1

*Уважаеми Г-н Председател,*

*Уважаеми членове на Държавната изпитна комисия, Колеги и гости.*

Казвам се Елица Венчова и темата на дипломна работа, която ще Ви представя е 3D сканираща система със структурирана светлина за реконструкция - Scan-y.

Дипломната работа е разработена под ръководството на проф. Цунижев.

2

Основната цел на дипломната работа е да представи реализиране на 3D скенер със структурирана светлина с цел реконструкция.

За изпълнение на тази задача са формулирани следните подзадачи -

* функционален и технически анализ на приложната област,
* разглеждане на принципа за калибриране, използването на шаблони и получаването на дълбочина,
* Разглеждане на функциите и проектната реализация на Scan-y
* Тестване и настройки на системата
* Възможно развитие и подобрения

3

Първо ще представя две основни дефиниции:

**3D скенер** е устройство, което се използва за сканирането на различни физически обекти от триизмерното пространство и получаването на данни за тяхната геометрия и понякога за цвета в цифров формат.

**Реконструкция** на обект предоставя компютърно четима информация за 3D структурата на обект. Използва се освен с цел преизползване и съхранение на данните, позволява анализирането и създаването на подобрен вариант на сканирания обект.

4

Скенерите се разделят на два вида.

**Контактни**, които използват система от рамена. Върхът се допира до различни точки на обекта и чрез измерване на ъглите на рамената, се създава координатна система от тези точки.

**Безконтактните** се разделят да два типа - пасивни и активни.

**Пасивни** използват естествената светлина при сканирането на сцената. Най-често са съставени от две камери поставени под различен ъгъл, които подобно да човешките очи, генерират стереоскопично изображение. Например при дясното изображение, ако изключим проектора се получава пасивен тип скенер.

**Активните** се отличават с това, че не зависят от естествената светлина, а използват собствен източник като лазер или проектор. След малко ще видим как се получават пространствени данни при този тип. Основното е, че **освен източник на светлина, е необходима поне една камера**.

**Предимствата** на активните методи са, че няма опасност на наранят обекта чрез допир, и че не зависят от наличието или липсата на естествена светлина.

Техен **недостатък** е, че не са подходящи за сканиране на движещи се обекти или динамична сцена. Това ги прави идеален метод за нашата цел реконструкция.

Разбира се, има **още много методи** за сканиране като компютърна томография, излъчване на инфрачервена светлина, които са водещи в своите приложни области.

5

* Основна функция на 3D системите за сканиране, разбира се, е, че всички връщат пространствени данни
* Нещо, което не е задължително и зависи основно от избрания метод на сканиране, но и от реализацията, е наличието на **текстура** или **цвят**. Например е необходима допълнителна разработка за добавяне на камера, за да може един контактен скенер да дава информация за цвета на точката. В случая на активен тип сканиране тази камера вече е налична.
* Основните видове **интерфейси** са няколко. Хардуерен, който е много интуитивен за потребителя, но за добавяне на нова възможност често се изисква нов прототип. Web или настолно приложение са най-често използваните. Особено при web приложение предимството е, че е необходим единствено браузър и свързаност към мрежата, в която е скенера. По идея Scan-y трябва да се използва чрез web приложение, но все още не е реализирано. За момента единственият начин за използване е чрез команден ред като се извиква желаната функционалност с определените параметри.
* **Броят оси** намаляват броя необходими сканирания, когато целта е получаване на 360° резултат. Например на снимката скенера има две оси - една която завърта поставения предмет около центъра и една, която дава възможност за поглед отгоре. Реализацията на Scan-y е само с една ос, която завърта обекта около центъра.
* **Процесът на заснемане** може да бъде ръчен или автоматично - ръчното заснемане предоставя по-голяма свобода и възможност за наблягане на определени места, но автоматичното е по-малко ангажиращо и намалява възможността за пропуски. При Scan-y заснемането е автоматично. Потребителят задава параметри като код на шаблон, размер на стъпка на завъртане и др.

6

Техническите компоненти на един 3D скенер са:

* **Контролна платформа**, която изпълнява управлението на сензорите, актуаторите и обработката на получените данни.
* **Камера** се използва за заснемане на сцената,
* **Докато източник на светлина и въртящата платформа** променят сцената, за да може да се получи информация за обекта от различните гледни точки.
* **Паметта** е необходима, за да се съхраняват снимките и евентуално генерираните пространствени данни.
* **Комуникации** - това са интерфейсите между платформата, отделните компоненти на система, а също и за потребителския интерфейс.
* И **нефункционални характеристики** като резолюция, кадри в секунда, точност, размер на сканирания обекта и др.

7

* Тук по визуален начин е показано как работи заснемането със структурирана светлина.
* На първото изображение се вижда разположението на обекта, камерата и проектора.
* **Въпреки, че проекторът не е камера**, то той може да бъде разгледан като камера на обратно, защото принципът на действие на двете е много подобен, но камерата приема светлината, а проектора я излъчва.
* На второто изображение **графично е показан този принцип**, чрез който се постига **един вид стереоскопично сканиране**.

Отляво е показано **какво излъчва проектора**, това е шаблон, който е съставен от **черни и бели райета**.

Тук горе е **показана сцената** и **по-точно как камерата я вижда**. Вижда се, че обекта е начупен и съответно линиите от шаблона също се начупват. На следващият слайд ще видим принципа на декодиране и получаването на **съответствие между точките C,B и A**.

* Най-отдолу са показани два по-разпространени шаблона - двоичен, който представлява множество от изображения съдържащи се от черно-бели линии, които с всяко следващо намаляват ширината си и увеличават броя. Вторият е фазово отместване. То се състоя от три синусоидни шаблона като всеки следващ има отместена на фаза с някаква стойност като например 2pi/3.

8

* **Тук ще видим как тези множества от изображения се декодират**. На горното изображение всеки от редовете е цяло изображение като е премахната височината за спестяване на място.<<<<<<>>>>>> За кодирането на **16 пиксела** ширина са необходими **4 шаблона**. Ако стойността на черното и бялото ги приемем съответно за 1-ци и 0-ли, то в първата колона, където е само черно, се получава 1111. Така кодът в маркирания пиксел се получава 0110. Разбира се, това кодира уникално само колоната. За да бъде засечен пиксела и по ред, трябва да се прожектират **хоризонтални шаблони.**
* Шаблонът, който разгледахме е Binary шаблон, но за реализацията на Scan-y е използван Gray code. **Принципът на декодиране е същия**, но местата, на които е по-вероятно да се получи грешка са места, на които има преход между черно и бяло. **Това е предимството на Gray code, преходите са много по-малко**. В примерното сравнение на долното изображение се вижда, че при Binary преходите са 11, а при Gray code са само 7. Реално **Gray code е подобрена версия на Binary** като всеки следващ код се различава само с една стойност спрямо предходния.

9

* На този слайд е показан процеса на сканиране.
* Първата стъпка е генерирането на шаблоните.
* Последователно тези шаблони се прожектират и се заснемат от камерата.
* Горните стъпки се изпълняват за всички гледни точки.
* След това за всяка гледна точка се декодират шаблоните - вертикалните и хоризонталните. Тук са **по два, защото** допълнително се заснемат и **обърнати шаблони**(бялото е черно и черното е бяло), **които помагат да се потвърди стойността на пиксела.**
* За всяко декодирана точка, се намира отместването ѝ спрямо очакваното. Това разстояние се превръща в облак от точки като се запазва и информацията за цвета.
* Накрая се генерира .PLY, които съдържа пространствените данни и е четим от програми като AutoCAD, MeshLab и др.
* Последното е визуализирано изображение на 3D облак от точки с MeshLab.

10

Тук е показана техническата реализация на Scan-y.

* В центърът е **контролната платформа**, която е Raspberry Pi 3. Тя управлява всички останали компоненти и генерирането на пространствени данни.
* **Камерата** PiCamera е свързана чрез лентов кабел към специален за нея интерфейс.
* **Проекторът** е свързан и се подава съдържание към него чрез HDMI интерфейс.
* За **памет** се използва MicroSD карта, която по дизайн се използва от Raspberry за съхранение и допълнително съдържа операционната система.
* **Въртящата платформа** се задвижва с **Nema 14** стъпков мотор и драйвър **Pololu** А4288. Допълнително за задвижване на стъпковия мотор е **необходимо захранване 12V**, които не могат да се получат от Raspberry и е необходимо отделно захранване, което преобразува 220V в 12V.
* **На снимката се виждат платформата с обекта**, разположените под различен ъгъл проектор и камера. Контролната платформа е при камерата. Последната част е захранване на Raspberry-то и стъпковия мотор.
* Цялата реализация е поставена в **кутия с покривало**, за да може максимално да се намали ефекта от естествената светлина.

11

* Това е диаграмата на класовете реализирани в Scan-y. Предполагах, че ще се вижда по-добре.
* **Реализацията е подобна на физическата архитектура**. В центъра е клас StructuredLight. В него се съдържат основите методи, които управляват функциите в системата - **Калибриране на проектора, на камерата, стерео калибриране и Сканиране.**
* За всеки от **сензорите** и **актуаторите** има реализиран **отделен клас** - Камера, Проектор и Въртяща платформа. В тях се реализирани **функциите за заснемане**, **визуализиране на шаблони**, **завъртане на платформата** и др.
* Генерирането на шаблоните е изнесено в отделен клас. Причината е, че в момента основният използван метод е Gray code, но в него е реализирано генериране и на други шаблони като Binary, Phase shift и др. В бъдеще лесно може се добави избор на друг шаблон, като единствено трябва да се реализира декодирането.
* **Последните два класа** са Recontruct3D, който декодира шаблоните и генерира пространствените данни и
* Клас ProccessPointClouds. Той е помощен клас на Reconstruct3D, в който са изнесени методите за обработка на облаци от точки. В момента в него има единствено методи за записване и четене на .PLY файлове, но в бъдеще там трябва да се реализират методи за **сливане на облаците от точки** , генериране на mesh и създаване на stl файлове.

12

* В дясно на изображението са показани стъпките за реализиране на 3D сканираща система и статус на изпълнение на всяка от тях.
* Стъпките са разделени на 3 части.
* В първата е **Дефиниране на проекта**, която включва анализ на приложната област и изискванията и избор на управляващ модул, технологии и сензори и актуатори.
* Втората част е **Реализация на системата** - Тук са включени стъпките за управление на сензорите и актуаторите, генериране на шаблоните, калибриране, което е реализирано с OpenCV библиотеката, но не работи много добре и е необходим допълнителен анализ за подобряване. Затова е отбелязано със звездичка, Функционалността на сканирането, което остава да бъде допълнена със сливането на различните гледни точки и генериране на mesh. Има изграден интерфейс, чрез който може да се извикват всички дефинирани функционалности. И нещо, което все още не е започнато е създаването на web базиран потребителски интерфейс.
* Последната част **Тестване** е сложена като отделна по-скоро само за нагледност. Тестването е по-тясно свързано с реализацията. В момента са реализирани unit тестове за създадените методи и софтуерно-хардуерни тестове, които тестват управлението на хардуерните компоненти. Това, което не е направено е крайното валидиране на системата, включително Тестове за точност.
* **Възможни бъдещи разширения** са например
  + - **комбинирането с друг метод** на сканиране, който да компенсира недостатъците на Gray Code или при комбинацията с Phase shift шаблоните е възможно постигане на точност по-малка от един пиксел.
    - С **допълнителна камера** поставена огледално от другата страна на проектора, ще може да се заснемат зони, които заради характера на повърхността не могат да бъдат осветени, когато са видими за първата камерата.
    - Могат да се добавят и допълнителни обработки на **Морфологично затваряне**, което изчиства артефакти от предния фон на изображението и други, които помагат за намаляването на шума.

13

Благодаря!

Въпрос 1:

**Как бихте определили вашата реализация като съотношение цена и възможности?**

Трудно е да определяне цената за реализация на Scan-y, защото освен проектора и лепилото за кутията, всичко останало го имах налично преди това. Все пак ориентировъчно може да се каже, че цената на компонентите е общо 430лв. **плюс стойност за кутията**, която най-вероятно ще бъде променена като например с 3D принтирани части.

Като съотношение цена-качество определено Scan-y ще попадне в **бюджетната категория** скенери като за подобряване на качеството определено трябва да се добавят допълнителни логики, които **предварително определяне на обекта** и да се прилагат определени логики. Като например за полупрозрачни и отразяващи обекти **Фазовото отместване** дава много по-добри резултати. Също и както вече споменах **методи за премахване на шума**.

Въпрос 2:

1. **Известно ли е съотношението между размерите на полученото сканиране и реалните мерки на обекта, или може ли да се извади реално съотношение пиксел-размер?**

Не, съотношението не е известно. Причината е, че бъде много по-лесно да се направи **когато се тества точността на системата**. Принципът е, че при калибрирането на камерата се използва **физическа дъска**, на която е **известна ширината на един квадрат**. По този начин може да се измери разстоянието до задния фон, които по принцип може да се разглежда **като 0 или най-голямата дълбочина**, която може да се получи. Когато имаме колко пиксела на това разстояние на какъв размер отговаря, след това е възможно да се изчисли и колко е **разстоянието между пикселите на по-преден фон**.