#Run_04_StructureFromMotion(baseDir,binDir)
#Run_05_PrepareDenseScene(baseDir,binDir)

#Run_06_CameraConnection(baseDir,binDir)

```

```
#Run_07_DepthMap(baseDir,binDir,numImages,3)
#Run_08_DepthMapFilter(baseDir,binDir)
#Run_09_Meshing(baseDir,binDir)
#Run_10_MeshFiltering(baseDir,binDir)

#Run_11_Texturing(baseDir,binDir)
return 0
```

```
main()
```