Chana Chadad | 316094101
Adi Kobi | 316034347

Final Project Computer Engineering

-PICIT-

Don't take a photograph, make it!

(Ansel Adams)

Chana Chadad | 316094101 | chana.chadad@mail.huji.ac.il

Adi Kobi | 316034347 | adi.kobi@mail.huji.ac.il

Advisor Prof. Shmuel Peleg | peleg@mail.huji.ac.il

:

Chana Chadad | 316094101
Adi Kobi | 316034347

Problem Definition

With the introduction of digital cameras and smart phones into our lives, we are never satisfied with one single photograph. Every event on our lives of documented in a series of nearly-identical photos, which are shared on social media. Yet, there is always the one person who is dissatisfied. **We are here to satisfy everyone!**

Over the course of this project, we wish to develop an algorithm which will allow the generation of the perfect photograph, which will include each person's favorite moment.

For example, if three people are photographed in three consecutive pictures at a birthday party, each photograph will disappoint one of them. In the first picture, Amir is blinking. In the second, the birthday girl is looking away, and in the third, a balloon is blocking Keren's face. By integrating the three photographs, we aim to create the perfect picture, incorporating the best shot of each of them.

Background

In order to understand this report, the reader should have basic knowledge in image processing.

related work is Photomontage processing, Geometric image rearrangement, Pyramids, homography transform, blending, interesting points and descriptors.

Related Work

http://grail.cs.washington.edu/projects/photomontage/photomontage.pdf

the article presented a framework that allows a user to easily and quickly create a digital photomontage. The system combines graph-cut optimization and a gradient domain image-fusion algorithm with an intuitive user interface for defining local and global objective

http://www.cs.huji.ac.il/~peleg/papers/iccv09-shiftmap.pdf

Shift-map editing. change the features of the image, while saving the natural look of it.

http://webee.technion.ac.il/people/anat.levin/papers/blendingTR.pdf

image stitching, In order to define and get the best possible stitching, the article represent several formal cost functions for the evaluation of the quality of stitching.

https://www.cs.huji.ac.il/~yweiss/Colorization/colorization-siggraph04.pdf

A simple colorization method that requires neither precise image segmentation, nor accurate region tracking. The method is based on a simple premise: neighboring pixels in space-time that have similar intensities should have similar colors.

In this article we learn about another way to indicated the boundaries between parts of the image.

Solution

Overview

First alternative:

The first alternative for solving the problem is that when we take a picture, the camera will take multiply pictures and then choose the best one, by recognize blinking, lack of eye contact and more.

Advantages: choose in real time the picture, without involving the user.

Disadvantage: you can't merge between different images, and the algorithm doesn't know your taste, and what count a good picture for the individuals.

Second alternative:

Another alternative is using filters.

Advantages: variety of options, remove flaws like acne.

Disadvantage: you can't save the nature look and can't merge between different images.

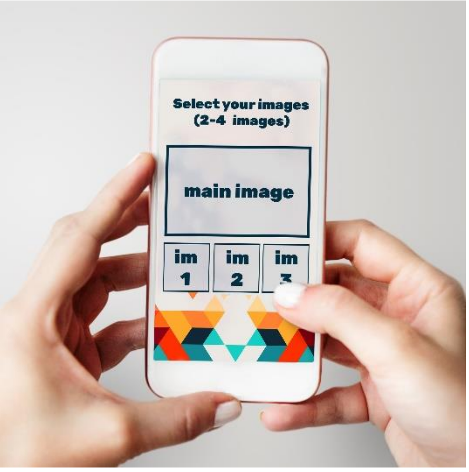
Proposed Solution

In our project the user will take few pictures (about 3), then he will be able to choose the parts from each image that will be in the final image. The program merges all the pictures in a natural way and create the final outcome- the best image!

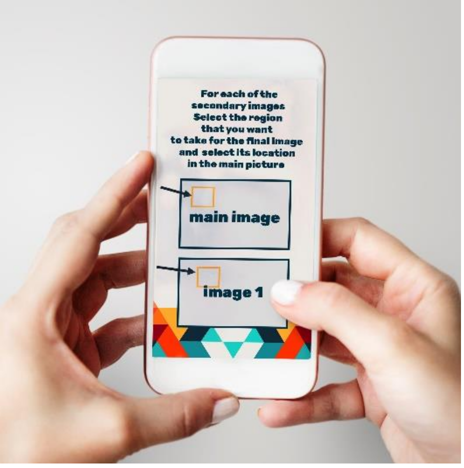
During the implementation we will use algorithms of image processing such as min-cut, pyramids blending, detected feature points, RANSAC, stitching. In this way we will get the best results for blending between images. We chose this approach because now we can create an image that look natural, the users can choose their own moments and create an integration of multiply images.

Chana Chadad | 316094101
Adi Kobi | 316034347

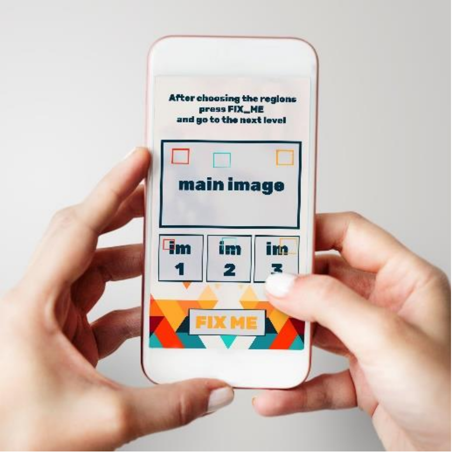
Key Components



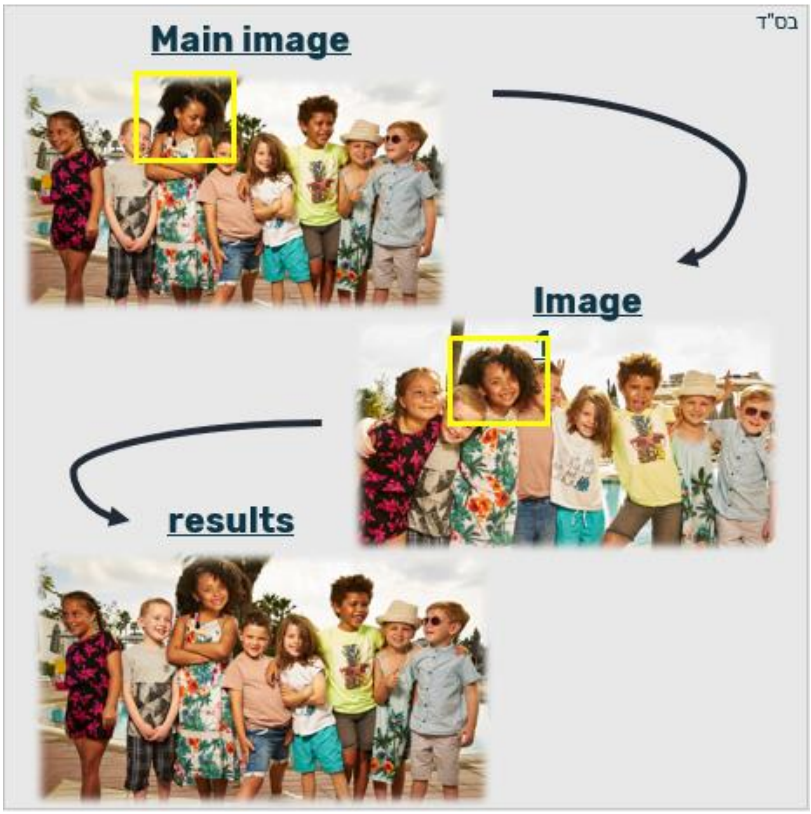
Select your images
(2-4 images)

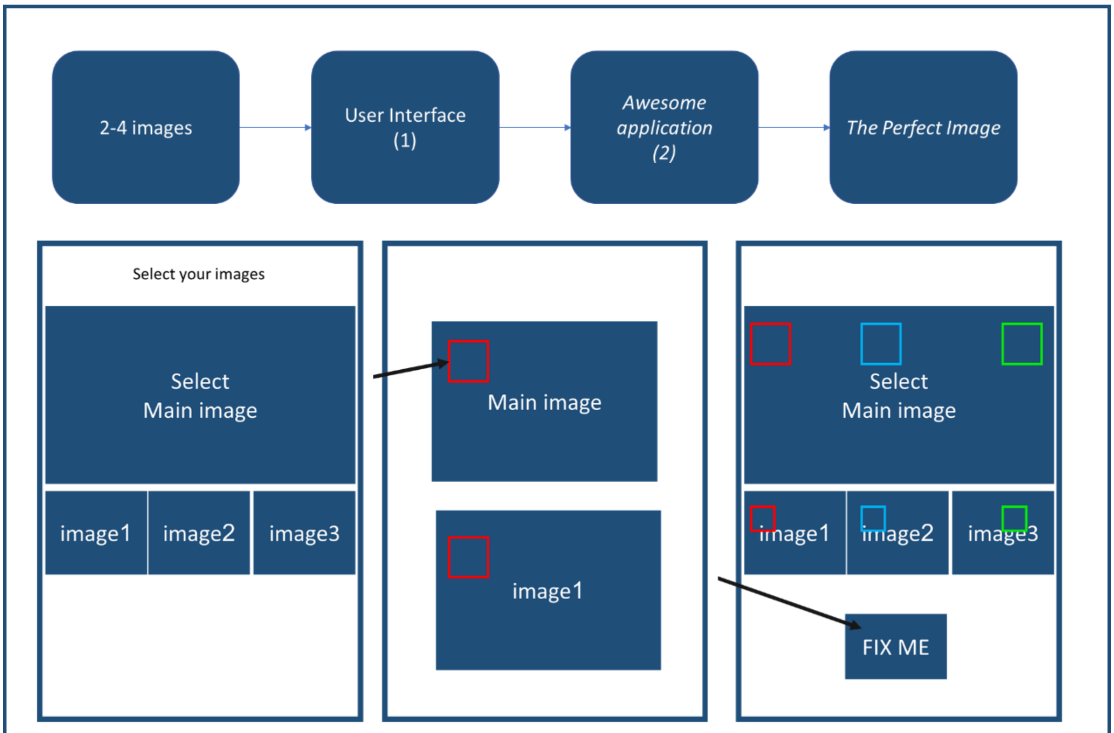


For each of the
secondary images
Select the region
that you want
to take for the final image
and select its location
in the main picture



After choosing the regions
press FIX_ME
and go to the next level





1- as we see in the images above.

2- The app will run some image processing algorithms and create the final image.

Solution Feasibility

various technologies we plan to use in your project:

Python/cpp libraries like:

- Numpy
- Open CV
- Matplotlib
- Scipy
- Skimage
- Image io
- Pillow

For the application:

django - a high-level Python Web framework or android studio, or Xcode

https://developer.android.com/studio/?gclid=Cj0KCQjwh6XmBRDRARIsAKNInDHxNznQkCnb8kdPfYoPW_HMEu_hXd2BOUz93_Q7eC52hjQ-XzI04UAaAoGVEALw_wcB

https://developer.apple.com/xcode/

different challenges and uncertainties in our project:

we want that our application will be in daily use, so we need to make sure that the performance will be quick, clean and accurate.

Time:

The process of checking the parts of the images to find the best fit for the integration requires an observation on all the image's pixels, finding interesting points, processing and handling them. All this may take a long time, which will significantly impair the user experience. Our goal is to keep the use easy as we can, so we will have to invest thought in finding the fastest way that give us good performance.

Accuracy:

We want that the combination of the pictures will look natural, as this is the original picture that was taken. To do this we will have to look for the ideal way to make the blending and the integration.

After implement every step- the algorithms and the app, We will make user experience testing and improve our work accordingly.

Chana Chadad | 316094101
Adi Kobi | 316034347

Plan for Spring Semester 2019

Skateboard-Bike-Car

We are not sure that our algorithms will be fast and accurate enough, so to test this we will first check with simple algorithm and small images.

We are not sure our app will be comforting to use, so we will create a paper prototype of our app (that include the levels of the use) and test it on some of our friends.

Our plain:

1. First step, we want to get familiar with the algorithms that we will use, learn MinCut and Shift-Map.
2. Then we will implement the basic algorithms and test them on simple images on the computer.
3. Know the target- choose the platform for the app and start implement the connection between the algorithm that we run on the computer to the app platform.
4. Test the implementation on more complicated data.
5. user experience testing -Try our app on different people, get feedback and improve the app.
6. Design the application.
7. Get the complete product.

Preliminary Knowledge

Academic literature:

http://grail.cs.washington.edu/projects/photomontage/photomontage.pdf

the article presented a framework that allows a user to easily and quickly create a digital photomontage. The system combines graph-cut optimization and a gradient domain image-fusion algorithm with an intuitive user interface for defining local and global objective

http://www.cs.huji.ac.il/~peleg/papers/iccv09-shiftmap.pdf

Shift-map editing. change the features of the image, while saving the natural look of it.

http://webee.technion.ac.il/people/anat.levin/papers/blendingTR.pdf

image stitching, In order to define and get the best possible stitching, the article represent several formal cost functions for the evaluation of the quality of stitching.

HUJI courses: (we already took those courses)

Image processing

Algorithms

Computer vision

Software and Hardware Requirements

Learn how to use android studio or IOS in order to build mobile-based image processing.

Preliminary Assignments

Read articles and choose the methods and algorithms to use in our implementation.

Overview on the material of image processing.

Then, we will Implement 2 of the algorithms that we may choose and learn from the articles- min cut and shift-map.

Semester Schedule

During the meeting with our advisor Prof. Shmuel Peleg we decide on a schedule for this semester. Prof. Shmuel emphasize that before implement the algorithms, we need to invest time in learning them deeply.

May:

- learn the first algorithm- min cut.

June:

- implement Min-cut algorithm.
- Learn the second algorithm- Shift-Map.
- Implement the Shift-Map algorithm.

Chana Chadad | 316094101
Adi Kobi | 316034347

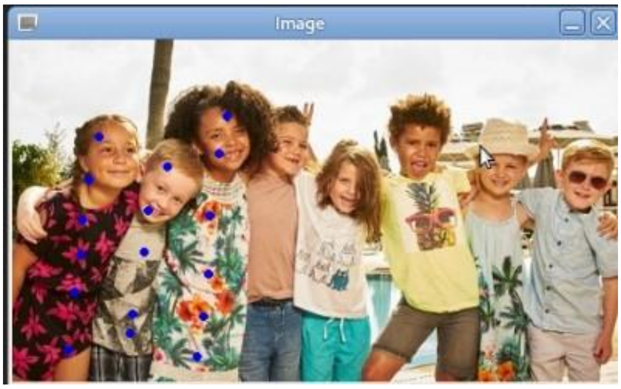
Status Update

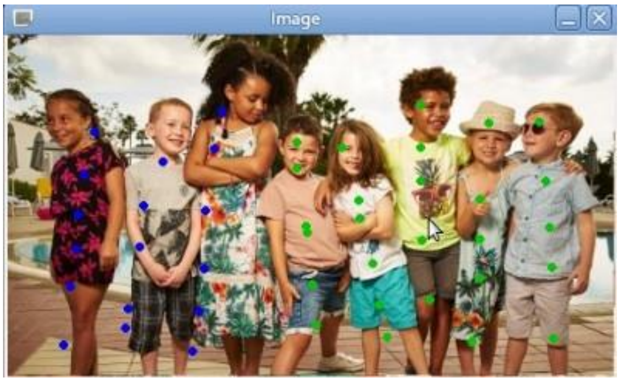
Prof. Shmuel emphasized that before implementing the algorithms, we need to invest time in learning them deeply. As we say, we want to try different ways and algorithms for the implementation of the project and choose from them the final algorithm. We learned the articles: " Interactive Digital Photomontage" and " Shift-Map Image Editing".

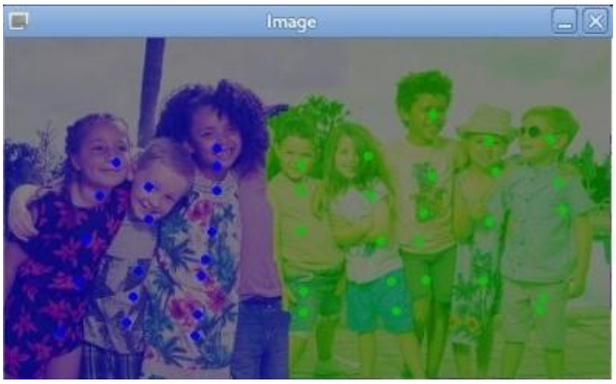
We upload the relevant code of the first algorithm to the git and create some examples for this implementation. the next goal is to implement the second article.

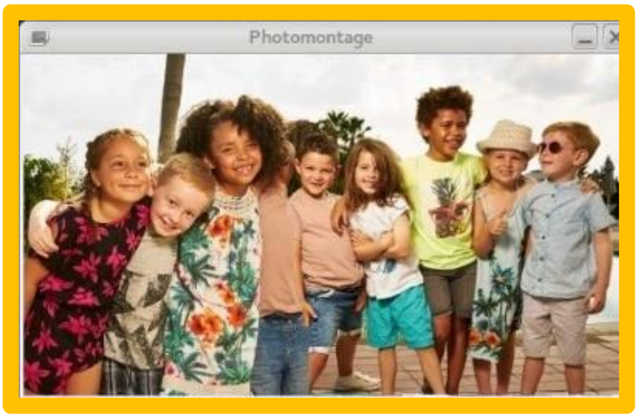
we add to this report one example. We want to mention, in the Photomontage algorithm we work differently than we thought and see in our first vision- now the user doesn't need to choose one of the images to be the main image and parts to replace from it. instead it, the user chooses the parts that he wants from each image, and the algorithm combine all the parts together to one big image.

We can see that the results are not so good (the boy in the middle), but compatible with the purpose of the skateboard- create basic implementation. In the next year our aim is to improve this method, implement the other method and choose the best one. This is a change in our plan- in the first report we aim to finish both methods, but we understood that those things take more time, and we need to get into the small details in order to achieve results. we learned the articles as well and have an implementation of one method, we will implement the second one next year.









Evaluation and Verification

We want to mention, we try to consult with the course staff regarding the sentence- "We want to see actual numbers and quantitative targets", because we didn’t understand as well the meaning of actual numbers regarding to our kind of project (we can see how it relate more to machine learning projects, as in the example with the accurate rate.), but we didn’t get answer to our emails , nevertheless we try our best in it;

We want to test our combining methods on different image, different situations or seams locations and kinds. We want our project to fit varied kinds of images, in order to achieve that we test it on set of images and try to find our strengths and our weakness by looking on the results.

We think that the best way to do this will be separate into some levels:

1. Test on 10-20 images, find the strengths and weakness, improve according to them.
2. Do the first step again. Try different seams, test accuracy and execution speed.
3. Go outside! user experience testing -Try it on different people, get feedback and improve accordingly. This level can be done after we create the app and merge everything together.
4. Check everything on a very complicated images, even some images that we can't get good results using photoshop, this level is another type of verification, in a "creative" way. In this part we want to check that our system isn't "going crazy" in edge cases and return reasonable results even on weird images.
5. Another level of verification is to compare our results to results that different people did with different methods (articles and works online, or even by manual editing work).

Plan for Next Year

MVP- create program that take 2-4 images and create one final image according to the user choosing. In this level, the program is only on the computer and this is the simple prototype of our project.

Bike: after creating the Skateboard- first sketch of the merge algorithms implementation that work only on the computer, we want to pass to the bike part that will contain basic application and communication with the user. The bike part creates an interface for the user and connect between the program to this interface.

Car: in this part, we will examine the results even more, improve it and the user experience. In addition, we will try to achieve better performance- meaning faster and more accurate results. Also, we will work on the appearance of the application itself and the convenience of use.

Date:	Task:	Priority:
October	Implement the second article.	P0
November	Working on the code of the algorithms	P0
December	Learn the application platform and get experience in creating general and basic app	P0
January	Continue working on the application, build base that will fit to our needs.	P0
February	GUI for the algorithm. Combine the code of the algorithm inside the application	P0
March	Finishing touches for the demo-presentation	P0
April	Improve the accurateness, the appearances and the time of the performances.	P1
May	Improve the accurateness, the appearances and the time of the performances.	P1
June	Create the final presentation :)	P0

We think that our project can be very useful for a lot of people so we hope that the ending app will results good images with good performance, and mort then this, it will be fun, beautiful and easy to use.

Can't wait to finally see our dream taking shape and coming true :)