J Intell Robot Syst (2012) 68:43–52
DOI 10.1007/s10846-012-9667-8

A Robotic-Driven Disassembly Sequence Generator for End-Of-Life Electronic Products

Ahmed ElSayed · Elif Kongar · Surendra M. Gupta · Tarek Sobh

Received: 29 November 2011 / Accepted: 2 March 2012 / Published online: 16 March 2012

Abstract In this study, we propose an intelligent automated disassembly cell for online (real time) selective disassembly. The cell is composed of an industrial robotic manipulator, a camera, range sensing and component segmentation visual algorithms. The cell prototype allows for robotic sensory-driven disassembly under uncertainty. An online genetic algorithm model for selective disassembly is also proposed for optimal and near/optimal disassembly sequencing.

A. ElSayed (✉)
Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06604, USA
e-mail: aelsayed@bridgeport.edu

E. Kongar
Departments of Mechanical Engineering and Technology Management, University of Bridgeport, Bridgeport, CT 06604, USA
e-mail: kongar@bridgeport.edu

S. M. Gupta
Department of Mechanical Engineering and Industrial Engineering, Northeastern University, Boston, MA 02115, USA
e-mail: gupta@neu.edu

T. Sobh
School of Engineering, University of Bridgeport, Bridgeport, CT 06604, USA
e-mail: sobh@bridgeport.edu

Keywords Robotic disassembly · Visual control · Optimum sequence generation · End-Of-Life products

1 Introduction

Advanced technology products are regularly rendered technically obsolete within a few years of commercialization due to the rapid pace of technological developments. Thus, electronic products are frequently discarded before their materials degrade. These complex End-of-Life (EOL) products contain a broad spectrum of materials including precious metals such as silver, and valuable materials such as copper. Therefore, efficient recovery of materials in the electronic EOL products is essential for economic reasons. Furthermore, the conceptual lifetime of an electronic product depends primarily on the pace of superseding technological change that makes obsolete the otherwise fully-functioning product. Hence, the discarded product is likely to contain one or more usable components. The economically and environmentally sustainable option is to reuse these components in technically valid products. EOL processing options, e.g., reusing, recycling, and remanufacturing, are effective ways to regain the materials and the components in electronic EOL products. Regardless of the motivation, most

EOL processing options necessitate a certain level of disassembly. Disassembly operations are very complex, time-consuming, and expensive. Therefore, limiting the disassembly operations to recyclable materials and reusable components in the EOL product is crucial in order to render electronic waste recovery operations economically viable. Responding to this need, generating disassembly algorithms for selective disassembly [1, 2] is the first major research topic presented in this paper.

Electronic waste recovery, EOL processing options, and disassembly of electronic products are well documented [3–16]. However, related literature falls short in providing realistic methods for product recovery. Despite the ever growing research on disassembly of electronic products, the proposed methods generally assume that the part locations are known in the product structure. Furthermore, the majority of these algorithms function under the assumption of a known hierarchical disassembly path, e.g., for a personal computer (PC) disassembly, the front or side cover must be removed to access the power supply, and the power supply must be removed to access the processor and the fan, etc. However, EOL electronic products—specifically PCs—are unlikely to preserve their original product structure throughout their useful lives. The bill-of-materials (BOM) is likely to be altered by the user for a variety of reasons such as repair, upgrade, or personal configuration preferences. Hence, the EOL product is likely to embody missing, added or replaced parts. This is also true for the fastener structures; when a part is taken out or replaced with another, the type and the location of fasteners are likely to change. RAM (random access memory) slots and PCI (peripheral component interconnect) slots are good examples for this type of uncertainty since these parts are more frequently replaced compared to the rest of the BOM.

In this study, we propose an intelligent automated disassembly cell utilizing a setting composed of an industrial robotic manipulator, a camera, range sensing and component segmentation visual algorithms to design a prototype of a robotic sensory-driven disassembly cell. Figure 1 depicts the Mitsubishi Industrial Micro-Robot System Model RV-M1 and HP Pavilion 6370Z Desktop PC with an additional hard drive and three RAM modules.

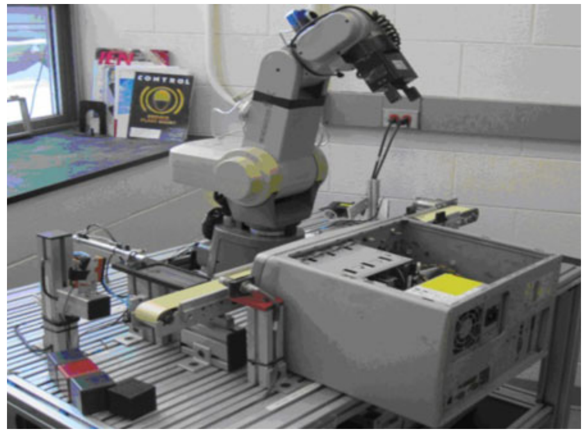

Fig. 1 The RISC laboratory MITSUBISHI MOVEMASTER industrial robotic manipulator, camera, and EOL PC

The proposed model aims at handling uncertainty in the EOL product structure and consists of two modules: (1) a sensory-driven visual and range acquisition and recovery system, and (2) an online genetic algorithm (GA) model. For the sensory processing system, a sensory module is designed to acquire and recognize part descriptions and coordinates via the usage of range and 2-D sensors; complemented with the appropriate pattern matching vision algorithms (Fig. 2). The visual part-recognition software module output is then fed into the GA algorithm that generates online optimal and/or near-optimal disassembly

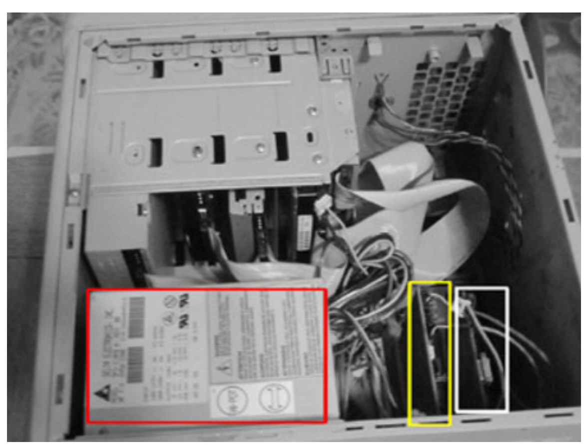

Fig. 2 Camera output for the power supply, sound card, and modem card detection within the EOL PC

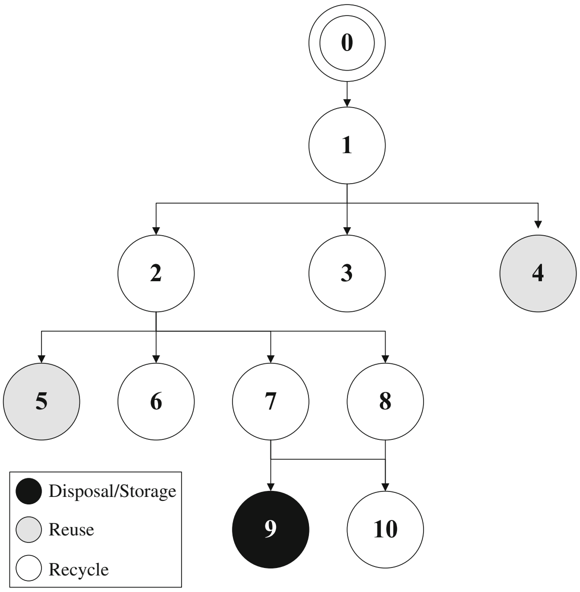

Fig. 3 The Bill-of-Materials of the PC with corresponding END-OF-LIFE processing options

sequences for the detected parts in the product structure (Fig. 3).

In Fig. 3, "0" is the robot reference point where the disassembly operation will start. Table 1 lists the remaining component information with corresponding material, disassembly time, and disassembly method information. Basically, if the component is subject to reuse, non-destructive disassembly is selected as the appropriate disassembly operation to preserve the physical structure of the component. If the component is subject to recycling, (i.e., the material value is important,) the faster disassembly option—destructive disassembly—is preferred. The components that do not contain any market value are subject to storage or proper disposal and are not disassembled unless the precedence relationships mandate their disassembly for the remaining components.

2 Background

Disassembly sequencing problems are combinatorial and NP-complete, prohibiting utilization of exhaustive search techniques. Therefore, genetic algorithms (GA) have been gaining popularity for such problems. One of the multi-objective optimization applications was proposed by Valenzuela-Rendón and Uresti-Charre [17]. Keung et al. [18] also applied a multi-objective GA approach to a tool selection model. Lazzerini and Marcelloni [19] used GA in scheduling assembly processes. In the area of disassembly, Kongar and Gupta [20] proposed a GA for disassembly sequencing problems, while McGovern and Gupta [21] applied genetic algorithm to disassembly line balancing. For a thorough review of work done in the area of product recovery, see Ilgin and Gupta [16], Gungor and Gupta [15], Lee et al. [22], and Lambert and Gupta [23].

Precedence relationships that must be preserved during disassembly operations constitute another factor that increases the computational complexity of disassembly sequence planning problems. Sanderson et al. [24] proposed a methodology considering precedence relationships in assembly sequence planning. Seo et al.

Table 1 Components of the eol pc with corresponding material and disassembly operation information

i	Description	Material	Disassembly	
			Time dt_i (s)	Method[a]
0	Robot reference point			
1	Side cover	Aluminum (A)	3	D
2	Power supply	Copper (C)	6	D
3	Sound card	Plastic (P)	3	ND
4	Modem card	Plastic (P)	3	ND
5	CPU	Plastic (P)	5	ND
6	Hard drive	Aluminum (A)	4	ND
7	CD drive	Aluminum (A)	4	ND
8	Zip drive	Aluminum (A)	4	ND
9	RAM	Plastic (P)	3	ND
10	Drives slot	Aluminum (A)	2	D

[a] *D* Destructive, *ND* Non-destructive

[25] proposed a GA for optimal disassembly sequence generation considering economical and environmental factors. Bierwirth et al. [26] and Bierwirth and Mattfeld [27] proposed precedence preservative crossover (PPX) technique for scheduling problems. The algorithm, which is also employed in this study, preserves the precedence relationships in the product structure. Hui et al. [28] utilized a genetic algorithm to determine good disassembly sequences by converting disassembly sequence planning problem into a searching problem on an information-enhanced graph. Shimizu et al. [29] developed a prototype system for strategic decision-making on disassembly for recycling at the design stage of the product life cycle.

In the area of automated disassembly, Torres et al. [30, 31] presented a personal computer disassembly cell that is able to handle a certain degree of automatism for the non-destructive disassembly process. This work was then followed by Pomares et al. [32] who generated an object-oriented model required for developing a disassembly process. Gil et al. [33] proposed a flexible multi-sensorial system for automatic disassembly using cooperative robots. As a follow-up work, Torres et al. [34] presented a task planner for a disassembly process based on decision trees.

In this paper, we present a GA-based technique for generating online adaptive disassembly sequences for partial disassembly [35]. Online adaptive systems proposed by Milani [36], allow modeling highly evolutionary domains where solution environment are subject to changes over time. Therefore, the model is very suitable for disassembly sequencing problems since the solution environment changes following each disassembly action due to the modifications in the bill-of-materials.

3 Materials and Methods

3.1 Visual Processing via Template/Pattern Matching

Template/pattern is defined as anything fashioned, shaped, or designed to serve as a model from which something is to be made: a model, design, plan or outline, where as *matching* is the act of comparing in respect of similarity; to examine the likeness or difference [37]. The proposed visual processing algorithm utilizes a coarse-to-fine 3-D recovery mechanism. First, in order to reduce the search space for the 2-D camera-based algorithm, Hokuyo UHG-08LX laser range sensing and coarse scene segmentation are performed. The resulting crude $2\frac{1}{2}$-D map of the major components of the PC is then accepted as the search space. The time required for laser scanning is approximately 2.3 s. A 352×288 pixel CCD-array camera is used to capture the online 2D images from the PC under disassembly. Initially, the actual BOM templates sizes (e.g. 407×229 pixels for the power supply, 87×212 pixels for the modem, and 69×245 pixels for the sound card) are used to perform the matching process over a captured image of a resolution of 1000×750 pixels. However, the time required for the matching process was detected to be higher than the minimum disassembly time of the BOM (e.g. the time required for matching the power supply was 1.475 s, 0.837 s for the modem, and 0.877 s for the sound card). This required the robot arm to become idle after the removal of each part. Thus, the resolutions of the stored patterns were reduced and a pyramid structure was utilized with captured images from the PC under disassembly to detect each possible part. The minimum resolution for the captured image used in the matching process was 160×120 pixels, and the resolution of the BOM templates varied. Table 2 depicts the used resolution and the corresponding detection times of each part of the BOM. A correlation/convolution-based 2-D

Table 2 BOM parts resolutions and detection times

Part	Resolution	Detection time (s)
Side cover	74×42	0.078
Power supply	82×49	0.0312
Sound card	9×65	0.0156
Modem card	10×62	0.0156
CPU	63×10	0.0312
Hard drives	14×23	0.0156
CD drive	35×28	0.0156
Zip drive	18×21	0.0156
RAM	62×4	0.0156
Drives slot	32×25	0.0156

template matching module is then executed, after the appropriate scaling, rotation and histogram-equalization algorithms are implemented, in order to filter visual noise and to speed up the template matching process. Components that are matched with a high confidence level are then tagged and a cross-reference—given the calibrated camera parameters—is then performed to recover the planar parameters of the recognized parts. The depth parameters are recovered from the 2½-D map stored by the laser sensor, in order to guide the manipulator end-effector (gripper) for eventual disassembly.

The correlation algorithm used in the matching process is the normalized cross-correlation using the equation provided in [39–41]:

$$\delta(u, v) = \frac{\sum_{x,y}\left(f(x,y) - \overline{f_{u,v}}\right)\left(t(x-u, y-v) - \bar{t}\right)}{\sqrt{\left(\sum_{x,y}\left(f(x,y) - \overline{f_{u,v}}\right)^2\right)\left(\left(t(x-u, y-v) - \bar{t}\right)^2\right)}}$$

where:

t	is the template image under test of a size a × b.
$f(x, y)$	is the captured image from the EOL PC of a size M × N.
$\bar{t}$	is the mean of the template image.
$f_{u,v}$	is the mean of $f(x, y)$ in the region under the template.

The calculation involves following steps:

1) Calculate the correlation complexity (it is bounded by $O\left(N.M.\log_2 M\right)$ since M and N values are significantly larger than a and b).
2) Calculate the local sums by pre-calculating running sums using recursive formulae. The computational complexity is of the order of $O(3.M.N)$.
3) Use local sums to normalize the cross-correlation to calculate the correlation coefficients. The normalization complexity is constant, $(O(1))$.

Thus, from the given equation we can see that reducing the dimension of the captured image from the EOL PC will reduce the time required for calculation of the 2D FFT while reducing the calculation of the local sums which are used in the normalization process.

3.2 Genetic Algorithm

The disassembly sequence generation module is an improved version of previously published work and obtains near-optimal and/or optimal disassembly sequences in a step-wise manner. The algorithm proposed in [2] functions under the assumptions of: (1) the BOM preserves its original structure, (2) the part coordinates are available, and (3) the precedence relationships for removal are known. The algorithm then disassembles the end-of-life PC to remove the parts demanded for reuse and/or recycling. The proposed algorithm relaxes the above assumptions and utilizes only the demand information to launch the sensory vision module. The module generates 2 and 2½-D maps of the EOL PC and produces part matches. The recognized parts and their coordinates are then fed into the GA module that generates a near-optimal and/or optimal disassembly sequence for the detected parts. The disassembly sequence with the appropriate part coordinates is then transferred to the robot arm manipulator. After the removal of each identified part the dynamically active sensory modules detect the parts that become accessible. If no new part is detected the current sequence is continued. If a new part is detected a new sequence is generated for the newly accessible parts and the remaining parts detected in the previous sequence.

The steps of the proposed GA can be summarized as follows (Table 3).

Initial Population The initial population consists of hundred (ncr = 100) random chromosomes that satisfy the precedence relationships and any other constraints imposed by the product structure.

Table 3 Steps of the proposed genetic algorithm

Step 1.	Start with a population of (ncr) random individuals each with chromosome length (chl).
Step 2.	Calculate the fitness F(ch,gn) of each chromosome (ch) in the generation (gn).
Step 3.	Permute the current chromosomes.
Step 4.	Select the first 40 chromosomes for the next generation.
Step 5.	Select 60% of the remaining chromosomes for crossover.
Step 6.	Randomly calculate the crossover probability. If probability holds, perform precedence preservative crossover (PPX) operation, and generate the children for the next generation.
Step 7.	Randomly calculate the mutation probability. If probability holds, perform the mutation operation on a randomly selected number (rnd) of chromosomes starting from the first chromosome, generate the output of the mutation as a new population; if not, define the current population as the new population.
Step 8.	Return to Step 2 until the new generation contains ncr chromosomes. Then replace the old population with the new generation.
Step 9.	If the termination condition is satisfied, STOP, else return to Step 2.

Crossover The precedence preservative crossover (PPX) methodology is utilized for the crossover operation. Here, in addition to the parents ($Parent_1$ and $Parent_2$), two additional strings ($Child_1$ and $Child_2$) pass on the precedence relationship based on the two parental permutations to two new offspring while making sure that no new precedence relationships are introduced. A vector, representing the number of operations involved in the problem is randomly filled with elements of the set. This vector defines the order in which the operations are successively drawn from $Parent_1$ and $Parent_2$.

The PPX algorithm generates an empty offspring. The leftmost operation in one of the two parents is selected in accordance with the order of parents given in the vector. After an operation is selected it is taken out from both parents and is appended to the offspring. This step is repeated until both parents are empty and the offspring contains all operations involved.

Mutation The population is subjected to mutation operation with a given probability. If the probability threshold holds, perform the mutation operation on a randomly selected number of chromosomes (rnd), where ($0<rnd<n-1$), and swaps any two random components in all selected chromosomes without violating the precedence relationships and the overall feasibility (Table 4).

Table 4 System parameters of the proposed genetic algorithm

Parameter	Value/explanation
Initial population	Random and feasible
Population size	ncr = 100
Length of the chromosome	chl = n * 5 (n changes in each level)
Max. number of generations	100
Crossover operator	Precedence preservative crossover (PPX)
Mutation operator	Applies to a random number (rnd) of chromosomes ($0<rnd<n-1$) and swap any two components
Crossover probability	0.60
Mutation probability	0.005
Selection procedure	Roulette selection
Regeneration procedure	Selected chromosomes are cloned to keep the population size constant
Fitness parameters	Basic disassembly time, travel time for robot arm in 3D space, penalty for method change, penalty for pairs
Assumptions	Every component is assumed to have one joint that connects the component with the rest of the product structure

Fitness Evaluation The fitness value is obtained by the overall disassembly time. Disassembly time involves the time spent for the disassembly operations, the travel time for the robot arm in 3D space, the penalty for method change, and the award for recycling pairs. The first time parameter is the basic disassembly time for component i in sequence j, (dt_{ij}).

The second function (tt_{ij}) is the penalty (in seconds) for each travel time to disassemble component i in sequence j, which includes a function of the distance traveled between the (j−1)th and jth sequences and the robot arm speed factor (sf):

$$tt_{ij} = \frac{\sqrt{\left(X_{i(j-1)} - X_{ij}\right)^2 + \left(Y_{i(j-1)} - Y_{ij}\right)^2 + \left(Z_{i(j-1)} - Z_{ij}\right)^2}}{sf}$$

The proposed model assumes that the end effector speed for the robot arm is a constant value of 7 cm/sec. In addition, the time spent for the robot arm angle change (for all three angles) is assumed to be embedded in the disassembly time for each component.

The third criterion in the fitness function is the penalty for disassembly method change (mt_{ij}), for the time spent for tool change during the disassembly operations. For each disassembly method change, the sequence is penalized by 1 s:

$$mt_{ij} = \begin{array}{ll} 0, & \text{If no method change is required, } (e.g.\, ND \text{ to } ND) \\ 1, & \text{If method change is required, } (e.g.\, ND \text{ to } D) \end{array}$$

In addition, the algorithm searches for a "recycling pair" and does not penalize the sequence if the two adjacent components are made of the same material and if they are both demanded for recycling.

Let T_j denote the cumulative disassembly time after the disassembly operation in sequence j is completed for component i. In the case where there is no recycling pair, the overall disassembly time for sequence j is calculated as:

$$T_j = T_{j-1} + dt_{ij} + tt_{ij} + mt_{ij}, \text{ for } j = 1, \ldots, n-2.$$

In the case where there is a recycling pair, the travel and method change times are omitted from the equation. Hence, the overall penalty for sequence j can be calculated as:

$$T_i = T_{j-1} + dt_{ij}, \text{ for } j = n-1 \tag{1}$$

In this proposed GA model, the objective is to minimize the total fitness function (F) by minimizing (i) the traveled distance, (ii) the number of disassembly method changes, and (iii) by combining the identical-material components together, eliminating unnecessary disassembly operations. Let $F(ch, gn)$ denote the total fitness for chromosome ch in generation gn. Hence, the total time to disassemble all the components can be calculated as in Eq. 2:

$$F\,(ch,\ gn) = \sum_{j=0}^{n-1} dt_{ij} + \sum_{j=0}^{n-2} ct_{ij} + \sum_{j=0}^{n-2} mt_{ij}, \quad \forall j,\ j = 0, \ldots, n-1. \tag{2}$$

Selection and Regeneration Procedure After every generation, the chromosomes obtain a certain expectation depending on their fitness values. A roulette wheel is then implemented to select the sequence of parents that will be included in the next generation (the higher the fitness value the higher the chance to be selected). This method aims at allowing the parents in the current generation to be selected for the next generation without getting trapped in the local optima. In addition, a new population is generated eliminating the weak chromosomes.

Termination The algorithm is terminated when the maximum number of generations is exceeded or no further improvement is obtained.

4 Results

The proposed algorithm optimizes the disassembly sequence for the above defined fitness function at every step when a new template is detected. For the provided EOL PC, the algorithm starts with the generated sequence (2 3 4), and removes component 2 (power supply). Detection is then reactivated after the removal, and components 5, 6, 7, and 8 are detected. A new sequence is then generated to include newly detected components and the components that are not yet removed from the BOM (components 3 and 4). Following this, a new sequence is generated (7 8 6 3 5 4). After the removal of component 7, the detection is reactivated and component 9 is detected, followed by a new sequence generation to include the still intact components (8 6 3 5 4) and newly detected component (9). This is followed by a new sequence generation. The algorithm continues until all the reusable and recyclable components are taken out from the EOL product structure.

Table 5 summarizes the model results and provides detection times along with the time it takes to optimize the sequence for each template. The best sequence obtained in each step, the corresponding fitness values are also provided in the table.

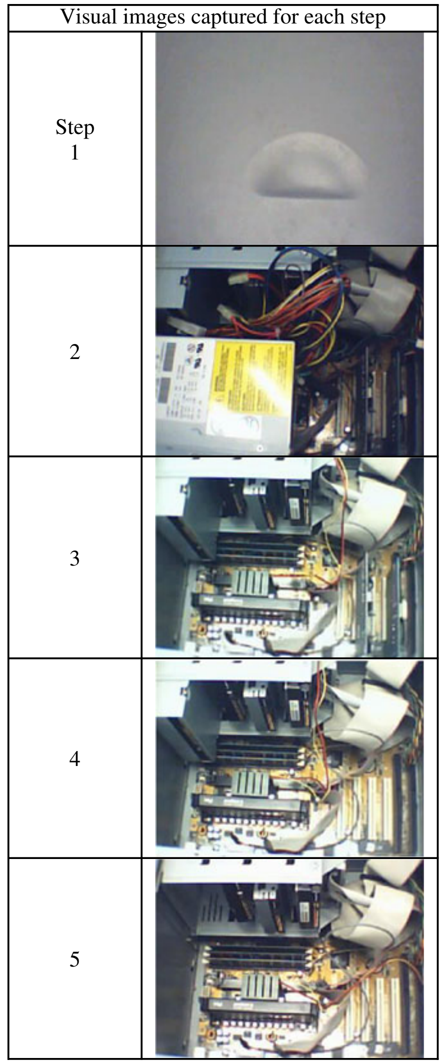

Fig. 4 Visual images captured for each disassembly step

Figure 4 depicts the visual images captured by the camera for each disassembly step.

Table 5 Results of the proposed genetic algorithm

Step	Detection time (s)	Optimization time (s)	Best sequence	Fitness value (s)
1	0.45	N/A	1	24.74
2	0.30	0.4680	2 3 4	49.16
3	0.12	0.1248	7 8 6 3 5 4	61.89
4	0.08	0.3276	8 6 3 4 5 9	60.99
5	0.02	0.3744	10 6 3 4 5 9	64.24

5 Conclusions and Future Work

From the given numerical results of the detection and optimization time we can conclude that the total time required for both detection and optimization of each step is less than the minimum disassembly time of the BOM, which means that the proposed method provides for real time disassembly systems. Since our Normalized Cross-correlation method uses 2D FFT for calculating the correlation and local sums for the normalization, the proposed method performance can be enhanced by using parallel processing techniques on the newer multi-core processors. One of the main advantages of the proposed method is its ability to handle uncertainty, allowing flexibility in the disassembly sequence generation. Furthermore, the method releases unrealistic assumptions and constraints in the GA population allowing improvement in the optimization computations. The algorithm produces reliable and accurate disassembly sequences. The $2^1/_2$-D module utilized to map the depth via a range finder increase both the accuracy and computation time for the visual segmentation and matching framework. The area of automated disassembly research is relatively new, and even though few studies demonstrate electronic disassembly cells [30–34, 38], this research project is one of very few environmentally driven and economically benign disassembly applications combining robotics research with much needed sustainability endeavors. In addition, the online selective disassembly sequencing algorithm proposed in the study further improves the existing body of research in the area of disassembly. Future work will include improving the system capability to capture 3-D models.

References

1. Kongar, E., Gupta, S.M.: Disassembly sequencing using genetic algorithm. Int. J. Adv. Manuf. Technol. **30**, 497–506 (2006)
2. ElSayed, A., Kongar, E., Gupta, S.M.: A genetic algorithm approach to end-of-life disassembly sequencing for robotic disassembly. In: Northeast Decision Sciences Institute Conference, NEDSI 2010, Alexandria, VA (2010)
3. Moyer, L., Gupta, S.M.: Environmental concerns and recycling/disassembly efforts in the electronics industry. J. Electron. Manuf. **7**, 1–22 (1997)
4. Ammons, J.C., Realff, M.J., Newton, D., Cerav, S.: Determining reverse production systems infrastructure for electronic assemblies. In: Proceedings of the 1999 IEEE International Symposium on Electronics and the Environment, Danvers, Massachusetts, pp. 257–262 (1999)
5. Biddle, D.: End-of-Life Computer and Electronics Recovery Policy Options for the Mid-Atlantic States, 2nd edn. Center for Solid Waste Research, Mid-Atlantic Consortium of Recycling and Economic Development Officials (MACREDO), Philadelphia, PA, March (2000)
6. Lambert, A.J.D., Gupta, S.M.: Demand-driven disassembly optimization for electronic products. J. Electron. Manuf. **11**, 121–135 (2002)
7. Linton, J.: Electronic products at their end-of-life: options and obstacles. J. Electron. Manuf. **9**, 41–51 (1999)
8. Sodhi, M.S., Reimer, B.: Models for recycling electronics end-of-life products. OR Spektrum **23**, 97–15 (2001)
9. Veerakamolmal, P., Gupta, S.M.: Analysis of design efficiency for the disassembly of modular electronic products. J. Electron. Manuf. **9**, 79–95 (1999)
10. Wong, L.: Barriers to implementation of electronic end-of-life product programs in the U.S., investigating the environment: research for environmental management. University of California at Berkeley, Environmental Sciences Group Major, Berkeley, CA, Compilation of Research Reports from the Fall 2000/ Spring 2001 Senior Research Seminars (2001)
11. Zhou, M., Caudill, R.J., Sebastian, D.: Multi-lifecycle product recovery for electronic products. J. Electron. Manuf. **9**, 1–15 (1999)
12. Kongar, E., Gupta, S.M.: A multiple objective tabu search approach for end-of-life product disassembly. Int. J. Adv. Operat. Manage. (IJAOM) **1**, 504–531 (2009)
13. Kongar, E., Gupta, S.M.: End-of-life processing via data envelopment analysis. In: Mantri, J.K. (ed.) Research Methodology on Data Envelopment Analysis (DEA), vol. Chapter 12, pp. 197–215. Universal Publishers, Boca Raton, Florida (2008)
14. Kongar, E., Mueller, D.: A comparative study on production and end-of-life cycles of anthropogenic iron cycles. In: 5th International Logistics and Supply Chain Congress 2007, Istanbul, Turkey (2007)
15. Gungor, A., Gupta, S.M.: Issues in environmentally conscious manufacturing and product recovery: a survey. Comput. Ind. Eng. **36**, 811–853 (1999)
16. Ilgin, M.A., Gupta, S.M.: Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art. J. Environ. Manag. **91**, 563–591 (2010)
17. Valenzuela-Rendon, M., Uresti-Charre, E.: A non-generational genetic algorithm for multiobjective optimization. In: Seventh International Conference on Genetic Algorithms, pp. 658–665 (1997)
18. Keung, K.W., Ip, W.H., Lee, T.C.: The solution of a multi-objective tool selection model using the GA approach. Int. J. Adv. Manuf. Technol. **18**, 771–777 (2001)
19. Lazzerini, B., Marcelloni, F.: A genetic algorithm for generating optimal assembly plans. Artif. Intell. Eng. **14**, 319–329 (2000)
20. Kongar, E., Gupta, S.M.: Disassembly to order system under uncertainty. OMEGA **34**, 550–561 (2006)
21. McGovern, S.M., Gupta, S.M.: A balancing method and genetic algorithm for disassembly line balancing. Eur. J. Oper. Res. **179**, 692–708 (2007)

22. Lee, D.-H., Kang, J.-G., Xirouchakis, P.: Disassembly planning and scheduling: review and further research. J. Eng. Manuf. **215**, 695–709 (2001)
23. Lambert, A.J.D., Gupta, S.M.: Disassembly Modeling for Assembly, Maintenance, Reuse, and Recycling. CRC Press, Boca Raton (2005)
24. Sanderson, A.C., Homem de Mello, L.S., Zhang, H.: Assembly sequence planning. AI Mag. **1990**, 62–81 (1990)
25. Seo, K.-K., Park, J.-H., Jang, D.-S.: Optimal disassembly sequence using genetic algorithms considering economic and environmental aspects. Int. J. Adv. Manuf. Technol. **18**, 371–380 (2001)
26. Bierwirth, C., Mattfeld, D.C., Kopfer, H.: On permutation representations for scheduling problems. In: Voigt, H.M., et al. (eds.) Parallel Problem Solving from Nature—PPSN IV. Lecture Notes in Computer Science, vol. 1141, pp. 310–318. Springer, Berlin, Germany (1996)
27. Bierwirth, C., Mattfeld, D.C.: Production scheduling and rescheduling with genetic algorithms. Evol. Comput. **7**, 1–18 (1999)
28. Hui, W., Dong, X., Guanghong, D.: A genetic algorithm for product disassembly sequence planning. Neurocomputing **71**, 2720–2726 (2008)
29. Shimizu, Y., Tsuji, K., Nomura, M.: Optimal disassembly sequence generation using a genetic programming. Int. J. Prod. Res. **45**, 4537–4554 (2007)
30. Torres, F., Gil, P., Puente, S.T., Pomares, J., Aracil, R.: Automatic PC disassembly for component recovery. Int. J. Adv. Manuf. Technol. **23**, 39–46 (2004)
31. Torres, F., Puente, S.T., Aracil, R.: Disassembly planning based on precedence relations among assemblies. Int. J. Adv. Manuf. Technol. **21**, 317–327 (2003)
32. Pomares, J., Puente, S.T., Torres, F., Candelas, F.A., Gil, P.: Virtual disassembly of products based on geometric models. Comput. Ind. **55**, 1–14 (2004)
33. Gil, P., Pomares, J., Puente, S.V., Diaz, C., Candelas, F., Torres, F.: Flexible multi-sensorial system for automatic disassembly using cooperative robots. Int. J. Comput. Integr. Manuf. **20**, 757–772 (2007)
34. Torres, F., Puente, S., Diaz, C.: Automatic cooperative disassembly robotic system: task planner to distribute tasks among robots. Control Eng. Pract. **17**, 112–121 (2009)
35. ElSayed, A., Kongar, E., Gupta, S.M., Sobh, T.: An online genetic algorithm for automated disassembly sequence generation. In: Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2011, Paper No: DETC/MESA 2011-48635, 28–31 August 2011, Washington, DC, USA (2011)
36. Milani, A.: Online genetic algorithms. Int. J. Inf. Theor. Appl. **11**, 20–28 (2004)
37. Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. John Wiley & Sons Inc (2009)
38. Gil, P., Torres, F., Ortiz, F.G., Reinoso, O.: Detection of partial occlusions of assembled components to simplify the disassembly tasks. Int. J. Adv. Manuf. Technol. **30**, 530–539 (2006)
39. Lewis, J.P.: Fast Normalized Cross-Correlation. Industrial Light & Magic (1995)
40. MathWorks: Matlab and simulink for technical computing. http://www.mathworks.com (2011). Accessed November 2011
41. Lewis, J.P.: Fast Template Matching. In: Vision Interface, p. 120–123 (1995)