A Collective-Based Adaptive Symbiotic Model for Surface Reconstruction in Area-Based Stereo

John Yannis Goulermas, *Member, IEEE,* and Panos Liatsis, *Member, IEEE*

***Abstract*—This paper proposes a novel optimization algorithm for image-space matching and three-dimensional space analysis, using an adapted scheme of evolutionary computation that employs the concept of symbiosis in a collective of homogeneous populations. It is applied to the automatic generation of disparity surfaces used for depth estimation in stereo vision. The global task of approximating the complete disparity surface is decomposed to a large number of smaller local problems, each solvable by a smaller processing unit. Coevolution is sustained in such a way as to counteract the arbitrary decomposition of the original super-problem, so that the local evolutions of all the subproblems become interlocked. This, in the long run, provides a consistent global solution, and it does so via an asynchronous and massively parallel architecture. The entire surface is partitioned to a set of adjoining patches represented by distinct species or populations, with phenotypes corresponding to different polynomial functionals. The credit assignment functions take into account both self and symbiotic terms in an adaptive and dynamic manner, in order to produce disparity patches that are fit within their own domain and at the same time fit in association with their symbionts. This persistent propagation of local interactions to a global scale throughout evolution generates a unified disparity surface composed of the many smaller patch surfaces.**

***Index Terms*—Image-space matching, parallel optimization, population collective, stereo correspondence problem, surface approximation, symbiosis.**

I. Introduction

MACHINE VISION is the science concerned with the computational modeling, processing, and interpretation of the two-dimensional (2-D) projections of three-dimensional (3-D) physical scenes by computer algorithms. A significant field of machine vision relates to the recovery of the original physical 3-D information, so that visible surface reconstruction and depth measurements become possible. Stereo vision refers to one such mechanism of obtaining surface estimations using nonintrusive camera-based sensing to acquire multiple views of the same real-world scene, in a manner similar to the way many biological beings with eyes of overlapping fields of view perceive depth. In order to measure the distance between the observer (camera system) and each of the scene elements from the pair of captured images, it is required that certain image features (which correspond to the 2-D projections of 3-D physical primitives) in one retina are matched with their conjugate features in the other retina. These pairs give rise to a set of disparities or parallax values between the two image retinas, which by using certain geometrical transformations and camera parameters can produce the sought measurements [1]–[3]. In the last two decades, stereo vision has been receiving a rapidly increasing appeal in a multitude of application domains. These include vehicle navigation, teleoperation, robotics, manufacturing, medicine, terrain analysis, entertainment, etc.

The problem of finding conjugate pairs between images is called the stereo correspondence problem (SCP), which in essence is an optimization problem of a very large solution space. Many different optimization methods with diverse properties have been employed in the past, such as local search [4], dynamic programming [5]–[7], relaxation [8], variational methods [1], [9], [10], gradient methods [11], graph methods [12], hybrid techniques such as ones combining graph and gradient methods [13], [14], neural networks [15], [16], simulated annealing [17], evolutionary optimizations [18]–[21], and others. In general, there are two kinds of matchers: *feature-based* and *area-based*. The former rely on the geometric invariance of the scene projections, match extracted features such as edge structures, and generate sparse disparity measurements. The latter are based on the photometric invariance, that is, preservation of the intensity profiles between the views, and generate dense disparity maps. This kind of matcher can be more computationally demanding as compared with the feature-based ones their search space is significantly broader. Area-based matchers, however, do not need interpolating steps to fill the gaps and their naturally dense output is more suitable to modern applications of stereo, such as view synthesis and image-based rendering [22]. Comprehensive details for all such techniques can be found in the reviewing texts [22]–[28] and the references therein. In this paper, we relate to the latter type and produce dense disparity output.

Previous applications of genetic algorithms (GAs) to the SCP, include [18], which used a GA to find the optimum disparity from a set of discrete values precalculated by aggregation windows of different sizes. A block decomposition scheme was used for efficiency, however, the search was bounded by the quality of the precalculated values and no interblock smoothness was imposed. Reference [19] also used a block decomposition optimising each block with a separate GA execution. However, the lack of dependency between blocks generated discontinuities in the final disparity map. Also, all block chromosomes were composed of discrete per-pixel disparity values. Reference [20] used block partitioning but with simple interactions between blocks to smoothen the transitions. However,

Manuscript received April 8, 2002; revised June 15, 2003.

J. Y. Goulermas is with The Centre for Virtual Environments, Business House, University of Salford, Salford M5 4WT, U.K. (e-mail: j.y.goulermas@salford.ac.uk).

P. Liatsis is with the Control Systems Centre, Department of Electrical Engineering and Electronics, University of Manchester Institute of Science and Technology (UMIST), Manchester M60 1QD, U.K. (e-mail: panos.liatsis@umist.ac.uk).

Digital Object Identifier 10.1109/TEVC.2003.817460

1089-778X/03$17.00 © 2003 IEEE

chromosomes were again discrete, and their quality depended on a region extraction preprocessing scheme used to preset their structure. Also, synchronous serial execution was used for the GAs, and since chromosomes consisted of portions of constant disparities, surfaces were biased toward frontoparallel ones. The method proposed in [21] used a quadtree to adaptively decompose the image to areas of different sizes represented as tree leafs with explicit smoothness handling between the leafs. However, each area had a constant discrete disparity value and their scheme was not arranged for parallelization.

Our proposed algorithm, which is an extension of [26] and [29], uses a particular type of GA, called symbiotic (SGA), and is distinctly different from previous approaches. The entire disparity surface map is decomposed into a set of equal-sized patches with local only support, and the optimization of each patch is assigned to a single processing unit. Although such surface decomposition is again employed for efficiency, we use continuous surfaces that support subpixel disparity accuracy and modeling of arbitrary surfaces. The objective function of every population is configured in a way that takes into account two types of objectives. The first is composed of the self scores of the intrapatch intensity profiles and the geometrical scene constraints. The second, the symbiotic score, enforces interpatch continuities on the participating surfaces. While each population searches for a best-fit surface within its own domain, it continuously takes into account currently available solution quality information from the neighboring populations; these populations are considered to constitute the symbiont species of each evolving patch. In this way, the incessant propagation of local information allows each GA to be directed to a region of its solution space that exhibits high surface compatibility with other proximal patches.

We design symbiosis in a way that counteracts the global disparity surface decomposition and enables a parallelisable solution of the super-problem. Evolution of each population is implemented by an autonomous processing unit: however, coevolutionary dependencies are sustained through shared elements in the credit assignment functions of all participating SGAs. Importantly, each species/patch does not have to interact with all others in the entire collective but only with its immediate neighbors. Despite such locality, global consistency is propagated from short-term regional interactions to gradually more distant populations, so that a globally acceptable solution is achievable in the long term. Toward the end of evolution, the best patch surfaces from all populations are collected to integrate the solution of the super-problem, i.e., the complete disparity map.

In addition to the typical advantages of evolutionary optimization [30] (e.g., global search, no need for continuity, differentiability, unimodality, etc.), our novel framework incorporates certain advantages over the standard GA, as well as some previous coevolutionary models described in Section II. First, it allows for the use of a very large number of simple populations, so that it can be implemented in a massively parallel system. Each processing unit (CPU or network node) can assume evolution of one or more populations. Also, communication between the interacting units is frugal. Not only each population needs to communicate with just its nearby neighboring species, but communication is restricted to the simple transmission of partial phenotypic information of some representative members. Furthermore, each objective function evaluation does not require the repetitive composition of the global disparity solution for the individual credit assignments, but only the data provided by the aforementioned local communication. Finally, synchronization between the members is unnecessary. Even if some processing units miss transmitted information or fail temporarily, the overall performance does not falter.

The structure of this paper is as follows. In Section II, a concise survey of coevolutionary applications is given. Section III presents the details of the stereo matching problem and the proposed decomposition scheme. Section IV explains the chromosome encoding scheme, while Section V describes the GA operation sequencing and the employed genetic operators. The method used to hybridise the initial populations of the collective is outlined in Section VI, while the description of the credit assignments is given in Section VII. Section VIII describes some design and implementation issues. Finally, Sections IX and X include results and conclusions, respectively.

II. Previous Coevolutionary Paradigms

Computational models utilizing coevolutionary strategies have been studied in various cases. The first perhaps application of coevolution in [31] used a population of real coded patterns to play a simple game of poker. Each pattern encoded evolution parameters and betting probabilities. During coevolution each pattern played a number of games against an opponent selected from the population. The losing patterns were eliminated, while the winning ones were allowed to reproduce. Reference [32] employed parasitic GAs to search for minimal sorting networks of fixed cardinality. The two involved populations were the networks (hosts) and the test cases (parasites) and the fitness of a host depended on its ability to solve tests, while the fitness of a parasite depended on the frequency it was solved incorrectly by the networks. References [33]–[35] coevolved one population of constraints and one of solutions to solve constraint satisfaction problems. The fitness of a solution depended on the number of satisfied constraints over a number of interpopulation encounters, while the fitness of a constraint depended on the number of times it made a solution to fail. Reference [36] also used one population of test cases and one of solutions to evolve cellular automata with density classification capabilities. The work evaluated the usefulness of resource-sharing fitness functions, where solutions received high fitness if they could solve test cases that were difficult to a large set of other solutions. Reference [37] used a population of neural networks (game strategies) to evaluate board positions in a tree search playing checkers. At each generation, all parents and offspring competed for survival, by having each network played against other population members and preserving the best.

In the field of optimization, [38] used n symbiotic populations for minimization of n-dimensional functions, with each population encoding a different decision variable. The fitness of each member was calculated by partnering that member with good representatives from the other populations. A similar configuration was also applied in rule-based robot learning [39]. Different partnering methods were considered to make fitness

calculations more realistic. Reference [40] also used two coevolving populations for the minimax problem; one for minimization on one decision variable, while the other for maximization of the other variable. Reference [41] applied an evolution strategy to coevolve two populations, one of decision variables and one of Lagrange multipliers to solve efficiently generic nonlinear constrained problems, formulated as a zero-sum minimax problem. Reference [42] used coevolution of one host and one parasitic population to enhance the search for useful schemata during evolution. The host GA searched for good solutions to the problem at hand, while the parasitic one explored the solution space for schemata that improved the search within the host. A probabilistic gene transcription mechanism from the parasites to hosts was employed for fitness evaluation. The work was applied to function optimization, while it was extended in [43] for constraint satisfaction problems.

Other applications involve the work of [44] and [45], which evolved neural networks by using one population to encode connectivity and weights of individual neurons and a second population to encode complex combinations of neurons from the first population, in order to form complete network solutions which were evaluated upon their classification ability. Reference [46] used symbiosis for designing a fuzzy controller by evolving a population of different types of fuzzy rules. Fitness evaluation was performed by selecting a set of rules, then, evaluating the performance of the controller with the problem data, and finally, accumulating part of the overall fitness value to the participating rules. Reference [47] applied coevolution to the design of minimax controllers for uncertain environments by using one population of controllers and one of plants. Each controller individual was scored according to its performance over the plant parameter population, while each plant was scored according to its simulation results with each given controller. References [48] and [49] proposed a generalized society model for coevolution, which was applied to radial-basis neural networks for function approximation. The genome of each population encoded basis functions of the same type (centers and variances) and the fitness for each member took into account its similarity with representatives from other populations. Reference [50] used the idea of shared memory in the coevolution of two populations; one of painters and one of whitewashers for the solution of the room-painting problem. The shared memory enhanced the efficiency of the partnering strategy by storing fertile partnerships instead of discarding them as soon as fitnesses were evaluated. Reference [51] used a particular epistatic problem to test various partnering strategies for generational and steady-state GAs, such as random, best, selection-based, distributed, and joined. A different type of work from [52] used a computational model to demonstrate the learning capability of the immune system. This was modeled as a population containing bit-coded antibodies, capable of memorizing pattern information. First, an antigen population immunized (trained) the system, and then, a different antigen population was used to assess the learning of pattern classification problems.

In the field of machine vision, coevolution has also been applied successfully. The work of [53] used $n+1$ coevolving populations to estimate motion and layer information of n video objects. n populations encoded the motion transform parameters of each object and another population encoded the layering of the objects. Fitness evaluation involved repetitive partnering of selected members from all populations to derive a predicted frame and compare it with the actual segmented video sequence. Reference [54] provided an antagonistic coevolution to segmentation of textured areas. There were two populations encoding textural descriptors for two types of texture, which competed for territories of different texture. The fitness value was the combination of the responses of the descriptors applied on their own textures, as well as the textures of the opposing population. Reference [55] applied symbiotic optimization of snakes to object tracking, where different populations were used to search for adjoining snake segments. Fitness evaluations used self criteria to minimize the snake's internal energy, as well as symbiotic criteria based on the proximity of segment control points and their Fourier descriptors. Reference [56] applied symbiosis to the discrete optimization problem of sparse feature-based correspondence. A different population was assigned to each scan line to solve a weighted bipartite graph matching problem, but the cost function took into account edge-linking similarity information from the adjacent species.

III. Problem Definition and Partitioning

The input to the algorithm is a stereo pair $\{L, R\}$ with each image of dimensions $h \times w$ pixels. Matching pairs up a left pixel (y_l, x_l) with a right pixel (y_r, x_r) and gives rise to a disparity function $d(y_l, x_l) = x_l - x_r$ (y and x are used to denote image rows and columns, respectively). Image L is used as the reference retina. Without loss of generality, we assume canonical camera configuration with vertically registered retinas, so that the epipolar constraint [25], [28] forces within scan-line matching, that is, $y_l = y_r$. A user-defined disparity range $\Delta = [d_{\min}, d_{\max}]$ is also enforced so that search is restricted by $x_r \in [x_l - d_{\max}, x_l - d_{\min}]$.

As mentioned earlier, we employ a partitioning of the global surface spanning the entire retina to small rectangular patches. We have also considered various alternatives for such partitioning. We could, for instance, create populations each corresponding to a single image row and apply symbiotic dependency to interrow smoothness between each epipolar row and its adjacent ones. Although this would give rise to h GAs, the solution space of each GA could be very large, since each chromosome could be of length w, giving a discrete search space (ignoring occlusions and uniqueness for simplicity) of size $(d_{\max} - d_{\min} + 1)^w$. Since we wish to achieve massive parallelism, a higher number of populations each solving a simpler problem is more prudent. Comparing this with the partitioning to rectangular patches of size $h_p \times w_p$, we obtain a much larger number of populations, each of a smaller search space of size $(d_{\max} - d_{\min} + 1)^{h_p \cdot w_p}$.

The simple partitioning scheme we used is shown in Fig. 1, where each patch is centered at the point (j, i) of a uniformly spaced grid and overlaps with its adjacent ones by one pixel. We use $d_{ji}(y, x)$ to denote a 2-D continuous function with domain limited by the patch boundaries (note that such surface depends on both coordinates y and x at each image location. In addition, if the epipolar constraint were not used, two such independent

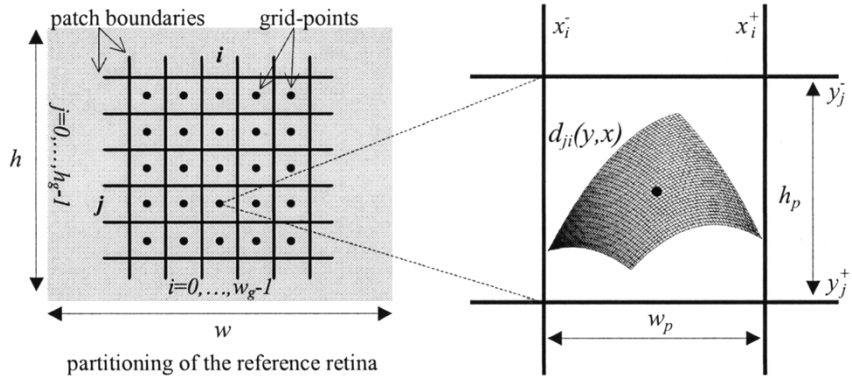


Fig. 1. Partitioning of L into $h_g \times w_g$ rectangular patches. The disparities within each patch are computed by the values of $d_{ji}(y,x)$ in the domain $[y_j^-, y_j^+] \times [x_i^-, x_i^+]$.

functionals would be needed to model the disparity vector). The range of each $d_{ji}(y,x)$ corresponds to the set of disparities spanning the pixel domain $[y_j^-, y_j^+] \times [x_i^-, x_i^+] \equiv [j \cdot (h_p - 1), (j+1) \cdot (h_p - 1)] \times [i \cdot (w_p - 1), (i+1) \cdot (w_p - 1)]$. Image points (y,x) are indexed by $y = 0, \ldots, h-1$ and $x = 0, \ldots, w-1$, while patch centers (j,i) by $j = 0, \ldots, h_g - 1$ and $i = 0, \ldots, w_g - 1$, with $h_g = \lceil h/(h_p - 1) \rceil$ and $w_g = \lceil w/(w_p - 1) \rceil$.

IV. Genome Encoding

To obtain phenotypes that are low-order disparity surfaces, we model the chromosomes as bivariate polynomials of degree N

$$d_{ji}(y,x) = \sum_{a+b \leq N} z_{ab} \cdot y^a \cdot x^b \tag{1}$$

where z_{ab} is the coefficient of each term; for example, the biquadratic polynomial is given by $z_{20} \cdot y^2 + z_{02} \cdot x^2 + z_{11} \cdot y \cdot x + z_{10} \cdot y + z_{01} \cdot x + z_{00}$. We choose to model such a chromosome as a coefficient vector, for instance, $[z_{20}, z_{02}, z_{11}, z_{10}, z_{01}, z_{00}]$, since real-valued representations have been shown to result in faster and more consistent searches for continuous problems [57]–[59]. In addition, we have the benefit of continuous subpixel disparity measurements. The length $\tau(N)$ of such chromosomes is $\binom{N+2}{N} = (N+1) \cdot (N+2)/2$, that is, the number of positive integer solutions of $a + b \leq N$.

Note that modeling local disparity with low-order polynomials has also been used in [60]–[62] but in a different optimization context. Low-order functionals are preferred here as they naturally incorporate the smoothness constraint. If the disparity surface has an unconstrained form, then explicit regularization terms must be added to the cost function. This is because the SCP is an ill-posed problem and regularization theory is needed to recover acceptable solutions [63]. Various smoothness terms with different nullspace properties have been used, with typical examples including the square-Laplacian $(d_{yy} + d_{xx})^2$ [1], the quadratic-variation $d_{yy}^2 + 2 \cdot d_{yx}^2 + d_{xx}^2$ [10], [11], and the square-gradient $d_y^2 + d_x^2$ [63]. The computational overhead of these operators, however, is avoided here by employing low-order surfaces.

V. GA Configuration

We allocate a separate population P_{ji} at each (j,i) grid point, responsible for controlling the disparity surface $d_{ji}(y,x)$. Every population contains a fixed number of g_{size} chromosomes of $\tau(N)$ real values each and at each generation g_{child} members are replenished. The calculation of the objective value of every k_{th} chromosome $c_{jik} \in P_{ji}$ is described in Section VII. This configuration gives a collective of populations that are algorithmically homogeneous; it is only the problem instance that changes and, not the type of the underlying problem. All GAs are set-up similarly to the *mod-GA* proposed in [59] with each generation completing in the following steps.

Step 1) Select g_{size}-g_{child} distinct survivors for the next generation.

Step 2) Select parents from the current population and breed g_{child} offspring from them, with each parent allowed to mate only once.

Step 3) Replace the current population with the survivors and the new offspring and evaluate objective values accordingly.

This scheme has certain advantages over standard generational replacement techniques [59], as both parent and child have a good chance of appearing in the next generation. Also, the finite population slots are exploited better since survivors are distinct. The g_{child} parameter defines deterministically the ratio of population replacement. In the above steps, we add the *elitism* operator to preserve the best members as unconditional survivors. The fitness allocation mechanism uses *linear ranking* [64] and employs Baker's ranking formula [65] to explicitly control selective pressure and the balance between exploration and exploitation (using the parameter $\eta_{\max}$). This scheme has shown to mitigate significantly the *super-individual* and *close-race* effects [57], [66].

A. Search Operators

In order to have better control over the search, we split the breeding in step 2) above. Offspring are produced either by crossover or by mutation, i.e., these two operators are not applied sequentially. At each generation t, every GA performs a number of cycles n of selecting the next genetic operator to be applied until all new g_{child} offspring are collected. At

each cycle, crossover is applied with probability p_c and mutation with probability $(1 - p_c)$. As explained in Section V-A1, we use four crossover operators: three quadratic ones generating two offspring each and one binary generating one offspring. Thus, the average of individuals generated by crossover is $7/4 = 1.75$. Mutation (see Section V-A2) always generates one new offspring. Since $p_c \cdot n \cdot 1.75 + (1 - p_c) \cdot n$ is equal g_{child}, crossover and mutation produce $175 \cdot p_c/(0.75 \cdot p_c + 1)\%$ and $(1 - p_c) \cdot 100/(0.75 \cdot p_c + 1)\%$ of all the offspring, respectively. We use an initial value of $p_c = 0.6$, which gives 72% and 28% offspring for the two operators, respectively.

We have adopted a simple experimental heuristic to improve search. Since the initial generations contain adequate diversity, the exploitation facility of crossover is more desirable. Later, when diversity is decreased, the exploration of new regions by the mutation operator is more useful. To take this into account, p_c is halved automatically upon a predefined generation count t_c. Thus, for generations $t_c, \ldots, t_{\max}$, a value of $p_c = 0.3$ changes the contributions from crossover and mutation to 43% and 57% of g_{child}, respectively.

1) Crossover: As we have used a real-valued chromosome representation, standard point crossovers employed in bit-string GAs may not be adequate. Previous research on function optimization [58], [59], [67] has shown the usefulness of specialised operators for real-valued representations. In this paper, for two selected parents c_{jik} and $c_{\text{jil}} \in P_{ji}$, we use the following crossover library.

C-I: Classic *χ-point* [59]: χ break-sites are selected randomly and c_{jik} and c_{jil} contribute accordingly to produce two offspring. Since $\tau(N)$ is small, a small fixed symmetric value of $\chi = 2$ is adequate. This crossover generates offspring at the corners of the hypercube defined by the two parents.

C-II: Line arithmetical [58], [59], [67]: A random uniform number $r \in [0, 1]$ is generated and then, two offspring are produced through the linear combinations of the parental vectors: $r \cdot c_{jik} + (1 - r) \cdot c_{\text{jil}}$ and $(1 - r) \cdot c_{jik} + r \cdot c_{\text{jil}}$. This crossover produces offspring lying along the line segment joining the two parent points in the $\tau(N)$-dimensional solution space. Note that C-II is a general case of the *average arithmetical* crossover for fixed $r = 0.5$.

C-III: Intermediate arithmetical [59], [67]: This works as C-II, however, a different r_m is generated for each m_{th} allele $c_{jik}[m]$ and $c_{\text{jil}}[m]$ of the two parents, for $m = 1, \ldots, \tau(N)$. The two offspring are given by the combinations $r_m \cdot c_{jik}[m] + (1 - r_m) \cdot c_{\text{jil}}[m]$ and $(1 - r_m) \cdot c_{jik}[m] + r_m \cdot c_{\text{jil}}[m]$. This generates offspring within the boundaries of the hypercube defined by the two parents.

C-IV: Heuristic arithmetical [58], [59]: This operator generates only one offspring given by $r \cdot (c_{jik} - c_{\text{jil}}) + c_{jik}$, for a random $r \in [0, 1]$ given that $f(c_{jik}) \leq f(c_{\text{jil}})$. The generated offspring lie at the line passing through the parents similarly to C-II, but the comparison of the objective values biases the operator to search toward the most promising direction.

As seen by the geometric interpretation of the above operators, they support combinations of the polynomial coefficients z_{ab} in various ways. If the two parents are near fit because of some promising depth attributes of their phenotypic polynomial surfaces, such as curvature components or intercept, then the children have potentially the chance of inheriting these attributes, by either exchanging (C-I) or by arithmetically recombining (C-II, C-III, C-IV) the related coefficient groups. We observed a faster search when this combination of crossovers was used instead of any single one. Once the GA has decided to use crossover as the next breeding operation, a scheme from the library is selected in random. Note that all crossover schemes guarantee offspring with alleles within the permissible alphabet $\Phi = [z_{\min}, z_{\max}]$ (see Section VI), apart from C-IV. A maximum of five attempts for reselecting r is allowed for this operator until Φ is respected.

2) Mutation: The mutation operator also uses a library of varying schemes with different properties, exploiting the real-valued representation. A single parent member is selected and each allele is mutated with a probability $p_m = 0.25$ (with a minimum of one mutant allele per member). The following mutation schemes are adopted.

M-I: Uniform: The m_{th} mutant allele $c_{jik}[m]$ is reassigned a random value from $[z_{\min}, z_{\max}]$.

M-II: Nonuniform [59]: The new allele value $c'_{jik}[m]$ depends on the age of the population and is calculated by

$$c'_{jik}[m] = \begin{cases} c_{jik}[m] + J(t, z_{\max} - c_{jik}[m]) \\ c_{jik}[m] - J(t, c_{jik}[m] - z_{\min}) \end{cases} \tag{2}$$

where the two choices are decided in random with a probability of 0.5. The added/subtracted value is given by

$$J(t, z) = z \cdot \left(1 - r^{(1 - t/t_{\max})^b}\right) \tag{3}$$

where t is the current generation, $t_{\max}$ is the length of evolution, b is a system dependent parameter set to 4.0, which reflects the degree of uniformity, and $r \in [0, 1]$ is a uniform random number. This function gives a value within $[0, z]$ which is closer to zero with higher probability, as more generations elapse. Thus, in the early stages, the search space is sampled uniformly, while later on, a local fine-tuning is performed.

M-III: Boundary [59]: This changes the mutant allele to either $z_{\min}$ or $z_{\max}$ with a probability of 0.5.

When an individual is selected for mutation, one of M-I, M-II, and M-III schemes is chosen with probabilities 0.4, 0.4, and 0.2, respectively; we do so since M-III is more disruptive. This composite mutation scheme was found sufficient in sampling new points of the coefficient space and managing diversity loss.

VI. Population Initialization

In order to restrict the range of the search during initialization and evolution, we provide a chromosome alphabet $\Phi = [z_{\min}, z_{\max}]$ that represents the set of all permissible values for the gene coefficients z_{ab}. To set its limits to some reasonable values, we use the disparity range Δ and assign $z_{\max} = -z_{\min} = 1.5 \times \max(|d_{\max}|, |d_{\min}|)$. We also scale the domain $[y_j^-, y_j^+] \times [x_i^-, x_i^+]$ of each patch to $[0, 1]^2$. Then, to obtain the initial populations, we randomly select from Φ all

coefficients z_{ab} excluding the constant one z_{00}. Subsequently, z_{00} is recalculated via

$$z_{00} = \bar{d}_{ji} - \frac{1}{w_p \cdot h_p} \cdot \sum_{y=y_j^-}^{y_j^+} \sum_{x=x_i^-}^{x_i^+} \sum_{0<a+b\leq N} z_{ab} \cdot y^a \cdot x^b \quad (4)$$

where $\bar{d}_{ji} \in \Delta$ is an estimated value of the average disparity of patch (j,i). If (4) produces $z_{00} \notin \Phi$, then the operation can be repeated. The proposed heuristic combines a simple and very fast mechanism for generating random solutions near the feasible regions of the search space without, however, sacrificing the genomic diversity of the initial populations. Note that those few elements (y,x) that may have disparities outside Δ are handled by the penalty terms of the credit assignment (see Section VII).

A. Hybridization Phase

It is beneficial to hybridise the initial populations, so that the search is deployed from prosperous regions of the feasible solution spaces. In GAs, various methods for hybridization have been used [57], [59]. Here, we adapt an inexact but fast stereo matcher (the *bidirectional local search* (BLS) similar to the one used in [4] and [68]), to calculate approximate disparities for each patch and generate reasonable $\bar{d}_{ji}$ estimates for (4). In BLS, each pixel (y,x_l) in L is assigned its most preferred (within Δ) pixel (y,x_r) in R and only if this preference is mutual then the pairing is accepted. For preference evaluation, we use the *zero-mean normalized cross-correlation* (S_{ZNCC}) score [68] as in (5), shown at the bottom of the page, where the comparison is performed within $(2\cdot k+1)\times(2\cdot k+1)$ windows (with $k=4$) centered at the two pixels and $\bar{L}(y,x_l)$ and $\bar{R}(y,x_r)$ are the average intensities.

Using the above, we employ the following hybridization procedure.

Step 1) Apply BLS to all $h_g \cdot w_g$ patches independently and for each patch evaluate whether the number of its pixels with defined disparities is above a threshold $\gamma \cdot h_p \cdot w_p$, where γ is a user-defined percentage.

Step 2) For each patch (j,i) that passes the test, set its $\bar{d}_{ji}$ value equal to the median of its disparity distribution.

Step 3) For all the patches that failed the test, calculate their $\bar{d}_{ji}$ values using the average values of their eight-connected neighbors, which have defined disparities. If some patches do not have such neighbors at all, iteratively apply this step, until all missing values are gradually filled up.

The above test is used to reject disparities which are of low confidence and possibly incorrect, while the median value is used for rejection of the outliers. This heuristic was found to be very effective in providing the SGAs with good estimates for the starting populations. Note that although there is a single $\bar{d}_{ji}$ for all the chromosomes of each population P_{ji}, the starting diversity of P_{ji} is high, since the coefficients of its members can have quite different magnitudes.

VII. Credit Assignment Terms

A. Self Contributions

The primary goal of each population P_{ji} is to locate a surface d_{ji} that optimally conforms with the photometric similarity of the two images by matching each point $(y,x) \in [y_j^-, y_j^+] \times [x_i^-, x_i^+]$ in L with a point $(y, x-d_{ji}(y,x))$ in R. Given the k_{th} chromosome c_{jik} in P_{ji}, we use (6), shown at the bottom of the page, minimizing error term f_M to achieve this. The above objective penalises the out-of-range disparities by a per pixel constant cost ξ, since the problem constraints are violated (note that, for simplicity, we use $c_{jik}(y,x)$ to denote the disparity value d_{ji} corresponding to the coefficients of c_{jik} at pixel (y,x)). The pair $\{L_0, R_0\}$ denotes the original pixel intensity tables, while $\{L_y, R_y\}$ and $\{L_x, R_x\}$ are the vertical and horizontal gradient fields of the images, respectively. These are calculated by convolving the original images with the corresponding derivatives of a Gaussian kernel of scale $\sigma = 0.4$ in order to reduce the effects of noise and quantization. Since the values in the three R_n tables are accessed at a subpixel level, we use a simple parabolic interpolation along the row. We also truncate the responses of the absolute differences in (6) to a small value of ρ_I for each of the three terms prior to summation, in order to limit the contribution of large errors [22]. Finally, note that although we used the sum of absolute differences, any dissimilarity/similarity metric could have been used, such as correlation or metrics based on rank statistics [4], [69], [70], which exhibit robustness to linear

$$S_{\text{ZNCC}}(y,x_l,x_r) = \frac{\sum_{j=-k}^{k}\sum_{i=-k}^{k}\left(L(y+j,x_l+i)-\bar{L}(y,x_l)\right)\cdot\left(R(y+j,x_r+i)-\bar{R}(y,x_r)\right)}{\sqrt{\sum_{j=-k}^{k}\sum_{i=-k}^{k}\left(L(y+j,x_l+i)-\bar{L}(y,x_l)\right)^2\cdot\sum_{i=-k}^{k}\sum_{j=-k}^{k}\left(R(y+j,x_r+i)-\bar{R}(y,x_r)\right)^2}} \quad (5)$$

$$f_M(c_{jik}) = \frac{1}{3\cdot w_p\cdot h_p}\cdot\sum_{y=y_j^-}^{y_j^+}\sum_{x=x_i^-}^{x_i^+}\begin{cases}\xi, & \text{if } c_{jik}(y,x)\notin[d_{\min},d_{\max}]\\ \sum_{n=0,y,x}\left|L_n(y,x)-R_n(y,x-c_{jik}(y,x))\right|, & \text{otherwise.}\end{cases} \quad (6)$$

and certain nonlinear intensity variations but are more computationally expensive.

A second error term can be designed by taking into account certain scene constraints. The uniqueness constraint (UC), for instance, requires a one-to-one matching. In the above formulation, this is not guaranteed since a right point can be the conjugate of multiple left points. A further improvement can be obtained using the ordering constraint (OC). Both constraints are in fact constraints on the possible scene objects and their use can reduce the search space of the SCP. The UC implies that each physical point occupies a unique position in the 3-D space and assumes opaque scene objects, while the OC disallows thin occluding scene objects at large separations that cause projection reversals [2].

Here, we satisfy both constraints simultaneously as follows. Suppose that at two nearby points (y, x) and $(y, x + \Delta x)$ in L, where $\Delta x > 0$, the corresponding disparities are $d(y, x)$ and $d(y, x + \Delta x)$. Then, the OC implies that their conjugate points in R preserve their order, that is

$$(x + \Delta x) - d(y, x + \Delta x) > x - d(y, x) \Rightarrow \frac{d(y, x + \Delta x) - d(y, x)}{\Delta x} < 1 \quad (7)$$

which, for $\Delta x \to 0$, restricts the horizontal derivative to an upper bound of 1.0. This has also been observed in [60] and used in the energy minimization framework of [11]. Here, we explicitly incorporate the violations of this term via the penalty term f_O defined as

$$f_O(c_{jik}) = \frac{1}{w_p \cdot h_p} \cdot \sum_{y=y_j^-}^{y_j^+} \sum_{x=x_i^-}^{x_i^+} \begin{cases} 1, & \text{if } \frac{\partial d_{ji}(y,x)}{\partial x} \geq 1 \\ 0, & \text{otherwise} \end{cases}. \quad (8)$$

In this way, the OC is not enforced as a hard constraint as in dynamic-programming-based algorithms, since if the stereo pair contains objects that cause reversals, the photometric term f_M has potentially the ability to outweigh f_O and admit a disparity surface that violates the OC. However, since the majority of the scene objects are expected to abide by the OC, an additional improvement can be achieved.

B. Symbiotic Contributions

As mentioned in Section IV, intrapatch smoothness is managed effectively by the low-order polynomials. However, due to the difference between the views and the noise, the surface continuity across the patches breaks down. Increasing the size of the patches may alleviate this problem, however, it causes loss of resolution and increases the errors at object boundaries, where discontinuities must be allowed. Thus, since minimizing the self errors f_M and f_O is a necessary but not adequate optimization strategy, we have to impose extra terms to enforce interpatch regularization. We design such terms based on symbiotic interlocking in order to proscribe each patch from evolving independently of its neighbors. Fig. 2 pictures such dependency between each population P_{ji} and its four-connected symbiont species $P_{j\pm1\ i\pm1}$.

The first such term attempts to suppress the strong *mosaic effect* by enforcing zeroth-order C_0 (or positional) continuity at the boundary disparities with each of the four neighbors in the

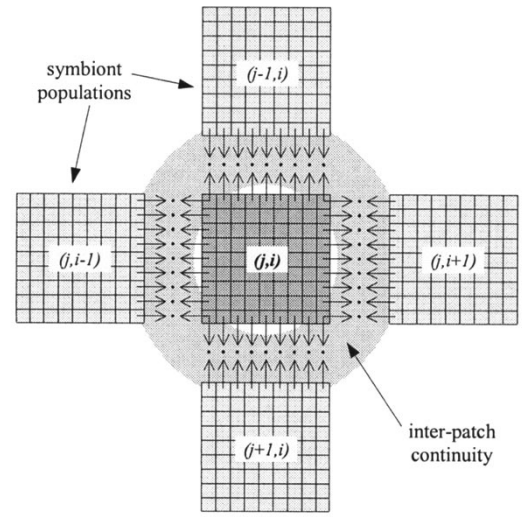


Fig. 2. Symbiotic dependency between species P_{ji} and its four symbionts. Regularizing forces at the patch boundaries can enforce surface continuity disrupted by the problem decomposition.

top ($y = y_j^-$), bottom ($y = y_j^+$), left ($x = x_i^-$), and right ($x = x_i^+$) sides, as shown in Fig. 2. This is given by the error term

$$\begin{aligned} f_{C0}(c_{jik}) = \frac{1}{\sum \beta} \Bigg\{ & \beta(c_{jik}, c^*_{ji-1}) h_p \underbrace{\sum_{y=y_j^-}^{y_j^+} \left| c_{jik}(y, x_i^-) - c^*_{ji-1}(y, x_{i-1}^+) \right|}_{\text{left adjacency}} \\ & + \beta(c_{jik}, c^*_{ji+1}) h_p \underbrace{\sum_{y=y_j^-}^{y_j^+} \left| c_{jik}(y, x_i^+) - c^*_{ji+1}(y, x_{i+1}^-) \right|}_{\text{right adjacency}} \\ & + \beta(c_{jik}, c^*_{j-1i}) w_p \underbrace{\sum_{x=x_i^-}^{x_i^+} \left| c_{jik}(y_j^-, x) - c^*_{j-1i}(y_{j-1}^+, x) \right|}_{\text{top adjacency}} \\ & + \beta(c_{jik}, c^*_{j+1i}) w_p \underbrace{\sum_{x=x_i^-}^{x_i^+} \left| c_{jik}(y_j^+, x) - c^*_{j+1i}(y_{j+1}^-, x) \right|}_{\text{bottom adjacency}} \Bigg\} \end{aligned} \quad (9)$$

which measures the weighted sum of absolute side disparity discrepancies between the surface of chromosome c_{jik} under evaluation and the currently best local solutions $c^*_{j\pm1\ i\pm1}$ of each of the four symbionts $P_{j\pm1\ i\pm1}$. β in (9) are adaptive weights, which are discussed in Section VII-C. f_{C0} is scaled by the sum of all $2 \cdot (h_p + w_p)$ weights β to normalize them to a sum of unity.

The effect from the minimization of f_{C0} is illustrated in Fig. 3(a) for a simplified 1-D analog. The two end points of the curve d_i try to obtain similar values with the adjacent

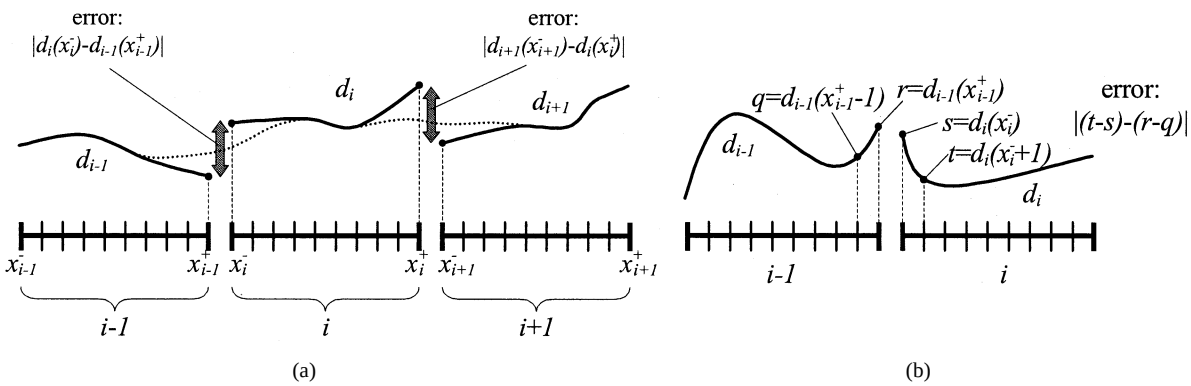


Fig. 3. One-dimensional (1–D) analogues for the enforcement of C_0 and C_1 continuities. (a) f_{C0} minimizes the disparity jumps at the patch boundaries. A likely minimization of the errors between the patches i and $i \pm 1$ could produce the dotted joining curves. (b) Minimization of the f_{C1} error reduces the peak effect.

sides of each neighbor $d_{i\pm1}$, in order to attenuate the disparity jumps. Since the curvature of each surface is limited by the polynomial degree N, each curve is forced toward altering its entire profile to create a surface that is continuous with the neighboring ones. A likely outcome is shown as a dotted curve in Fig. 3(a). In this way, by evolving all populations in parallel, local interactions are propagated to a global scale to yield a globally smooth disparity map. The benefit gained by f_{C0} is twofold; not only patch boundary discontinuities are reduced, but also each surface itself is regularised to the matches that are consistent with its neighbors. For example, if d_i has many mismatched elements, its neighbors $d_{i\pm1}$ have a good chance of rectifying its profile to support more consistently their own possibly stronger surface profiles.

Since the lack of first-order C_1 continuity can give rise to a *peak effect* at the patch boundaries, we also use a second term f_{C1} to produce more smoothly flowing surfaces. The corresponding 1-D case of Fig. 3(b) exemplifies the situation. Although the closest endpoints x_{i-1}^+ and x_i^- of d_{i-1} and d_i have similar values, their rapidly and similarly sloping neighboring points x_{i-1}^+-1 and x_i^-+1 cause a peak, which is likely unwanted due to the inherent smoothness of the physical world. The minimization of f_{C1} ensures suppression of this effect by aggregating such errors at discrete side pixel locations [finite differences are used instead of derivatives since (6) has already calculated the disparity values at those pixels] as in (10), shown at the bottom of the page.

Note that the above two discontinuities are also undesirable in computer graphics, where boundary-based object representations are sought. When, for instance, splines are used to model different regions of an object's surface, similar surface continuities are needed. In these cases, rigid conditions, such as equality and colinearity are imposed on selected spline control points [71], [72]. For the current work, we could use

$$
\begin{aligned}
f_{C1}(c_{jik}) = \frac{1}{\sum\beta}\Bigg\{ & \beta(c_{jik}, c^*_{ji-1}) \cdot h_p \cdot \underbrace{\sum_{y=y_j^-}^{y_j^+} \left| c_{jik}(y, x_i^- + 1) - c_{jik}(y, x_i^-) - c^*_{ji-1}(y, x_{i-1}^+) + c^*_{ji-1}(y, x_{i-1}^+ - 1) \right|}_{\text{left adjacency}} \\
& + \beta(c_{jik}, c^*_{ji+1}) \cdot h_p \cdot \underbrace{\sum_{y=y_j^-}^{y_j^+} \left| c_{jik}(y, x_i^+ - 1) - c_{jik}(y, x_i^+) - c^*_{ji+1}(y, x_{i+1}^-) + c^*_{ji+1}(y, x_{i+1}^- + 1) \right|}_{\text{right adjacency}} \\
& + \beta(c_{jik}, c^*_{j-1i}) \cdot w_p \cdot \underbrace{\sum_{x=x_i^-}^{x_i^+} \left| c_{jik}(y_j^- + 1, x) - c_{jik}(y_j^-, x) - c^*_{j-1i}(y_{j-1}^+, x) + c^*_{j-1i}(y_{j-1}^+ - 1, x) \right|}_{\text{top adjacency}} \\
& + \beta(c_{jik}, c^*_{j+1i}) \cdot w_p \cdot \underbrace{\sum_{x=x_i^-}^{x_i^+} \left| c_{jik}(y_j^+ - 1, x) - c_{jik}(y_j^+, x) - c^*_{j+1i}(y_{j+1}^-, x) + c^*_{j+1i}(y_{j+1}^- + 1, x) \right|}_{\text{bottom adjacency}} \Bigg\}
\end{aligned}
\tag{10}
$$

splines instead of polynomials and enforce such smoothness symbiotically in exactly the same manner.

The final symbiotic error term f_C is another form of continuity and measures the local coherency between patch surfaces by enforcing nearby patches to have similar disparity values. This coherency error uses a local geographic neighborhood $\aleph(P_{ji})$ around every patch P_{ji} to aggregate the absolute differences of average disparity values between the current chromosome c_{jik} and the elite ones c^*_{mn} of each symbiont as

$$f_C(c_{jik}) = \frac{\sum\limits_{P_{mn}\in\aleph(P_{ji})} \beta(c_{jik}, c^*_{mn}) \cdot |\hat{c}_{jik} - \hat{c}^*_{mn}|}{\sum \beta}. \quad (11)$$

The symbols $\hat{c}$ denote the averaged patch disparities and $\aleph(P_{ji})$ contains all patches within a fixed distance of approximately 30 pixels around and excluding P_{ji}.

It can be seen that in (9)–(11), we have solely used the best members c^* from each symbiont. This partnering strategy (see variations in Section II) is a rather greedy one, since instead of evaluating a chromosome with the currently elite representatives, more auxiliary evaluations with various combinations of weaker representatives could be used to allow for more realistic symbiosis. However, experimentation showed that such an expensive scheme is not needed. The reason lies in the fact that we have used (see ensuing subsections) symbiosis in an adaptive manner that compensates for the greedy behavior of this strategy.

Another important point to be mentioned about these three terms, is that the continuity and coherency assumptions cannot hold near object boundaries. At these areas, nearby patches have to be allowed to diverge in order to model the depth separations between scene object. For this reason, we make the three error terms robust by truncating each of the absolute difference contributions to a small fixed value of ρ_D. In this way, discontinuities between different surfaces will be allowed and not smoothed out.

The above five error families f_M, f_O, f_{C0}, f_{C1}, and f_C are embedded in the evolutionary optimization framework of the previous sections using two types of weights; the β ones that balance the individual symbiotic contributions and the λ ones used to combine/scalarise all the errors.

C. Adaptive Symbiotic Forces

The weight $\beta(c_1, c_2)$ adjusts dynamically the effect of evaluating c_1 (the current member c_{jik} under evaluation) with respect to c_2 (the best member $c^*_{j\pm1\ i\pm1}$ or c^*_{mn} in (9) and (10) or (11), respectively). It can be envisaged as a measure of the *symbiotic force* between the two species members. Each such weight is the direct product of three further quantities $\beta_i \in [0, 1]$ described below.

β_1 is a monotonically decreasing function of the difference $u(c_1, c_2)$ of the average pixel intensities in L within the two patches corresponding to c_1 and c_2. This is because the more similar the intensities of the two patches, the more likely they belong to the same object region. In such cases, the disparity discrepancies between c_1 and the neighboring symbionts need more minimization. This weight is defined as the truncated line

$$\beta_1 \equiv \max\left(1 - \frac{|u(c_1, c_2)|}{u_{\max}}, 0\right) \quad (12)$$

where $u_{\max}$ is a fixed threshold which depends on the intensity differences between all the symbiotically associated patch pairs in the collective.

β_2 is designed to decrease when the symbiont c_2 has higher error than c_1. It is defined as

$$\beta_2 \equiv \frac{E_{\text{self}}(c_1)}{E_{\text{self}}(c_1) + E_{\text{self}}(c_2)} \quad (13)$$

where $E_{\text{self}}(c)$ consists of the combined errors f_M and f_O of c as defined in (15). In this way, continuity or coherency as dictated by a symbiont c_2 is less trustworthy when its internal energy E_{self} is higher than the energy of c_1 (note that only the self-errors E_{self} are used here since the E_{symb} part of (15), which combines f_{C0}, f_{C1} and f_C is not calculated yet for c_1).

Finally, β_3 regulates the dominance between a symbiont c_2 and its peers involved in the evaluation of c_1. We define it as

$$\beta_3 \equiv 1 - \frac{E_{\text{total}}(c_2)}{\sum\limits_{c} E_{\text{total}}(c)} \quad (14)$$

where E_{total} is the total error defined in (16), which combines (6) and (8)–(11). The denominator is summated over all symbionts c of c_1 (that is the four $c^*_{j\pm1\ i\pm1}$ for (9) and (10) or all the c^*_{mn} in $\aleph(P_{c1})$ for (11)). In this way, β_3 controls the influence of the participating symbionts by biasing coevolution to be more compliant with the fitter species. Note that although β_1 is fixed during evolution as it depends solely on the image intensities, β_2 and β_3 adapt constantly to the errors. Such adaptive behavior has the advantage of making the global propagation of all local optimizations more realistic and avoid the local optima effectively.

D. Dynamic Multiobjective Optimization

The need for simultaneous satisfaction of the five error families f_M, f_O, f_{C0}, f_{C1}, and f_C gives rise to a multiobjective (or vector) optimization problem; such problems are typically studied using Pareto optimality theory [73]. In the context of evolutionary optimization, there are various techniques to enable recovery of multiple solutions from the Pareto optimal set [74]–[81]. In this paper, however, due to the large-scale of the problem, we use a weighted-sum approach from the field of classical multiobjective theory [73] to recover a single Pareto optimal point. We directly scalarise the five terms in a single-valued performance index as the mixture of the following two energies:

$$\begin{aligned} E_{\text{self}}(c_{jik}) =& f_M(c_{jik}) + \lambda_O \cdot f_O(c_{jik}) \\ E_{\text{symb}}(c_{jik}) =& \lambda_{C0} \cdot f_{C0}(c_{jik}) + \lambda_{C1} \cdot f_{C1}(c_{jik}) \\ &+ \lambda_C \cdot f_C(c_{jik}) \end{aligned} \quad (15)$$

where λ_O, λ_{C0}, λ_{C1}, and λ_C are fixed user-defined weights.

However, at the beginning of evolution, the E_{symb} energy may not be as effective as later on. Also, some unfortunate phenotypic arrangement in the initialization phase among nearby

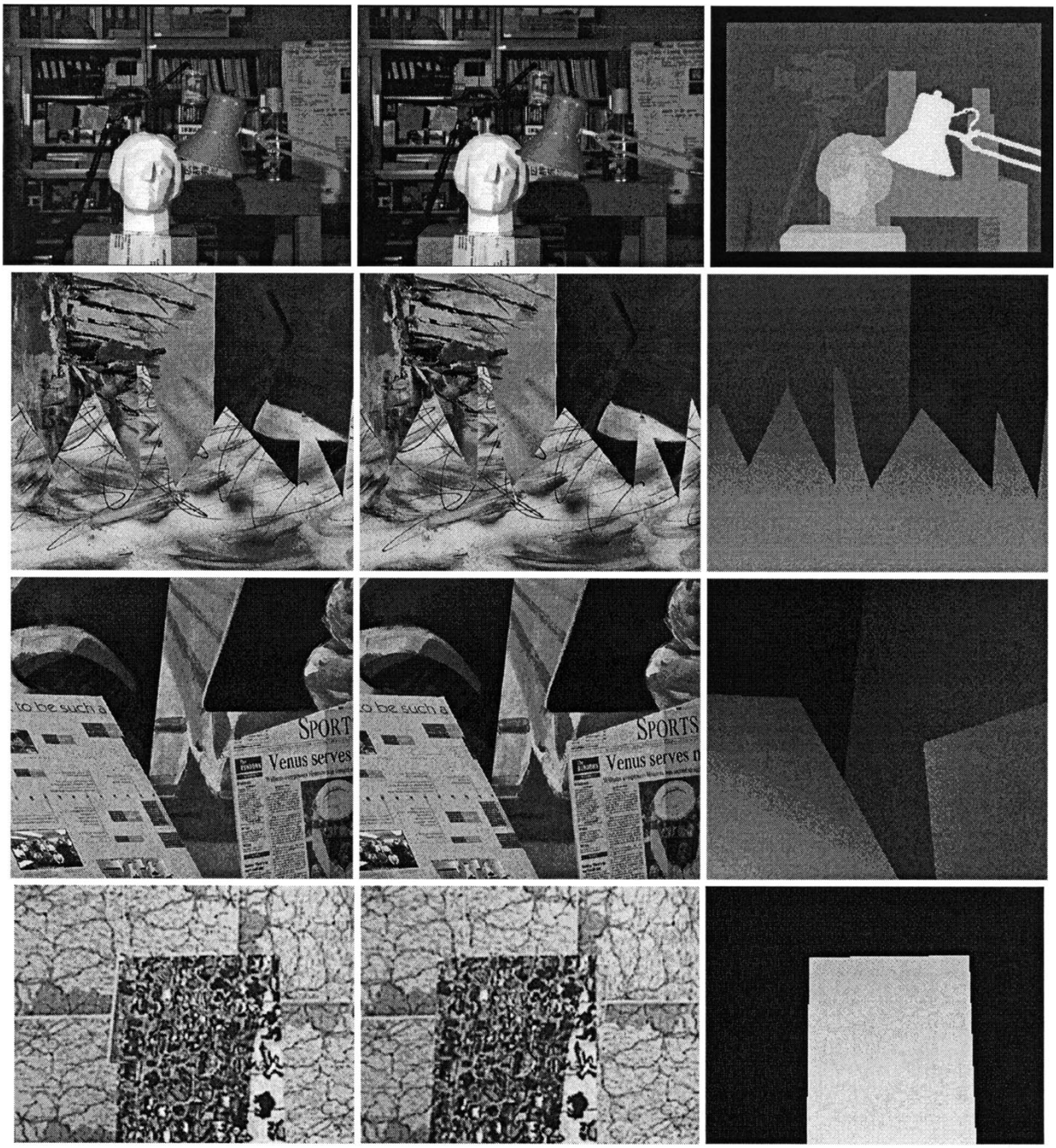


Fig. 4. Test images of the Tsukuba, Sawtooth, Venus, and Map pairs, from top to bottom. The first two columns show each stereo pair, while the third column the provided ground truths.

patches of low symbiotic errors, may lock optimization to local minima owing to the nature of our decomposition. For these reasons, we combine E_{self} and E_{symb} using a time-varying weighting as

$$\begin{aligned} E_{\text{total}}(c_{jik}) =& E_{\text{self}}(c_{jik}) + \lambda(t) \cdot E_{\text{symb}}(c_{jik}) \\ \lambda(t) =& \min\left(\frac{t}{t_s}, 1.0\right) \end{aligned} \tag{16}$$

where t_s is a user-defined generation threshold and t is the current generation. In this way, the *symbiotic strength* $\lambda(t)$ biases the optimization toward the self energies first and gradually introduces the importance of symbiotic interactions. At generation t_s, the symbiosis culminates and for the remaining evolution

TABLE I
ATTRIBUTES OF THE FOUR STEREO PAIRS

	Tsukuba	Sawtooth	Venus	Map
Dimensions *h*×*w*	*288×384*	*380×434*	*383×434*	*216×284*
Range Δ	[*0,15*]	[*0,19*]	[*0,19*]	[*0,29*]
Scaling	*16*	*8*	*8*	*8*
Border	*18*	*10*	*10*	*10*

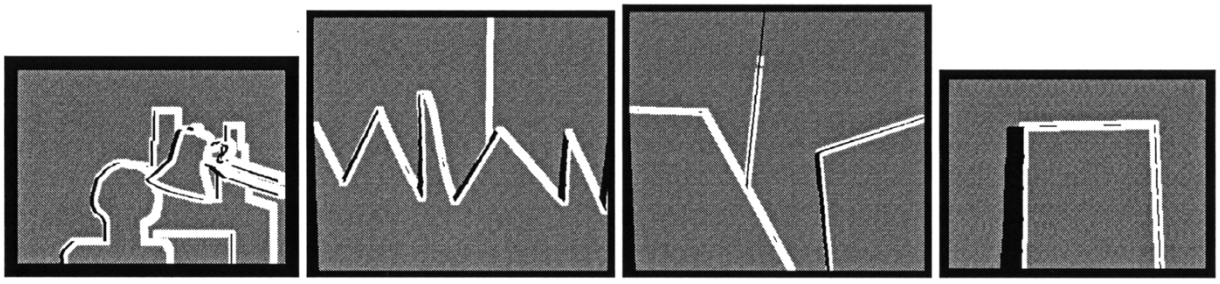


Fig. 5. The provided occluded areas (black) and the discontinuity areas (white). The surrounding black border is ignored from evaluation.

TABLE II
ALGORITHM PARAMETERS USED FOR THE EXPERIMENTS

η_{max}	t_{max}	g_{size}	g_{child}	p_c	p_m	t_c	t_s	γ	N	$h_p \times w_p$	ρ_I	ρ_D	ξ	λ_O	λ_{C0}	λ_{C1}	λ_C
1.6	300	50	25	0.6	0.25	100	200	0.3	1	4×4	6.0	2.0	200	15.0	1.0	3.0	1.5

$\lambda_{C0}, \lambda_{C1}, \lambda_C$ assume their full strength, so that at $t_{\max}$ the collective converges to a global equilibrium.

$E_{\text{total}}(c_{jik})$ is the principal GA objective function used in each of the $w_g \cdot h_g$ populations of the collective. Note that the time-varying strength $\lambda(t)$ and the error-adaptive β forces give rise to a nonstationary optimization environment. Such optimizations of dynamic problems have attracted recent interest in various domains [82]. A primary problem in such cases is that convergence may lead to population diversity loss, making, thus, the system insensitive to environmental changes after some generation and various mechanisms have been devised to counteract this effect. In this paper, this is another reason that (as described in Section V-A), we increase mutation by halving the crossover probability p_c during evolution.

VIII. IMPLEMENTATION AND EXECUTION SCHEDULING

We have prototyped the proposed algorithm in C++, where the collective of species is implemented as a queue of GA population objects. Each object is evolved by one generation in turn and the entire queue is processed for a total of $t_{\max}$ cycles. However, in order to demonstrate the behavior of a real-world asynchronous parallel implementation, the queue is perturbed at each cycle to model the fact that one processing unit (processor or network node) may be slower than others or fail temporarily to communicate. The scheduler object reshuffles the queue at every cycle, so that the order of evaluation is changing. We have experimented with different perturbation schemes, with arbitrary eliminations and/or duplications of population objects within the queue, with no noteworthy differences in the produced results. Due to the way symbiosis is sustained, the system is resistant to the lack of such synchronization.

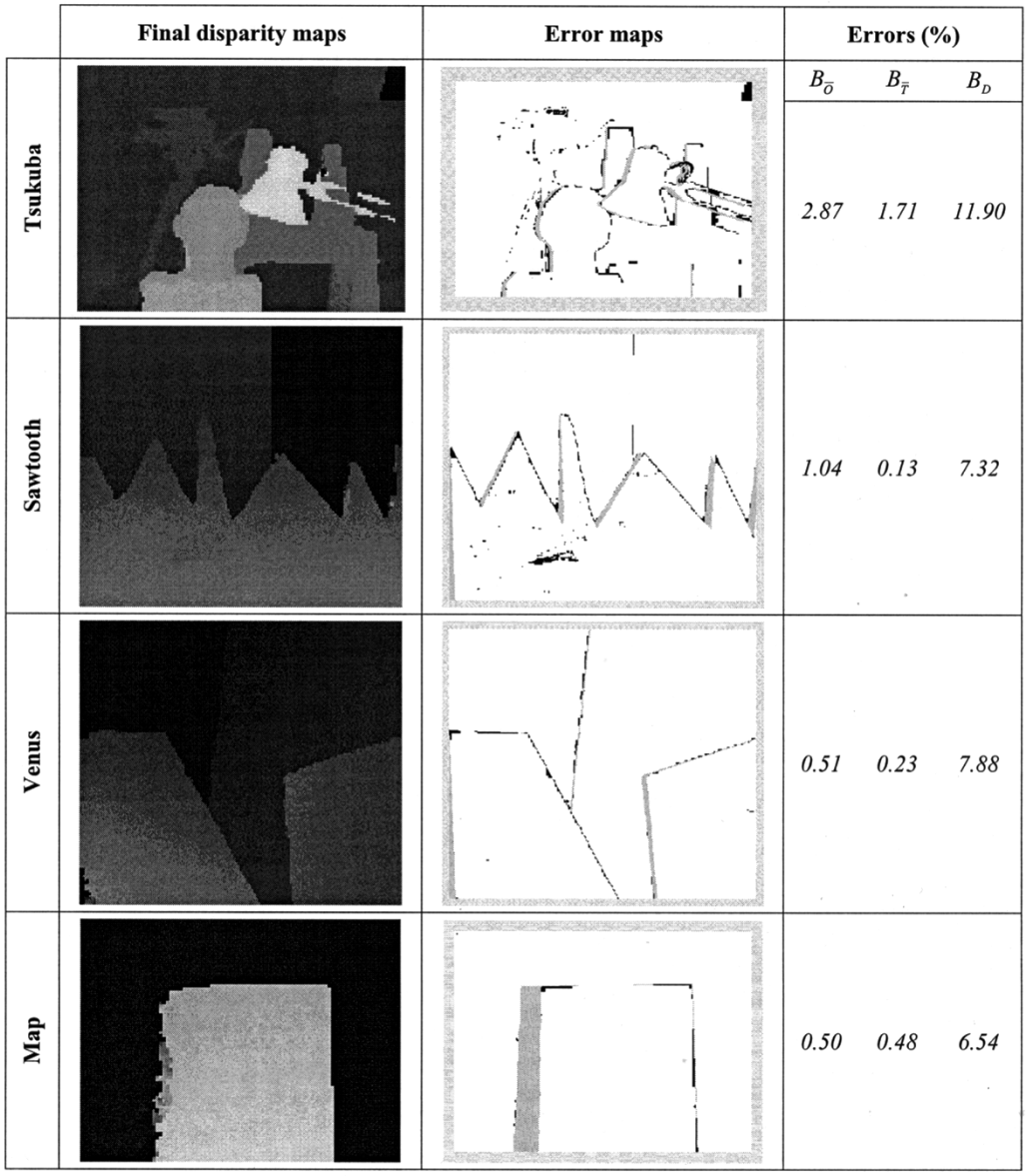
The only interspecies communication arises when some c_{jik} needs evaluation of its $E_{\text{symb}}(c_{jik})$ energy term. As discussed before, each related species P_{mn} has to make available to its symbionts the following data pertaining to its currently top member c^*_{mn}: the $4(h_p + w_p - 4)$ disparity values calculated from the two outermost pixel perimeters of its patch, its average disparity $\hat{c}^*_{mn}$, and the values of $E_{\text{self}}(c^*_{mn})$ and $E_{\text{symb}}(c^*_{mn})$. We designed each population object to maintain a memory buffer to store all these values. Their update occurs when the P_{mn} object completes the generation. We have implemented no buffer synchronization for these values. When a member c_{jik} needs to have its error calculated, the above values are read from all its P_{mn} symbionts without waiting. In an asynchronous implementation, however, it may be the case that an object P_{mn} is currently updating its internal buffer, when some P_{ji}'s are in need of its contents. To test this effect (and also evaluate the time savings in a parallel version—see Section IX-D), we have also implemented a parallel version of the above queue setup using multithreaded programming. A user-defined number of threads access and evolve different objects from the queue simultaneously and in random order. Again, the system behaved robustly despite the lack of thread locking and synchronization, which would undermine parallelism.

A final remark relates to the number of function evaluations, which is an essential ingredient in the complexity estimation of an evolutionary approach. Assuming an average of $t_{\max}$ generations for each of the $h_g \cdot w_g$ populations, the collective needs a total of $h_g \cdot w_g \cdot (g_{\text{size}} + t_{\max} \cdot g_{\text{child}})$ evaluations for E_{self} and $h_g \cdot w_g \cdot g_{\text{size}} \cdot (1 + t_{\max})$ evaluations of E_{symb}; this gives an overhead of $h_g \cdot w_g \cdot t_{\max} \cdot (g_{\text{size}} - g_{\text{child}})$ additional evaluations for the latter term. This is because reevaluation of E_{self} is not needed for old population members but only for the new g_{child} offspring. Symbiotic terms E_{symb}, however, need reevaluation for all members both old and new at each generation since the symbiont populations are in flux and their buffers' contents may change. Some lazy evaluation scheme can be devised by checking whether the symbionts' top member have changed in order to save time, but for simplicity, we have not implemented such a scheme here.

IX. EXPERIMENTS AND RESULTS

A. Test Imagery and Evaluation Criteria

Until recently, it had been exceedingly difficult to evaluate the quality of the SCP algorithms found in the literature due to the lack of common test imagery and accurate ground truth



	Final disparity maps	Error maps	Errors (%)		
			$B_{\bar{O}}$	$B_{\bar{T}}$	B_D
Tsukuba			2.87	1.71	11.90
Sawtooth			1.04	0.13	7.32
Venus			0.51	0.23	7.88
Map			0.50	0.48	6.54

Fig. 6. Resulting disparity maps for the four test cases. Error maps contain bad pixels (in black), correct pixels (in white) and excluded occlusions and borders (in grey). The percentages for the three types of errors are shown in the right column.

information. The latest reviewing work of [22], however, has provided the means for a systematic evaluation of different matchers with controlled imagery of known ground truth. In this paper, we use the same test cases and evaluation criteria for quantitative evaluation of the produced disparity maps, as well as for comparison with other methods.

Fig. 4 shows the left and right images of the four stereo pairs, referred to as Tsukuba, Sawtooth, Venus, and Map. The right column of the figure shows the four actual scaled disparity maps used to estimate the disparity errors. Table I summarizes the dimensions, disparity search ranges, borders and scaling factors of the disparity maps for these pairs. In addition to the ground truth, there is information about occlusions, discontinuities and texture. In Fig. 5, for instance, occluded areas are shown in black and white areas represent surface discontinuities. The black surrounding borders are excluded from evaluation because image boundaries hinder matching.

Error statistics for each test case in [22] are gathered within three areas:

- $B_{\bar{O}}$: all pixels in nonoccluded regions;
- $B_{\bar{T}}$: all pixels in regions without texture;
- B_D: all pixels near discontinuities.

The first type of error is calculated across the entire image, while the other two within regions where matching is difficult in order to provide a more focused analysis to problematic image regions. In all cases, the occlusions shown in Fig. 5 are excluded from evaluation. The error in each of these areas is expressed as the percentage of bad pixels, that is, pixels with computed disparities different from the actual ones. The first error, for example, is defined as

$$B_{\bar{O}} = \frac{1}{|\bar{O}|} \sum_{(y,x)\in\bar{O}} [|d_{\text{true}}(y,x) - d_{\text{computed}}(y,x)| > \delta_{\text{error}}] \tag{17}$$

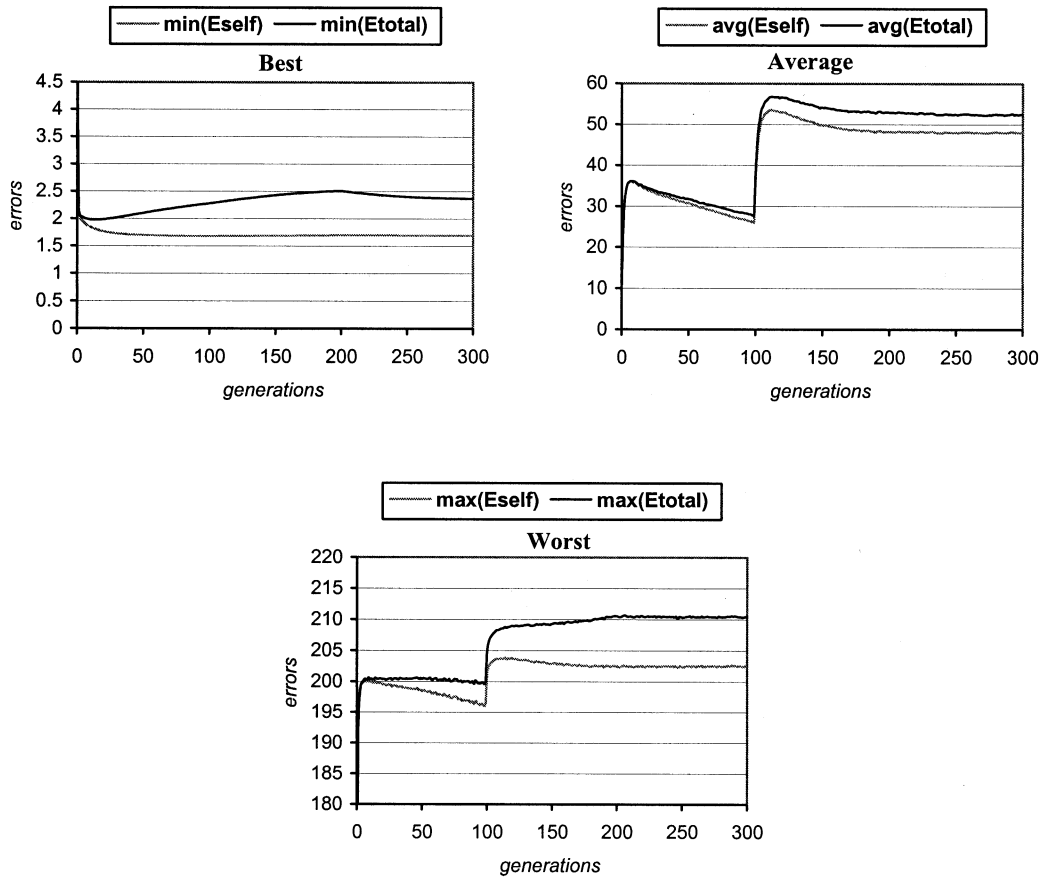


Fig. 7. Evolution of Tsukuba; Progression of best, average, and worst values of the E_{self} and E_{total} energy terms averaged over the entire collective.

where a threshold $\delta_{\text{error}} = 1.0$ is used. $B_{\bar{T}}$ and B_D are similarly defined for the textureless and the discontinuity regions.

B. Resulting Disparity Maps and Error Analysis

The behavior of our algorithm[1] depends on the various parameters described in Sections III–VII. Manual fine-tuning of these parameters has produced the values shown in Table II. We use these values for all experiments, unless otherwise stated. Although better results could be obtained by varying these parameters with each image pair, we keep them fixed in order to comply with the criteria of [22] and allow a more realistic evaluation without assuming any *a priori* knowledge.

The final disparity maps of the algorithm are shown in Fig. 6 and can be compared qualitatively with the true maps of Fig. 4. It can be seen that disparities are generally smooth within the object surfaces while the depth separations are preserved adequately. The erroneous pixels are shown in the error maps in black, where occluded regions are excluded and shown in grey. Tsukuba contains certain mismatches near discontinuity areas due to the large amount of detailed objects, such as the camera and its stand and the lamp. Sawtooth contains a blotch of errors in the bottom untextured area and other mismatches near discontinuities. Venus and Map appear with smaller errors which are also concentrated in the discontinuity areas. Such mismatches are due to the square patches we use, which although of small 4×4 size still cause problems as they have to obtain membership with either the object or its background surface. Overall the disparity maps appear fairly accurate within the majority of the surfaces. This is verified from the quantitative results in the right side of Fig. 6. Discontinuity regions have larger B_D error percentages, textureless $B_{\bar{T}}$ values are adequately small, while the overall error estimations $B_{\bar{O}}$ show the accuracy ranging between 97% and 99.5%.

From the above results, the symbiotic cooperation is successful and despite its local enforcement, its propagation to global scale is effective. Below, we examine in detail the convergence characteristics of such collective symbiotic interactions. Fig. 7 plots the progress of the E_{self} and E_{total} errors of (15) and (16) (for Tsukuba only as the other tests have similar curve profiles). Since we cannot examine each population independently, we average the statistics over all the $h_g \cdot w_g$ species. The three graphs refer to the best (c^*_{mn}), average, and worst members in each patch and for each generation. The best shows the (typical in GAs) rapid decrease of both errors within the first few generations. Subsequently, while the E_{self} assumes a stable level, E_{total} increases slowly until $t_s = 200$. This is owed to the increasing symbiosis strength

[1]It can be downloaded from http://www.nicve.salford.ac.uk/~yannis/software/casgas.zip, and it supports both serial and parallel execution modes, as well as different GA and stereo parameter input and images for further evaluation.

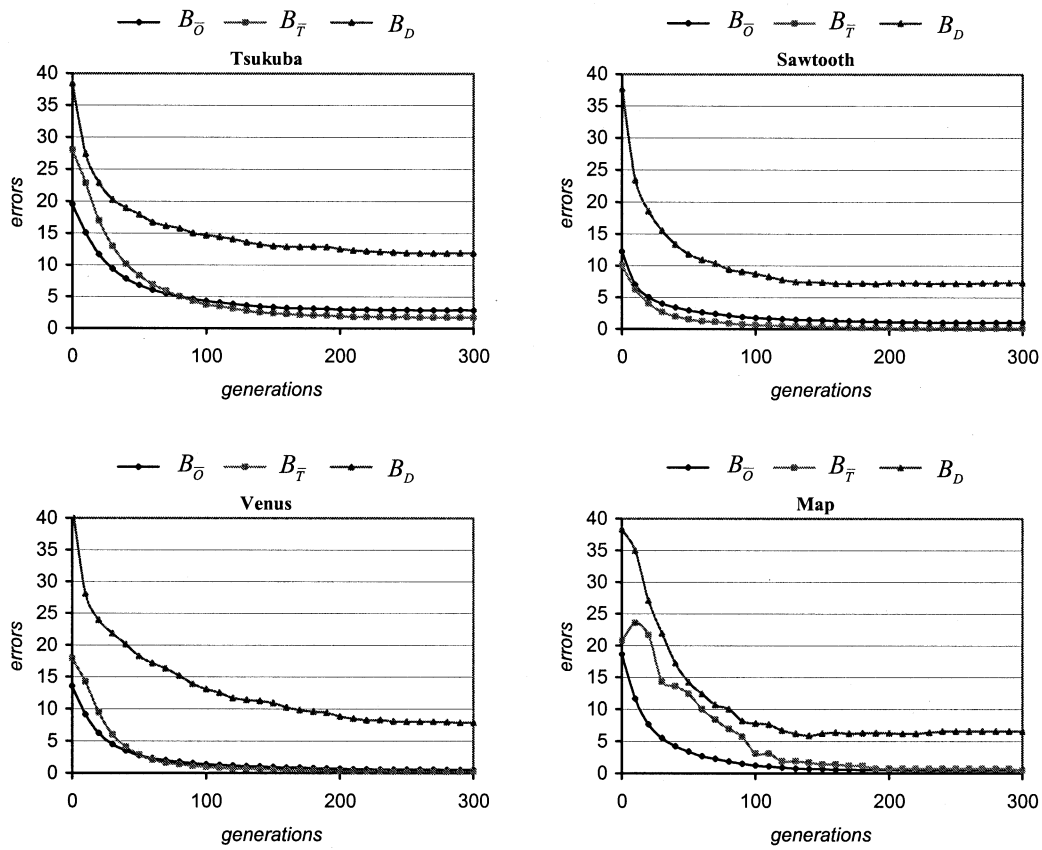


Fig. 8. Behavior of the $B_{\bar{O}}$, $B_{\bar{T}}$, and B_D errors during evolution (measured in steps of ten generations).

$\lambda(t)$ of (16). Following that, there is consistent decrease of E_{total} until termination at $t_{\max} = 300$. For both the average and worst graphs, we observe a short initial error increase, due to the initial population genes being readjusted to the symbiotic terms (at $t = 0$, the calculation of E_{symb} is solely based on the ranking of E_{self}). The steep error increase at $t_c = 100$ is due to the increase of the mutation rate explained in Section V-A. After that, the average obtains a stable error decrease.

For the particular problem we solve, one cannot guarantee convergence solely from the error reduction shown in Fig. 7. Unlike classic function extremization problems, reduction of E_{total} cannot guarantee minimization of the $B_{\bar{O}}$, $B_{\bar{T}}$, and B_D errors. This is for two reasons: First, the structure of the self/symbiotic cost functions of (6) and (8)–(11) is a matter of subjective problem design. Second, symbiotic interactions are very complex and cannot guarantee avoidance of oscillatory behavior or local minima. Although the results show adequate final error levels, it is important to examine the reduction of the actual problem errors during evolution. Fig. 8 shows the convergence of such errors for all experiments. It can be seen that the reductions are very consistent (the initial irregularity of $B_{\bar{T}}$ in Map is due to the very small amount of 420 textureless pixels) and within the first 100 generations the mismatches are dramatically reduced. This verifies the correctness of the proposed performance index designs.

For a qualitative alternative of the above graphs, Figs. 9 and 10 show the disparity and error maps for the Tsukuba and Venus for a selected number of generations. The gradual reductions of erroneous matches are clearly visible.

C. Comparisons With Other Algorithms

We compare the proposed algorithm with 23 others evaluated in [22] and the Middlebury page.[2] The evaluation makes use of the imagery and the three error types used in Sections IX-A and B to compare a large set of best-performing recent stereo matchers. Table III reproduces some of these comparisons with optimization methods such as graph cuts [12], dynamic programming [6], [7], GAs [21], and hybrid methods [13], [14]. The table shows the errors for each algorithm and for each image individually ($B_{\bar{T}}$ is excluded from the evaluation of the Map, as it is textured almost everywhere). In addition to the error percentages, the table contains the relative comparison rank within each error column. Each algorithm in the table is sorted according to its overall rank.

It can be seen that due to the different scene and image properties of each data set, each algorithm's output quality may vary between data sets and error types. The differences between the best [14] and the worst [7] algorithms illustrate roughly the error ranges of the complete table. For all errors, our algorithm stands in the top half of the table with a current overall rank of 5. It

[2]The *Middlebury Stereo Vision Research Page* is accessible at http://www.middlebury.edu/stereo. It contains comparative results additional to the ones published in [22] and supports uploading and evaluation of disparity maps online. The reader can browse the comparison data in detail for all 24 entries (including the one proposed here), as well as download extra test imagery.

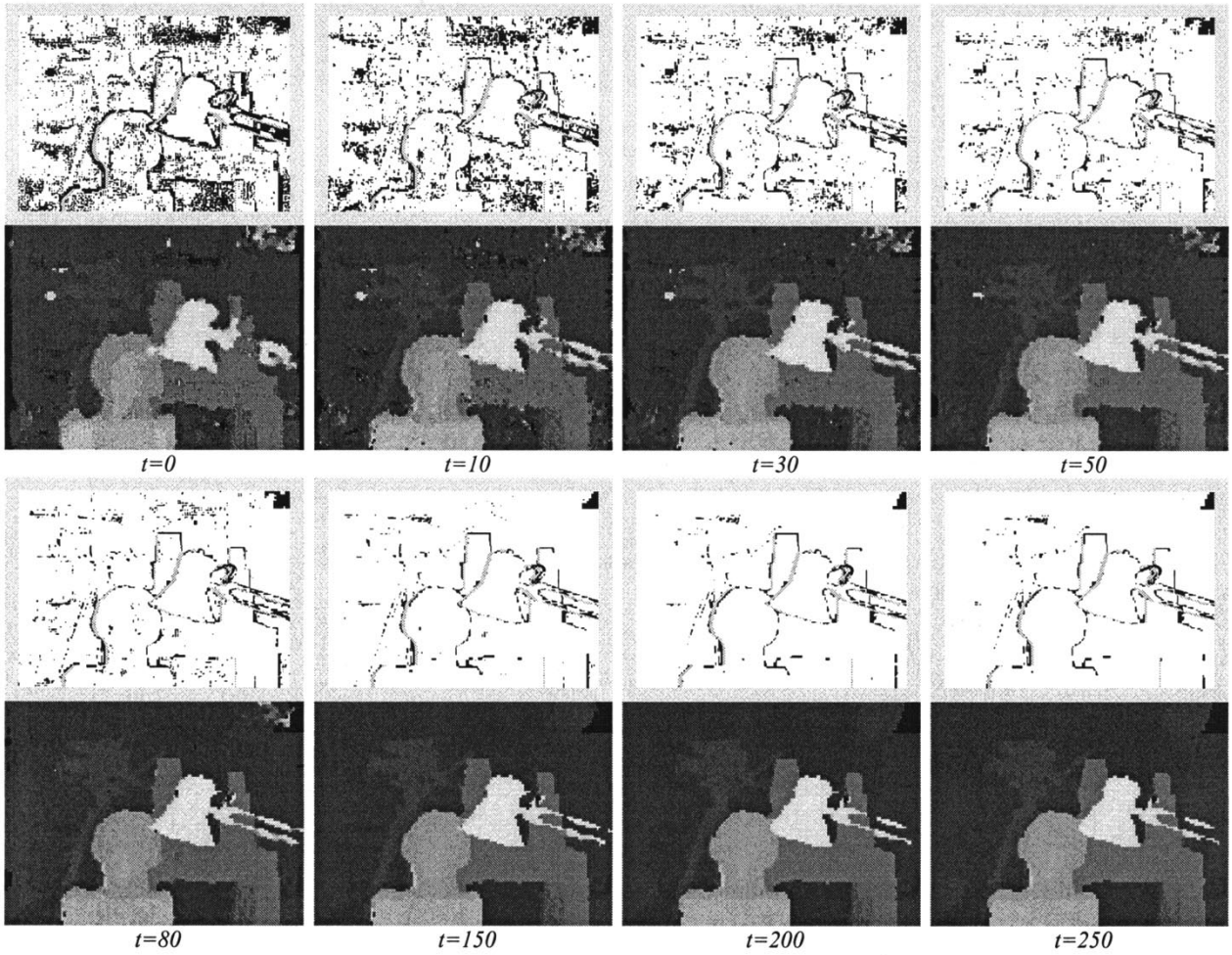


Fig. 9. Sequence of the Tsukuba evolution for selected cycles t.

should be noted that the only other evolutionary method from the entire set of the 24 methods is the one of [21] at rank 16.

The majority of matchers calculate disparity values from a discrete (integer or fractional) set and need post-processing for subpixel accuracy. From all methods in the table, only [13] and [14] use continuous disparities by fitting planes and splines, respectively, to the visible surfaces. Our method also uses continuous estimation via piecewise surface fitting. However, it does not use repetitive alternations of segmentation (with graph cuts) and fitting (with gradient) as in [13] and [14] but single steps of repetitive evolution. This provides the advantage of asynchronous large-scale parallelization.

D. Speed Issues and Parallelization

Concerning the speed of the proposed algorithm, it belongs to the relatively computationally demanding ones. Execution on an AMD 1.45-GHz processor takes for the four images 84, 127, 129, and 46 min in the above order, for the coevolution, plus 1–1.5 min for the hybridization and initialization phases. Some of the other methods, such as ones based on local search or dynamic programming are considerably faster. However, those methods calculate discrete disparities, while methods which calculate continuous surfaces of unrestricted shape can be much slower due to the larger search spaces. Reference [14], for instance, also reports similarly high execution times (120–480 min on a 0.45 GHz UltraSparc II). In general, the more accurate methods are slower, but implementation speedups can greatly affect execution. For instance, for simplicity in our algorithm we have not implemented any precomputation/cost storage schemes for (6) or any lazy evaluation schemes for the symbiotic terms, which could reduce the load of the repetitive GA cost assignments drastically.

Since, to the best of our knowledge, the proposed method is the only continuous method of asynchronous parallelization properties, we exemplify the speedup gain from such parallelism by experimenting with the simple multithreaded model described in Section VIII. Fig. 11 shows averaged statistics from multiple executions of the test data on a quad-processor machine. Because we use slower processors (Intel 0.85 GHz), we run a simpler problem with larger patches of sides $h_p = w_p = 7$. We examine cases for a number of threads up to the number of available CPUs so that no thread needs to stay idle. We let the preemptive multitasking of the operating system (Windows) to do the scheduling of the threads and we have not forced thread affinities, exclusive CPU usages, or thread locking.

It can be calculated from the chart that the average time gain for two, three, and four threads over the case of a single thread is

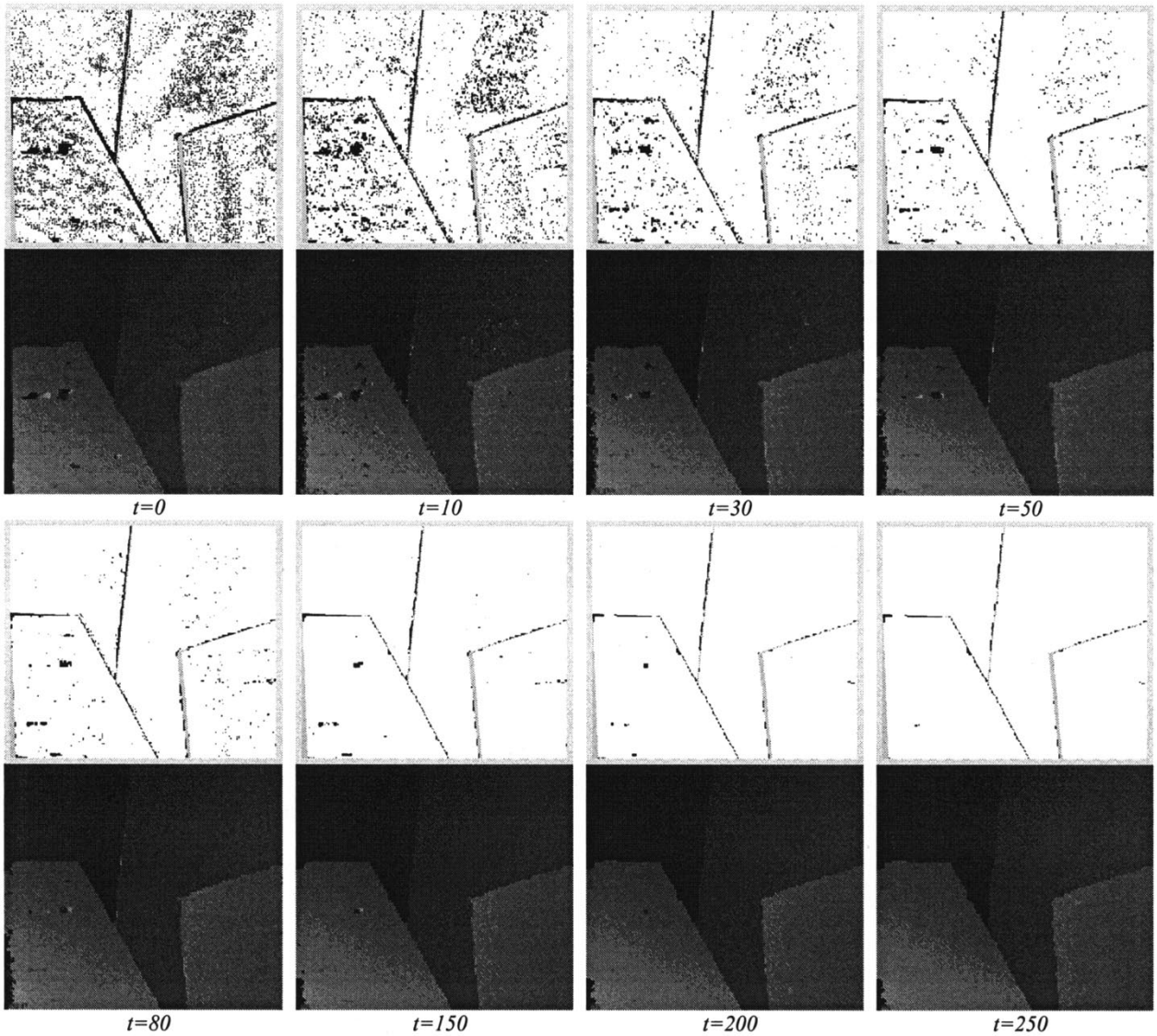


Fig. 10. Sequence of the Venus evolution for selected cycles t.

TABLE III
SOME OF THE CURRENT (05/2003) 24 ENTRIES OF THE MIDDLEBURY EVALUATION TABLE. SUBSCRIPTS ON THE RIGHT SIDE OF EACH ERROR PERCENTAGE DENOTE PERFORMANCE RANK FOR THE INDIVIDUAL ERROR COLUMN. THE LAST COLUMN GIVES THE RANK OF OVERALL PERFORMANCE

Algorithm	Tsukuba			Sawtooth			Venus			Map		Rank
	$B_{\bar{O}}$	$B_{\bar{T}}$	B_D	$B_{\bar{O}}$	$B_{\bar{T}}$	B_D	$B_{\bar{O}}$	$B_{\bar{T}}$	B_D	$B_{\bar{O}}$	B_D	
Layered [14]	1.58_{3}	1.06_{5}	8.82_{3}	0.34_{1}	0.00_{1}	3.35_{1}	1.52_{6}	2.96_{14}	2.62_{2}	0.37_{9}	5.24_{9}	1
Our method	2.87_{9}	1.71_{9}	11.90_{8}	1.04_{6}	0.13_{5}	7.32_{10}	0.51_{1}	0.23_{1}	7.88_{5}	0.50_{11}	6.54_{11}	5
GC+occl. [12]	1.27_{2}	0.43_{2}	6.90_{2}	0.36_{2}	0.00_{1}	3.65_{2}	2.79_{16}	5.39_{17}	2.54_{1}	1.79_{17}	10.08_{16}	7
Multiw Cut [13]	8.08_{21}	6.53_{18}	25.33_{22}	0.61_{4}	0.46_{12}	4.60_{4}	0.53_{2}	0.31_{2}	8.06_{6}	0.26_{4}	3.27_{4}	10
Genetic [21]	2.96_{10}	2.66_{11}	14.97_{16}	2.21_{16}	2.76_{20}	13.96_{18}	2.49_{14}	2.89_{13}	23.04_{20}	1.04_{15}	10.91_{18}	16
Dyn. prog. [6,22]	4.12_{15}	4.63_{17}	12.34_{12}	4.84_{23}	3.71_{23}	13.26_{16}	10.10_{24}	15.01_{24}	17.12_{15}	3.33_{22}	14.04_{21}	21
Max. surf. [7]	11.10_{24}	10.70_{22}	41.99_{24}	5.51_{24}	5.56_{24}	27.39_{23}	4.36_{19}	4.78_{16}	41.13_{23}	4.17_{23}	27.88_{23}	24

52.7%, 36.7%, and 28.8%, respectively. This shows that, despite the simple parallel implementation, it scales well to the number of processors utilized. The small overhead (from the ideal linear gain of 50%, 33.3%, and 25%, respectively) can be assigned to OS thread context switching or memory bus bandwidth as there is no interthread synchronization. Other more sophisticated parallel implementation platforms could provide speedup much closer to linear, perhaps for a larger number of CPUs.

TABLE IV
ERROR AND TIME (MINS, EXCL. INITIALISATIONS) RESULTS OF EXPERIMENTS WITH ALTERING THE DEFAULT PARAMETERS OF TABLE II. ONLY THE DIFFERING PARAMETERS ARE DISPLAYED AND RELATED ONES ARE GROUPED INTO PARTS

	Parameters	Tsukuba				Sawtooth				Venus				Map		
		$B_{\bar{O}}$	$B_{\bar{T}}$	B_D	time	$B_{\bar{O}}$	$B_{\bar{T}}$	B_D	time	$B_{\bar{O}}$	$B_{\bar{T}}$	B_D	time	$B_{\bar{O}}$	B_D	time
	default (Table II)	2.87	1.71	11.90	84	1.04	0.13	7.32	127	0.51	0.23	7.88	129	0.50	6.54	46
(a)	no hybridisation	2.76	1.34	12.77	84	1.07	0.16	7.58	127	0.70	0.25	7.74	129	1.62	12.54	46
(b)	$h_p=w_p=5$	3.24	1.57	15.31	41	1.27	0.16	11.06	62	0.47	0.05	7.76	63	0.85	12.02	23
	$h_p=w_p=6$	3.51	1.79	17.27	30	1.41	0.07	12.88	45	0.62	0.25	10.08	46	1.06	14.88	16
	$h_p=w_p=7$	4.01	1.73	20.33	21	1.40	0.17	12.31	32	0.67	0.24	11.16	32	1.38	19.40	12
	$h_p=w_p=4$, $N=2$	4.00	3.31	15.12	100	2.27	0.69	10.94	144	1.64	1.40	12.61	145	5.32	19.66	49
	$h_p=w_p=9$, $N=2$	4.33	2.27	19.95	15	2.05	0.47	16.73	23	1.42	1.06	19.07	23	1.52	10.19	8
	$h_p=w_p=11$, $N=3$	5.93	4.40	25.57	14	3.20	1.17	21.18	21	2.54	1.81	22.77	21	6.00	35.06	8
(c)	$\lambda_{C0}=0$	3.85	3.35	13.68	84	1.28	0.18	8.05	127	0.82	0.71	8.57	129	0.94	8.48	46
	$\lambda_{C1}=0$	3.98	3.60	14.14	84	1.60	0.54	9.66	127	1.12	0.87	11.08	129	1.20	9.21	46
	$\lambda_{C0}=\lambda_{C1}=0$	10.00	15.12	19.81	74	3.23	3.39	13.61	113	5.44	9.72	15.55	113	4.22	18.79	43
	$\lambda_C=0$	6.94	10.06	14.98	34	1.87	0.84	7.45	51	2.97	4.60	14.95	54	1.41	9.76	19
	$\lambda_{C0}=\lambda_{C1}=\lambda_C=0$	18.01	31.14	23.12	25	7.25	12.19	16.36	38	14.83	28.59	22.90	38	13.17	32.51	14

E. Tests With Different Parameters

In this section, we examine the effect various important parameters carry (again, all four image sets are run with fixed parameters). Part (a) in Table IV shows results without using the hybridization procedure of Section VI. In this case all initial populations are given genes selected from the alphabet Φ in random. Although some columns show similar or slightly better errors, overall, the lack of hybridization gives more mismatches. This ascertains the advantage of using informed initialization using correlation. In the graphs of Fig. 8, it can be seen that the image errors at generation $t = 0$ are around 40 for B_D and 20% for $B_{\bar{O}}$ and $B_{\bar{T}}$. Examination of the same starting errors without hybridization show that they have considerably higher averages of around 85%. This, on the other hand, verifies that the proposed dynamic optimization behaves robustly and is capable of generating very large error reductions even when the starting surface coefficients are entirely random.

Part (b) in Table IV contains errors and execution times for different patch sizes $h_p \times w_p$ and polynomial degrees N. These parameters are interrelated, as in general, larger patches require higher degrees as more varying surfaces are required to model the increased level of elevation detail larger areas may enclose. However, larger patches give higher errors in regions with abrupt depth variations, for example in cases where a discontinuity occurs not at the patch boundary, but halfway its area and the patch has to assume membership with either the object or the background. Also, if details are finer than surface curvatures allow, they can get oversmoothed. Furthermore, in textureless areas high degree surfaces can produce unnecessary irregularities. These are the reasons that small 4×4 planar patches give the best results overall. However, if high accuracy is not required, using larger patches can significantly reduce the execution time as the size of the $h_g \times w_g$ collective decreases.

Finally, in part (c) in Table IV, we examine the benefit of using the various symbiotic terms f_{C0}, f_{C1}, and f_C, by removing subsets of these terms from the calculation of E_{symb} in (15). The first three rows examine exclusion of positional and first-order continuity and both, while the fourth row examines coherency only. The last row is of particular interest as it ignores from optimization all symbiotic interactions and makes evolution of all patches totally independent.

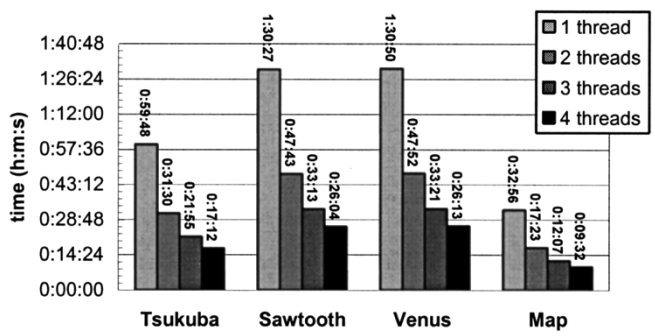


Fig. 11. Average times of running the four test sets with 7×7 patches on an Intel 0.85 GHz quad-CPU PC. The number of threads corresponds to the number of CPUs used concurrently.

From observing all rows, exclusion of any of these terms results to higher errors. Exclusion of both f_{C0} and f_{C1} causes as expected more mismatches than either one alone. The lack of coherency f_C also causes mismatches, but fewer than the exclusion of both continuities. Finally, excluding all symbiotic terms produces as expected very high errors in all images and all regions of interest $\bar{O}$, $\bar{T}$ and D. To demonstrate the nature of such errors visually, we use Fig. 12, which contains error and disparity maps for the last three rows of Table IV. When continuity is not optimized it can be seen that mismatches are grainy and isolated as they mainly occur at the patch boundaries. The middle row of the figure shows that exclusion of coherency causes blotchy errors since some patches obtain entirely wrong elevations. The combination of all errors appears in the maps of the last figure row, where optimization is driven exclusively by the E_{self} term. Comparison of these results with those of Fig. 6 verifies the importance of all symbiotic terms within E_{symb}.

The corresponding execution times in Table IV illustrate the additional computational load of symbiosis (exclusion of either f_{C0} or f_{C1} does not change the times as both terms are

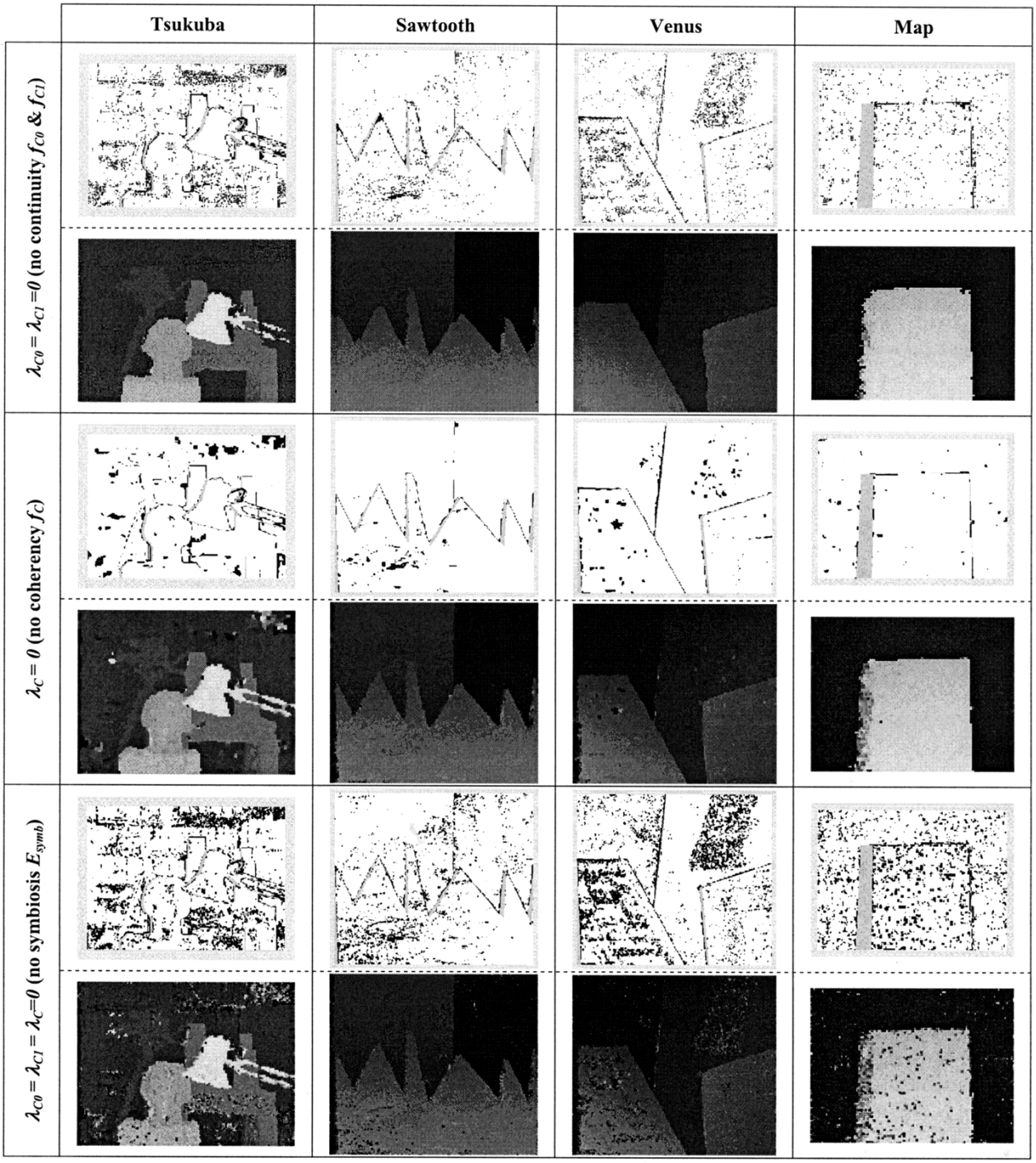


Fig. 12. Resulting disparity and error maps for cases with the symbiotic objectives of continuity and coherency excluded from optimization.

calculated by the same module). When both continuities are excluded, there is some time reduction, but it can be seen that exclusion of the f_C term speeds up optimization by a factor of 2.5. This is because, in the current implementation, the calculation of the β forces within the large $\aleph(\cdot)$ neighborhood of (11) can be very expensive.

F. Symbiotic Versus Nonsymbiotic Optimization

The above paragraphs discussed a nonsymbiotic configuration (with $\lambda_{C0} = \lambda_{C1} = \lambda_C = 0$), where all populations/species are evolved autonomously based on the index E_{self} only. Although this setup has identical population architecture, it does not solve the same problem due to the exclusion of all terms within E_{symb}. To test our proposed algorithm with a nonsymbiotic one that solves an identical problem, i.e., the smooth piecewise surface fitting, we have implemented a modified architecture. Instead of using $h_g \cdot w_g$ populations to code each patch coefficients independently, we use a single population with each chromosome being a concatenation of the $h_g \cdot w_g \cdot \tau(N)$ coefficients from all patches. In this way, the population matrices

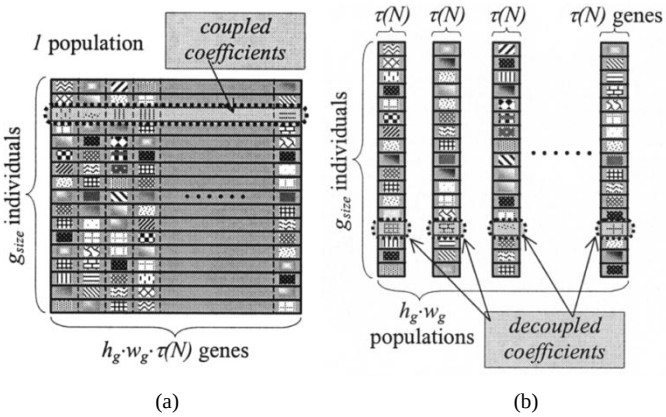


Fig. 13. Difference in the population memory organizations between (a) nonsymbiotic and (b) symbiotic models.

remain in essence the same. As exemplified in Fig. 13, the main difference is that the symbiotic setup has all the genes/coefficients decoupled and allowed to evolve independently within each patch territory, while the nonsymbiotic couples all coefficients to single genotypic strings.

Mutation and crossover operators in this single GA implementation are applied in the same way as described in Section V-A. All five objective terms of (6) and (8)–(11) now become self-energies as they are calculated from intrapatch data alone and without any external genotypic dependency. Also, the previous symbionts of a patch become the neighboring surfaces within its own host chromosome. All terms are aggregated to the single performance index

$$E_{\text{total}}(c_k) = \sum_{j=0}^{h_g-1} \sum_{i=0}^{w_g-1} \{f_M(c_{kji}) + \lambda_O \cdot f_O(c_{kji}) + \lambda(t) \cdot [\lambda_{C0} \cdot f_{C0}(c_{kji}) + \lambda_{C1} \cdot f_{C1}(c_{kji}) + \lambda_C \cdot f_C(c_{kji})]\} \quad (18)$$

where c_k is the k_{th} member of the GA, and c_{kji} is the portion of $\tau(N)$ patch coefficients at coordinates (j, i). To make the comparison objective, we have used similarly calculated β and λ weights, as well as the hybridization phase.

Regarding the time complexity of such a scheme, it is clear that only new population members need evaluation of their E_{total}. This requires a total of $h_g \cdot w_g \cdot (g_{\text{size}} + t_{\max} \cdot g_{\text{child}})$ evaluations for each f term of (18), which avoids the symbiotic overhead of recalculating f_{C0}, f_{C1}, and f_C for the old $g_{\text{size}} - g_{\text{child}}$ members at every generation (see Section VIII). For the current parameters, this can save a 30% of the total evolution time.

Despite the fewer evaluations of this nonsymbiotic architecture, experimentation showed that its optimization capabilities are overly limited. Using the same parameters as in Table II, the eleven error measurements at $t_{\max}$ were between 22% and 48%, while experimentations with larger values of $t_{\max}$, g_{size}, and g_{child} could only achieve less than a 5% improvement. These results make such an approach impractical, even when compared with the nonsymbiotic version of Section IX-E. The reason for this inefficiency can be understood by observing that the problem decomposition and the proposed symbiotic model solves $h_g \cdot w_g$ dynamic problems, each with a search space subset of $\Re^{\tau(N)}$, while the nonsymbiotic version solves a single static problem with a search space in $\Re^{h_g \cdot w_g \cdot \tau(N)}$. Consider, for instance, the 380×434 pixels Sawtooth set with 4×4 planar patches, which yields a collective of 127×145 species. In this case, the search space for the nonsymbiotic approach is $\Re^{55,245}$ whose dimensionality prohibits efficient search with any reasonable population sizes and evolution times. The symbiotic model only solves 18,415 problems in $\Re^3$.

X. Conclusion

This paper proposed a novel algorithm for solving the SCP using piecewise continuous surfaces with a parallel evolutionary optimization based on the concept of symbiosis. The original super-problem was decomposed to a large set of small patches, each corresponding to a separate species/subproblem optimisable by a simple GA. All species in the homogeneous collective were evolved concurrently with local mutualistic interactions adaptively propagated to yield global optimality. Results showed that such a model manages to counteract the detrimental effect from the arbitrary decomposition successfully and that it outperforms other equivalent nonsymbiotic optimizations and also compares competitively with the currently available SCP solvers. Specific advantages include asynchronous massive parallelization, robust handling of local minima, rapid and stable convergence, and computation of continuous disparity/depth values.

Future improvements include more versatile decomposition, as the current limitation of having equally sized rectangular patches gives errors at discontinuity regions. Patches with adaptively varying shapes could directly alleviate this problem. Alternatively, adaptive segmentation, such as the graph-cut based one used in [13] and [14], could be implemented as a second level collective embedded in the existing configuration, so that heterogeneous collectives segment and surface fit concurrently. The proposed method could also be applied to problems other than the SCP, such as problems decomposable to a very large number of smaller subproblems whose performance indices can be designed with local interactions alone (e.g., function approximation).

Acknowledgment

The authors are grateful to D. Scharstein and R. Szeliski and the Middlebury Stereo Vision Research Page [22] for allowing us to use their test images and results, as well as the Editor-in-Chief D. B. Fogel and the anonymous referees for their insightful comments and suggestions, which improved our work significantly.

References

[1] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
[2] O. Faugeras, *Three-Dimensional Computer Vision*. Cambridge, MA: MIT Press, 1993.
[3] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. New York: McGraw-Hill, 1995.
[4] J. Banks and M. Bennamoun, "Reliability analysis of the rank transform for stereo matching," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 870–880, Dec. 2001.
[5] I. J. Cox, S. L. Higorani, S. B. Rao, and B. M. Maggs, "A maximum likelihood stereo algorithm," *Comput. Vis. Image Understanding*, vol. 63, no. 3, pp. 542–567, 1996.

[6] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *Int. J. Comput. Vis.*, vol. 33, no. 3, pp. 181–200, 1999.
[7] C. Sun, "Fast stereo matching using rectangular subregioning and 3-D maximum-surface techniques," *Int. J. Comput. Vis.*, pp. 99–117, 2002.
[8] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 675–684, July 2000.
[9] L. Alvarez, R. Deriche, J. Sanchez, and J. Weickert, "Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach," *J. Vis. Commun. Image*, vol. 13, no. 1–2, pp. 3–21, 2002.
[10] M. A. Gennert, "Brightness-based stereo matching," in *Proc. IEEE 2nd Int. Conf. Computer Vision*, 1988, pp. 139–143.
[11] G. Q. Wei, W. Brauer, and G. Hirzinger, "Intensity- and gradient-based stereo matching using hierarchical Gaussian basis functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 1143–1160, Nov. 1998.
[12] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 508–515.
[13] S. Birchfield and C. Tomasi, "Multiway cut for stereo and motion with slanted surfaces," in *Proc. Int. Conf. Computer Vision*, 1999, pp. 489–495.
[14] M. Lin and C. Tomasi, "Surfaces with occlusions from layered stereo," Ph.D. dissertation, Robotics Laboratory, Stanford Univ., Stanford, CA, 2002.
[15] J. J. Lee, J. C. Shim, and Y. H. Ha, "Stereo correspondence using the hopfield neural network of a new energy function," *Pattern Recognit.*, vol. 27, no. 11, pp. 1513–1522, 1994.
[16] S. Taraglio and A. Zanela, "Improving a real-time neural-based stereo vision system," *Real-Time Imaging*, vol. 7, pp. 59–76, 2001.
[17] J. P. P. Starink and E. Backer, "Finding point correspondences using simulated annealing," *Pattern Recognit.*, vol. 28, no. 2, pp. 231–240, 1995.
[18] H. Saito and M. Mori, "Application of genetic algorithms to stereo matching of images," *Pattern Recognit. Lett.*, vol. 16, no. 8, pp. 815–821, 1995.
[19] Y. S. Kim, K. P. Han, E. J. Lee, and Y. H. Ha, "Robust 3-D depth estimation using genetic algorithm in stereo image pairs," in *Proc. IEEE Conf. Circuits and Systems*, 1996, pp. 357–360.
[20] K. P. Han, K. W. Song, E. Y. Chung, S. J. Cho, and Y. H. Ha, "Stereo matching using genetic algorithm with adaptive chromosomes," *Pattern Recognit.*, vol. 34, pp. 1729–1740, 2001.
[21] M. Gong and Y.-H. Yang, "Genetic-based stereo algorithm and disparity map evaluation," *Int. J. Comput. Vis.*, vol. 47, pp. 63–77, 2002.
[22] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1–3, pp. 7–42, 2002.
[23] S. T. Barnard and M. A. Fischler, "Computational stereo," *ACM Comput. Surveys*, vol. 14, no. 4, pp. 553–572, 1982.
[24] U. R. Dhond and J. K. Aggarwal, "Structure from stereo—A review," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 1489–1510, Nov.–Dec. 1989.
[25] L. L. Grewe and A. C. Kak, "Stereo vision," in *Handbook of Pattern Recognition and Image Processing*, T. Y. Young, Ed. New York: Academic, 1994, pp. 239–317.
[26] J. Y. Goulermas, "Evolutionary Techniques for the Stereo Correspondence Problem," Ph.D. dissertation, Control Systems Centre, Univ. Manchester Inst. Sci. Technol. (UMIST), Manchester, U.K., 2000.
[27] G. A. Jones, "Constraint, optimization and hierarchy: Reviewing stereoscopic correspondence of complex features," *Comput. Vis. Image Understanding*, vol. 65, no. 1, pp. 57–78, 1997.
[28] T. Ozanian, "Approaches for stereo matching: A review," *Mod. Identif. Control*, vol. 16, no. 2, pp. 65–94, 1995.
[29] J. Y. Goulermas and P. Liatsis, "Coevolutionary strategies in area-based stereo," in *Proc. VIPromCom-01*, 2001, pp. 65–74.
[30] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[31] J. Reed, R. Toombs, and N. A. Barricelli, "Simulation of biological evolution and machine learning: I. Selection of self-reproducing numeric patterns by data processing machines, effects of hereditary control, mutation type and crossing," *J. Theoretical Biol.*, vol. 17, pp. 319–342, 1967.
[32] W. D. Hillis, "Coevolving parasites improve simulated evolution as an optimization procedure," in *Proc. Artificial Life II*, 1991, pp. 313–324.
[33] J. Paredis, "Coevolutionary computation," *Artif. Life*, vol. 2, no. 4, pp. 355–375, 1995.
[34] ——, "Coevolutionary lifetime learning," in *Proc. PPSN IV, Lecture Notes in Computer Science*, vol. 1141, 1996, pp. 72–80.
[35] ——, "Coevolutionary algorithms," in *The Handbook of Evolutionary Computation*, T. Back, D. Fogel, and Z. Michalewicz, Eds. London, U.K.: Oxford Univ. Press, 1997.
[36] J. Werfel, M. Mitchell, and J. P. Crutchfield, "Resource sharing and coevolution in evolving cellular automata," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 388–393, Nov. 2000.
[37] K. Chellapilla and D. B. Fogel, "Evolving an expert checkers playing program without using human expertise," *IEEE Trans. Evol. Comput.*, vol. 5, pp. 422–428, Aug. 2001.
[38] M. A. Potter and K. A. DeJong, "A cooperative coevolutionary approach to function optimization," in *Proc. PPSN, Lecture Notes Computer Science*, vol. 886, 1994, pp. 249–257.
[39] K. A. DeJong and M. A. Potter, "Evolving complex structures via cooperative coevolution," in *Proc. 4th Conf. Evolutionary Programming*, ch. 48, 1995, pp. 307–317.
[40] H. J. C. Barbosa, "A coevolutionary algorithm for a game approach to structural optimization," in *Proc. 7th Int. Conf. Genetic Algorithms*, 1997, pp. 545–552.
[41] M. J. Tahk and B. C. Sun, "Coevolutionary augmented lagrangian methods for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 114–124, July 2000.
[42] H. Handa, N. Baba, O. Katai, T. Sawaragi, and T. Horiuchi, "Genetic algorithm involving coevolution mechanism to search for effective genetic information," in *Proc. IEEE Int. Conf. Evolutionary Computation*, ch. 127, 1997, pp. 709–714.
[43] H. Handa, O. Katai, N. Baba, and T. Sawaragi, "Solving constraint satisfaction problems by using coevolutionary genetic algorithms," in *Proc. Int. Conf. Evolutionary Computation*, 1998, pp. 21–26.
[44] D. E. Moriarty and R. Miikkulainen, "Hierarchical evolution of neural networks," in *Proc. Int. Conf. Evolutionary Computation*, ch. 145, 1998, pp. 428–433.
[45] N. Richards, D. E. Moriarty, and R. Miikkulainen, "Evolving neural networks to play go," in *Proc. 7th Int. Conf. Genetic Algorithms*, 1997, pp. 768–775.
[46] C. F. Juang, J. Y. Lin, and C. T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 290–302, Apr. 2000.
[47] A. V. Sebald and J. Schlenzig, "Minimax design of neural net controllers for highly uncertain plants," *IEEE Trans. Neural Networks*, vol. 5, pp. 73–82, Jan. 1994.
[48] Q. Zhao, "EditEr: A combination of IEA and CEA," in *Proc. Int. Conf. Evolutionary Computation*, 1997, pp. 641–645.
[49] ——, "A general framework for cooperative coevolutionary algorithms: A society model," in *Proc. Int. Conf. Evolutionary Computation*, 1998, pp. 57–62.
[50] N. Puppala, S. Sen, and M. Gordin, "Shared memory based cooperative coevolution," in *Proc. Int. Conf. Evolutionary Computation*, ch. 145, 1998, pp. 570–574.
[51] L. Bull, "Evolutionary computing in multi-agent environments: Partners," in *Proc. 7th Int. Conf. Genetic Algorithms*, 1997, pp. 371–377.
[52] J. E. Hunt and D. Cooke, "Learning using an artificial immune system," *Network Comput. Appl.*, vol. 19, pp. 189–212, 1996.
[53] E. L. Dixon, C. P. Markhauser, and K. R. Rao, "A new object motion estimation technique for video images, based on a genetic algorithm," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 886–985, Aug. 1997.
[54] C. Jacquelin, A. Aurengo, and G. Hejblum, "Evolving descriptors for textural segmentation," *Pattern Recognit.*, vol. 30, no. 7, pp. 1069–1079, 1997.
[55] C. Ooi and P. Liatsis, "Obstacle tracking using symbiotic snakes," in *Proc. Inst. Elect. Eng. Conf. Advanced Driver Assistance Systems*, 2001, pp. 58–62.
[56] J. Y. Goulermas and P. Liatsis, "Hybrid symbiotic genetic optimization for robust edge-based stereo correspondence," *Pattern Recognit.*, vol. 34, pp. 2477–2496, 2001.
[57] L. Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand, 1991.
[58] A. H. Wright, "Genetic algorithms for real parameter optimization," in *Foundations of Genetic Algorithms*, G. Rawlins, Ed. San Mateo, CA: Morgan Kaufmann, 1991, pp. 205–218.
[59] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. New York: Springer-Verlag, 1996.
[60] R. D. Eastman and A. M. Waxman, "Using disparity functionals for stereo correspondence and surface reconstruction," *Comput. Vis. Graph. Image Processing*, vol. 39, pp. 73–101, 1987.

[61] W. Hoff and N. Ahuja, "Surfaces from stereo: Integrating feature matching, disparity estimation and contour detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 121–136, Feb. 1989.
[62] P. F. Mclauchlan, J. E. W. Mayhew, and J. P. Frisby, "Stereoscopic recovery and description of smooth textured surfaces," *Image Vis. Comput.*, vol. 9, no. 1, pp. 20–26, 1991.
[63] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314–319, 1985.
[64] D. Whitley, "The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 116–121.
[65] T. Back, *Evolutionary Algorithms in Theory and Practice*. London, U.K.: Oxford Univ. Press, 1996.
[66] P. J. B. Hancock, "An empirical comparison of selection methods in evolutionary algorithms," *Evol. Comput., Lecture Notes Comput. Sci.*, vol. 865, pp. 80–94, 1994.
[67] H. Muhlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm. I: Continuous parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 25–49, 1993.
[68] P. Fua, "Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities," *IJCAI*, vol. 63, no. 3, pp. 1292–1298, 1991.
[69] R. Zabih and J. Woodfill, "Non-Parametric local transforms for computing visual correspondence," in *Proc. 3rd European Conf. Computer Vision*, 1994, pp. 151–158.
[70] D. N. Bhat and S. K. Nayar, "Ordinal measures for image correspondence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 415–423, Apr. 1998.
[71] A. Watt, *3D Computer Graphics*, 2nd ed. Reading, MA: Addison-Wesley, 1993.
[72] J. D. Foley, A. vanDam, S. K. Feiner, and J. F. Hughes, *Computer Graphics—Principles and Practice*, 2nd ed. Reading, MA: Addison-Wesley, 1996.
[73] K. M. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer, 1999.
[74] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
[75] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 100–116, Apr. 2003.
[76] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 117–132, Apr. 2003.
[77] A. Jaszkiewicz, "Do multiobjective metaheuristics deliver on their promises? A computational experiment on the set-covering problem," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 133–143, Apr. 2003.
[78] D. A. Van Veldhuizen, J. B. Zydallis, and G. B. Lamont, "Considerations in engineering parallel multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 144–173, Apr. 2003.
[79] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 174–188, Apr. 2003.
[80] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer, "Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 189–203, Apr. 2003.
[81] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 204–223, Apr. 2003.
[82] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Norwell, MA: Kluwer, 2001.

John Yannis Goulermas (M'97) was born in Greece in 1970. He received the B.Sc. degree (honors, Class I) in computation from the University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., in 1994, the M.Sc. degree in research, and the Ph.D. degree from the Control Systems Centre, Department of Electrical Engineering and Electronics, UMIST, in 1996 and 2000, respectively, working in the area of machine vision.

He has worked for two years in industry in the area of financial/pricing modeling and optimization. He is currently a Senior Research Fellow with the Centre for Virtual Environments, University of Salford, Salford, U.K., working in the area of biomechanics and human gait analysis. His main research interests include pattern recognition, artificial intelligence, machine vision, and optimization.

Panos Liatsis (S'87–M'91) received the Diploma in electrical engineering from the University of Thrace, Xanthi, Greece, and the Ph.D. degree from the Control Systems Centre, University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K.

He is currently a Lecturer in the Department of Electrical Engineering and Electronics, UMIST. His main research interests are computer vision, neural networks, and genetic algorithms, with applications to vision-based vehicle guidance, sensor integration, and medical informatics. He is a Member of the Technical Chamber of Greece and a European Engineer (Eur Ing).