

3-й Всероссийский фестиваль научно технического творчества

«РобоМИР»

Продвижение и популяризация объектов культурного всемирного наследия, входящих в состав ЮНЕСКО

Автогид

Авторы проекта:

команда «Мегафон»

Сомов Иван

Ретин Сергей

Соколов Максим

Руководители проекта:

Ворон Александр Максимович

Царькова Наталья Владимировна

г. Москва, 2013

Описание

Санкт-Петербург — важный экономический, научный и культурный центр России, крупный транспортный узел. Исторический центр Санкт-Петербурга и связанные с ним комплексы памятников входят в список объектов Всемирного наследия ЮНЕСКО; это один из самых важных в стране центров туризма. Объект «Исторический центр Санкт-Петербурга» стал одним из первых в практике ЮНЕСКО случаев предоставления статуса памятника огромному культурно-природному ландшафту, на территории которого проживают сотни тысяч людей. Значительное историко-культурное наследие позволяет превратить туризм в базовую отрасль городской экономики. Санкт-Петербург по итогам 2010 года занял 7 место среди самых посещаемых и популярных у туристов городов Европы.

Наш проект – Автогид – был разработан для удовлетворения высокого спроса на качественные, увлекательные и интересные экскурсии по историческому центру Санкт-Петербурга.



Задачи проекта

Перед разработкой и реализацией мы поставили следующие задачи перед нашим проектом:

- Повышение качества информационного наполнения экскурсий. Эта задача решается за счет использования базы данных, которая хранит информацию о памятниках, знаменательных местах города, интересных исторических фактах.
- Повышение вовлеченности туристов в экскурсионный маршрут. Турист, использующий «Автогид», может задавать интересующий его маршрут во время экскурсии. Это позволяет туристу изучать именно те места и памятника, которые ему интересны.



Особенности проекта

Использование современных информационных технологий позволяет:

- достичь высокого уровня безопасности туристов
- защитить памятники от наезда
- предоставить туристам новые услуги, например, запись панорамы интересующего памятника.
- быстро находить памятники (технология компьютерного зрения)
- представлять информацию о памятнике на видео (технология дополненной реальности)
- задавать вопросы об интересующих памятниках и получать ответы на них (технология машинного распознавания речи и технология синтеза речи)



Принцип действия

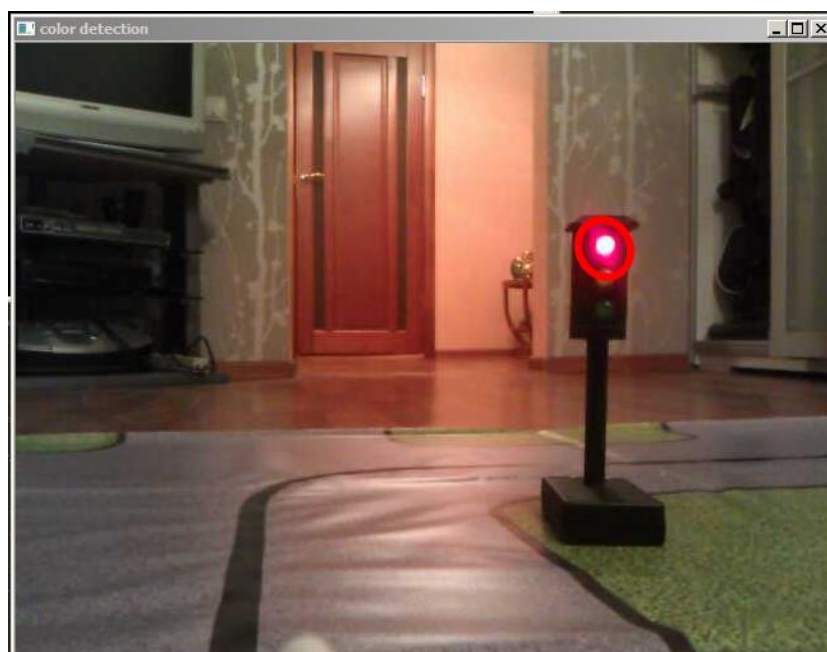


В качестве прототипа транспортного средства, которое будет перевозить туристов, используется робот на базе Lego Mindstorms NXT. Навигация роботом осуществляется самим туристом. Активная система безопасности не позволяет туристу наехать на памятники, выехать за пределы дороги или проехать перекресток на красный сигнал светофора. Для определения расстояния до памятников используется датчик ультразвука. Для определения границы дороги используются 2 датчика освещенности. Еще одним сенсором робота является web камера.



Видеопоток с камеры передается на компьютер, где происходит обработка сигнала: поиск памятников, определение сигналов светофоров и границ дороги. Если на видео был обнаружен памятник, тогда его описание и фотографии с помощью технологии дополненной реальности будут добавлены на личное устройство туриста – планшет или смартфон. Также турист сможет создать панораму понравившегося памятника и сохранить ее себе на память.

Для анализа изображения использовалась библиотека компьютерного зрения OpenCV.



Если турист хочет узнать о памятнике больше, то он может задать вопрос в устной естественной форме. При этом его речь будет распознана, затем Автогид найдет интересующую информацию из базы данных и сформирует голосовой ответ. Для распознавания и синтеза речи использовался Google Voice API.



Листинг программы

```
from nxt.locator import *
from nxt.motor import*
from nxt.sensor import*
import cv2
from time import*
import numpy as np
from rgb2 import*
from features import detect_surf
from panoram import*

def connect(nameb):
    brick=find_one_brick(name=nameb)
    return brick

def draw_str(dst, (x, y), s):
    cv2.putText(dst, s, (x+1, y+1), cv2.FONT_HERSHEY_PLAIN, 1.0, (0, 0, 0), thickness
= 2, lineType=cv2.CV_AA)
    cv2.putText(dst, s, (x, y), cv2.FONT_HERSHEY_PLAIN, 1.0, (255, 255, 255),
lineType=cv2.CV_AA)

def console_writing(i,count):
    #print i
    if excurs[i]==0:
```

```

        if i==1:
            print 'A rostra column is a type of victory column, originating in ancient
Greece and Rome where they were erected to commemorate a naval military victory'
        if i==2:
            print 'The Bronze Horseman is an equestrian statue of Peter the Great in
Saint Peterburg, Russia. Commissioned by Catherine The Great, it was created by the
French sculptor Falconet in 1833'
        if i==3:
            print 'Just a few minutes walk down the Griboedov Canal from Nevsky
Prospect and Kazan Cathedral, St. Petersburgs Bank Bridge is one then worlds most
beautiful pedestrian bridges, thanks to the glorious sculptures of golden-winged
griffons by famous local sculptor Pavel Sokolov. At 1.85 meters, it is also the
narrowest in the city, a miniature architectural gem.'
        if i==4:
            print 'Quay with Sphinxes is a quay at the Universitetskaya Embankment in
Saint Peresburg, in front of the Imperial Academy of Arts. It is remarkable for the
two answered Sphinxes that were brought from Egypt to Russia at the height of
Egyptomania in 1832. The quay was completed in 1834.'

    print ' '
    excurs[i]=1

    draw_str(img, (20, 20*(count+2)), 'Press ENTER for continue')
    while cv2.waitKey(1) <> 13:
        block_all()
        cv2.imshow('img', img)

def moving():
    '''
    while us.get_distance() < 5:
        block_all()
        print 'ultra'
    '''

    global Bool
    print Bool
    power=10
    dpower=10

    common=0
    dm=0
    key=cv2.waitKey(1)

    left.run(common+dm, regulated=True)
    right.run(common-dm, regulated=True)

    if (Bool > 0) and (key == 2621440):
        left.run(power, regulated=True)
        right.run(power, regulated=True)
        sleep(2)
        block_all()
        print 'back'
        Bool=0
    if (Bool > 0) and (detect(40) == 1):
        Bool=0
        left.run(-power, regulated=True)
        right.run(-power, regulated=True)
        sleep(2)

def block_all():
    left.brake()
    right.brake()

```



```

def light_if(level):
    znacl=right_light.get_input_values().scaled_value
    znac2=left_light.get_input_values().scaled_value
    #print znacl,znac2
    if (znacl<level) or (znac2<level):
        block_all()
        return 1
    else:
        return 0

if __name__=='__main__':
    nameb='165-6'
    brick=connect(nameb)

    left=Motor(brick,PORT_B)
    right=Motor(brick,PORT_C)
    level=200

    right_light=Color20(brick,PORT_1)
    right_light.set_light_color(Type.COLORRED)

    left_light=Color20(brick,PORT_3)
    left_light.set_light_color(Type.COLORRED)

    us=Ultrasonic(brick,PORT_2)

    clock()
    global template_name
    global template
    global excurs
    excurs=[0,0,0,0,0,0]

    template_name=['Rostra','Petr_I','Grifon','home']
    #template_name=['home']
    cap=cv2.VideoCapture(0)
    max_kol=[15,15,10,15]
    kol=[0,0,0,0,0,0]
    global Bool
    Bool=0

    while cv2.waitKey(1)<>27:
        #print '#####'
        st=clock()
        _,img=cap.read()
        global template
        if Bool==0:
            for i in xrange(len(template_name)):
                st=clock()
                template=cv2.imread(template_name[i]+'.jpg')
                try:
                    k=detect_hmg(img,template)
                except:
                    block_all()
                    print 'error'
                if k>=max_kol[i]:
                    kol[i]=1
                else:
                    kol[i]=0
                #print template_name[i]+' : ',kol[i]
                Bool=Bool+light_if(level)
            count_detected=0

```

```

        for i in xrange(len(template_name)):
            if kol[i]==1:
                count_detected+=1
                draw_str(img, (20,20*(count_detected+1)),template_name[i]+'
detected')
                console_writing(i+1,count_detected)
                draw_str(img, (20,20*(count_detected+2)),str(clock()-st))
cv2.imshow('img',img)
moving()
'''if clock(>20:
    panoram()'''
#print Bool
#cv2.imshow('template',template)
#print clock()-st
block_all()
cv2.destroyAllWindows()
cap.release()
#'''

```

Выводы

Была разработан робот «Автогид», который позволяет проводить экскурсии по историческому центру Санкт-Петербурга. Анализ результатов тестирования робота наглядно продемонстрировал практическую возможность реализации автоматизированных экскурсий с высоким качеством. Робот перемещается исключительно в пределах дорог, активная система безопасности позволила избежать наездов робота на памятники. При обнаружении памятника робот выводит краткую справку о нем, фотографии, панораму. Разный вид предоставляемой информации – текстовый, графический, голосовой – позволяет туристу полностью удовлетворить тягу к изучению истории одного из самых красивых городов мира - Санкт-Петербурга.