Localisation by Mobile Phone Images

Is goede lokalisatie met slechte foto's mogelijk?

Afstudeerproject Bachelor AI

Thomas Mensink
Martijn Liem
Daan Odijk

Datum
1 juni - 25 juni 2004

Begeleider
Ben Kröse

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam

Localization by Mobile Phone Images

Graduate project Bachelor
Artificial Intelligence
University of Amsterdam
June 2004

Thomas Mensink, Martijn Liem and Daan Odijk
{tmensink,mliem,dodijk}@science.uva.nl

Abstract. In this paper we describe methods for localization based on mobile phone images. We use techniques proposed by researchers in the fields of robot localization and image retrieval. We combine these two fields and extend their methods with our own extension for computing and end result. Furthermore we will describe a method to compute the confidence and reduce the number of absolute errors to a minimum.

1. Introduction

Localization is a central issue in the research for intelligent autonomous systems. Besides localizing mobile intelligent autonomous systems, localization techniques can also be useful for localizing Personal Digital Assistants or Mobile Phones. These can serve, for instance, as interactive museum guides.

Many current localization systems use stereo vision or omni-directional camera's to make an accurate estimation of the location the system is at. Ulrich and Nourbakhsh [1] introduced a model for histogram based localization using an omni-directional camera and a moving robot. The camera has a steady position on top of the robot and captures his environment with one image every second. They localize at room-level, where the position in a room is not important.

In this paper we focus on localization by images from a low quality webcam, which has a resolution and color-depth comparable to a mobile phone camera.

There are two big differences between mobile intelligent autonomous systems and our domain. First, mobile systems have the advantage of being able to register their complete path of movement, while we only have an image of the current location without any context. Second, all the described systems make use of very complex hardware like stereo or omni-directional camera's. Our webcam is not able to make either stereo or omni-directional images, so the captured images contain much less information about the location.

Tico, Haverinen and Kuosmanen [2] introduced a system for image retrieval from a large database of newspaper images, based on weighted hue en intensity histograms. By weighting their histograms, they largely reduced the amount of noise in the histograms, which results in higher quality image retrieval. Therefore we will use the same weighted hue and intensity histograms for our system.

Our system will try to estimate the location of a sample image, captured by our webcam. The following section describes the creation of the histograms used by our system. The third section outlines the comparison between histograms. Section 4 describes our method to determine our confidence for localization. Section 5 describes our experiments and results and in section 6 we draw our conclusions. Finally, in section 7 we make recommendations for further research.

2. Creating and comparing histograms

To be able to get results with a high level of correctness and certainty, we make use of five different histograms to discriminate between images. From the HSI (Hue, Saturation, Intensity) color space we only use the hue and intensity histograms, from the RGB (Red, Green, Blue) space we use all three.

For the RGB histograms we used standard, non-weighted histograms. The main problem using the RGB color space is that an object appears to have different colors under different levels of luminance. When shadows are cast over a red wall, the wall will appear to have a dark red color, while under full sunlight the wall would appear brightly red. These differences in color appearance result in large differences in RGB values. This makes it difficult to compare images purely based on RGB histograms.

This problem can be avoided by making use of the HSI color space. The way this color space classifies colors is quite similar to the way humans perceive colors. Instead of looking at the separate color components of a color, you take into account the luminance level and the level of chromaticity in the color. By dividing the space in a color component (hue), a chromaticity component (saturation) and a luminance component (intensity) the dark red wall and the bright red wall will have the same value for at least the hue component. This makes it possible to identify them as the same wall.

According to Tico et al. [2], you only need the H and I histograms to be able to distinguish between different images. They state that, while the hue histogram alone should be enough to describe the color content of an image, you also need to discriminate between chromatic and achromatic regions in the image to get a good overview of the color contents of an image. This can be done by the weighted intensity histogram.

Because the color of an achromatic image region will only introduce noise into the hue histogram, these regions should not be taken into account when making this histogram. After all, an achromatic region will have a hue level which has no effect on the appearance of the color. Therefore, such pixels should have less influence on the hue histogram, which can be achieved by weighting the hue values according to the level of chromaticity. The saturation component will not be sufficient to determine the level of chromaticity, while this component is too much influenced by the level of intensity. When you take a pixel with RGB values (0.01, 0, 0), the pixel will look completely black to a human observer. Nevertheless, the saturation component will achieve the maximum value for this pixel. A pixel with these RGB values will have a large influence on the weighted intensity, so the weighted intensity describes the achromatic contents of an image.

Our method of classifying a pixel as chromatic or achromatic is based on the standard deviation of the RGB values of that pixel, Tico et al. [2]. Per pixel, we compute the standard deviation (s^*) of the red, green and blue values. Because achromatic colors are defined by having nearly the same red, green and blue values, the standard deviation of these values can tell the level of chromaticity of the pixel. This parameter achieves high values in chromatic pixels and low values in achromatic ones. When this value is normalized between 0 and 1, it can be used to weigh the hue and intensity histograms. The values of the weights can be computed using the following function:

$$\mu(s^*) = \begin{cases} 0 & \text{if} \quad 0 \le s^* < a \\ 2\left(\dfrac{s^*-a}{b-a}\right)^2 & \text{if} \quad a \le s^* < \dfrac{a+b}{2} \\ 1-2\left(\dfrac{s^*-b}{b-a}\right)^2 & \text{if} \quad \dfrac{a+b}{2} \le s^* < b \\ 1 & \text{if} \quad b \le s^* < 1 \end{cases} \tag{1}$$

In this function a and b are threshold values to discriminate between chromatic and achromatic pixels.

The weights the hue and intensity values of a certain pixel (m, n) have in the histograms, is computed using (1) as follows:

$$w_H(n,m) = \mu(s^*(n,m))$$
$$w_I(n,m) = 1 - \mu(s^*(n,m)) \tag{2}$$

Here w_H is the weight of the hue value for pixel (n, m) and w_I is the weight of the intensity value. As you can see in (2), pixels with high chromaticity levels get higher hue weights while getting lower intensity weights and vice versa.

Using (1) and (2) we now can compute the weighted H and I histograms. Let L_H and L_I denote the number of bins used for each histogram. Assuming the values for H and I are quantized such that $H(n,m) \in \{0,1,...,L_H - 1\}$ and $I(n,m) \in \{0,1,...,L_I - 1\}$, the weighted hue histogram W_H and weighted intensity histogram W_I are defined as

$$W_X(l) = \frac{\sum_{n,m} w_X(n,m)\delta(X(n,m),l)}{\sum_{n,m} w_X(n,m)} \tag{3}$$

for each $l = 0, 1, \ldots, L_X - 1$, where δ is the Kronecker delta function and X stands either for H or I. This function simply takes all weighted pixel values quantized in bin l divided by the sum of all values, so the histogram is normalized as well. When comparing these weighted histograms to standard hue and intensity histograms, they clearly are much more fluent and contain much less noise. This will make a comparison between two images with slightly different histograms much more simple. In figure 2.1 you can see the difference between a weighted and a non-weighted hue histogram of a single picture. The weighted histogram clearly shows two peaks at the positions of the orange (8) and blue (56) colors, without the noise from the non-weighted histogram.

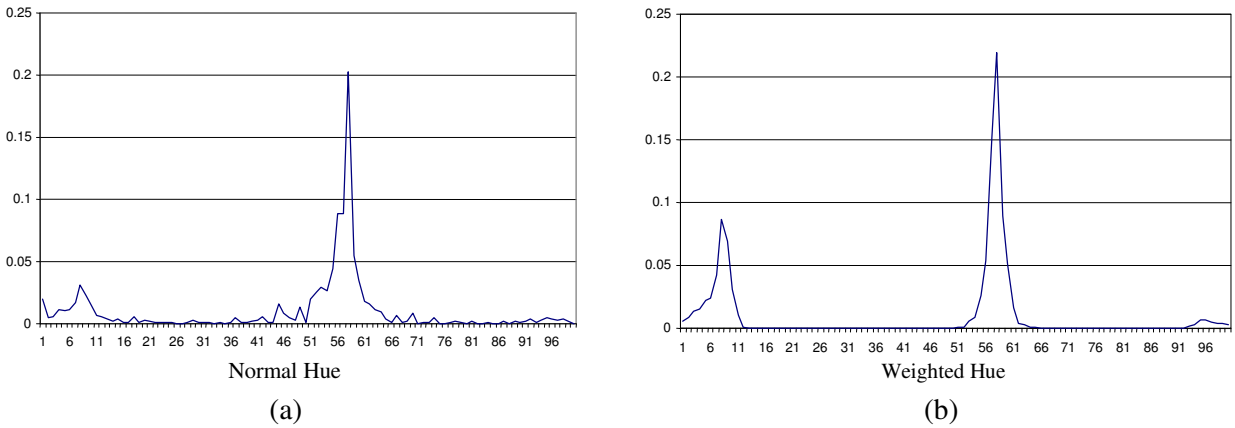

Figure 2.1: Example of the difference between a non-weighted hue histogram (a) and a weighted histogram (b) for the image shown in figure 5.2.

3. Histogram Comparison

For computing distances between histograms we use the Jeffrey divergence as proposed by Ulrich and Nourbakhsh [1]. This method works slightly better than the χ^2 statistics. The Jeffrey divergence is computed as follows:

$$d(H,K) = \sum_i \left(h_i \log \frac{2h_i}{h_i + k_i} + k_i \log \frac{2k_i}{h_i + k_i} \right) \tag{4}$$

where H and K are histograms and h_i and k_i separate histogram bins. The total distance is the sum of the distances per bin.

For each candidate location and each type of histogram, the distance between the input histogram and the reference histograms is computed using (4). For each candidate location the minimum distance is then determined for each type of histogram.

Ulrich and Nourbakhsh [1] use about the same procedure as described above. However, they use these distances to let each type of histogram cast a vote for a candidate location. For each vote a confidence value is computed as well. Their final vote can only be confident when all types of histograms vote for the same candidate location with sufficient confidence.

This voting system did not seem to work at all for our system. The first big difference is that they use an omni-directional camera and thus have a larger field of view. This way, each image contains a lot more information about the environment compared to our non-omni-directional images. Second, our system should be able to find the correct candidate location under all sorts of lighting circumstances. The method of Ulrich and Nourbakhsh [1] is very sensitive for changes in illumination.

We propose a new method to overcome our objections regarding their voting system. We look at histogram comparison as comparing the distances between histograms using the Jeffrey divergence. Once we have the histogram with the smallest distance per candidate location for all types of histograms we compute a compiled distance. This compiled distance will be computed in a 5 dimensional space, because we use 5 types of histograms. In this 5 dimensional space we can use the Euclidian distance to find the nearest candidate location.

By scaling the dimensions used for the compiled distance, we can assign weights to the different types of histograms. This way we can make sure that the type of histogram with the most valuable information has more influence in the compiled distance. A more formal description of the system is given below.

$$cd_i = \sqrt{\sum_{j=1}^{5} \left(W_j * \min_m \left\{ d\left(H_j, K_{i,m,j}\right) \right\} \right)^2} \tag{5}$$

cd_i: compiled distance for candidate location i
d: distance as computed by Jeffrey's divergence
H_j: sample histogram of type j (H, I, R, G, B)
$K_{i,m,j}$: reference histogram m in candidate location i of type j
W_j: weight for histogram of type j

For each candidate location we have a compiled distance to the input image. The candidate location with the smallest compiled distance will be the one closest to the sample image.

4. Confidence

Our method of computing the confidence is based on that of Ulrich and Nourbakhsh [1]. The confidence measure c is computed as follows:

$$c = 1 - \frac{cd_m}{\min_{i \neq m}\{cd_i\}} \tag{6}$$

cd_m: minimum compiled distance of all candidate locations
cd_i: minimum compiled distance of all other candidate locations

Confidence values range between 0 and 1. The higher the confidence value is, the more confident we are about a chosen location. The confidence measure achieves high values for candidate locations matching the sample image much better than any other candidate location. The confidence value is low if the second best candidate location matches the sample image similarly well as the best candidate location. If no candidate location matches the sample image well, a high confidence value is unlikely. Thus if the compiled distance is unable to reliably classify an input image its uncertainty is reflected by a low confidence value.

Our system does not only give a confidence value between 0 and 1, it also classifies as confident or uncertain. We do this by comparing the confidence value to a certain threshold value. If it is above this value the system is confident, if it is below this value it is uncertain. This way we can tune the system to virtually never cast a confident vote for a false location.

5. Experiments and results

We tested our localization system in an environment with 4 different rooms. We used the library, the study room, the entrance hall and the staircase from the Euclides building [6] of the University of Amsterdam. These rooms all have difference sizes. The figures 5.1 to 5.4 below give an impression of the rooms used. At each location we made pictures using a very low quality webcam connected to a laptop.

Figure 5.1: The library

Figure 5.2: The entrance hall

Figure 5.3: The study room

Figure 5.4: The staircase

The obtained images were of a resolution of 352 by 288 pixels and a color-depth of 24 bits. We made a total of around 850 reference images. For every reference image we derived the Red, Green and Blue histograms and the weighted Hue and Intensity histograms as described in the chapters above. For all histograms we used a bin size of 16, except for the intensity histogram where we used only 4 bins. The threshold values for discriminating between chromatic and a-chromatic colors used in formula (1), were set at a = 0.05 and b = 0.1. These values were chosen based on various testing results.

For the weights used in the compiled distance we used 2 for red, green and blue, 1 for intensity and 4 for the hue histogram. These values seemed to be most reasonable, according to the importance of the different histograms.

We also made 47 sample images from the different locations to test our system. Of those 47 images, there were 8 images captured in a different lighting condition. They were taken in the library and

entrance hall with some or all of the lights switched off. For all 47 images the histograms were made likewise the reference set. Next they were compared with our database of reference images as described in section 3. For every image, the system calculated the location were the image was taken. Overall the localization was correct for 37 out of the 47 (79%) images. The system did quite good for the darker images as well, where only one out of eight images was classified wrong. This is shown in Table 5.1.

	Normal		Darker		Total	
Right	30	(77 %)	7	(88 %)	37	(79 %)
Wront	9	(23 %)	1	(12 %)	10	(21 %)
Total	39		8		47	

Table 5.1: Localization results

From the 47 sample images the weighted hue and intensity histograms were able to localize 32 images correctly. The RGB histograms however were only able to localize 27 of them correctly.

Besides the best candidate location we also computed the confidence of that localization. We compared this to a threshold value to state if we were confident or uncertain.

Table 5.2 shows the results for confident and uncertain localizations based on different threshold values.

	Threshold value		
	$c \geq 0.26$	$c \geq 0.275$	$c \geq 0.40$
Confident: Right	28 (60 %)	26 (55 %)	20 (43 %)
Uncertain: Right	9 (19 %)	11 (24 %)	17 (36 %)
Uncertain: Wrong	8 (17 %)	9 (19 %)	10 (21 %)
Confident: Wrong	2 (4 %)	1 (2 %)	0 (0 %)

Table 5.2: Results when stating the confidence with different threshold value

We used different threshold values to give an impression of how much better the system is able to work if you allow it to make some small mistakes. As shown in table 5.3, when using 0.26 as the threshold value 63% is classified as confident, while only 43% is classified confident for a threshold of 0.40. When you look at the number of confident wrongfully localized samples, 7% is classified wrong at a threshold of 0.26 against no wrongly classified samples using a threshold of 0.40.

	Threshold value		
	$c \geq 0.26$	$c \geq 0.275$	$c \geq 0.40$
Confident (percentage of total)	30 (63 %)	27 (57 %)	20 (43 %)
Right when confident	28 (93 %)	26 (96 %)	20 (100 %)
Wrong when Confident	2 (7 %)	1 (4 %)	0 (0 %)

Table 5.3: Rightly or wrongfully confident at different threshold values

6. Conclusion

In this paper we introduced a method for localization based on images from a low quality webcam. The method used is derived from a combination of a method for localization with omni-directional camera and a method for retrieving images from a large database. The used combination shows an accuracy of 79 percent in the given test conditions. This was way beyond our expectations. The results of only using the weighted hue histogram are very high, because the hue has the most (light independent) color information.

With the addition of the confidence features we added, it is possible to reduce the wrongfully localized images to a minimum. This does however imply that, in the case no errors are allowed, the system will be limited to casting a confident vote for a candidate location in only 43 percent of the cases. Allowing the system to make a mistake in 7 percent of the confident cases, will increase the number of confident votes up to 30 (63 percent). The threshold value should depend on the domain in which the system operates.

We are aware that our research as well as our findings are limited by the images we took and the way we constructed the system. First of all we used only four rooms in a single building, which is quite limited. Our estimates are, however that adding another room which doesn't resemble one of the other rooms too much, the results will not be enormously influenced. Secondly, although we did indeed experiment with different light conditions more research is needed to draw extensive conclusions here. Thirdly, our sample images were limited in number and difficulty. More sample images would have been useful, as well as more difficult images, like images not taken in one of the rooms in our reference set.

7. Further research

Although we are quite content with the results of our system, we do see ways to improve it which might need further research.

One important improvement is to make the system ask for a new image when it is uncertain. This new image can either be processed without looking at the image(s) processed earlier or with some sort of composed result. This way the system is extended to be truly useful in real world situations.

In our experiments we make use of only one camera. To get more unique data per position, it is possible to use two simple camera's, positioned with a predefined angle between them, so you have a combination of two unique histograms per position. This will largely reduce the chance of finding similar data for multiple candidate locations and therefore highly improve the results gained by the system.

We have seen that the HSI color space works better than the RGB color space. It might very well be possible that some other color space might work better alone or in complement to our current system. It can also be interesting to look at a way to normalize the RGB color space to make it less light affected by changes in illumination.

We compare our sample image to all the reference images in the database. There might be a way of clustering the reference images, which will increase the speed and the quality of the system. Maybe the system is improved by a tree like data structure or clustering the images in to groups (either human or computer based).

We can think of a lot more improvements such as a better camera and completely different methods to compare the images, like vanishing points. We leave this to the imagination of our reader.

8. References

1. Iwan Ulrich and Illah Nourbakhsh, Appearance-Based Place Recognition for Topological Localization
2. Marius Tico, Taneli Haverinen, Pauli Kuosmanen, A method of Color Histogram creation for image retrieval
3. José-Joel Gonzalez-Barbosa, Simon Lacroix, Rover localizaition in natural environments by indexing panoramic images
4. J.M. Porta, J.J. Verbeek, B.J.A. Kröse, Active appearance-based robot localization using stereo vision
5. François Ennesser and Gérard Medioni, Finding Waldo, or focus of attention using local color information
6. Euclides Building, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam