For office use only	Team Control Number	For office use only
T1 _____	2012050	F1 _____
T2 _____		F2 _____
T3 _____	Problem Chosen	F3 _____
T4 _____	A	F4 _____

2020
MCM/ICM
Summary Sheet

An MCM Paper Made by Team 2012050

Here is the abstract of your paper.
Firstly, that is ...
Secondly, that is ...
Thirdly, that is ...
Finally, that is ...

Contents

1 Introduction

1.1 Problem Background

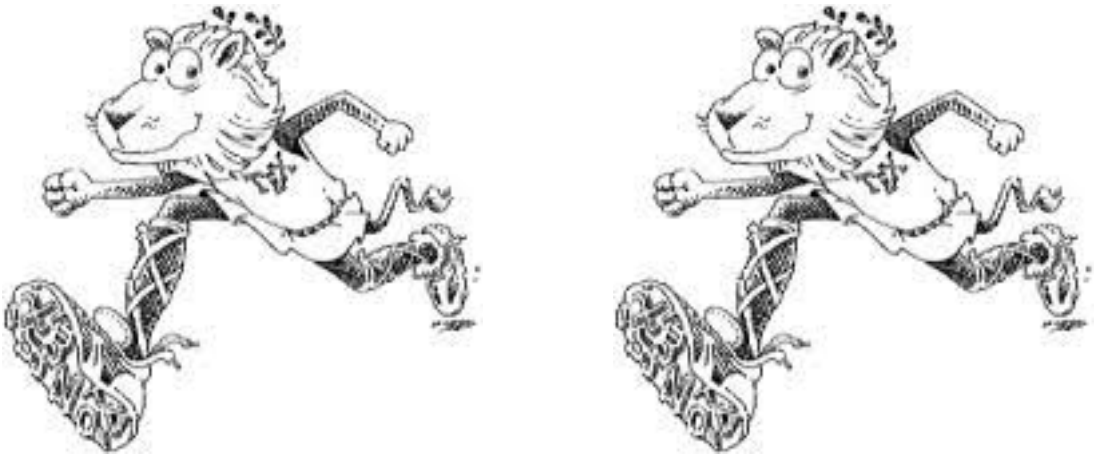
Here is the problem background [1] ...

Two major problems are discussed in this paper, which are:

- Doing the first thing.
- Doing the second thing.

1.2 Literature Review

A literatrue say something about this problem ...



(a) ElegantLaTeX Logo (b) ElegantLaTeX Logo

Figure 1: ElegantLaTeX Logo

Figure 1 is the logo of the Tex.

Figure 2: logo

1.3 Our work

We do such things [2] ...

1. We do ...
2. We do ...
3. We do ...

2 Preparation of the Models

2.1 Assumptions

2.2 Notations

The primary notations used in this paper are listed in **Table 1**.

Table 1: Notations

Symbol	Definition	Unit
A	the first one	cm
b	the second one	cm
α	the last one	cm

3 The Models

3.1 Model 1

3.1.1 Detail 1 about Model 1

$$e^{i\theta} = \cos\theta + i\sin\theta. \tag{1}$$

$$\iiint_D \mathrm{d}f = \max_D g \tag{2}$$

Algorithm 1: GF(4) 3D reconstruction

Input: $\mathcal{X} \in \mathbb{R}^{l_1 \times l_2 \times l_3}, K_c, K_p, R, T$
Output: $Coord_{i,j}$
Initialize all $GF^{(i,j)}s$
for each $X_{ij}^k (N_0 \leq i \leq N_1, M_0 \leq j \leq M_1, k \in (r,g,b))$ **do**
  $d = \max(|\sum_{i=-\epsilon}^{\epsilon} I(x^k + i, y^k) - \sum_{j=-\epsilon}^{\epsilon} I(x^k, y^k + j)|)$;
  if $d > t$ **then**
    $C_{ij} = -1$
  else
    $Candidate_{ij} = -3$
for each $Candidate_{ij}^k (N_0 \leq i \leq N_1, M_0 \leq j \leq M_1)$ **do**
  if $Candidate_{ij} == -1$ **then**
    $\rho_C = \frac{n\sum_{i=1}^{n} M_{Ci} M_{Ci'} - \sum_{i=1}^{n} M_{Ci} \sum_{i=1}^{n} M_{Ci'}}{\sqrt{n\sum_{i=1}^{n} M_{Ci}^2 - (\sum_{i=1}^{n} M_{Ci})^2} \sqrt{n\sum_{i=1}^{n} M_{Ci}'^2 - (\sum_{i=1}^{n} M_{Ci'})^2}}$;
    if $\rho_C > t$ **then**
      $GridPoint_{ij} = -1$
for each $GridPoint_{ij}^k (N_0 \leq i \leq N_1, M_0 \leq j \leq M_1)$ **do**
  $FeaturePoint_{i,j} = BFS(GridPoint_{i,j}, FLAG)$;
  if $FeaturePoint_{i,j} == -1$ **then**
    if $\sum_{i=-\epsilon}^{\epsilon} I(x^k + i, y^k) - \sum_{j=-\epsilon}^{\epsilon} I(x^k, y^k + j) > 0$ **then**
      $FeaturePoint_{i,j} = -1$
    else
      $FeaturePoint_{i,j} = -2$
for each $FeaturePoint_{ij}^k (N_0 \leq i \leq N_1, M_0 \leq j \leq M_1)$ **do**
  if $FeaturePoint_{i,j} \neq -1$ *and* $FeaturePoint_{i,j} \neq -2$ **then**
    $s = \sqrt{1 - \frac{rg+gb+rb}{r^2+g^2+b^2}}$;
    $h_r = \frac{2r-g-b}{2\sqrt{(r-g)^2+(r-b)(g-b)}}$;
    $h_g = \frac{2g-r-b}{2\sqrt{(g-r)^2+(g-b)(r-b)}}$;
    $h_b = \frac{2b-g-r}{2\sqrt{(b-g)^2+(b-r)(g-r)}}$;
    $k = s - \sqrt{1 - \max(h_r, h_g, h_b)}$ **if** $k < 0.2$ **then**
      $FeaturePoint_{i,j} = 0$
    else
      $FeaturePoint_{i,j} = \max(r, g, b)$
for each $FeaturePoint_{ij}^k (N_0 \leq i \leq N_1, M_0 \leq j \leq M_1)$ **do**
  if $FeaturePoint_{i,j} == -1$ *or* $FeaturePoint_{i,j} == -2$ **then**
    $(u_1 m_{31}^1 - m_{11}^1)X_W + (u_1 m_{32}^1 - m_{12}^1)Y_W + (u_1 m_{33}^1 - m_{13}^1)Z_W = m_{14}^1 - u_1 m_{34}^1$;
    $(v_1 m_{31}^1 - m_{21}^1)X_W + (v_1 m_{32}^1 - m_{22}^1)Y_W + (v_1 m_{33}^1 - m_{23}^1)Z_W = m_{24}^1 - v_1 m_{34}^1$;
    $(u_1 m_{31}^2 - m_{11}^2)X_W + (u_1 m_{32}^2 - m_{12}^2)Y_W + (u_1 m_{33}^2 - m_{13}^2)Z_W = m_{14}^2 - u_1 m_{34}^2$;
    $Coord_{i,j} = (X_W, Y_W, Z_W)$

4 Strengths and Weaknesses

4.1 Strengths

- First one...
- Second one ...

4.2 Weaknesses

- Only one ...

References

[1] Zhan Song and Chi Kit Ronald Chung. Determining both surface position and orientation in structured-light-based sensing. IEEE Transactions on Pattern Analysis & Machine Intelligence, 32(10):1770–1780, 2010.

[2] Haibo Lin, Lei Nie, and Zhan Song. A single-shot structured light means by encoding both color and geometrical features. Pattern Recognition, 54:178–189.

Apendix: The source codes

This MATLAB program is used to calculate the value of variable a.

Program 1: `temp.m`

```
a = 0;
for i = 1:5
    a = a + 1;
end
```

This LINGO program is used to search the optimize solution of 0-1 problem.

Program 2: `temp.lg4`

```
model:
sets:
WP/1..12/: M, W, X;
endsets
data:
M = 2 5 18 3 2 5 10 4 11 7 14 6;
W = 5 10 13 4 3 11 13 10 8 16 7 4;
enddata
max = @sum(WP:W*X);
@sum(WP: M * X)<=46;
@for(WP: @bin(X));
end
```

Program 3: `temp.py`

```
employees = []
for id in employee_ids:
    employee = fetch_employee(id)
if employee:
    employees.append(employee)
```