Optimization of Background Subtraction for Image Enhancement

Larry Venetsky, Ross Boczar, Robert Lee-Own

Advanced Technologies Branch, Code 4.8.1.4
NAVAIR, Naval Air Engineering Center
Lakehurst, NJ 08733

ABSTRACT

Analysis of foreground objects in scenery via image processing often involves a background subtraction process. This process aims to improve blob (connected component) content in the image. Quality blob content is often needed for defining regions of interest for object recognition and tracking. Three techniques are examined which optimize the background to be subtracted - genetic algorithm, an analytic solution based on convex optimization, and a related application of the CVX solver toolbox. These techniques are applied to a set of images and the results are compared. Additionally, a possible implementation architecture that uses multiple optimization techniques with subsequent arbitration to produce the best background subtraction is considered.

Keywords: background subtraction, image processing, genetic algorithm, convex optimization, CVX

1. INTRODUCTION

Background subtraction is a popular machine vision technique for extracting foreground objects in an image. These regions of interest may be extracted for better contextual understanding of an image.[1] This technique is useful in applications where the image acquisition cameras have a constant focal length and do not pan and tilt, thereby keeping the camera focused on a stationary background with moving foreground objects. There are many techniques aimed at finding the best possible background to subtract from the incoming image stream in order to obtain the objects of interest for further processing.[2] Zivkovic and Heijden proposed an adaptive Gaussian mixture model toward this task,[3] while Elgammal et al. used a nonparametric density estimation approach.[4] Previous work, based on an adaptive contrast adjustment scheme, attempted to account for changes in the scene and update the background accordingly.[5] The method proved unsuccessful in cases where the ambient lighting conditions changed too rapidly. Thus, static optimization methods which leveraged metrics of effectiveness used in the previous work were proposed.

Three methods for background optimization were examined with the goal of finding a more robust de-noising architecture consisting of the fusion of several background synthesis techniques — genetic algorithm, an analytical solution based in convex optimization, and a related CVX solver toolbox implementation.

These algorithms were used for the application of extracting foreground objects from aircraft carrier deck images. The objects were typically aircraft, related support equipment, or personnel. The incoming image frames were composed of multiple sub-images ('cells') with sometimes very large differences in pixel intensities. This is because each cell was produced by a different camera under varying ambient lighting conditions as well as varying degrees of wear and tear from deployment.

Figure 1 illustrates the results of our algorithm. The first row of images illustrates typical deck scenery with aircraft positioned at different locations on the deck. The second row shows both the original and optimized background images, while the third row demonstrates foreground extraction via subtraction of the background from the input image. The goal of this research was to examine several optimization techniques and understand how to use these techniques in concert by taking advantage of the best performer at various deck conditions.

Section 2 illustrates the use of the genetic algorithm, while Section 3 provides a framework and the results for convex optimization solutions to this background optimization problem.

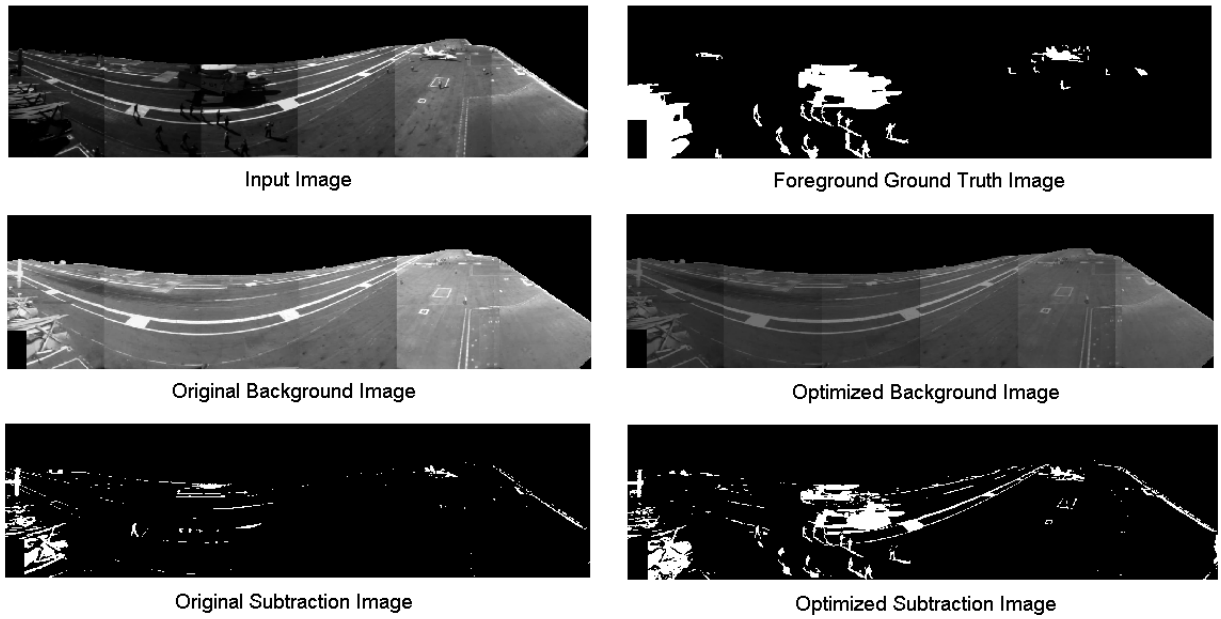

Figure 1. Unoptimized vs. optimized image subtraction.

2. GENETIC ALGORITHM

The genetic algorithm introduced by J. Holland in the 1970s became a very popular technique for solving many optimization problems where a large number of sometimes conflicting criteria needed to be considered at the same time. The algorithm was inspired by evolutionary theory as applied to optimization over a given search space. A fitness function is needed to direct the drive for improved performance for the entire population of candidate solutions. The genetic algorithm optimizes over a n-dimensional search space using a random search biased toward the most promising area in the space. This reduces the optimization time compared to a purely random search for a globally or locally optimal solution.[6]

The canonical genetic algorithm starts with initialization of several candidate solutions (known as the population) to either a random or predetermined state. A fitness function is then used to evaluate the goodness of each member of the population. The selected fitness function allows for a tournament in which individuals with higher fitness have higher probability of survival in the competition, thus simulating the survival of the fittest concept in natural selection.[7] Next, the survivors of the competition go through the crossover mating process that establishes new offspring, modeling generational trait inheritance in nature. To avoid the convergence of the entire population to the current best solution (a local maxima), Holland proposed the mutation process to impose diversity in each generation of the population. The mutation randomly modifies coding bits (simulating DNA mutation) with some generally low probability. Mutation has two consequences: diversification of the population and better exploration of the search space, allowing for discovery of other promising solutions. The new generation of the population resulting from the crossover and mutation processes will again be evaluated by the fitness function, enter tournament, and continue the genetic algorithm optimization until some stopping criteria are met. The genetic algorithm shown in Figure 2 is guaranteed to increase the average fitness of the population by virtue of the schema theorem.

The essence of the schema theorem (1), first proposed by Holland as the fundamental theorem of genetic algorithms, is that schema with high fitness values and small bit string lengths grow exponentially with time. This guarantees improvement of the average fitness of the population as a whole.[6]

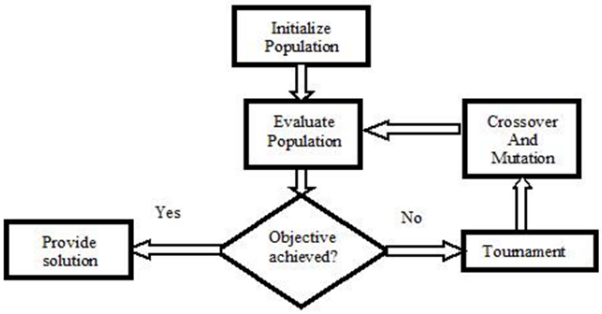

Figure 2. Canonical genetic algorithm.

$$\mathbb{E}\left[N_{h,t+1}\right] \geq N_{h,t}\frac{f_h}{\bar{f}_t}\left[1 - p_c\frac{\delta_h}{l-1} - o_h p_m\right] \tag{1}$$

$\mathbb{E}\left[N_{h,t+1}\right]$ – expected number of instances of schema h with high fitness value in generation $t+1$
$N_{h,t}$ – number of instances of schema h with high fitness value in generation t
f_h – observed fitness of schema h in generation t
$\bar{f}_t$ – average fitness of population in generation t
p_c – crossover probability
δ_h – defining length of schema h
l – number of bit positions schema h
o_h – order of schema h (number of fixed positions)
p_m – mutation probability

An essential part of optimization by genetic algorithm was evaluating each individual by a fitness function. In our application, fitness was a combination of criteria representing the potential correlation between the input image and the candidate background to be subtracted.

2.1 Notation

The vectors $\mathbf{f}, \mathbf{x} \in \mathbb{R}_+^6$ represent the average cell pixel intensities of the N-cell input image and N-cell background image respectively ($N = 6$), where grayscale pixel values range from 0 to 255. The vector $\mathbf{m}$ represents the fitness function where m_j is the jth component of the fitness function. The vector $\mathbf{1}$ represents $[111\cdots1]^T$. $||\cdot||_p$ is the p-norm.

2.2 Fitness function

In the previous work,[5] close inspection revealed that an indicator of whether or not the method would be able to account for the changes was the difference between the mean pixel intensity of the input scene and the mean pixel intensity of the current background. If the deviation was minimal, the approach would successfully modify the background to match the input and produce a clear blob image upon subtraction and post-processing. With a larger deviation between the input scene and working background the adaptive system would nearly always fail. Deviation of mean pixel intensity appeared to be such a strong indicator of effectiveness that it was theorized that a fitness function could be constructed using pixel intensity as a metric for judging predicted performance of a background subtraction and post-processing scheme.

Three criteria were defined for evaluating the match of the candidate background to the input image. Our objective was to find optimal image intensities for the background cells $\{x_i\}$, to which the original background image could be adjusted to match.

The first fitness component (2) is the difference in average frame intensity between the input and background frames. This component ensures the two images were relatively close in average intensity.

$$\begin{aligned} m_1 &= \left| \frac{1}{N} \sum_{i=1}^{N} f_i - \frac{1}{N} \sum_{i=1}^{N} x_i \right| \\ &= \frac{1}{N} \left| \mathbf{1}^T (\mathbf{f} - \mathbf{x}) \right| \end{aligned} \tag{2}$$

The second fitness component (3) is the average cell-by cell intensity difference magnitude.

$$\begin{aligned} m_2 &= \frac{1}{N} \sum_{i=1}^{N} |f_i - x_i| \\ &= \frac{1}{N} ||\mathbf{f} - \mathbf{x}||_1 \end{aligned} \tag{3}$$

The third fitness component (4) represens the deviation of each background cell from the average of the entire background image.

$$\begin{aligned} m_3 &= \frac{1}{N} \sum_{i=1}^{N} \left| x_i - \frac{1}{N} \sum_{j=1}^{N} x_j \right| \\ &= \frac{1}{N} \left|\left| \mathbf{x} - \frac{1}{N} \mathbf{1}^T \mathbf{x} \mathbf{1} \right|\right|_1 \\ &= \frac{1}{N} \left|\left| \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{x} \right|\right|_1 \end{aligned} \tag{4}$$

The m_3 component reduces the standard deviation of the background solutions in an attempt to avoid overfitting to the input image. This is under the assumption that the background image should be relatively uniform, and that intensity variations in the input image correspond to the presence of foreground objects. For example, this component also ensures that a background cell would not be incorrectly brightened to match a corresponding input cell with a large number of bright objects if the rest of the cells were darker.

Additionally, a fourth object-filtering component, shown by (5), was defined. This fitness component is defined by a ratio of the number of large objects (aircraft- and support equipment-sized blobs) to the number of small objects (typically noise). The object filtering combined with the first three components led to a composite fitness function that aimed to copmletely describe the optimization.

$$m_4 = \alpha \log \left(\frac{1 + \sum_{i=1}^{N} L_i}{1 + \sum_{i=1}^{N} S_i} \right) \tag{5}$$

where

$$\begin{aligned} \alpha &- \text{ scale factor} \\ L_i &- \text{ number of large objects in cell } i \\ S_i &- \text{ number of small objects in cell } i \end{aligned}$$

The threshold between large and small objects was defined as 5 percent of the total number of pixels containing available deck area in a given cell. This threshold was derived empirically from the available data. This threshold must be determined by the specific application at hand.

The first set of experiments were performed without the object filtering, using only the fitness function $g(\mathbf{m})$.

$$g(\mathbf{m}) = -\log \left(\sum_{i=1}^{3} \exp m_i \right) \tag{6}$$

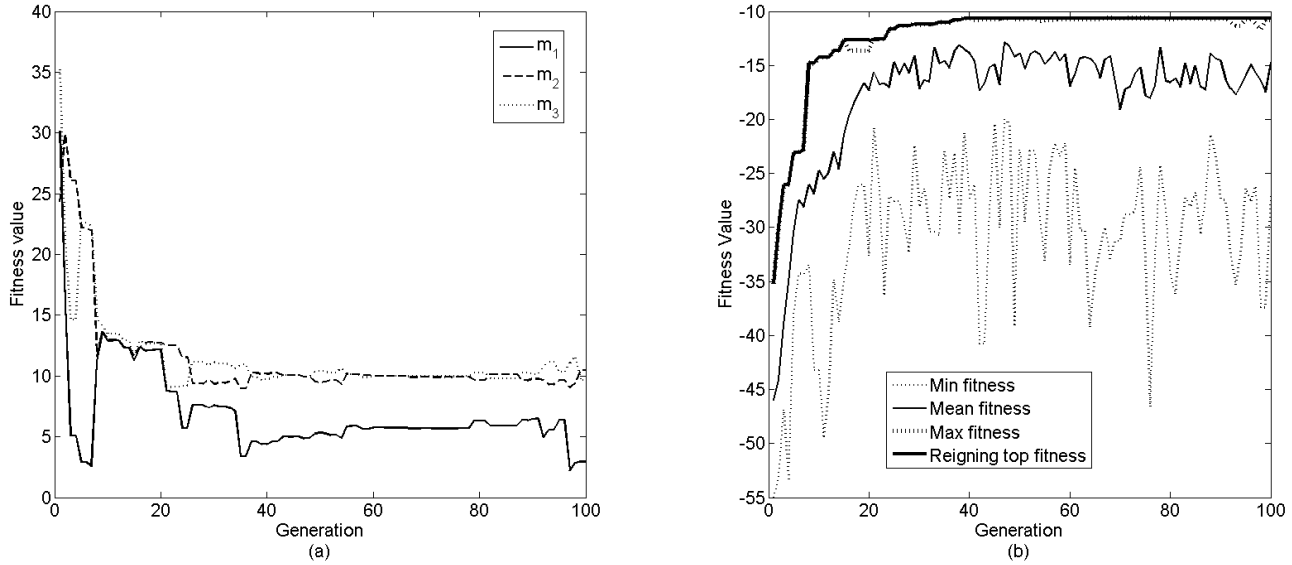

Figure 3. (a) Fitness components of best in population; (b) Overall population fitness.

The fitness function, known as the (negative of the) log-sum-exp function, is an analytic approximation of the max function. It can be easily shown that

$$\max x_i \leq \log\left(\sum_{i=1}^{n} \exp x_i\right) \leq \max x_i + \log n \tag{7}$$

This fitness function was chosen primarily because of its convexity[8] and its tendency to force the genetic algorithm to adjust the worst performing fitness component at each generation step.

Next, the fourth fitness component was incorporated, as shown in (5), to weight the optimization more toward solutions that revealed large objects, resulting in a new fitness function $h(\mathbf{m})$:

$$h(\mathbf{m}) = g(\mathbf{m}) + m_4 \tag{8}$$

$$= -\log\left(\sum_{i=1}^{3} \exp m_i\right) + \alpha \log\left(\frac{1+\sum_{i=1}^{N} L_i}{1+\sum_{i=1}^{N} S_i}\right) \tag{9}$$

2.3 Methodology and results

Figure 3 shows the results of a typical genetic algorithm optimization run. Figure 3(a) shows the progression of the fitness vector $\mathbf{m}$ over successive generations. Figure 3(b) shows an increase in population fitness as optimization process goes forward in time.

2.3.1 Noise thresholding

After subtraction, the algorithms returned (typically noisy) grayscale images that needed to be converted into a foreground/background binary image. Though post-subtraction noise removal and thresholding were not the focus of this research, an implemented algorithm was necessary in order to evaluate the subtraction methods. Thus, a modification of Otsu's method (minimization of intra-class variances) was used.[9] The subtracted images were transformed according to the power law seen in Figure 4. With normalized mean pixel intensity denoted by μ, contrast was increased via contrast adjustment on above-average intensity pixels only according to the law $f(x) = \left(\frac{x-\mu}{1-\mu}\right)^{\gamma}$. The new image was then passed through Otsu's method in MATLAB in order to create a binary image. Limitations of this algorithm are discussed in the Conclusion.

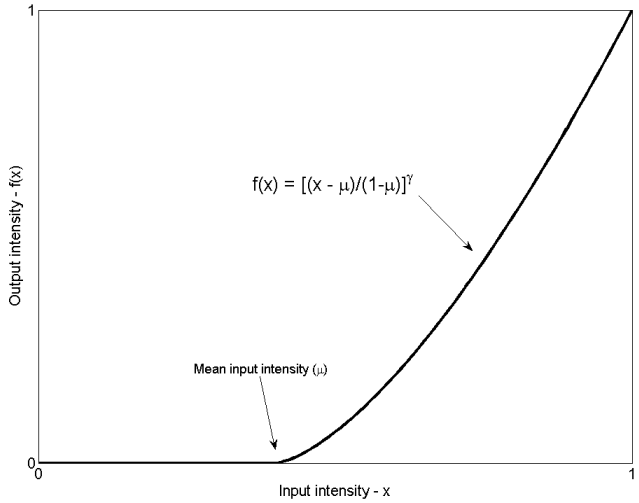

Figure 4. Noise thresholding.

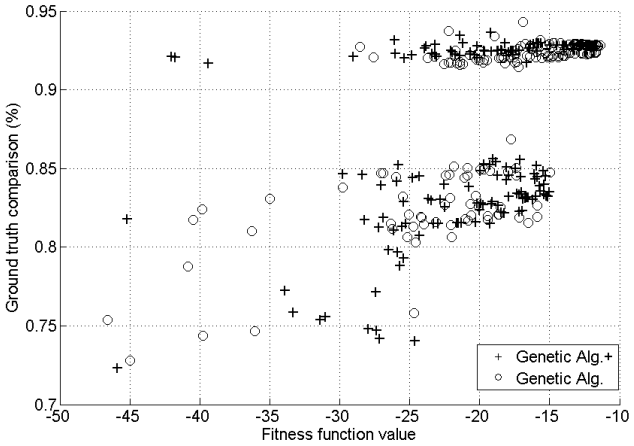

Figure 5. Validation of fitness function.

2.3.2 Performance metric

Optimal foreground images were manually generated using the GIMP image editing program. These "ground truths" were the reference images for the results of each optimization method. Both the result of the optimization and the ground truth image were binary images, so the metric of performance was the percentage of matching pixels between optimized and ground truth images. After running the algorithms over a variety of deck densities (defined by percentage of deck space occupied by foreground objects) and lighting conditions, it was possible to consider an arbitration algorithm based on the best performing algorithm.

2.3.3 Fitness verification

Figure 5 confirms the validity of the optimizations for both genetic algorithm implementations (three- and four-component fitness functions). Each point on the graph represents the fitness value of an individual candidate solution at some point in the optimization process, for both the three-component (genetic algorithm) and four-component (genetic algorithm+) fitness functions. Roughly speaking, high fitness corresponds to a higher probability of high "true fitness," defined by comparison of the optimization result to the ground truth foreground image.

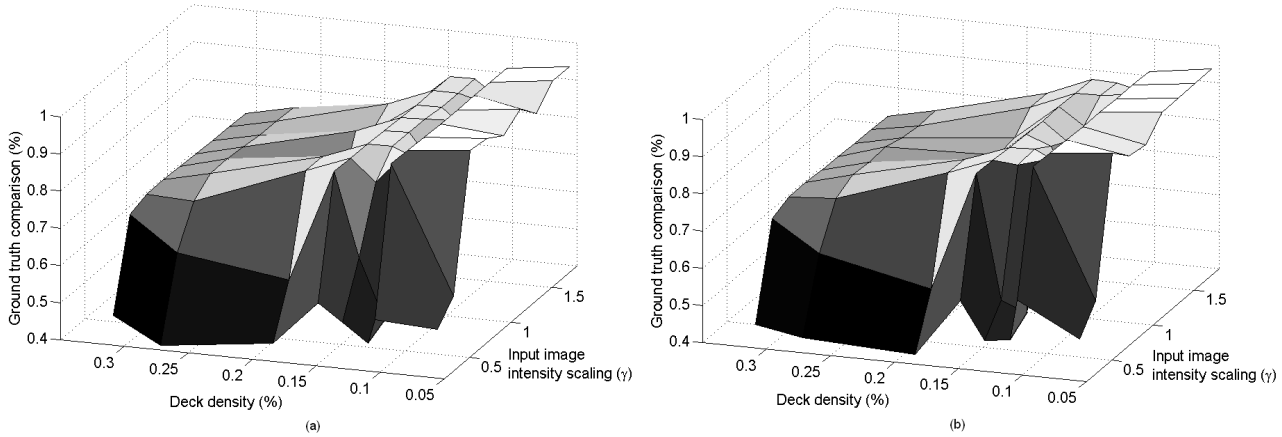

Figure 6. (a) Genetic algorithm (three-component) results; (b) Genetic algorithm (four-component) results.

2.3.4 Results

For testing the overall performance of the genetic algorithm, a set of eight images was used, each with varying deck conditions including deck object density. Changing the image intensities via gamma scaling (contrast adjustment) allowed for the creation of a set of 72 synthetic image. Gamma scaling brightens or darkens the image according to the law $f(x) = x^{\gamma}$, where $x \in [0, 1]$ represents pixel intensity and $\gamma \in (0, \infty)$ is the scaling parameter. Our images were adjusted over the range $\gamma = 0.2 - 1.8$. Figures 6(a) and 6(b) show the overall results contour for the GA and GA+ algorithms, respectively. Performance was calculated using comparison to ground truth.

The algorithms performed best at low deck density and gamma scalings corresponding to enhancing contrast/darkening of the image. It was interesting to note that addition of the fourth component of the fitness function only affected a small area of the results contour.

The parameters used in the genetic algorithm optimization are shown in Table 1.

3. CONVEX OPTIMIZATION

A convex optimization formulation of the background optimization problem is now presented. This forumlation can subsequently be then modified to admit the classical least-squares solution to a quadratic program (QP).

Table 1. Genetic algorithm parameters.

Parameter	Description	Value
number of generations	-	100
population size	-	15
string length (bits)	length of encoded string for each individual	24
number of elite individuals	highest-fitness individuals that skip tournament	3
mutation probability	-	0.015
crossover probability	-	0.75
tournament odds	probability that higher fitness wins in tournament	0.75

3.1 Convex formulation

Given the input and background vectors $\mathbf{f}, \mathbf{x}$, the problem of maximizing three-component fitness can be posed as an optimization problem:

$$\max_{\mathbf{x} \in \mathbb{R}_+^6} g(\mathbf{m}) \tag{10}$$

$$s.t. \qquad m_1 = \frac{1}{N} \left| \mathbf{1}^T (\mathbf{f} - \mathbf{x}) \right| \tag{11}$$

$$m_2 = \frac{1}{N} ||\mathbf{f} - \mathbf{x}||_1 \tag{12}$$

$$m_3 = \frac{1}{N} \left|\left| \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{x} \right|\right|_1 \tag{13}$$

Though our objective function is concave (equivalent to minimizing a convex function), the constraints (11)-(13) are nonconvex. However, due to the monotonicity of the objective function, the problem can be relaxed to

$$\max_{\mathbf{x} \in \mathbb{R}_+^6} g(\mathbf{m}) \tag{14}$$

$$s.t. \qquad m_1 \geq \frac{1}{N} \left| \mathbf{1}^T (\mathbf{f} - \mathbf{x}) \right| \tag{15}$$

$$m_2 \geq \frac{1}{N} ||\mathbf{f} - \mathbf{x}||_1 \tag{16}$$

$$m_3 \geq \frac{1}{N} \left|\left| \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{x} \right|\right|_1 \tag{17}$$

The problem (14)-(17) is now convex, and can be quickly solved using numerical methods. It can be shown that the problems (10)-(13) and (14)-(17) have the same optimal solution.

The optimization was solved using CVX, the convex solver toolbox for MATLAB developed in part by Stephen Boyd. CVX is a modeling system for disciplined convex programming (DCP). DCP involves convex optimization problems that are described using a limited set of construction rules, which enables the problems to be analyzed and solved efficiently.[10]

3.2 Analytic solution

If problem (10)-(13) is slightly modified, it admits an analytic solution. Two changes were made — the objective became a positive linear weighting of the metrics, and the constraints were changed to reflect squared-error. The consequences of both 2-norm and 1-norm error are well understood in the literature.[11]

$$\min_{\mathbf{x} \in \mathbb{R}_+^6} \mathbf{w}^T \mathbf{m} \tag{18}$$

$$s.t. \qquad m_1 = \frac{1}{N} \left(\mathbf{1}^T (\mathbf{f} - \mathbf{x}) \right)^2 \tag{19}$$

$$m_2 = \frac{1}{N} ||\mathbf{f} - \mathbf{x}||_2^2 \tag{20}$$

$$m_3 = \frac{1}{N} \left|\left| \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{x} \right|\right|_2^2 \tag{21}$$

Substituting the constraints into the objective and simplifying reveals a quadratic objective:

$$\begin{aligned}
\mathbf{w}^T\mathbf{m} &= w_1\left(\frac{1}{N}\left(\mathbf{1}^T(\mathbf{f}-\mathbf{x})\right)^2\right) \\
&\quad + w_2\left(\frac{1}{N}||\mathbf{f}-\mathbf{x}||_2^2\right) \\
&\quad + w_3\left(\frac{1}{N}\left|\left|\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathbf{x}\right|\right|_2^2\right) && (22)\\
&= \frac{w_1}{N}\left(\mathbf{x}^T\mathbf{1}\mathbf{1}^T\mathbf{x} - 2\mathbf{f}^T\mathbf{1}\mathbf{1}^T\mathbf{x}\right) \\
&\quad + \frac{w_2}{N}\left(\mathbf{x}^T\mathbf{x} - 2\mathbf{f}^T\mathbf{x}\right) \\
&\quad + \frac{w_3}{N}\left(\mathbf{x}^T\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)^T\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathbf{x}\right) \\
&\quad + \text{const.} && (23)\\
&= \mathbf{x}^T\hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{b}}^T\mathbf{x} + \text{const.}, && (24)
\end{aligned}$$

where

$$\begin{aligned}
\hat{\mathbf{A}} &= \frac{w_1}{N}\mathbf{1}\mathbf{1}^T + \frac{w_2}{N}\mathbf{I} \\
&\quad + \frac{w_3}{N}\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)^T\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right) && (25)\\
&= \frac{w_1}{N}\mathbf{1}\mathbf{1}^T + \frac{w_2}{N}\mathbf{I} + \frac{w_3}{N}\left(\mathbf{I}-\frac{2}{N}\mathbf{1}\mathbf{1}^T + \frac{1}{N^2}\mathbf{1}\mathbf{1}^T\mathbf{1}\mathbf{1}^T\right) && (26)\\
&= \frac{w_1}{N}\mathbf{1}\mathbf{1}^T + \frac{w_2}{N}\mathbf{I} + \frac{w_3}{N}\left(\mathbf{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right) && (27)\\
&= \frac{w_2+w_3}{N}\mathbf{I} + \left(\frac{w_1}{N} - \frac{w_3}{N^2}\right)\mathbf{1}\mathbf{1}^T, && (28)\\
\hat{\mathbf{b}}^T &= -\frac{2w_1}{N}\mathbf{f}^T\mathbf{1}\mathbf{1}^T - \frac{2w_2}{N}\mathbf{f}^T\,. && (29)
\end{aligned}$$

A further relaxation of the problem, by requiring $\mathbf{x} \in \mathbb{R}^6$ instead of $\mathbf{x} \in \mathbb{R}_+^6$, does not change the optimal solution. It can be easily shown for the relaxed problem that the optimal solution $\mathbf{x}^*$ satisfies

$$0 \le \min\{f_i\} \le x_i^* \le \max\{f_i\}, \forall\, i, \tag{30}$$

and is therefore the optimal solution of the original problem. Thus, this is an unconstrained quadratic optimization problem, whose solution corresponds to the least-squares minimizer of $||\hat{\mathbf{A}}\mathbf{x} - \hat{\mathbf{b}}||_2$. Under some conditions, the unique solution is

$$\mathbf{x}^* = -\frac{1}{2}\hat{\mathbf{A}}^{-1}\hat{\mathbf{b}}, \tag{31}$$

if $\hat{\mathbf{A}}$ is positive semidefinite.[11] This is satisfied by the choice of weights (see Appendix A).

3.3 Results

Figures 7(a) and 7(b) show the results contours for the CVX and analytic solutions, respectively. In our case, $\mathbf{w} = [111]^T$.

In general, the convex optimization solutions performed better than the genetic algorithm solutions at low gamma (low contrast) images. Neither method performed well on our high-object density image set.

Also, the convex optimization algorithms performed faster than the genetic algorithm solutions, as seen in Table 2. The algorithms were run on a Dell Core 2 Duo 2GHz laptop with 8GB RAM, running MATLAB 2011b. The majority of the GA+ processing time was a result of connected component processing for centroid identification and blob classification, due to the addition of the m_4 fitness component.

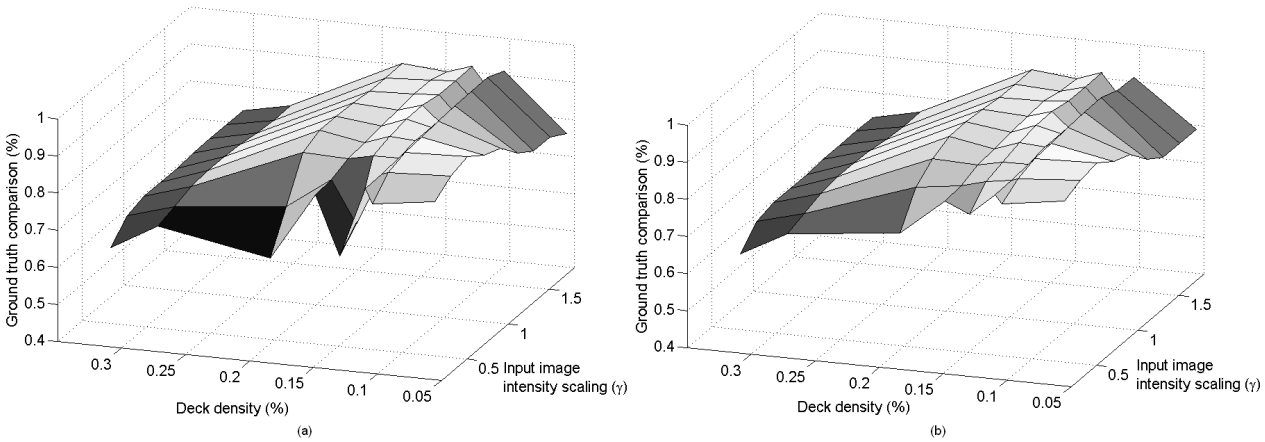

Figure 7. (a) CVX solver results; (b) Analytic solution results.

Table 2. Processing time

Algorithm	**Approx. Processing Time (s)**
GA (100 gen.)	10
GA+ (100 gen.)	150
CVX	1-2
Analytic	∼0

3.4 Overall Results

Using the best performing algorithm at each image in the object density/gamma scaling space, as shown in Figure 8(a), an "optimal" results contour can be formed (Figure 8(b)).

Some individual algorithms performed well in certain spaces; for example, the analytic solution matched 90 percent of the ground truth image on for images with 11 to 18 percent deck density, averaged over all gamma scalings ($\gamma = 0.2 - 1.8$).

Figures 9(a)-(d) show the loss for each individual algorithm, defined as the difference in performace (subobtimality) with respect to the theoritcal optimal contour.

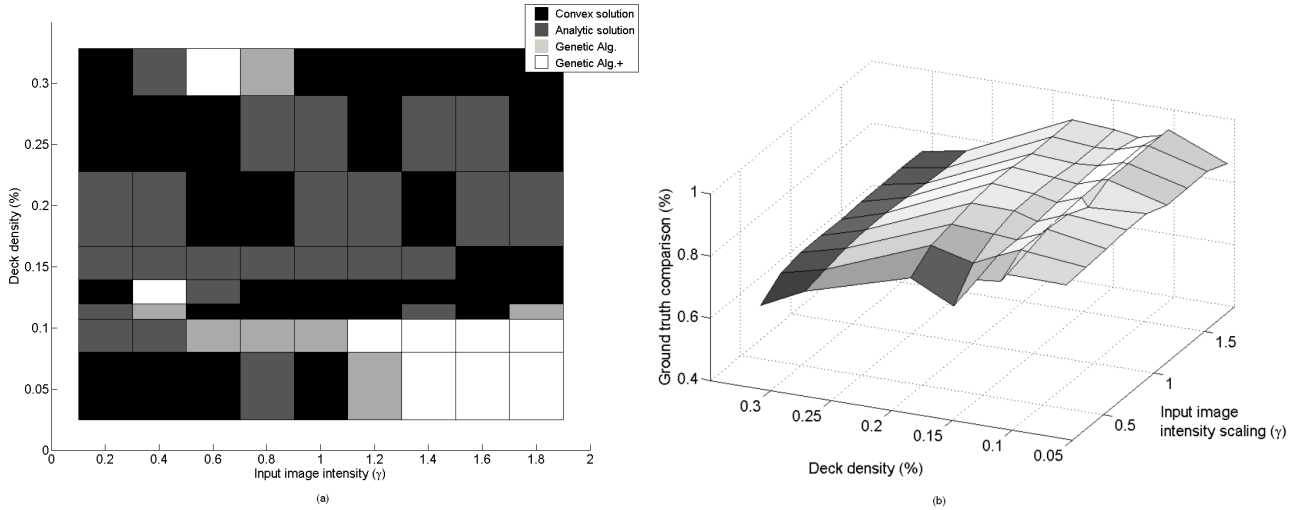

Figure 8. (a) Best performer at each image; (b) Optimal contour.

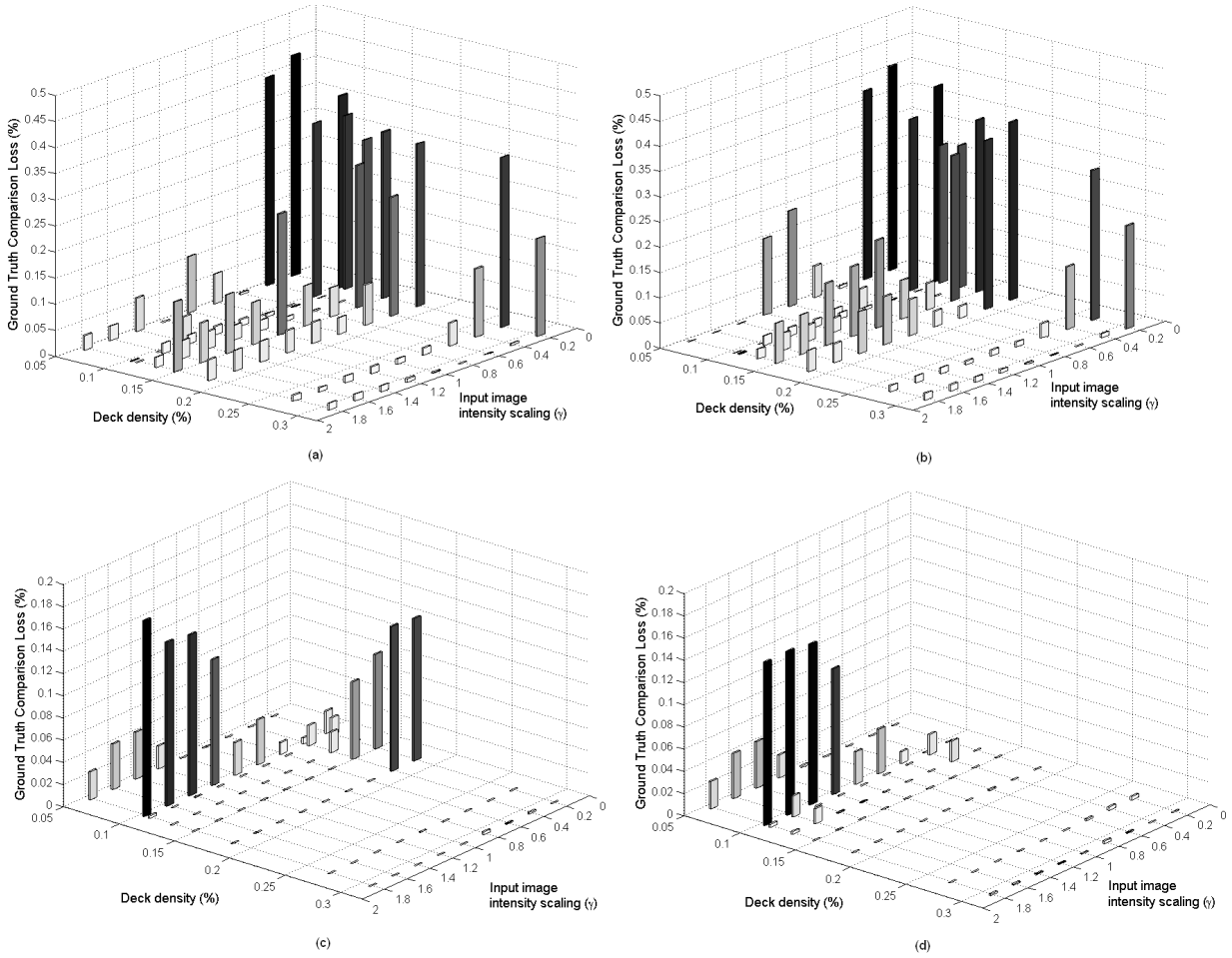

Figure 9. (a) GA loss; (b) GA+ loss; (c) CVX loss; (d) Analytic solution loss.

4. CONCLUSION

In this paper, three formulation methods for optimizing image background subtraction were presented: genetic algorithm, convex optimization, and a quadratic program admitting a least-squares solution. These algorithms were run on a set of representative images, and the results were aggregated to form an optimal results contour.

Since the optimal results contour was generated with what is essentially *training* data, the optimal results contour can not be used as a metric of test performance, and a thorough cross-validation study needs to be performed in order to truly evaluate the benefits of fusing multiple algorithms. However, the results suggest that using different approaches on different parts of the object density/deck brightness image space would lead to better overall results. For example, one could apply a Bayesian model selection framework to a set of training data.[12] Using the results contours, one could represent each algorithm as a probabalistic model M_i, and using the training data, generate a prior $p(M_i)$ over the input image space. Then, for a series of test data D, the results of each algorithm would represent likelihoods $p(D|M_i)$. The selected output image would then be the result of the algorithm that maximizes the posterior probability,

$$p(M_i|D) \propto p(M_i)p(D|M_i)\,. \tag{32}$$

Further work includes the discussed study of potential decision fusion/arbitration schemes in order to boost overall performance, a thorough cross-validation study of the α parameter in (9) (which requires a larger set

of training images), and better implementation of thresholding/noise reduction algorithms for post-subtraction foreground object extraction.

In general, the thresholding algorithm shown in Figure 4 was suitable for all subtraction methods. However, this algorithm failed when the genetic algorithm method could not find a good match for the image, as the subtracted image then had a high noise floor before being passed to the thresholding algorithm. Further noise reduction and thresholding techniques, in the form of adapting or implementing the large existing body of work, are an avenue for future research.

APPENDIX A. VALIDATION OF OPTIMAL SOLUTION

Semidefiniteness of $\hat{\mathbf{A}}$:

$$\mathbf{x}^T\hat{\mathbf{A}}\mathbf{x} = \mathbf{x}^T\left(\frac{w_2+w_3}{N}\mathbf{I} + \left(\frac{w_1}{N} - \frac{w_3}{N^2}\right)\mathbf{1}\mathbf{1}^T\right)\mathbf{x} \tag{33}$$

$$\triangleq \alpha\mathbf{x}^T\mathbf{x} + \beta\mathbf{x}^T\mathbf{1}\mathbf{1}^T\mathbf{x} \tag{34}$$

$$= \alpha\mathbf{x}^T\mathbf{x} - \frac{\alpha}{N}\mathbf{x}^T\mathbf{1}\mathbf{1}^T\mathbf{x} + \left(\beta + \frac{\alpha}{N}\right)\mathbf{x}^T\mathbf{1}\mathbf{1}^T\mathbf{x} \tag{35}$$

$$= \alpha\left(\sum_{i=1}^{N} x_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)^2\right) + \left(\beta + \frac{\alpha}{N}\right)\left(\sum_{i=1}^{N} x_i\right)^2 \tag{36}$$

$$\geq 0, \qquad \text{if } \alpha \geq 0 \text{ and } \beta + \frac{\alpha}{N} \geq 0 \tag{37}$$

The last step follows from Cauchy-Schwarz:

$$|\mathbf{x}^T\mathbf{1}|^2 \leq \mathbf{x}^T\mathbf{x}\cdot\mathbf{1}^T\mathbf{1} \tag{38}$$

$$\frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)^2 \leq \sum_{i=1}^{N} x_i^2 \tag{39}$$

Since the weights are positive,

$$\alpha = \frac{w_2+w_3}{N} \geq 0 \tag{40}$$

$$\beta + \frac{\alpha}{N} = \frac{w_1}{N} - \frac{w_3}{N^2} + \frac{1}{N}\frac{w_2+w_3}{N} = \frac{w_1}{N} + \frac{w_2}{N^2} \geq 0 \tag{41}$$

Therefore, the solution is valid. Also, choosing stricly positive weights ensures $\hat{\mathbf{A}}$ is positive definite: both terms in (36) are zero if and only if $\mathbf{x} = \mathbf{0}$. Thus, the solution is unique.[11]

ACKNOWLEDGMENTS

The authors thank Dr. James Sheehy for his support of this work. We also thank Dr. James Hing and Mr. Mark Husni for their reviews and comments, and Thomas Young for his discussions related to the background subtraction method presented in Hing, Venetsky, Severe and Young.[5]

REFERENCES

1. H. Fukushima and J. Watanabe, "An image processing technique for background subtraction and its application to comet austin 1989," *Publ. Natl. Astron. Obs. Japan* **2**, pp. 185–189, 1991.
2. D. Parks and S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, pp. 192–199, sept. 2008.
3. Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters* **27**(7), pp. 773–780, 2006.
4. A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *Computer VisionECCV 2000* , pp. 751–767, 2000.
5. J. Hing, L. Venetsky, J. Severe, and T. Young, "Toward automated visual tracking of aircraft on a carrier flight deck," *NAVAIR Journal for Scientists and Engineers* **1**, Aug. 2012.
6. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA, 1992.
7. H. Spencer, *The Principles of Biology*, Williams and Norgate, 1864.
8. S. Boyd and L. Vandenberghe, *Convex Optimization*, p. 72. Cambridge University Press, New York, NY, USA, 2004.
9. N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on* **9**, pp. 62 –66, jan. 1979.
10. CVX Research Inc., "CVX: Matlab software for disciplined convex programming, version 2.0 beta," Sept. 2012. http://cvxr.com/cvx.
11. S. Boyd and L. Vandenberghe, *Convex Optimization*, pp. 140–141. Cambridge University Press, New York, NY, USA, 2004.
12. C. M. Bishop, *Pattern Recognition and Machine Learning*, pp. 161–162. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.