Boosting Cross-Modality Image Registration

Adrian Barbu
Department of Statistics
Florida State University
Tallahassee, Florida 32306, USA
Email: abarbu@stat.fsu.edu

Razvan Ionasec
Siemens Corporate Research
755 College Rd. E
Princeton, NJ 08540, USA
Email: razvan_ionasec.ext@siemens.com

***Abstract*—Cross-modality image registration is a difficult problem because the same structures have different intensity patterns in the two modalities, making straightforward methods based on SSD or cross-correlation not applicable. This paper presents a learning based approach to cross-modality image registration. First, it describes a method to map the image registration problem into a problem of binary classification. Then, it presents a method to select a number of image registration algorithms from a larger pool and combine them by AdaBoost into a boosted algorithm that is more accurate than any of the algorithms in the pool. Finally, it presents a method named virtual boosting that allows to directly obtain the result of the boosted algorithm without performing any parameter search. In our cross-modality image registration application, the algorithm pool consists of many feature-based registration algorithms with different configurations. An experimental validation on the registration of thousands of aerial video frames with satellite images from Google Maps showed that the boosted algorithm has a 20-30% smaller error than the best registration algorithm from the pool (based on SIFT features). More generally, the method presented can be applied to combine a number of algorithms aimed at solving the same problem into a boosted algorithm that is more accurate than any of them.**

I. Introduction

Cross-modality image registration is an intensely studied problem with applications in fields such as surveillance and medical imaging. It is a difficult problem because the same structures have different intensity patterns in the two modalities, making straightforward methods based on SSD or cross-correlation not applicable. Instead, invariants are usually sought, either as similarity measures that find a mapping between the two intensity patterns, or as landmarks that are the same for the two modalities. Many registration methods from the first category are based on the maximization of the mutual information [16], sometimes using a learned prior [6]. However, these methods are computationally expensive, making them not applicable to real-time or near real-time situations. In the second category are methods that extract a number of feature points (e.g. SIFT features [2]) and match them using simple algorithms to quickly obtain the desired registration. These feature-based methods however can quite often give erroneous results since they do not take the entire image into consideration when obtaining the desired result and they have no confidence measure of the result obtained.

There are other direct image methods such as those based on phase correlation [10], [4], [3], which use the Fast Fourier transform to quickly obtain a similarity map. These methods are very efficient and can be very effective for certain types of noise patterns; however they are ineffective in the case of our cross-modality registration as it can be observed from the phase correlation map from image 2.

978-1-4244-3461-9/09/$25.00 © 2009 IEEE

Fig. 1. Example of an two aerial images (up) containing large parallax due to tall buildings and a satellite image (from Google Maps [5]) of the same area.

Comprehensive surveys of the early and later image registration techniques can be found in [1] and [17].

In this paper, we show how the registration problem can be mapped into a problem of binary classification trained in a supervised manner, with one positive example for each correct registration of an image pair, and negative examples for any incorrect registration. This framework allows us to learn a robust and accurate registration algorithm from thousands of manually registered training images. The registration algorithm is learned using a larger pool of feature-based registration algorithms from which a small number is selected and combined using AdaBoost.

II. Cross Modality Image Registration

The problem that we will study in this paper is the registration of aerial video sequences obtained by an unmanned plane to satellite images of the same area. The unmanned plane has a

GPS locater that records its position and altitude every second with an accuracy of about 10 meters.

This is a difficult registration problem because of two reasons:

- It is cross-modality, with the satellite images being obtained at a different time, from a different location (distance and angle), under different illumination conditions and with a different type of camera than the aerial video images.
- The registration is desired to be accurate at the street level while the images contain tall buildings producing large parallax deformations, as shown in Figure 1.

Fig. 2. Video image, rotated and scaled satellite image and phase correlation map. The true registration is at (15,29) and the map ranges from -100 to 100 in both x and y directions.

Methods that work directly on the images, using image gradients, are not effective in this case because the gradient maps or the image edges are too different. This can be observed in Figure 3 where we show that the edge maps of the left video frame from Figure 1 and a rotated, rescaled and cropped satellite image that was aligned with the video image.

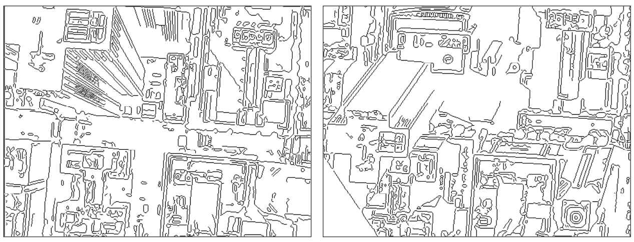

Fig. 3. Canny edge detection of the left aerial image in Figure 1 and a rotated, scaled and cropped version of the satellite image of the same area.

A. Registration as a Binary Classification Problem

In this paper we will study rigid registration, either described as a 6-dimensional affine transformation or as a two dimensional translation. For the sake of clarity, we will focus on the two dimensional translation first and at the end of the paper we will describe the extension to affine registration.

By taking advantage of the given GPS and altitude information, the orientation of each aerial image can be approximately estimated from the GPS trajectory. Also, the scale can be approximately obtained from the plane altitude. This way, a 4-parameter aerial-satellite registration can be obtained from a two dimensional displacement (dx, dy) of the rotated and scaled aerial image with respect to the satellite image. For example, a displacement of $(0, 0)$ means the center of the rotated and scaled video image is overlapped with the center of the satellite map.

Using these approximations, the desired registration is contained in the 2-dimensional space of displacements (dx, dy) described above. The problem that needs to be solved is therefore the following: given a pair $I = (A, S)$ of an aerial image A and a satellite image S together with the rotation and scale information from the GPS, find the two dimensional displacement $(dx, dy) \in [-M, M]^2$ that best registers the two images at the street level.

We approach this problem as a supervised learning problem, in which the training examples consist of pairs (I_i, u_i), with $I_i = (A_i, S_i)$ being a satellite-aerial pair and $u_i = (dx_i, dy_i)$ being some registration displacement. The positive examples contain the correct displacements $u_i = (dx_i, dy_i), i = 1, ..., N$ for the N image pairs $I_i = (A_i, S_i), i = 1, ..., N$. The negative examples will be obtained as $(A_{k(j)}, S_{k(j)}), (dx, dy)$ such that $k(j) \in \{1, ..., N\}$ is random and (dx, dy) is any value such that $(dx, dy) \neq (dx_{k(j)}, dy_{k(j)})$.

Our database consists of two video sequences containing 2300 frames and about 230 satellite images obtained from Google [5] from the GPS locations. Therefore, there is one satellite image to every 10 video frames.

We manually annotated the true registration of the frames and divided the dataset into two disjoint sets: a training set of 1800 frames and a test set of 1500 frames.

III. Learning Image Registration by AdaBoost

A trained classifier that can separate the positives from the negatives described in the previous section would be able to answer for each pair I, u whether the displacement $u = (dx, dy)$ is the actual registration between the two images in $I = (A, S)$. Hence, the classifier behaves like a similarity function that gives a probability that the aerial image is registered with the satellite given a displacement $u = (dx, dy)$. In order to find the true registration between the aerial image A and the satellite image S, the classifier has to be queried for all possible locations $u \in [-M, M]^2$, as illustrated in Figure 4. Even though this is feasible for the two dimensional registration, it becomes computationally prohibitive for the affine registration, where u will be a six-dimensional vector, and its search space will be very large.

This approach faces this computational challenge because the search task is separated from the learning task, limiting the types of features that can be used in the classifier design to features that can be applied everywhere in the search space. However, there exist many other types of features that do not satisfy this criterion and are automatically excluded from the feature pool. Examples include edge detection results, interest points, morphological operations, Hough transforms, etc.

In this paper, we propose to extend the scope of the boosting framework by integrating the parameter space into the learning task and we can show how to obtain boosting results without any search in the parameter space.

In this different view that we propose, a feature together with its associated search becomes a weak algorithm that gives a response map in the search space. Then the most relevant algorithms to the task will be chosen by AdaBoost from the pool of weak algorithms using positive and negative samples. This generalizes the voting or bagging schemes to combine algorithms previously used in the literature by providing a principled way (supervised learning) to choose the right algorithms and their weights to obtain a robust result.

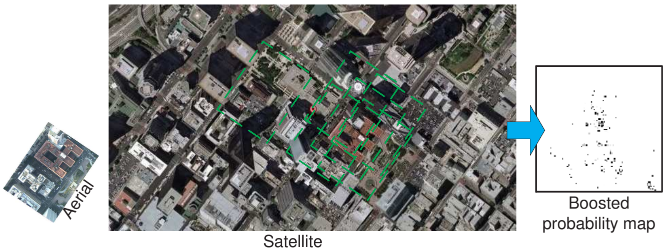

Fig. 4. For detection using Adaboost, the trained classifier must be queried at all locations in the parameter space, obtaining a probability map (right image). In this example, the classifier parameters are the (dx, dy) registration location of the aerial image.

In our image registration problem, we can use simple feature-based or phase correlation based registration algorithms as weak algorithms. From these weak algorithms a subset will be chosen by AdaBoost and the result will be a boosted algorithm that is more accurate than any of the algorithms in the pool.

A. Overview of Adaboost and the detection process

In this paper we will concentrate on the Adaboost [7] algorithm, but the same method can be applied to other boosting methods, for example LogitBoost [7]. Adaboost starts with a training set of positive ($s_i^+, i = 1, .., n^+$) and negative ($s_j^-, j = 1, .., n^-$) samples and a pool of features F_k. The training samples are aligned (e.g. rotated, scaled and cropped), and assigned a weight $w_i^+ = 0.5/n^+, w_j^- = 0.5/n^-$. The training algorithm proceeds in a greedy manner as follows:

AdaBoost Training

1. For each feature, a sign $\sigma_k \in \{-1, 1\}$ and threshold t_k are found to minimize the training error
$$\epsilon_k = \sum_i w_i^+ \delta[\sigma_k(F_k(s_i^+) - t_k) > 0] + \sum_j w_j^- \delta[\sigma_k(F_k(s_j^-) - t_k) < 0] \quad (1)$$
2. The feature k_1 with the smallest training error $e_1 = e_{k_1}$ is selected.
3. The feature weight is $\alpha_1 = -\log(e_1/(1.0 - e_1))$
4. For each sample, its weight is multiplied by $\beta_1 = \epsilon/(1-\epsilon)$ if it is misclassified, and divided by β_1 if it is well classified.
5. Steps 1-4 are repeated until the desired number N of features are selected.

For each chosen feature F_m, a corresponding weak classifier has been found, namely

$$h_m(s) = \delta[\sigma_m(F_m(s) - t_m) > 0] \quad (2)$$

The obtained classifier is

$$H(s) = \text{sign}(\sum_m \alpha_m h_m(s) > T) \quad (3)$$

When the classifier is used for detection, all parameters that were used to align the data (e.g. position, scale, rotation) must be found. All these parameters lie in a search space S of dimensionality equal to the number of parameters. For each set of parameters $s \in S$, the classifier is queried given the current image, and the probability for that location is recorded.

Thus the detection process can be written as follows:

AdaBoosting
1. For each parameter $s \in S$:
2. Compute features $F_m(s), m = 1, ..., N$.
3. Compute weak classifiers $h_m(s), m = 1, ..., N$.
4. Obtain detection result
$H(s) = \text{sign}(\sum_m \alpha_m h_m(s) > T)$

This search can also be performed in a coarse to fine fashion, for speed.

B. Boosting Feature Maps

Assuming that no coarse to fine search takes place (e.g. considering only one scale of the coarse to fine pyramid), the detection process involves the computation of all the features $F_m(s)$ of the classifier at each point s in the search space S. This is computationally equivalent to obtaining the response of each feature F_m on the whole space S. Thus the whole detection process can be written as follows:

Boosting Algorithms
1. For each $m = 1, ..., N$:
2. Compute the feature map $F_m(S) = \{F_m(s), s \in S\}$
3. Compute the weak classifier map by thresholding
$h_m(S) = \delta[\sigma_m(F_m(S) - t_m) > 0]$
4. Obtain the detection result in the whole search space
$H(S) = \text{sign}(\sum_m \alpha_m h_m(S) > T)$

In this view, a *weak classifier*, applied to the search space S, produces a thresholded feature map, as seen in Figure 5. This thresholded feature map can be considered as a *weak algorithm*. Other weak algorithms can be constructed that are not obtained by thresholding feature maps. For example, in our image registration problem, some of the weak algorithms will be interest point-based registration algorithms that obtain one or a few sparse locations in the whole search space.

There is little work that combines algorithms in a supervised way. In [11], the authors use supervised learning to choose one image segmentation algorithm from a database of algorithms and to find the optimal segmentation parameters. In contrast, our work is aimed at using a number of algorithms concurrently and train a more robust algorithm by supervised learning.

Observe that if the feature output $F_m(S)$ is not binary, it must be thresholded with a value t_m that is determined during training.

$$h_m(S) = \delta[\sigma_m(F_m(S) - t_m) > 0] \quad (4)$$

This is equivalent to a using a potential function [15] to correctly interpret the feature value. Without the threshold (or without the potential function in general), overly confident features can have an uncontrolled influence on the boosted output.

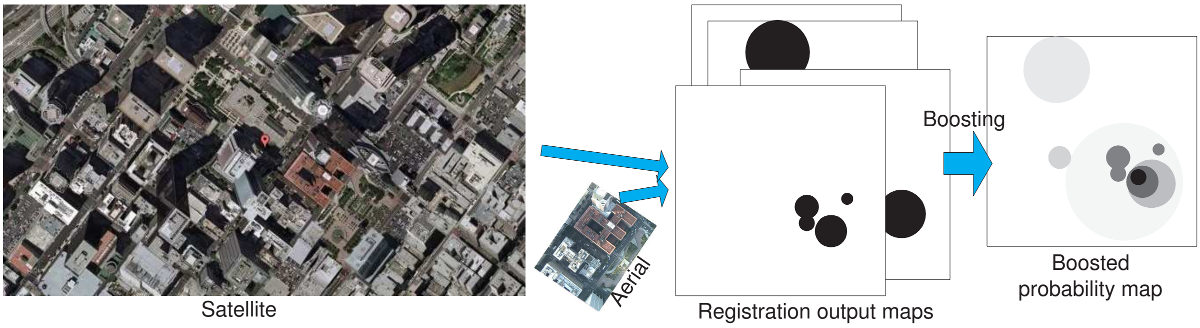

Fig. 5. Equivalently, each weak classifier can be applied in the whole search space obtaining a number of weak classifier maps. Detection map can be obtained from the weak maps by boosting (i.e. as a weighted sum of thresholded maps). Other fast algorithms can be used to provide such weak maps, for example feature-based registration algorithms in the case of image registration.

Also observe that the boosted algorithm outputs a discriminative probability map [7]:

$$P(S) = \frac{e^{h(S)}}{e^{h(S)} + e^{-h(S)}}, \quad , h(S) = \sum_m \alpha_m h_m(S) \tag{5}$$

Assume now that we are given a number of weak algorithms $A_k^{\theta_k}$, represented by their binary output in the space S and controlled by a set of parameters θ_k.

Training an Adaboost classification algorithm in this case is performed as follows:

Training a Boosted Algorithm

1. for each algorithm $A_k^{\theta_k}$, find parameters θ_k to minimize the training error

$$\epsilon_k = \sum_i w_i^+ \delta[A_k^{\theta_k}(s_i^+) = 1] + \sum_j w_j^- \delta[A_k^{\theta_k}(s_j^-) = 0] \tag{6}$$

2. Select the algorithm $A_{k_1}^{\theta_\cdot}$ with the smallest training error $\epsilon_1 = \epsilon_{k_1}$.
3. Its weight is $\alpha_1 = -\log(\epsilon_1/(1.0 - \epsilon_1))$
4. For each sample, multiply its weight by $\beta_1 = \epsilon/(1-\epsilon)$ if it is misclassified, divide the weight by β_1 if it is well classified.
5. Repeat steps 1-4 until the desired number N of classifiers are selected.

In what follows, we will use a number of feature-based and phase correlation based registration algorithms to train a robust registration algorithm that has a registration error about 20-30% smaller than the best algorithm in the pool.

C. The Algorithm Pool for Registration Boosting

From the precision of the GPS data and the grid of the satellite maps, we observed that the search for the registration displacement (dx, dy) can be restricted to a 200×200 pixel window centered at the map image center. Thus each weak algorithm will give a 200×200 pixel response.

The pool of weak registration algorithms from which the boosted algorithm will be trained contains phase-correlation based algorithms and feature-based registration algorithms. However, because of the cross-modality of the problem, the phase correlation algorithms provide a very weak output for our problem, as shown in Figure 2 and they were never selected in the boosting process.

The feature-based registration algorithms are composed of:

1) A *feature* part that finds a sparse set (300-1000) of interest points in each of the two images.
2) A *matching* part that is based on a set of local descriptors (e.g. SIFT) assigned to each interest point, a similarity measure to compare the local descriptors (e.g. SSD) and an algorithm to match the interest points of the two images, based on the descriptors and the similarity measure.
3) A method to generate *registration hypotheses* based on the obtained matches (e.g. RANSAC).

In our case, the interest points are picked using either the method described by Lowe [2] or Sojka [12], using different parameter values.

The local descriptors are based on gradient and intensity, as in [2]. As a consequence, the SIFT descriptors are among the descriptors used in the algorithm pool. The similarity measure is either based on SSD or VOD (variance of difference). The algorithm for matching interest points is the exhaustive comparison of all the interest points, but each match P_v of the video is restricted to have the displacement in the $(-100, 100)^2$ range.

In our boosting framework, each algorithm should generate a binary output in the search space. Each two-dimensional registration hypothesis $d(P, Q)$ generated by a weak algorithm has a degree of support from the interest point matches, i.e. the number $n(P, Q)$ of matches $P_i \to Q_i$ whose displacement $d(P_i, Q_i)$ is close to $d(P, Q)$, i.e. $|d(P_i, Q_i) - d(P, Q)| < \alpha$. The hypotheses are sorted in the decreasing order of the support. Then, each weak algorithm is based on a subset of the best hypotheses $d_i = (u_i, v_i)$ obtained from the matches. Here are some examples of combinations of hypotheses that we use:

1) The hypothesis with the largest support.
2) The hypothesis with the k-th largest support $k = 2, ..., 10$.
3) The n best hypotheses, $n = 2, ..., 10$.

Each of the combinations of hypotheses described above results in a weak algorithm (i.e. a binary map in the search space), having a number of control parameters $r_1, ...r_h$, where h is the number of hypotheses. The weak algorithm is a binary output obtained by placing disks

$$D_i(x) = I(|x - d_i| \leq r_i) \tag{7}$$

in the 200×200 search space at the locations given by the hypotheses, obtaining binary outputs as shown in Figure 6.

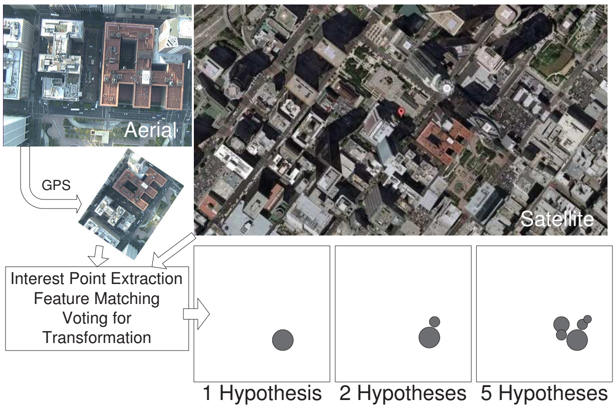

Fig. 6. The weak algorithms are obtained by matching different types of features (SIFT, Sojka) using different criteria (SSD, VOD) and different matching algorithms (exhaustive, RANSAC). The output can have one up to ten hypotheses.

This is only one way to construct a feature map from a sparse set of hypotheses, but there could exist other appropriate ways, depending on the problem at hand. The best control parameters $r_1, ..., r_h$ are obtained online during training, in a greedy manner in order to minimize the training error.

As one could see, there are many parameters and choices used to construct the weak algorithms. By choosing different combinations of parameters, and together with phase correlation algorithms with different filtering kernel types and sizes, we obtain about 60 weak algorithms that will be the pool from which the boosted registration algorithm will be trained.

Other registration algorithms [8], [9], [13] could also be included in the algorithm pool if desired.

D. Training the Boosted Registration Algorithm

To train the boosted algorithm, positive and negative examples must be generated.

There is one positive example for each aerial-satellite image pair (A, S), with the displacement $u = (dx, dy)$ corresponding to the true registration between the two images, obtained by manually registering the images.

For each aerial-satellite image pair (A, S), 1000 negative samples $u_i = (dx_i, dy_i)$ were randomly selected in the $200x200$ pixel search space, away from the positive example for that pair. To avoid large registration errors, a combination of the following two techniques can be used when generating the negative samples:

1) Implement a cascade of increasingly complex algorithms. A first boosted registration algorithm is trained with negatives that are far from the positives (e.g. at distance at least 30). This training will be easy. The false alarms obtained from this first stage whose distance to the corresponding positive is slightly smaller (e.g. at least 20) are used as negatives for another boosted algorithm. This procedure can be repeated several times until no improvement is observed.
2) Assign non-homogeneous weights to the negative samples. Samples that are at higher distance from the ground truth have higher weights.

The boosted registration algorithm was trained to contain 15 weak algorithms, in order to obtain a balance between speed and accuracy. An interesting observation is that the algorithm with the smallest error is Lowe's registration algorithm (defined as using SIFT features, 128 bit SIFT descriptors, SSD similarity measure and choosing the best match) only when training on areas with low parallax distortions. When training on the entire sequence containing large parallax distortions, the first algorithm that was selected is similar to Lowe's algorithm, but it outputs a map containing the 10 best hypotheses.

Some examples of probability maps obtained using the boosted algorithm are shown in Figure 7.

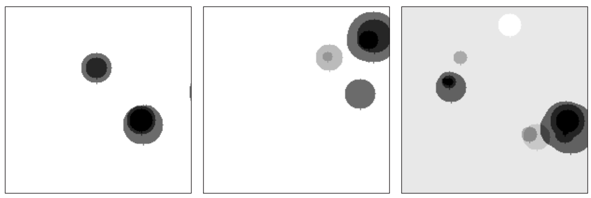

Fig. 7. Probability maps obtained by boosting the weak algorithms.

To select the final registration result, we find the regions of highest value in the probability map and we choose the location furthest away from the region boundary as the final registration output.

Time complexity. If the pool contains algorithms with a large range of computational complexities (e.g. phase correlation based and mutual information based algorithms), it is recommended to take a cascaded approach where in the first cascade level only the fast algorithms are boosted for an initial registration. Only on the locations where the maximum probability is over a threshold, a more powerful yet more expensive boosted algorithm is used to obtain the final registration output.

TABLE I
REGISTRATION ERROR OF THE REFERENCE (LOWE'S) ALGORITHM AND OUR BOOSTED ALGORITHM ON TRAINING AND TESTING DATA.

Algorithm	Median	80%	Max
Lowe, Dataset 1	5.01	7.45	23.70
Lowe, Dataset 2	5.62	7.73	16.70
Training, Dataset 1	4.76	7.10	21.64
Testing, Dataset 2	4.57	6.93	13.59

E. Experimental Validation

To test the robustness of our algorithm, we performed cross-validation by training on one sequence (Dataset 1) and testing on the other sequence (Dataset 2). The results are summarized in Table I.

As one could see, the error of the boosted algorithm is 18% smaller than the error of the Lowe's algorithm.

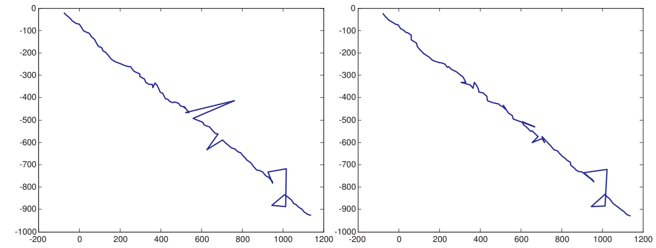

Fig. 8. Registration trajectories obtained using Lowe's registration (left) and the boosted algorithm (right). The large error at the bottom is due to large parallax of tall buildings that occupy the entire image.

For a qualitative assessment of the registration, we present in Figure 8, trajectories obtained using the Lowe's algorithm (left) and using our boosted algorithm. We see that except for one large error, the other errors have been greatly reduced. The large error is due to the very tall buildings shown in Figure 1.

The GPS trajectory and the manual annotation of the displacements dx, dy are shown in Figure 9.

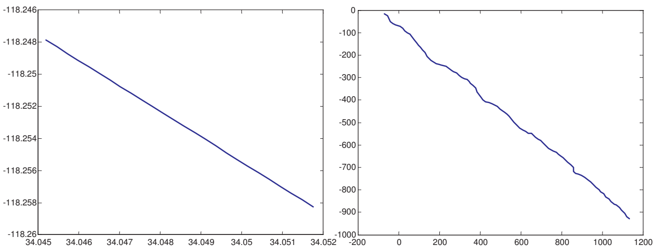

Fig. 9. GPS trajectory of the plane (left) and manually annotated registration trajectory relative to the first frame (right).

Since the plane is in motion, it makes sense to use a Kalman filter to obtain the final registration result. The results obtained by running a Kalman filter on Lowe's registration and on our boosted registration are shown in Figure 10, where the superiority of the boosted registration is even more evident.

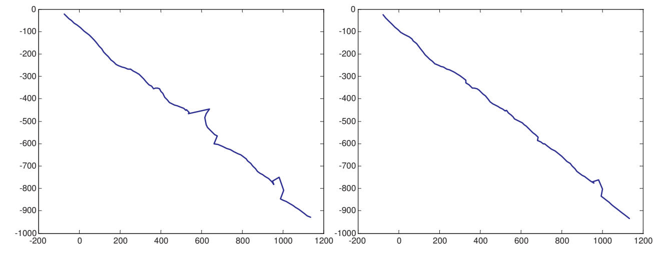

Fig. 10. Registration trajectories obtained using Lowe's registration (left) and the boosted registration algorithm (right), after smoothing with a Kalman filter.

IV. Extension to Higher Dimensional Spaces

If the GPS information is not available, the entire boosted registration framework can be extended to 6-parameter affine registration. Moreover, the concepts presented in this paper can be directly applied to any higher dimensional rigid/non-rigid parametric registration.

We represent a 6-parameter affine transformation by the displacements $(dx_i, dy_i), i \in \{1, 2, 3\}$ of three predefined points from the image (e.g. the image center and two corners), as shown in Figure 11. This way, the affine space can be viewed as isotropic, with all dimensions having the same unit of measure.

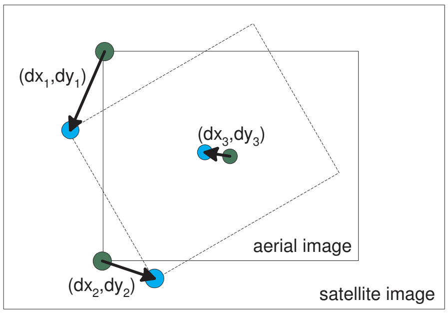

Fig. 11. An affine transformation is represented by the displacements (dx_i, dy_i) of three predefined points such as the image center and two adjacent corners.

Searching the six-dimensional registration space for the correct registration is too computationally expensive to be handled directly. In what follows, we present a new method to obtain the boosted registration map without discretizing the search space.

A. Virtual Boosting: Boosting Without Search

When the search space is high dimensional, memory and computation restrictions prohibit handling discrete versions of the outputs of the weak algorithms or the boosted probability maps. In some cases, the weak algorithm output can be represented compactly as a number of parametric objects. For example, the feature-based registration algorithms output a number of overlapping hyperspheres, which can be represented parametrically using their centers and radii.

The boosted log-probability map is a weighted sum of the weak algorithm outputs, so it can be again expressed parametrically as a number of weighted weak algorithm outputs, where in our case each such output is a number of overlapping hyperspheres.

To obtain the final registration location, we need to find the intersection of hyperspheres where sum of weights is maximum. Then we need to find the location inside this intersection that is furthest away from the boundaries, for robustness. In some cases one could perform an exhaustive search on a discrete grid and at each location find the hyperspheres that contain it and finally compute the boosted log-probability. But this is computationally intensive and it can become impractical if the parameter space is too large.

Instead, we propose another solution that avoids any search in the high dimensional space.

The intersection of hyperspheres whose sum of weights is maximum can be mapped to the *Maximum Weighted Clique* problem from combinatorial optimization:

1) Construct a graph whose nodes are the hyperspheres, having edges between hyperspheres that have non-zero intersection
2) The area of maximum probability is the intersection corresponding to the maximum weight clique if the intersection is non-empty.

In practice, we use the maximum weight clique algorithm from [14].

To find the final registration location inside the area of maximum probability, we rely on the convexity of the hyperspheres. Define the following cost function as the sum of distances to the hyperspheres $H_1 = (C_1, R_1), ..., H_k = (C_k, R_k)$ with centers C_i and radii R_i:

$$C(x) = \sum_i \max(||x - C_i|| - R_i, 0) \geq 0 \tag{8}$$

Clearly a location x is inside the intersection of the hyperspheres if and only if $C(x) = 0$. To find one such location, we use Powell optimization starting from the center of gravity of the intersections of two hyperspheres. Because $C(x)$ is a convex function, being the sum of convex functions, the algorithm is guaranteed to converge to the global minimum. If the hyperspheres intersect, that global minimum is zero.

To find the location inside this intersection of hyperspheres that is as far away from the boundary as possible, we alternatively find a location inside the intersection of hyperspheres and decrease the radii of all hyperspheres by 1 (remember that the affine transformation space is isotropic because it is represented as the displacements (dx_i, dy_i) of three predefined points). The algorithm is stopped when the minimum of $C(x)$ is not zero. Because of the simplicity and convexity of the cost function, finding the intersection location is very fast (about 0.2 seconds on a standard PC).

B. Experimental Validation of the Affine Registration

For the affine registration, we only used the feature-based weak algorithms. To balance between accuracy and computational cost, each weak algorithm obtains registration hypotheses from 1000 RANSAC iterations.

The training and testing datasets are the same as for the two-parameter registration. In Table II are displayed the performance of the standard Lowe's registration algorithm with RANSAC and our boosted registration method, on the training and testing (unseen) dataset. On the unseen dataset, the boosted registration observes a 30% smaller error than Lowe's algorithm, for the six-parameter affine registration.

TABLE II
REGISTRATION ERROR OF THE REFERENCE (LOWE'S) ALGORITHM AND OUR BOOSTED ALGORITHM FOR THE 6-PARAMETER AFFINE REGISTRATION.

Algorithm	Median	80%	Max
Lowe, Dataset 1	12.77	27.09	50.89
Lowe, Dataset 2	19.30	28.70	56.58
Training, Dataset 1	9.05	16.07	40.32
Testing, Dataset 2	15.98	20.06	39.81

V. CONCLUSIONS

In this paper we proposed a supervised learning framework for training a robust registration algorithm using a pool of inexpensive feature-based registration algorithms, all designed for the same task.

The same method can be used to train a robust tracking algorithm, in which case the phase correlation algorithms might be very effective. Moreover, registration to a set of representative views can also be incorporated in order to avoid drift.

In general, many problems have solutions that can be represented using a fixed number of parameters. The boosting framework presented in this paper can be used to combine multiple algorithms or variants designed to solve the same task to obtain a more robust result.

REFERENCES

[1] L. G. Brown. A survey of image registration techniques *ACM Computing Surveys (CSUR) archive* **24**, 4 (1992), pp 325 - 376

[2] M. Brown and D. G. Lowe. Recognising Panoramas. *ICCV*, 1218-1225, 2003

[3] X. Dai, Le Lu, G. Hager, Real-time Video Mosaicing with Adaptive Parameterized Warping, *CVPR* Demo Program, 2003

[4] E. De Castro , C. Morandi. Registration of translated and rotated images using finite Fourier transforms, *IEEE Trans. PAMI*, **9**, 5, pp.700-703, (1987)

[5] Google Maps. http://maps.google.com

[6] C Guetter, C Xu, F Sauer, J Hornegger - Learning based non-rigid multi-modal image registration using Kullback-Leibler divergence - MICCAI 2005

[7] J. Friedman, T. Hastie and R. Tibshirani, Additive logistic regression: a statistical view of boosting, Dept. of Statistics, Stanford Univ. Technical Report. 1998.

[8] D. J. Hirvonen, B. Matei, R. P. Wildes, S. C. Hsu: Video to Reference Image Alignment in the Presence of Sparse Features and Appearance Change. *CVPR* **2** 2001: 366-373

[9] M. Irani and P. Anandan. Robust multi-sensor image alignment. *ICCV* 1998 pp. 959966

[10] C.D. Kughn, and D.C. Hines. The phase correlation image alignment method. *Proc. IEEE Int. Conf. on Cybernetics and Society* pp. 163-165, (1975)

[11] V. Martin, M. Thonnat, N. Maillot. A Learning Approach for Adaptive Image Segmentation *ICVS* 2006.

[12] E. Sojka. A new algorithm for detecting corners in digital images. *18th Spring Conference on Computer Graphics (SCCG)* 2002

[13] N Prabal, P. Brian, N. Smith, Jody L.; Decker, Max L. Edge-based correlation image registration algorithm for the Multispectral Thermal Imager (MTI).*Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery X.* Edited by Shen, Sylvia S.; Lewis, Paul E. Proceedings of the SPIE, Volume 5425, pp. 197-207 (2004).

[14] P. R. J. Ostergard, A new algorithm for the maximum-weight clique problem, *Nordic Journal of Computing* **8** (2001), 424-436

[15] S. C. Zhu, Y. N. Wu and D.B. Mumford, FRAME: Filters, Random field And Maximum Entropy: Towards a Unified Theory for Texture Modeling, *IJCV*, 27(2) pp.1-20, 1998.

[16] WM Wells, P Viola, H Atsumi, S Nakajima, R Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1996

[17] B. Zitova, J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, **21**, 11, pp. 977-1000 (2003).